

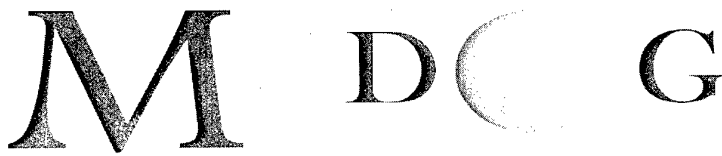
R+D 8957-AN-03
N 68171-00-M.5704

Proceedings of the

**INTERNATIONAL WORKSHOP ON
MULTIDISCIPLINARY DESIGN OPTIMIZATION**

7 - 10 August 2000, Pretoria, South Africa

Organized by



Multidisciplinary Design Optimization Group
Department of Mechanical Engineering, University of Pretoria



Under the auspices of
**The International Society of Structural and Multidisciplinary
Optimization (ISSMO)**

Supported by
**US Army Research Laboratory - European Research Office
The European Office of Aerospace Research and Development (EOARD)**

20010508 105

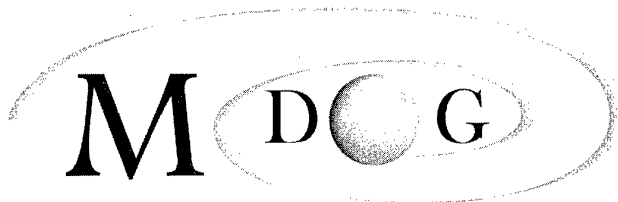
Proceedings of the

**INTERNATIONAL WORKSHOP ON
MULTIDISCIPLINARY DESIGN OPTIMIZATION**

7 - 10 August 2000, Pretoria, South Africa



Editors: Jan Snyman and Ken Craig



Multidisciplinary Design Optimization Group
Department of Mechanical Engineering, University of Pretoria

BIOGRAPHICAL SKETCHES OF KEYNOTE SPEAKERS

Panos Y. Papalambros
University of Michigan, Ann Arbor

Professor of Mechanical Engineering, University of Michigan. Diploma ME/EE NTU-Athens (1974), MS and PhD, Stanford (1976, 1979). At Michigan since 1979; Department Chair 1992-98. Areas of interest: design methodology, optimization and systems integration; ecologically conscious design. Over 100 publications; co-author of Principles of Optimal Design: Modeling and Computation (1988, 1991). Society memberships: ASME, INFORMS, MPS, SIAM, SME, SAE, AIAA, ISSMO and ASEE. Editorial Boards: Artificial Intelligence in Engineering Design and Manufacturing, Engineering Design, Engineering Optimization, Integrated Computer-Aided Engineering, Structural Optimization and Design Optimization. ASME Design Automation Award (1998). Fellow of ASME.

Edward J. Haug
University of Iowa, Iowa City

Carver Distinguished Professor of Mechanical Engineering and Director, NSF Center for Virtual Proving Ground Simulation, University of Iowa. BSME, University of Missouri-Rolla (1962); MS, Kansas State (1964); PhD, University of Missouri-Rolla (1966). US Army Weapons Command and Armaments Command (1966-76). Areas of interest: mechanical system analysis and design optimization, dynamics of mechanisms and machines, computational methods in mechanics. Author of more than 200 technical papers and author/editor of 14 books. Editor: Mechanics of Structures and Machines. Associate Editor of four additional journals. Director; four NATO Advanced Study Institute Workshops.

CONTENTS

Keynote Papers

- Trends and challenges in system design optimization
Panos Y. Papalambros & Nestor F. Michelena 1
- Derivatives for kinematic and dynamic analysis and optimization
Radu Serban & *Edward J. Haug* 16

Contributed Papers

- A feedback controller design methodology for vehicle suspension systems
M.M. Ali 26
- The minimisation of traffic noise over an irregular wall
Teo Bührmann 37
- Global optimization for noise and multiple local optima
Steven E. Cox, Raphael T. Haftka, C.A. Baker, B. Grossman, W.H. Mason & L.T. Watson 50
- Saddle points in design optimization
K.J. Craig & D.J. de Kock 60
- Optimal tundish design using CFD with inclusion modelling
D.J. de Kock and Ken Craig 70
- A general mathematical programming method for the determination of manipulator workspaces
L.J. du Plessis & A.M. Hay 79
- Minimum cost design of welded structures
József Farkas 89
- Particle swarms in size and shape optimization
P.C. Fourie & A.A. Groenwold 97
- Control of a three-link manipulator subject to inequality constraints
C. Frangos and Y. Yavin 107
- Rotordynamic analysis in the design of rotating machinery
G. Genta & E. Brusa 124
- Competing parallel algorithms and multiple local searches in global optimization
Albert A. Groenwold, J.F. Schutte & H.P.J. Bolton 134

Choosing optimal control policies using the attainable region approach S. Godorr, D. Hildebrandt, D. Glasser, C. McGregor & Brendon Hausberger	144
The Dynamic-Q optimization method: An alternative to SQP? J.A. Snyman & A.M. Hay	163
An optimisation approach to engine mounting design P.S. Heyns	173
Practical guidelines for training neural networks J.E.W. Holm	181
Economic design of welded I-beams with PWT and cellular plates Károly Jármai	192
The use of the Dynamic Leapfrog Algorithm in power system state estimation J.A. Jordaan & R. Zivanovic	202
Parameter estimation of polycrystal model through identification studies S. Kok , A.J. Beaudoin & D.A. Tortorelli	210
Optimisation of vehicle suspension characteristics A.F. Naudé & J.A. Snyman	220
Sound and vibration optimization of carillon bells and MRI scanners A.J.G. Schoofs , P.H.L. Kessels, A.H.W.M. Kuijpers & M.H. van Houten	230
Aeroelastic tailoring of aerodynamic surfaces and low cost wind tunnel model design Otto Sensburg , J. Schweiger, V. Tischler & V.B. Venkayya	240
The optimal design of a planar parallel platform for prescribed machining tasks W.J. Smit	254
The treatment of lock-up in the optimal design of serially linked manipulators performing prescribed tasks J.A. Snyman	264
Shape optimization for crashworthiness using distributed computing A. Akkerman, R. Thyagarajan, Nielen Stander , M. Burger, R. Kuhn & H. Rajic	270
The spherical approximation graph matching algorithm B.J. van Wyk & M.A. van Wyk	280
Optimization of heat sinks using mathematical optimisation J.A. Visser & D.J. de Kock	289
AUTHOR INDEX	299

TRENDS AND CHALLENGES IN SYSTEM DESIGN OPTIMIZATION

Panos Y. Papalambros

Nestor F. Michelena

Optimal Design Laboratory

Department of Mechanical Engineering

University of Michigan

Ann Arbor, Michigan, USA

ABSTRACT

Design optimization is now a mainstream discipline in high technology product development and a natural extension of the ever-increasing analytical abilities of computer-aided engineering. Business factors, such as globalization, outsourcing, supply-chain management, and rapid new product deployment, are placing increased emphasis on a "systems" approach to product design. Technological factors, such as the emergence of new technologies at the intersection of traditional ones (e.g., MEMS, nanotechnologies, and biotechnologies) and the widespread use of the internet, are also forcing increased emphasis on the same "systems" approach. As a result, there has been an increased need to study design optimization methods that can address effectively the "system" problem. In this article we view a system as a collection of entities that might be properly structured so that some form of decomposition is possible. The problem has two parts: finding a proper system partition that captures an appropriate structure, and solving the partitioned problem with a coordination method that guarantees some form of convergence that is meaningful for the original undecomposed problem. We review some key ideas in these two issues and we show how they can be used effectively in product development processes, such as target cascading, product family design, and combined design and control of "smart" artifacts.

1 INTRODUCTION

Perhaps one of the few agreed-upon characterizations of the world today is its "complexity." This complexity is nothing new, but the present emphasis stems from an often undeclared desire to deal with it directly. This desire in turn stems from our increased ability to deal with the complexity of the physical world, to a large degree due to the rapid growth of computing and information technology. The same trend is seen in the design of engineered artifacts. Here complexity is manifested by a gradual migration of efforts towards a "system" design. Occasionally the term "system" can be rigorously defined within a specific discipline, for example, an "input-output" system where the system is a function mapping a set of input quantities to a set of output quantities. The term is also used more casually in a variety of contexts.

Often the term "system" is used in contrast to the term "component." The context then is to address complexity derived from studying a collection of components that function jointly thus comprising a system. A system can be composed of other systems that are often referred as "subsystems," if one wants to emphasize the indivisible nature of a component. For example, a system can be an automobile—a collection of a great number of subsystems and components. A collection of variants of automobiles can be another definition of a system. This is one of the contexts that the term "system" is used in the present article.

Another frequent use of the term "system" is to suggest a more encompassing viewpoint—the "system view." Here the context is that an engineered artifact must be studied from various perspectives *simultaneously*, the assumption being that the artifact's functions may seem different from each perspective but their interactions are critical to the overall function of the artifact. Complexity is induced from the need to account for all these perspectives together. For example, the design of controllable artifacts requires that the embodiment design and controller design be studied simultaneously in order to produce efficient smart products. This is another context that "system" is used in the present article.

Design optimization assumes a decision-making paradigm for the design process. The formal *mathematical model* of the optimization problem is a statement of the form

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ &\quad \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ &\quad \mathbf{x} \in \mathcal{X} \subseteq \mathcal{R}^n \end{aligned} \tag{1}$$

where the scalar *objective function* $f(\mathbf{x})$ provides the comparison criterion among different alternatives, the vector-valued functions $\mathbf{h} = (h_1, h_2, \dots, h_{m_1})^T$ and $\mathbf{g} = (g_1, g_2, \dots, g_{m_2})^T$ are the *constraint functions* that determine whether a design is feasible, and $\mathbf{x}$ is the n -dimensional vector of the *design variables*, where n is finite. In many embodiment design problems the design variables take continuous real values in the n -dimensional real space $\mathcal{R}^n$. However, in many problems design variables may take only discrete values, such as standard sizes of cross-sections or configuration design problems. In model (1) the type of values used is described by the set $\mathcal{X}$, the *set constraint*. Models with continuous variables are generally easier to solve with techniques based on differential calculus.

Design problems often include two or more competing objectives, leading to the *multiobjective* or *multicriteria* problem

$$\begin{aligned} &\text{minimize } \mathbf{c}(\mathbf{x}) \\ &\text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ &\quad \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ &\quad \mathbf{x} \in \mathcal{X} \subseteq \mathcal{R}^n \end{aligned} \tag{2}$$

where $\mathbf{c}$ is the vector of I real-valued criteria c_i . The feasible values for $\mathbf{c}(\mathbf{x})$ constitute the *attainable set* $\mathcal{A}$. The multicriteria formulation is converted into a *scalar substitute problem* of the general form

$$\begin{aligned} &\text{minimize } f = \sum_i f_i(w_i) f_2(c_i, \mathbf{m}_i) \\ &\text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ &\quad \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ &\quad \mathbf{x} \in \mathcal{X} \subseteq \mathcal{R}^n \end{aligned} \tag{3}$$

where the scalars w_i and vectors $\mathbf{m}_i$ are preference parameters. Pareto optimality is a common preference structure. A point in the design space is a Pareto (optimal) point if there exist no feasible point that would reduce one criterion without increasing the value of one or more of the other criteria. The functions f , $\mathbf{h}$, and $\mathbf{g}$ can be explicit algebraic expressions representing an *analysis model* but may also be the formal statement of a complex procedure involving internal calculations and realized only as a computer program—what is often called a *simulation model*, such as numerical solutions of coupled differential equations. An *(optimal) design model* refers to Eq. (1) or (2). If the functions represent algebraic or equivalent relations of a finite design vector $\mathbf{x}$, then model (1) represents a *mathematical programming problem*. If differential or integral operators are involved and the variables $x_i = x_i(t)$, $t \in \mathcal{R}$, are defined in an *infinite dimensional space*, then we have a *variational problem*.

Designing complex engineered artifacts (and collections of them) most likely requires use of simulations coupled with optimization techniques. As complexity increases, our ability to employ intuition (even for understanding the computed trade-offs) declines rapidly. Furthermore, our ability to actually solve these system optimization problems becomes suspect as dimensionality increases. An approach towards solving these challenging problems is obvious: try to break down the problem to smaller ones that can be solved more easily and then compose the overall system solution from the solution of its parts. This decomposition approach can be effective but its practical and rigorous implementation is much more difficult than one might expect.

In this article we will first explore the issues associated with decomposition strategies for optimal system design. Then we will look at some specific implementations representative of current research efforts in this area. More specifically, we will briefly examine work on product development processes, such as platform design and target cascading, as well as

on simultaneous design and control strategies. Our aim here is to scope out some key ideas rather than present a comprehensive review. Thus references are relatively sparse and biased towards the work experiences of the authors.

A decomposition strategy is a process with two steps: partitioning and coordination. Partitioning the problem into smaller subproblems was originally done in an ad hoc manner. Newer developments allow us to do this step in a rigorous and flexible manner. Coordination of the solutions of the individual subproblems so that the solution of the overall problem is achieved is the core activity in a decomposition strategy. Coordination is difficult to perform rigorously and efficiently, and eventual success depends on the form of the problem partitioning. We will explore these issues in the next two sections.

2 PARTITIONING: BREAKING A PROBLEM DOWN TO PIECES

Partitioning methods are commonly classified as *object decomposition* (by physical components), *aspect decomposition* (by knowledge domains—the original motivation for multidisciplinary optimization or MDO), *sequential decomposition* (by directed flow of elements or information), and *model-based*. Object and aspect decomposition assume a “natural” decomposition of the problem. However, drawing “boundaries” around physical components and subassemblies is subjective, while division by specialties (knowledge domains) may be dictated by management considerations that fail to account for disciplinary coupling. Sequential decomposition presumes unidirectionality of design information flow that may contradict the cooperative behavior desirable in concurrent engineering. Model-based partitioning uses the mathematical functional representation of design objectives and constraints in the model(s) to identify unconnected or weakly-connected structures implicit in the mathematical design model, and is limited to the design decisions included in the model. However, the partitioning process can be automated and various ways to break down the problem can be quickly generated and evaluated.

The process starts with establishing the problem’s design function dependency on design variables, represented by a Boolean matrix termed the *functional dependence table* (FDT). Rows are labeled with relation/function names and columns are labeled with variable names. The entry in the i th row and j th column is “true” if the i th function depends on the j th variable; otherwise, it is “false.” A typical FDT is shown in Figure 1(a), the shaded boxes indicating a “true” Boolean value. Figure 1(b) shows the FDT for the same problem after x_1 has been selected as the linking variable, and rows and columns have been reordered to reveal two partitions of the problem: subproblem 1 with functions $\{f_2, f_4, g_2, g_4\}$ and local variables $\{x_2, x_4, x_6, x_8\}$, and subproblem 2 with functions $\{f_1, g_1, g_3\}$ and local variables $\{x_3, x_5, x_7\}$. (A hierarchical coordination strategy would consider a master problem on the linking variable x_1 and function f_3 .)

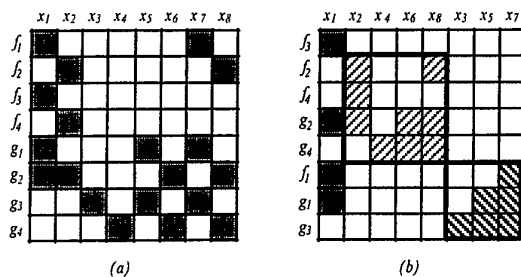


Figure 1. (a) Original form of an FDT and (b) derived form after reordering rows and columns to identify subproblems

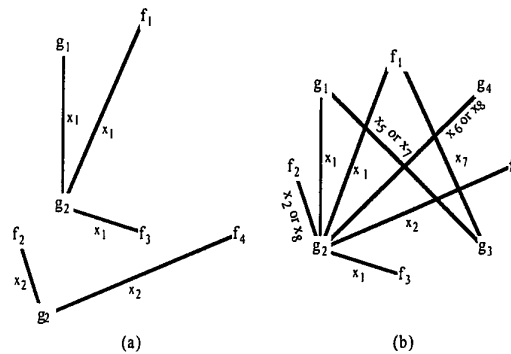


Figure 2. (a) Tree representation of function dependency on variables x_1 and x_2 , and (b) undirected graph representation of problem of Figure 1

The partitioning above can be made rigorous by modeling the decomposition problem as a network optimization problem (Michelena and Papalambros 1995). Mathematical relations are modeled as processing units of a communication network and design and state variables are communication links between these units. The optimal decomposition problem is then formulated as one of finding the communication links whose failure would reduce network reliability the most.

Extending this partitioning model to include other graph metrics, such as keeping subsystem sizes balanced and number of linking variables small, has led to partitioning methods that use hypergraph or integer programming formulations (Michelen and Papalambros 1997; Krishnamachari and Papalambros 1997).

Details of the network reliability, hypergraph partitioning, and integer programming formulations of problem partitioning follow.

Network reliability approach to problem partitioning

The partitioning problem is formulated as a multiobjective optimization problem with two conflicting objectives, namely, minimize the number of linking variables and minimize the size of the subproblems by maximizing the number of partitions. A design problem (and therefore its FDT) can be represented by an undirected, linear graph. This representation does not require a priori knowledge of input-output relations or causality between variables. A simplified version of the graph representation employed by Wagner and Papalambros (1993) is used in the network reliability formulation of problem partitioning. Wagner and Papalambros assigned a clique to each variable, and proved the equivalence of disjoint partitions in the FDT and connected components in its graph representation. In a clique-based representation, a clique connects vertices representing functions that depend on the same variable. Thus, if k functions depended on a variable, a k -clique was constructed for such a variable. In the network formulation a tree is used to connect functions that depend on the same variable. This tree-based representation is sufficient for the purpose of identifying connected components. An edge is labeled with the variable name(s) the functions associated with the edge's incident vertices depend on. Figure 2(a) shows trees associated with variables x_1 and x_2 in the FDT of Figure 1. Figure 2(b) shows the graph representation for the entire problem of Figure 1.

The undirected graph representation motivates formulating partitioning as network reliability optimization with two conflicting objectives: maximize the number of functioning links and minimize a measure of overall network reliability. Functioning links are identified with local variables, while failed links are identified with linking variables. Common sense indicates that the more connected a network is, the more reliable it is. This network optimization may be also viewed as selecting critical communication links, the linking variables, and assigning their control to a top decision-maker, the *master problem*. The control of the other links, the local variables, is left to low-level decision-makers, the *subproblems*. Critical links are those whose failure lessens the most the overall reliability of the network. Critical links are identified by finding the Pareto points shown in Figure 3. Solution point "A" corresponds to the case where every variable is considered a linking variable, so the problem is entirely disconnected. Solution point "B" corresponds to the case where every variable is considered a local variable, so the problem is maximally connected.

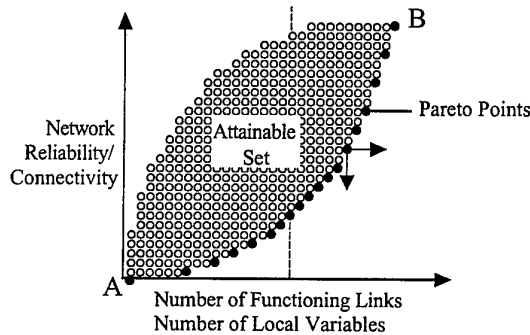


Figure 3. Pareto solution of partitioning problem

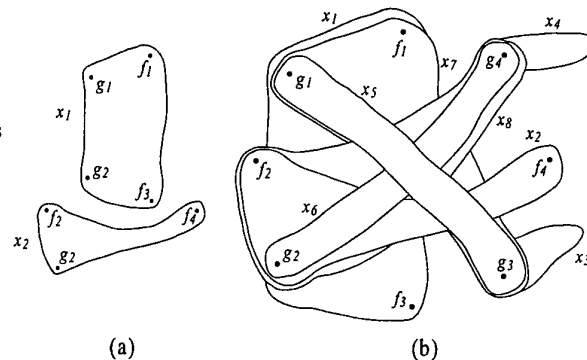


Figure 4. (a) Hyperedge representation of function dependency on variables x_1 and x_2 , and (b) hypergraph representation of problem of Figure 1

Two measures, all-terminal network state and network resilience, can be used as measures of network—and therefore design problem—connectivity. The all-terminal state ϕ_A of a network is one for totally connected graphs and zero for disconnected graphs, independent of the relative size of the parts. Network resilience ϕ_{PC} denotes the number of pairs of vertices in a network that are connected. Network resilience allows identifying partitions of similar size since a low value

indicates that the network is divided in many pieces of similar size. Efficient algorithms to compute these two measures of network connectivity were presented in Michelena and Papalambros (1995). A multiobjective problem with two conflicting objectives is thus formulated: (i) minimize the number of linking variables, i.e., failed links, by maximizing the sum of the indicator variables e_i (e_i is one if edge i is functioning and zero otherwise), and (ii) maximize the number of partitions by minimizing either the all-terminal network state or the network resilience.

$$\text{minimize } \left\{ - \sum_{i=1}^m e_i, \phi_A(e) \text{ or } \phi_{PC}(e) \right\} \quad (4)$$

$$e \in \{0, 1\}^m$$

Since ϕ_A is zero or one, Pareto points e^* are generated by factoring of ϕ_A and identifying factors that take a zero value with the fewest number of indicators e_i equal to zero. For ϕ_{PC} , m Pareto points e^* can be found by minimizing ϕ_{PC}

subject to the successive constraints $\sum_{i=1}^m e_i \geq m, m-1, \dots, 2, \text{ or } 1$, over the space $\{0, 1\}^m$. Alternatively, as shown in Michelena and Papalambros (1995), a method of objective weighting or a greedy algorithm can be used to generate the Pareto solutions.

Hypergraph partitioning approach to problem partitioning

The partitioning problem can be also formulated as a hypergraph partitioning problem. The resulting representation is robust enough to account for computational demands of the functions (simulations) in the model and for the strength of their interdependencies. This approach makes use of recent advances common to such diverse areas as graph theory, VLSI design, computational mechanics, and parallel computing.

A design problem can be represented by a hypergraph $H = (V, E_H)$ in which hyperedges in E_H are subsets of V . Vertices in V represent design relations (i.e., objective and constraints), and hyperedges represent design and intermediate variables. A hyperedge $e_i \in E_H$ represents a variable x_i if and only if for every vertex $v_i \in e_i$, the function associated with v_i depends on x_i . Figure 4(a) shows hyperedges associated with variables x_1 and x_2 in the FDT of Figure 1(a), respectively. Figure 4(b) shows the hypergraph representation for the entire problem of Figure 1. Optimal partitioning calls for (i) minimizing the interconnection between subproblems and (ii) balancing the size of the subproblems. The former is aimed at reducing the effort to coordinate individual subproblems, and the latter is aimed at matching available computational resources. Hence, the following hypergraph partitioning problem can be formulated (Michelena and Papalambros 1997):

Hypergraph K -Partitioning Problem. Given a hypergraph $H = (V, E_H)$ containing N vertices $V = \{v_1, v_2, \dots, v_N\}$ with positive weights $w_v(v_i)$, and M hyperedges $E_H = \{e_1, e_2, \dots, e_M\}$ with positive weights $w_e(e_j)$, a constant $2 \leq K \leq N$, and a partition load (or size) vector $\mathbf{m} = (m_1, \dots, m_K)$ such that $m_k \geq m_{k+1}$ and $\sum_{k=1}^K m_k = \sum_{i=1}^N w_v(v_i)$, find a partition of V into K

disjoint subsets $P^K = \{V_1, V_2, \dots, V_K\}$ that minimizes (i) the total weight of the hyperedges cut by P^K , $C(P^K)$, and (ii)

$\left| \sum_{v_i \in V_k} w_v(v_i) - m_k \right|$ for every k in $\{1, 2, \dots, K\}$. The hyperedges cut by P^K are $E_H^C(P^K) = \{e_j \in E_H \text{ such that there exist } v_{i1}, v_{i2} \text{ in } e_j, v_{i1} \in V_{j1} \in P^K, v_{i2} \in V_{j2} \in P^K, \text{ and } j_1 \neq j_2\}$. Thus, the total weight of the hyperedges cut by P^K is

$$C(P^K) = \sum_{e_j \in E_H^C(P^K)} w_e(e_j).$$

When this formulation is applied to design problem partitioning, vertex weights represent computational costs (e.g., CPU time or memory) for the design relations, edge weights depict coupling strength or amount of transferred data between computational (e.g., simulation) modules, and partition loads represent processing capabilities in a distributed computational environment. Solution methods for the hypergraph K -partitioning problem include iterative improvement partitioning methods, such as the Kernighan-Lin algorithm, and global partitioning methods, such as spectral algorithms.

Integer programming approach to problem partitioning

Hierarchical partitioning of a design problem has also been formulated as an integer program (Krishnamachari and Papalambros 1997). In the proposed "hierarchical decomposition synthesis" methodology, a hierarchically decomposed optimal design problem is obtained from studying first a general design problem (GDP) that contains only design relations but has no objective(s) defined. As in the hypergraph partitioning formulation, the integer linear programming (ILP) formulation assumes that there are two desirable characteristics of the decomposed design problem. The master problem and subproblems should be relatively small in size to facilitate comprehension and computation, and of approximately the same size to facilitate validation, parametric studies, and load balancing in case of parallel solution. The first characteristic leads to an objective function that attempts to minimize both the size of the master problem and the average size of the subproblems: minimize w_m (size of the master problem) + w_s (average size of the subproblems), where w_m, w_s are weights. The second characteristic imposes the constraint: $K_s \times (\text{size of the smallest subproblem}) \geq (\text{size of the largest subproblem})$, where $K_s \geq 1$ is a size factor. The size of a problem is defined as equal to the sum of the number of variables and the number of design functions that it contains.

In the formulation, each problem is designated by its variables and functions. All variables and functions in the problem must be assigned to the master problem or some subproblem. The master problem contains design relations that are functions exclusively of the linking variables. A local variable belongs to a subproblem if the function that depends on that variable is in the subproblem. Each function can belong to only one subproblem. Two different functions belonging to two different subproblems cannot have any common variables other than the linking ones. An integer linear programming (ILP) model is created whose zero-one variables indicate what problem the design variables and functions are assigned to. The model is kept linear by assuming that the number of subproblems K is fixed during optimization. The ILP model is advantageous since it represents a difficult but well-studied optimization problem. The global optimum may be found using branch and bound or cutting plane methods, and a lower bound on the ILP solution can be always obtained by solving the relaxed continuous LP (see, e.g., Papadimitriou and Steiglitz 1982). In the demonstration examples in Krishnamachari and Papalambros (1997) the partitioning models are first represented using AMPL (Fourer et al. 1993), and then solved using standard software from OSL (IBM 1990).

3 COORDINATION: PUTTING THE PIECES BACK TOGETHER

Coordination strategies based on engineering intuition are usually posed in an ad hoc manner and cannot guarantee that they converge to the same solution set as that of the undecomposed problem. Rigorous coordination strategies tend to make strong assumptions, such as linearity or convexity of all models, which may be unenforceable.

In the next subsections we will describe two approaches that attempt to strike a compromise between rigor and practical value: decomposition synthesis and sequential decomposed programming. In the third subsection we will describe the hierarchical overlapping coordination method, which has some promising characteristics for distributed product development.

Decomposition synthesis

One may pose the "system" problem so that a particular partitioning formulation is facilitated (Krishnamachari and Papalambros 1997). This idea motivates a *decomposition synthesis* strategy, where a problem is first partitioned without specifying a system objective. Then a system objective is composed using criteria from each subsystem so that the system solution will be a Pareto optimum of the subsystem solutions.

The methodology proposed in (Krishnamachari and Papalambros 1997) mainly focuses on synthesizing optimal design problems (ODP) that can be solved by a primal hierarchical decomposition method (Wagner and Papalambros 1993). A block-angular structure is first identified for the general design problem (GDP), which lacks a design objective. An ODP is then created that can be hierarchically decomposed based on this structure. Formally, a GDP is first cast into the form

$$\begin{aligned}
g_0(x_0) &\leq 0 \\
h_0(x_0) &= 0 \\
g_i(x_0, x_i) &\leq 0 \quad i=1, \dots, K \\
h_i(x_0, x_i) &= 0 \quad i=1, \dots, K \\
x &\in \mathcal{X} \subseteq \mathcal{R}^n
\end{aligned} \tag{5}$$

that has a master problem and K subproblems with a block-angular structure similar to that in Figure 1(b). This type of structure can be identified using the problem partitioning techniques described in Section 2 above. A hierarchically decomposed ODP is then synthesized by composing a weighted additive objective constructing criteria f_0 and f_i from constraints g_0 and g_i in Eq. (5), as shown in Eq. (6).

$$\begin{aligned}
&\text{minimize}_{x \in \mathcal{X} \subseteq \mathcal{R}^n} f_0(x_0, w_0) + \sum_{i=1}^K f_i(x_0, x_i, w_i) \\
&\text{subject to} \\
&g_0(x_0) \leq 0 \\
&h_0(x_0) = 0 \\
&g_i(x_0, x_i) \leq 0 \quad i=1, \dots, K \\
&h_i(x_0, x_i) = 0 \quad i=1, \dots, K
\end{aligned} \tag{6}$$

Sequential decomposed programming

In another approach, existing nonlinear programming algorithms (NLP) are modified to take advantage of problem structure without losing their basic convergence properties. This can be particularly effective in sequential optimization methods, hence the term *sequential decomposed programming*. This approach has been successfully applied to sequential quadratic programming and trust region algorithms (Nelson and Papalambros 1998).

Sequential decomposed programming accommodates the special type of structure known as hierarchically decomposable nonlinear programming and depicted in Eq. (6). Several (perhaps all) functions may depend on the vector of linking variables x_0 . Vectors x_i , $i \neq 0$ are vectors of local variables since there are K sets of functions, identified with the index i , $g_i(x_0, x_i)$, $h_i(x_0, x_i)$ and $f_i(x_0, x_i, w_i)$ that depend only on the linking variables x_0 and the respective local variables x_i . If the linking variables are held constant, then Eq. (6) can be posed as K smaller and separate nonlinear programs, called subproblems and shown in Eq. (7), that can be solved independently.

$$\begin{aligned}
&\text{minimize}_{x_i \in \mathcal{X}_i \subseteq \mathcal{R}^n} f_i(x_0, x_i, w_i) \\
&\text{subject to} \\
&g_i(x_0, x_i) \leq 0 \\
&h_i(x_0, x_i) = 0
\end{aligned} \tag{7}$$

In a typical hierarchical framework a master problem is solved in terms of x_0 . The predicted value for the optimal x_0 is passed to the subproblems (7), each of which is solved with respect to each x_i . The optimal value of x_i is returned to the master problem and the process is repeated until some convergence criterion is satisfied.

Nelson and Papalambros (1998) indicate that care must be taken with the extra step in which the linking variables are held constant in order to retain the properties of the original algorithm. The step that in the original algorithm is usually called "the direction finding problem" or "the approximate problem" is referred to here as "the coordination problem," because it represents a decision making process between the subproblems. To retain the global convergence properties of the original algorithm, coordination between subproblems is performed using an approximate problem (for example, a

quadratic program) that has the same form as in the original algorithm. That is, coordination uses all of the constraints and all of the variables in the original problem, as depicted in Figure 5. In order to retain any established local convergence properties, the subproblems are not solved independently when near a solution, i.e., the additional step is not used. Finally, the subproblems are used not only to improve their respective objective functions while maintaining or obtaining feasibility, but also to give better estimates of other quantities used in the algorithm, such as penalty parameters, Hessian estimates, and trust region radii.

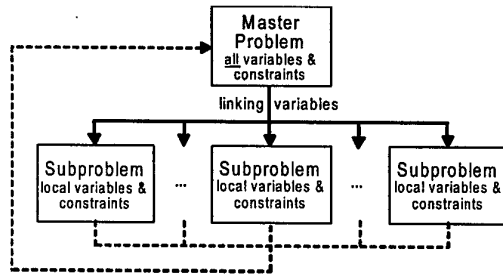


Figure 5. Sequential decomposed programming

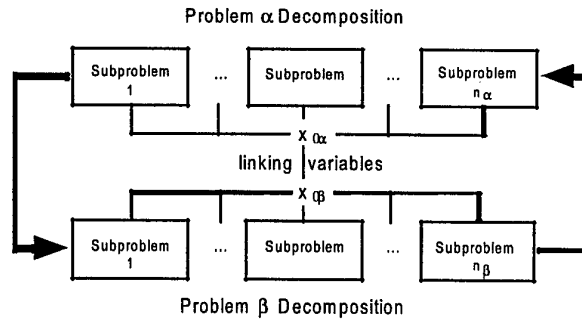


Figure 6. Hierarchical overlapping coordination

Hierarchical overlapping coordination

Hierarchical overlapping coordination (HOC) (Macko and Haimes 1978) simultaneously uses two or more design problem decompositions, each of them associated with different partitions of the design variables and constraints. Michelena, et al. (1999a) showed some extensions that can make the approach more applicable to design problems. A HOC strategy may reflect, for example, matrix-type organizations structured according to product lines or subsystems and the disciplines involved in the design process. Partitioning methods described in Section 2 can be used to generate the required problem decompositions. Coordination of the design subproblems is achieved by the exchange of information between decompositions.

Figure 6 illustrates how the HOC algorithm operates for two problem decompositions (α and β). Briefly, the algorithm can be described as follows:

- Step 1: Fix linking variables $x_{0\alpha}$, and solve Problem α by solving n_α independent subproblems, such as those in Eq. (7).
- Step 2: Fix linking variables $x_{0\beta}$ to their values determined in Step 1, and solve Problem β by solving n_β independent subproblems.
- Step 3: Go to Step 1 with the fixed values of $x_{0\alpha}$ determined in Step 2.
- Step 4: Repeat these steps until convergence is achieved.

The linking variables for one of the decompositions are fixed at values that result from the solution of a number of independent subproblems associated with the other decomposition. In general, the accumulation point achieved in Step 4 is not necessarily an optimal solution of the original problem. Convergence of the algorithm depends on the way the model decompositions interact with each other. A sufficient condition for convergence that can be computationally verified for decompositions of nonlinear convex problems was presented in Michelena et al. (1999a). This condition together with model partitioning methods can help in generating appropriate problem decompositions.

In the next two sections we will describe two applications of decomposition strategies applied to product development: setting design targets and designing collections of products simultaneously.

4 DESIGN TARGET CASCADING

An important phase in product development of complex artifacts is an early determination of key design targets for the major components or subsystems of the artifact. The process usually starts with a statement of design targets (or mission specifications) dictated by market and other top level considerations, see Figure 7. These top level targets must be "cascaded" to the rest of the system so that (i) major parts of the system can be designed independently using their own local targets, and (ii) consistency of local targets to each other and to the system as a whole is maintained. Clearly a top-down hierarchical dictation of targets will not work without some iterative "rebalancing" that will guarantee feasibility. A formalism employing a partitioning and coordination strategy becomes eminently appealing.

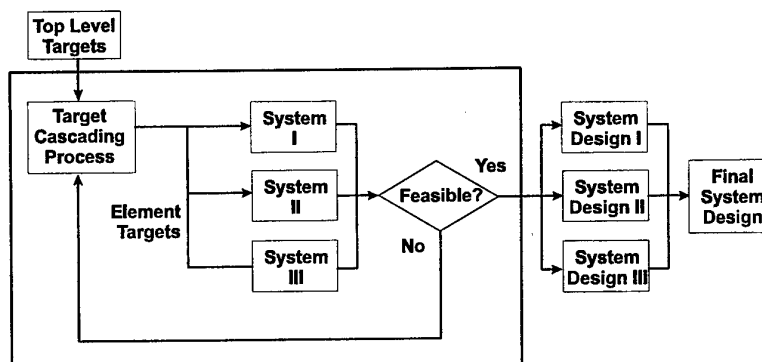


Figure 7. Target cascading as an element of a concurrent design process

A formal target cascading process can be stated as a mathematical optimization problem, assuming that appropriate models for the systems' functions are available. These models, in general, must be simple enough requiring no detailed design information but powerful enough to capture important interactions among subsystems and components, whose performance needs may be competing. The original design problem can be stated as follows:

$$\begin{aligned}
 &\text{minimize } \|T - R\| \\
 &\quad x \\
 &\text{where } R = r(x) \\
 &\text{subject to} \\
 &g_i(x) \leq 0 \quad i = 1, \dots, m_i \\
 &h_j(x) = 0 \quad j = 1, \dots, m_e \\
 &x_k^{\min} \leq x_k \leq x_k^{\max} \quad k = 1, \dots, n
 \end{aligned} \tag{8}$$

The objective is defined as the discrepancy between the target T and the response R obtained from the analysis model $r(x)$; g and h are inequality and equality design constraint vectors, and the design variable x is defined within lower and upper bounds, $x^{\min}$ and $x^{\max}$. Problem (8) can be reformulated as the multilevel problem shown in Figure 8. Since the "system level" is located in the middle of the overall hierarchy, this formulation is the most comprehensive, capturing all interactions, through linking variables, target responses from the lower level (superscript L), and target responses from the upper level (superscript U). At the system level, the problem can be stated mathematically as shown below in Eq. (9): minimize the deviations for system responses and subsystem linking variables, subject to system design constraints and tolerance constraints that coordinate subsystem responses and component design linking variables (Michelena et al. 1999b).

In the model below the objective function minimizes the discrepancy between current system level responses R_s and the targets set at the upper level R_s^U , as well as between subsystem linking variables y_{ss} and the targets set at the upper level y_{ss}^U . Therefore, R_s^U and y_{ss}^U are determined by solving an equivalent problem at the higher level. Target deviation tolerances are minimized to achieve consistent design with minimum discrepancies between the subsystem level responses R_{ss} and the target responses R_{ss}^L from the subsystem design problem, as well as between the component

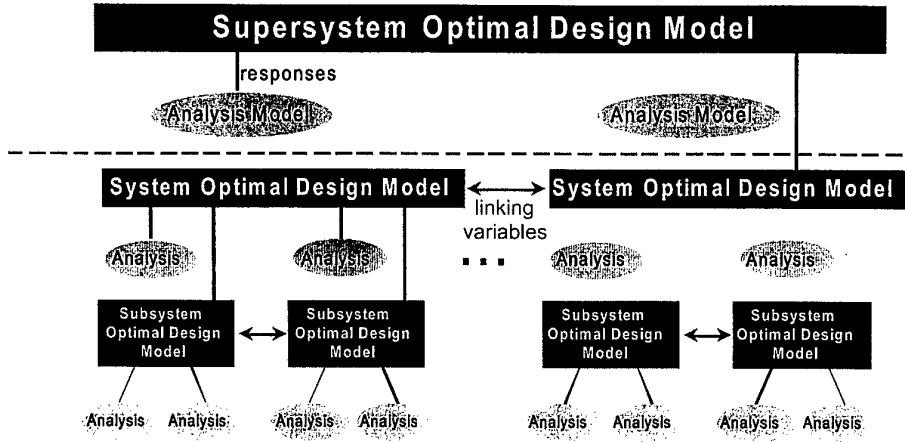


Figure 8. Multilevel representation of the target cascading problem (Kim et al. 2000)

level linking variables y_c and the target values y_c^L from the subsystem design problem. x_{ss} depicts subsystem local variables.

$$\begin{aligned}
 & \text{minimize} \quad \|R_s - R_s^U\| + \|y_{ss} - y_{ss}^U\| + \varepsilon_R + \varepsilon_y \\
 & x_{ss}', y_{ss}', y_c, R_{ss}', \varepsilon_R, \varepsilon_y \\
 & \text{where} \quad R_s = r_s \leftarrow_{ss} (R_{ss}', x_{ss}', y_{ss}') \\
 & \text{subject to} \\
 & \|R_{ss} - R_{ss}^L\| \leq \varepsilon_R, \|y_c - y_c^L\| \leq \varepsilon_y, \varepsilon_R \geq 0, \varepsilon_y \geq 0 \\
 & g_s(R_{ss}', x_{ss}', y_{ss}') \leq 0 \\
 & h_s(R_{ss}', x_{ss}', y_{ss}') = 0 \\
 & x_{ss}^{\min} \leq x_{ss}' \leq x_{ss}^{\max} \quad y_{ss}^{\min} \leq y_{ss}' \leq y_{ss}^{\max}
 \end{aligned} \tag{9}$$

A proper coordination strategy is still needed to solve this problem. This is not a trivial task. The formulation here is reminiscent of the proposed “collaborative optimization” approach (Braun and Kroo 1997). As has been shown recently (Alexandrov and Lewis 2000) collaborative optimization uses a model formulation and coordination strategy that can violate constraint qualifications a priori, so that Karush-Kuhn-Tucker optimality conditions would not be applicable. This issue remains a current research challenge.

The target cascading process has been applied successfully to some early studies in both academic and industrial settings (Michelena et al. 1999b, Kim et al. 2000). The approach is currently studied as part of the vehicle development process in automotive new product development. In a practical industry implementation a key element of success is the availability of appropriate complexity models. Even for the same product, targets change with time possibly in a period of months, so the overall process must be dynamic. One must envision a multiphase cascading process where the above formalism is applied to models of increased complexity and detail, as the product gets a more definitive characterization. In a sense one must have a cascade of models of increasing complexity in addition to a cascade of models in the hierarchy shown earlier. The challenges from both an engineering and a management viewpoint are obvious, indicative of the difficulties faced when realistic complexity is being addressed.

5 DESIGNING PRODUCT COLLECTIONS

Designing a collection of products simultaneously, as opposed to a single product, is an idea that has a variety of appeals. One attraction is the desire to avoid commitment to a single design until the later phases of the product development process when commitments are made closer to production, hence presumably with less risk. Another attraction is the desire to limit costs by imposing some commonality, thereby deriving a family of variants from a single "platform." Costs in manufacturing facilities and product development resources are reduced, and rapid exercise of product options to changing market needs is enhanced.

In general if two or more products share some parts the performance of the individual product will be likely compromised, compared to its performance without the commonality restriction. An optimization formulation can quantify this change in performance, and also compare the effect of sharing different sets of components. In a formal model, one must compare the best possible designs when sharing parts to the best possible designs when the products are not sharing parts. A single multicriteria optimization problem can capture the decision required in evaluating the entire platform (Nelson et al. 1999). Assuming that a criterion depends only on variables x_i , we pose the problem

$$\begin{aligned}
 &\text{minimize} && f_i(x_i) && i = 1 \dots p \\
 &&& g_i(x_i) \leq 0 && i = 1 \dots p \\
 &&& h_i(x_i) = 0 && i = 1 \dots p \\
 &\text{subject to} && x_{i,k_1} = x_{j,k_2} && (k_1, k_2) \in P_{ij} \\
 &&& && i, j = 1 \dots p \\
 &&& && i < j
 \end{aligned} \tag{10}$$

Let P_{ij} be a set of index pairs used to represent a set of equality constraints. If products i and j share some parts, then P_{ij} contains the index pairs of the design variables describing the common parts. Variants in a platform configuration is collectively defined by a distinct set of index pairs. To compare two different product platforms, solutions are compared with different sets of index pairs, say $\{P_{ij}\}$ and $\{Q_{ij}\}$. For a specific set of P_{ij} the solutions of the problem above are a Pareto set. Let different superscripts represent optimal values from different platform configurations. For two products A and B in a platform, a superscript *circle* represents optimal quantities for the null platform (f_A° and f_B°), i.e., independently designed products. A superscript *bullet* ($f_A^\bullet$

and $f_B^\bullet$) represents optimal quantities for the platform with the common parts. The individual minima f_i° of Eq. (10) are defined as the extreme values of the Pareto set, see Figure 9.

These are the solutions to Eq. (10) with only one of the scalar functions (f_A or f_B) used as an objective. The utopia point ($f_A^\bullet, f_B^\bullet$), is likely not as good as the point representing separate designs (f_A°, f_B°). The additional equality constraints of the product platform imply that the feasible space is smaller so the designer of a product platform should expect to give up some acceptable amount of performance (Nelson et al. 1999).

In a design strategy each possible pair of products that can share parts is identified. The individual optima for each configuration is computed to decide whether a trade-off study should be pursued. If so, the relevant multicriteria problem of Eq. (10) is solved to calculate the Pareto set. The design that offers the best value for all appropriate products while still allowing for the benefits of having a flexible product platform is then chosen. This strategy has been applied to vehicle platform design (Fellini et al. 2000) as a first step towards a rigorous portfolio development strategy. The point here, however, is that the problem in Eq. (10) is generally difficult to solve as the size of the problem increases, and in practice a hierarchical decomposition strategy must be employed, similar to what was discussed in previous sections.

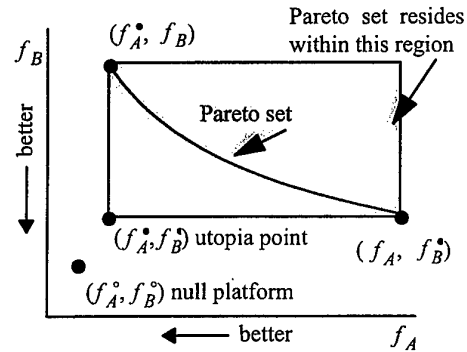


Figure 9. Pareto set for a platform with two products A and B sharing one or more components.

6 SYSTEM-LEVEL DESIGN AND CONTROL

We now turn our attention to “smart” artifacts, another example of system design where the ideas of the earlier sections have bearing. Design of a modern artifact is likely to include a control function that allows the artifact to adapt to changing environmental conditions. Traditionally, “design” and “control” have been treated as separate engineering functions. A little reflection reveals that designing the embodiment of an artifact and designing its controller are both design functions. Indeed, “control” means defining the control function and then designing the embodiment of a physical controller. A system-level design and control requires looking at these design activities in a simultaneous manner.

$$\text{minimize } f = a_1 d$$

variable: d

coupling parameter: b

simple parameters: $a = \{a_1, a_2, a_3\} \geq 0$

subject to:

$$g: a_2 b + a_3 - d \leq 0$$

output: $v = d^*$

optimum: $d^* = a_2 b + a_3$

(a) Optimal design problem

$$\begin{aligned} \text{minimize } J &= \int_0^{t_f} (x^T u_1 x + z^T u_2 z) dt \\ &= \frac{u_5^2 (u_1 + u_2 p^2)}{2(v - u_3 p)} [e^{2(v - u_3 p)t_f} - 1] \end{aligned}$$

variable: p

coupling parameter: v

simple parameters: $u = \{u_1, u_2, u_3, u_4, u_5, t_f\} \geq 0$

subject to:

$$\dot{x}(t) = vx + u_3 z = (v - u_3 p)x \Rightarrow x(t) = e^{(v - u_3 p)t}$$

$$y = u_4 x$$

$$z = -px$$

$$x(t=0) = u_5$$

$$l: -p \leq 0$$

$$\text{output: } b = x(t=1) = e^{(v - u_3 p)}$$

$$\text{optimum: } p^* = (u_1 v + \sqrt{(u_1^2 v^2 + u_1 u_2 u_3^2)}) / u_3^2$$

(b) Optimal gains problem

$$\text{minimize } \phi = w_1 f + w_2 J$$

where: $f = a_1 d$

$$\begin{aligned} \text{and } J &= \int_0^{t_f} (x^T u_1 x + z^T u_2 z) dt \\ &= \frac{u_5^2 (u_1 + u_2 p^2)}{2(v - u_3 p)} [e^{2(v - u_3 p)t_f} - 1] \end{aligned}$$

variables: d, p

coupling quantities: b, v

simple parameters: $a = \{a_1, a_2, a_3\} \geq 0$

$u = \{u_1, u_2, u_3, u_4, u_5, t_f\} \geq 0$

subject to:

$$\dot{x}(t) = vx + u_3 z = (v - u_3 p)x \Rightarrow x(t) = e^{(v - u_3 p)t}$$

$$y = u_4 x$$

$$z = -px$$

$$x(t=0) = u_5$$

$$l: -p \leq 0$$

$$g: a_2 b + a_3 - d \leq 0$$

$$\beta: e^{(v - u_3 p)} - b \leq 0$$

$$\gamma: d - v \leq 0$$

(c) Combined design and control problem

Figure 10. Model statements for the design, control and combined problem in the example in (Reyer and Papalambros, 2000)

A sequential design approach is often natural, as each task may be itself difficult and coupling them together could make the design task intractable. Problems with nonlinearities in the control tasks have particular difficulties. Decision criteria for each design task are often different and competing. Nevertheless, the initial design of the artifact affects its control characteristics, so if adaptive performance is important or expensive, control-related design criteria must be incorporated early in the design process. As products become increasingly complex, examining the tight coupling between embodiment design and control design becomes a critical step in the design process.

Designing a controller is generally a difficult *configuration design* problem. If a given configuration is assumed, an *Optimal Gain Problem* can be formulated that is time independent. In this context, the problem is broached in Reyer and Papalambros (1999). Here we can illustrate the main ideas using a simplified example from Reyer and Papalambros (2000). The term "simple" below means that the quantity in question is local to the problem (design or control), while the term "coupling" means that the relevant quantity appears in the other problem as well.

The design problem can be defined as shown in Figure 10(a). The model has one design variable d , one input coupling parameter b , several simple parameters a , and one output coupling parameter v . The problem is linear with a single unique global optimal solution which is constraint bound (i.e., fully determined by active constraints). The control problem is an optimal gains one with finite-dimensional variables as shown in Figure 10(b). The model has one control variable p , one input coupling parameter v , several simple parameters u , and an output coupling parameter b . For this simple problem the control strategy is equivalent to both a proportional feedback and an LQR controller. The control optimum is based on the LQR solution and has an analytical form. The system-level problem is defined with an objective that is the weighted sum of the design and control objectives, representing the Pareto solution of the two problems, Figure 10(c). The combined model has two variables d and p , two coupling quantities b and v , and two sets of simple parameters a and u . The model includes four inequality constraints in addition to the state constraints: one from the control side l ; one from the design side g , and two for coupling β and γ . The coupling constraints are *directed* (Papalambros and Wilde 1988, 2000)—a critical issue in understanding how the different strategies work with more details found in the cited reference.

The decomposition strategy ideas explored earlier can be used here to put intuitive ideas on a rigorous foundation. The partitioned problem has two linking variables, the coupling variables b and v , and two subproblems, one for design and one for control, Figure 11 and Figure 12. In the hierarchical representation the master problem has two variables, b and v , three constraints, g , β and γ , and the weighted combined objective. The design subproblem has one variable d , two constraints, g and γ , and the design objective f . Note that in the design subproblem, the control part of the system objective J , and the constraints l and β , are eliminated, since the design variable cannot affect them. Similarly, the control subproblem has one variable p , two constraints l and β , and the control objective J .

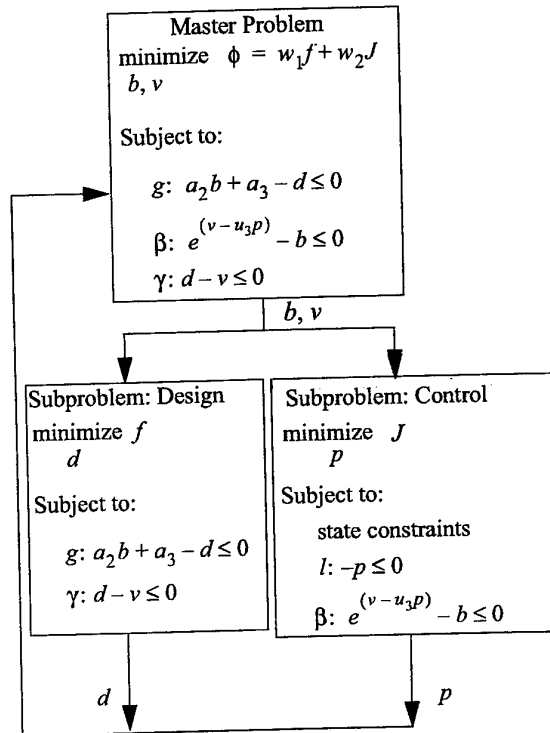


Figure 11. Partitioned concurrent strategy

	b	d	p	v
f		■		
J			■	■
g	■	■		
l			■	
β	■		■	■
γ		■		■

	b	v	d	p
f			■	
J		■		■
g	■		■	
γ		■	■	
β	■	■		■
l				■

Figure 12. Functional dependency tables for non-partitioned and partitioned problem

This partitioned problem must be solved with an appropriate coordination strategy. As explored in the cited reference, traditional design-control methods in general will not find the system optimum, either because they use no strategy equivalent to coordination or because the partition and the strategy employed miss elements of the combined problem shown above.

Even this simple example appears rather devious when addressing it from a system perspective. Problems of realistic complexity quickly become challenging. A deeper study of proper partitioning and coordination methods derived from the optimality conditions of the combined problem would be illuminating and hopefully lead to practical but rigorous system solution strategies.

7 CONCLUDING REMARKS

Partitioning can be now performed rigorously using the techniques described here. However, the optimal partitioning problem must be formulated having some intuition about the specific problem. Many different optimal partitions can be derived, some of which may be meaningful and some not. Therefore, the designer must still exercise judgment. In coordination the challenge is to prove "convergence" of the strategy. Convergence formally would mean that the non-partitioned problem and the partitioned one solved with coordination have the same solution set. This is often hard or impossible to prove for practical nonlinear problems. Alternatively, a problem may be defined in a meaningful *partitioned* form directly, as in the case of decomposition synthesis described above, where the system solution is by definition a Pareto solution of the subsystem problems. As complexity increases, the ideas described above can and should be extended to multilevel problems. The main extra difficulty there is controlling the computational cost.

Another challenge, occasionally insurmountable, is the presence of simulations in the models, namely, large, expensive implicit models, whose mathematical behavior as function generators is generally unknown, and frequently noisy. Dealing with noisy, expensive and likely non-convex functions is challenging, and particularly so for decomposition methods. Some discussion on this issues can be found in Sasena et al. (2000) and Fellini et al. (2000).

Finally, one should readily confess that decomposition strategies as described in the present context almost always incur additional computational cost, when compared with solving the undecomposed problem. Why then should one consider them at all? First, these methods may be the only hope to solve a problem whose complexity defies the alternative. But since coordination strategies may be ad hoc, even then a solution may be not known with certainty. The second reason for using a decomposition approach is simply to break down the problem to subproblems of sufficient size that a relatively experienced team of designers will have enough intuition to sanction the numerical results. From an optimization perspective, of course, the challenge is to develop such methods that are ever more robust and address an ever broader range of design problems.

ACKNOWLEDGEMENTS

This review article was composed based on joint work with a number of collaborators at the University of Michigan, including R. Fellini, H. M. Kim, R. Krishnamachari, S. Nelson, M. Parkinson, M. Sasena, and J. Reyer. The work described has been supported by a variety of sponsors, including the Automotive Research Center at the University of Michigan under the auspices of the US Army TACOM, Ford Motor Co., General Motors Corp., and the US Department of Energy. The authors gratefully acknowledge the support of the sponsors and the contributions of our colleagues.

REFERENCES

- Alexandrov, N. and Lewis, R. M., 2000. "Analytical and Computational Aspects of Collaborative Optimization," NASA/TM-2000-210104, Langley Research Center, Hampton, VA.
- Braun, R. D. and Kroo, I. M., 1997, "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," in *Multidisciplinary Design Optimization: State-of-the-Art*, N. Alexandrov and M. Hussaini (eds.), SIAM, pp. 98-116
- Fellini, R., Papalambros, P. Y., and Weber, T., 2000. "A Decomposition Strategy for Product Platform Design with Application to Automotive Powertrains," *AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization*, Long Beach, CA.
- Fourer, R., Gay, D., and Kernighan, B., 1993. *AMPL: A Modeling Language for Mathematical Programming*, The Scien-

- tific Press, South San Francisco.
- IBM, 1990. *Optimization Subroutine Library Guide and Reference*, IBM Corporation, New York.
- Kim, H. M., Michelena, N., Papalambros, P. Y., and Jiang, T., 2000. "Target Cascading in Optimal System Design," *ASME Design Automation Conference*, Baltimore, MD.
- Krishnamachari, R. and Papalambros, P. Y., 1997. "Optimal Hierarchical Decomposition Synthesis Using Integer Programming," *ASME J. of Mech. Design*, Vol. 119, No. 4, pp. 440–447.
- Macko, D. and Haimes, Y., 1978. "Overlapping Coordination of Hierarchical Structures," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-8, No. 10, pp. 745–751.
- Michelena, N. and Papalambros, P. Y., 1995. "A Network Reliability Approach to Optimal Decomposition of Design Problems," *ASME J. of Mech. Design*, Vol. 117, No. 3, pp. 433–440.
- Michelena, N. and Papalambros, P. Y., 1997. "A Hypergraph Framework to Optimal Model-Based Decomposition of Design Problems," *Comp. Optim. and Appl.*, Vol. 8, No. 2, pp. 173–196.
- Michelena, N., Papalambros, P. Y., Park, H. A., and Kulkarni, D., 1999a. "Hierarchical Overlapping Coordination for Large-Scale Optimization by Decomposition," *AIAA Journal*, Vol. 37, No. 7, pp. 890–896.
- Michelena, N., Kim, H. M., and Papalambros, P. Y., 1999b. "A System Partitioning and Optimization Approach to Target Cascading," *Proc. 12th Int. Conf. on Engineering Design*, U. Lindemann et al. (eds.), Vol. 2, pp. 1109–1112, Munich.
- Nelson, S. A. and Papalambros, P. Y., 1998. "A Modified Trust Region Algorithm for Hierarchical NLP," *Structural Optimization*, Vol. 16, pp. 19–28.
- Nelson, S. A., Parkinson, M. B., and Papalambros, P. Y., 1999. "Multicriteria Optimization in Product Platform Design," *ASME Design Automation Conference*, Las Vegas, NV.
- Papadimitriou, C. and Steiglitz, K., 1982. *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Papalambros, P. Y. and Wilde, D. J., 1988, 2000. *Principles of Optimal Design: Modeling and Computation*. 1st Ed., 2d Ed.; Cambridge University Press, New York.
- Reyer, J. A. and Papalambros, P. Y., 1999. "Optimal Design and Control of an Electric DC Motor," *ASME Design Automation Conference*, Las Vegas, NV.
- Reyer, J. A. and Papalambros, P. Y., 2000. "An Investigation into Modeling and Solution Strategies for Optimal Design and Control," *ASME Design Automation Conference*, Baltimore, MD.
- Sasena, M. J., Papalambros, P. Y., and Goovaerts, P., 2000. "Metamodeling Sampling Criteria in a Global Optimization Framework," *AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization*, Long Beach, CA.
- Wagner, T. C. and Papalambros, P. Y., 1993. "A General Framework for Decomposition Analysis in Optimal Design," *ASME Design Automation Conference*, DE-Vol. 65–2, New York, pp. 315–25.

DERIVATIVES FOR KINEMATIC AND DYNAMIC ANALYSIS AND OPTIMIZATION

Radu Serban
Department of Mechanical and Environmental Engineering
University of California-Santa Barbara
Santa Barbara, California 93106

Edward J. Haug
Department of Mechanical Engineering
The University of Iowa
Iowa City, Iowa 52242

ABSTRACT

Analytical formulas for kinematic derivatives needed in multibody system analysis and optimization are derived. A broad spectrum of problems, including implicit numerical integration, dynamic sensitivity analysis, and kinematic workspace analysis, require evaluation of at least three derivatives of kinematic constraint functions. In the setting of a formulation based on Cartesian generalized coordinates with Euler parameters for orientation, basic identities are developed that enable practical and efficient computation of all derivatives required for a large number of multibody mechanical system analyses. Efficiency of computation using the expressions derived has shown significant computational advantage in each of the analysis and optimization applications addressed.

1 INTRODUCTION

Three different areas of multibody system analysis are considered. The common requirement for numerical methods used in these types of analysis is the availability of higher order derivatives of kinematic constraint equations. The three types of analysis under consideration are as follows:

- (1) implicit numerical integration of the differential-algebraic equations (DAE) of motion for simulation of stiff mechanical systems
- (2) sensitivity analysis for design optimization, parameter estimation, and model correlation
- (3) kinematic workspace analysis of mechanisms

Next, each of these problems is briefly described and required derivatives for numerical solution are identified.

1.1 Kinematic and Dynamic Analysis

For a system described by generalized coordinates $\mathbf{q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]^T$, the equations of motion can be written in descriptor form as (Haug, 1989)

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q})\lambda = \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (1)$$

associated with the algebraic constraint equations

$$\Phi(\mathbf{q}) \equiv [\Phi_1(\mathbf{q}), \Phi_2(\mathbf{q}), \dots, \Phi_m(\mathbf{q})]^T = \mathbf{0} \quad (2)$$

The position constraints of Eq. 2 can be differentiated with respect to time to yield the kinematic velocity and acceleration constraint equations,

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (3)$$

$$\Phi_q(\mathbf{q})\ddot{\mathbf{q}} = -(\Phi_{qq}(\mathbf{q})\dot{\mathbf{q}})\dot{\mathbf{q}} \equiv \boldsymbol{\tau}(\mathbf{q}, \dot{\mathbf{q}}) \quad (4)$$

Here, subscript denotes partial derivative; e.g.,

$$\Phi_q = \left[\frac{\partial \Phi_i}{\partial q_j} \right]_{m \times n}$$

There are many methods for integrating the DAE of Eqs. 1 through 4. Hairer and Wanner (1996) present a comprehensive treatment of such methods. Numerically stiff systems, which often arise due to bushings and stiff compliant elements, require the use of implicit integration methods. Only these implicit methods exhibit the required stability to deal with such systems (Haug *et al.*, 1997a, 1997b; Negrut, 1998). Regardless of which implicit integrator is used, derivatives of all terms in Eqs. 1 through 4 with respect to both generalized coordinates and velocities are required. The derivatives to be evaluated are as follows:

$$\begin{aligned} & (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}})_q, (\Phi_q^T(\mathbf{q})\lambda)_q, (\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t))_q, (\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t))_{\dot{q}} \\ & (\Phi_q(\mathbf{q})\dot{\mathbf{q}})_q, (\Phi_q(\mathbf{q})\ddot{\mathbf{q}})_q, ((\Phi_{qq}(\mathbf{q})\dot{\mathbf{q}})\dot{\mathbf{q}})_q, ((\Phi_{qq}(\mathbf{q})\dot{\mathbf{q}})\dot{\mathbf{q}})_{\dot{q}} \end{aligned} \quad (5)$$

1.2 Dynamic Sensitivity Analysis

Numerous problems in multibody mechanical system analysis can be formulated as optimization problems with respect to dynamic behavior of the system. A nonlinear programming problem must then be solved. Optimization algorithms are most efficient if accurate derivative information is provided. Dynamic design sensitivity analysis of multibody systems thus represents the link between optimization tools and modelling and simulation tools.

Two methods are commonly used to generate sensitivity information. The most straightforward approach is the direct differentiation method, which differentiates the DAE of motion with respect to parameters (Krishnaswami *et al.*, 1983; Chang and Nikravesh, 1985; Haug, 1987). The second

method is the adjoint variable method, in which sensitivities are obtained as solutions of DAE that are adjoint to the equations of motion. The adjoint variable method has been extensively applied in both optimal control and optimal design (Haug and Ehle, 1982; Haug, 1987; Bestle and Eberhard, 1992; Bestle and Seybold, 1992). Both methods require derivatives of terms in the DAE of motion with respect to parameters. Use of the chain rule of differentiation in both methods reduces this problem to finding derivatives of these terms with respect to generalized states and velocities. Therefore, all derivatives noted above for implicit integration of the equations of motion are needed to generate the sensitivity equations.

1.3 Workspace analysis

The problem of defining achievable sets in the output space of a mechanism is known as kinematic workspace analysis. Domains of operation (accessible output, operational envelope), domains of interference, and domains of mobility can be treated in a unified fashion (Haug *et al.*, 1996). Most research has focused on describing mechanism workspaces by defining their boundaries. Analytical conditions for workspace boundaries in special mechanisms have been used by a number of authors (Tsai and Soni, 1981; Yang and Lee, 1983). For general mechanisms, Litvin (1980) used the implicit function theorem to obtain criteria for workspaces, relating them to singular configurations of mechanisms. Analytical criteria and numerical methods for mapping boundaries of workspaces, using Jacobian matrix row rank deficiency conditions, have been developed by Jo and Haug (1989), Wang and Wu (1993), Haug *et al.* (1996), and Adkins (1996). More recently, Snyman *et al.* (Snyman, du Plessis and Duffy, 1998; du Plessis, and Snyman, 1999; Hay and Snyman, 1999) have developed optimization methods for mapping boundaries of workspaces. In all these methods, the following quantities related to the kinematics of the underlying mechanism must be evaluated:

$$\left(\Phi_q(q) \gamma \right)_q, \left(\Phi_q(q)^T \beta \right)_q, \left(\left(\Phi_q(q)^T \beta \right)_q \alpha \right)_\beta, \left(\left(\Phi_q(q)^T \beta \right)_q \alpha \right)_q \quad (6)$$

where α , β , and γ are vectors of appropriate dimension.

1.4 Equations of Motion of Multibody Dynamics with Cartesian Coordinates

In Cartesian coordinates with Euler parameters (Haug, 1989) $\mathbf{p}_i = [e_{i0}, e_{i1}, e_{i2}, e_{i3}]^T \equiv [e_{i0}, \mathbf{e}_i]^T$ for orientation, the equations of motion of Eqs. 1 and 2 are

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}^T \mathbf{J}' \mathbf{G} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{r}} \\ \ddot{\mathbf{p}} \end{bmatrix} + \begin{bmatrix} \Phi_r^T & \mathbf{0} \\ \Phi_p^T & \Phi_p^{pT} \end{bmatrix} \lambda = \begin{bmatrix} \mathbf{F}^A \\ 2\mathbf{G}^T \mathbf{n}^A + 8\mathbf{G}^T \mathbf{J}' \mathbf{G} \mathbf{p} \end{bmatrix} \quad (7)$$

where $\mathbf{M} = \text{diag}\{m_i \mathbf{I}_3\}$ is a block-diagonal matrix containing the masses of the bodies,

$\mathbf{J}' = \text{diag}\{\mathbf{J}'_i\}$ is the block-diagonal inertia matrix of the system, $\mathbf{r} = [\mathbf{r}_1^T, \dots, \mathbf{r}_{n_b}^T]^T$ is the

vector of body positions, $\mathbf{p} = [\mathbf{p}_1^T, \dots, \mathbf{p}_{n_b}^T]^T$ is the vector of Euler parameters,
 $\mathbf{G} = [-\mathbf{e}, -\tilde{\mathbf{e}} + e_0 \mathbf{I}]$,

$$\tilde{\mathbf{e}} \equiv \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$$

and the kinematic and Euler parameter normalization constraints are

$$\Phi(\mathbf{r}, \mathbf{p}) = \mathbf{0} \quad (8)$$

$$\Phi^p(\mathbf{p}) \equiv [\mathbf{p}_1^T \mathbf{p}_1, \dots, \mathbf{p}_{n_b}^T \mathbf{p}_{n_b}]^T = \mathbf{0} \quad (9)$$

As shown in this paper, a relatively small number of basic identities enable computation of all required derivatives of kinematic terms arising in each of the analyses discussed above. Explicit forms for derivatives of basic quantities are obtained. Therefore, this approach is expected to be more efficient than using automatic differentiation methods (Bischof et al., 1996). When evaluating higher order derivatives, automatic differentiation codes differentiate a lower order derivative and thus cannot take advantage of simplifying identities that may exist, such as Euler parameter normalization constraints.

2 BASIC IDENTITIES

Derivatives of expressions used as basic building blocks in generating required kinematic derivatives are obtained in this section. The identities derived are subsequently applied to obtain higher order derivatives of constraint equations. When using Cartesian coordinates with Euler parameters to model a multibody system, the generalized inertia matrix has a simple block-diagonal form. Obtaining derivatives of terms involving this matrix is therefore straightforward.

In the Cartesian coordinate formulation, dependencies on states, velocities, and parameters that appear in the coefficients of the DAE of motion are well defined. The most complicated terms are those involving the Jacobian of the kinematic constraint equations. Derivatives with respect to Euler parameters are the principal focus, since dependency on position vectors is straightforward.

A quantity that arises in many terms of interest is the transformation of a vector that is defined in a Cartesian body-fixed reference frame to the global Cartesian reference frame. For an individual body (body number subscript suppressed), the orthogonal orientation transformation matrix is (Haug, 1989)

$$\mathbf{A}(\mathbf{p}) = (2e_0^2 - \mathbf{I}) + 2(\mathbf{e}\mathbf{e}^T + e_0 \tilde{\mathbf{e}}) \quad (10)$$

where $\mathbf{p} \equiv [\mathbf{e}_0, \mathbf{e}^T]$, $\mathbf{e}^T \equiv [e_1, e_2, e_3]$, and $\mathbf{p}^T \mathbf{p} = 1$ are Euler parameters defining orientation. A body-fixed vector $\mathbf{s}'$ that is represented in the local reference frame is represented in the global reference frame as

$$\mathbf{s} = \mathbf{A}(\mathbf{p})\mathbf{s}' = (2e_0^2 - \mathbf{I})\mathbf{s}' + 2(\mathbf{e}\mathbf{e}^T \mathbf{s}' + e_0 \tilde{\mathbf{e}}\mathbf{s}') \quad (11)$$

The derivative of this quantity with respect to the vector of Euler parameters is (Serban, 1998, Serban and Haug, 1998)

$$\frac{\partial}{\partial \mathbf{p}}(\mathbf{A}(\mathbf{p})\mathbf{s}') = \mathbf{B}(\mathbf{p}, \mathbf{s}') \equiv 2(\mathbf{s}'\mathbf{p}^T + \hat{\mathbf{B}}(\mathbf{p}, \mathbf{s}')) \quad (12)$$

where

$$\hat{\mathbf{B}}(\mathbf{p}, \mathbf{s}') = [(\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}})\mathbf{s}'; \mathbf{e}\mathbf{s}'^T - (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}})\tilde{\mathbf{s}}'] \quad (13)$$

Note that the orientation matrix of Eq. 10 is quadratic in $\mathbf{p}$, so it should be expected that the derivative of $\mathbf{s}$ with respect to $\mathbf{p}$ should be linear in $\mathbf{p}$. As seen in Eqs. 12 and 13, this is indeed the case. Moreover, these matrices have a commutativity property involving two 4-dimensional vectors, $\mathbf{p}_i$ and $\mathbf{p}_j$, namely (Serban, 1998, Serban and Haug, 1998),

$$\hat{\mathbf{B}}(\mathbf{p}_i, \mathbf{a}')\mathbf{p}_j = \hat{\mathbf{B}}(\mathbf{p}_j, \mathbf{a}')\mathbf{p}_i \quad (14)$$

$$\mathbf{B}(\mathbf{p}_i, \mathbf{a}')\mathbf{p}_j = \mathbf{B}(\mathbf{p}_j, \mathbf{a}')\mathbf{p}_i \quad (15)$$

As a direct consequence of these properties, the following result is obtained:

$$((\mathbf{A}(\mathbf{p}_i)\mathbf{s}')_{\mathbf{p}_j})_{\mathbf{p}_i} = \frac{\partial}{\partial \mathbf{p}_i}(\mathbf{B}(\mathbf{p}_i, \mathbf{a}')\mathbf{p}_j) = \frac{\partial}{\partial \mathbf{p}_i}(\mathbf{B}(\mathbf{p}_j, \mathbf{a}')\mathbf{p}_i) = \mathbf{B}(\mathbf{p}_j, \mathbf{a}') \quad (16)$$

Calculation of the derivative of $\mathbf{B}(\mathbf{p}, \mathbf{a}')^T \beta$ with respect to $\mathbf{p}$, for an individual body in the system, is also required. From Eqs. 13 and 12,

$$\frac{\partial}{\partial \mathbf{p}}(\hat{\mathbf{B}}(\mathbf{p}, \mathbf{a}')^T \beta) = \begin{bmatrix} \mathbf{a}'^T \beta & \mathbf{a}'^T \tilde{\beta} \\ \tilde{\mathbf{a}}'\beta & \mathbf{a}'\beta^T + \tilde{\mathbf{a}}'\tilde{\beta} \end{bmatrix} \quad (17)$$

$$\begin{aligned}\frac{\partial}{\partial \mathbf{p}}(\mathbf{B}(\mathbf{p}, \mathbf{a}')^T \beta) &= 2 \left\{ \frac{\partial}{\partial \mathbf{p}}(\mathbf{a}'^T \beta \mathbf{p}) + \frac{\partial}{\partial \mathbf{p}}(\hat{\mathbf{B}}(\mathbf{p}_i, \mathbf{a}')^T \beta) \right\} = 2 \begin{bmatrix} 2\mathbf{a}'^T \beta & \mathbf{a}'^T \tilde{\beta} \\ \tilde{\mathbf{a}}' \beta & \tilde{\mathbf{a}}' \tilde{\beta} + \mathbf{a}' \beta^T + \mathbf{a}'^T \beta \mathbf{I} \end{bmatrix} \\ &= 2 \begin{bmatrix} 2\mathbf{a}'^T \beta & \mathbf{a}'^T \tilde{\beta} \\ \tilde{\mathbf{a}}' \beta & \tilde{\mathbf{a}}' \tilde{\beta} + \tilde{\beta} \tilde{\mathbf{a}}' + 2\mathbf{a}'^T \beta \mathbf{I} \end{bmatrix} \equiv \mathbf{D}(\mathbf{a}', \beta)\end{aligned}\quad (18)$$

From the final form of Eq. 18, it can be verified that $\mathbf{D}$ is symmetric, which significantly enhances simplification of the derivatives summarized in Eqs. 5 and 6.

Finally, derivatives of $\mathbf{D}(\mathbf{a}', \beta)\alpha$ will be required. Expanding this expression, using this second form on the right of Eq. 18, and manipulating,

$$\mathbf{D}(\mathbf{a}', \beta)\alpha = 2 \begin{bmatrix} \alpha_1 \mathbf{a}'^T - \mathbf{a}'^T \alpha_2 \\ \alpha_1 \tilde{\mathbf{a}}' - \tilde{\mathbf{a}}' \alpha_2 + \alpha_2 \mathbf{a}'^T + \mathbf{a}' \alpha_2^T \end{bmatrix} \beta$$

The derivative of $\mathbf{D}(\mathbf{a}', \beta)\alpha$ with respect to β is thus

$$(\mathbf{D}(\mathbf{a}', \beta)\alpha)_\beta = 2 \begin{bmatrix} \alpha_1 \mathbf{a}'^T - \mathbf{a}'^T \alpha_2 \\ \alpha_1 \tilde{\mathbf{a}}' - \tilde{\mathbf{a}}' \alpha_2 + \alpha_2 \mathbf{a}'^T + \mathbf{a}' \alpha_2^T \end{bmatrix} \equiv \mathbf{E}(\mathbf{a}', \alpha)\quad (19)$$

3 KINEMATIC CONSTRAINT DIFFERENTIATION

A basic kinematic constraint between bodies numbered i and j is that points located on the respective bodies must remain in common, forming a spherical constraint. The resulting constraint equation and its Jacobian, which involves only generalized coordinates associated with bodies i and j are

$$\Phi^s = \mathbf{r}_j + \mathbf{s}_j - \mathbf{r}_i - \mathbf{s}_i = \mathbf{r}_j + \mathbf{A}(\mathbf{p}_j)\mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}(\mathbf{p}_i)\mathbf{s}'_i = \mathbf{0}\quad (20)$$

$$\Phi_q^s = [-\mathbf{I}, -\mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i), \mathbf{I}, \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)]\quad (21)$$

Expanding the product of the Jacobian and a vector $\alpha = [\alpha^1, \alpha^2, \alpha^3, \alpha^4]^T$, using Eq. 15, and taking the Jacobian of the resulting expression yields

$$\Phi_q^s \alpha = -\alpha^1 - \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)\alpha^2 + \alpha^3 + \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)\alpha^4 = -\alpha^1 - \mathbf{B}(\alpha^2, \mathbf{s}'_i)\mathbf{p}_i + \alpha^3 + \mathbf{B}(\alpha^4, \mathbf{s}'_j)\mathbf{p}_j$$

$$(\Phi_q^s \alpha)_q = [\mathbf{0}, -\mathbf{B}(\alpha^2, \mathbf{s}'_i), \mathbf{0}, \mathbf{B}(\alpha^4, \mathbf{s}'_j)]\quad (22)$$

Since this expression does not depend on $\mathbf{q}$,

$$\left((\Phi_q^s \alpha)_q \beta \right)_q = 0 \quad (23)$$

For $\alpha = \beta = \dot{\mathbf{q}}$, the derivative of $(\Phi_q^s \dot{\mathbf{q}})_q \dot{\mathbf{q}}$, which is called for in Eq. 5, can be developed by first expanding the expression, using Eq. 22 to obtain

$$(\Phi_q^s \dot{\mathbf{q}})_q \dot{\mathbf{q}} = -\mathbf{B}(\dot{\mathbf{p}}_i, \mathbf{s}'_i) \dot{\mathbf{p}}_i + \mathbf{B}(\dot{\mathbf{p}}_j, \mathbf{s}'_j) \dot{\mathbf{p}}_j \quad (24)$$

Using the commutativity property of Eq. 15, one of the derivatives required in Eq. 5 is simply

$$\left((\Phi_q^s \dot{\mathbf{q}})_q \dot{\mathbf{q}} \right)_q = [\mathbf{0}, -2\mathbf{B}(\dot{\mathbf{p}}_i, \mathbf{s}'_i), \mathbf{0}, 2\mathbf{B}(\dot{\mathbf{p}}_j, \mathbf{s}'_j)] \quad (25)$$

The derivative of $\Phi_q^T(\mathbf{q})\lambda$ with respect to $\mathbf{q}$ is required in both Eqs. 5 and 6 is. For the spherical constraint,

$$\Phi_q^{s^T}(\mathbf{q})\lambda = \begin{bmatrix} -\lambda \\ -\mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i)\lambda \\ \lambda \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j)\lambda \end{bmatrix} \quad (26)$$

Using Eq. 18,

$$\left(\Phi_q^{s^T}(\mathbf{q})\lambda \right)_q = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}(\mathbf{s}'_i, \lambda) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}(\mathbf{s}'_j, \lambda) \end{bmatrix} \quad (27)$$

Since this expression does not depend on $\mathbf{q}$, the last term in Eq. 6 is

$$\left(\left(\Phi_q^{s^T}(\mathbf{q})\beta \right)_q \alpha \right)_q = 0 \quad (28)$$

Using Eqs. 27 and 19, the third term in Eq. 6 is

$$\left(\left(\Phi_q^{s^T}(\mathbf{q})\beta \right)_q \alpha \right)_\beta = \begin{bmatrix} \mathbf{0} \\ -\mathbf{D}(\mathbf{s}'_i, \beta)\alpha^2 \\ \mathbf{0} \\ \mathbf{D}(\mathbf{s}'_j, \beta)\alpha^4 \end{bmatrix}_\beta = \begin{bmatrix} \mathbf{0} \\ -\mathbf{E}(\mathbf{s}'_i, \alpha^2) \\ \mathbf{0} \\ \mathbf{E}(\mathbf{s}'_j, \alpha^4) \end{bmatrix} \quad (29)$$

This completes the calculation of all terms in Eqs. 5 and 6 required for kinematic, sensitivity, and workspace analysis, for the spherical joint. The derivation of similar derivatives for each additional kinematic constraint can be carried out in an analogous manner (Serban, 1998, Serban and Haug, 1998).

4 CONCLUSIONS

Results presented in this paper enable accurate and efficient computation of higher order derivatives required in multibody analysis and optimization; including implicit numerical integration of the DAE of motion, workspace analysis, dynamic sensitivity analysis of mechanical systems, design optimization, and parameter estimation. A formulation to generate three derivatives of terms involved in the DAE of motion required in each of these analyses is presented. Derivative computations are developed in the Cartesian coordinate formulation with Euler parameters as orientation parameters. Similar methods can be applied to generate kinematic and kinetic derivatives when using joint (or relative) coordinates.

ACKNOWLEDGMENT

This research was supported by the US Army Tank-Automotive Command (TACOM), through the Automotive Research Center (Department of Defense contract number DAAE07-94-RO94).

REFERENCES

- Adkins, E.A., 1996, "Numerical Continuation and Bifurcation Methods for Mechanism Workspace and Controllability Analysis," Ph.D. dissertation, The University of Iowa
- Bestle, D. and Eberhard, P., 1992, "Analyzing and Optimizing Multibody Systems," *Mechanics of Structures and Machines*, Vol 20(1), pp. 67-92
- Bestle, D. and Seybold, J., 1992, "Sensitivity Analysis of Constrained Multibody Systems," *Archive of Applied Mechanics*, Vol. 62, pp. 181-190
- Bischof, C., Roh, L., and Mauer, A., 1996, "ADIC: A Tool for the Automatic Differentiation of C Program," Technical Report, Mathematics and Computer Science Division, Argonne National Laboratory
- Chang, C.O. and Nikravesh, P.E., 1985, "Optimal Design of Mechanical Systems With Constraint Violation Stabilization Method," *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, pp. 493-498
- du Plessis, L.J., and Snyman, J.A., 1999, "A Numerical Method for Determination of Dextrous Workspaces of Stewart Platforms", submitted to *International Journal for Numerical Methods in Engineering*

Hairer, E. and Warmer, G., 1996, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer Verlag, Berlin

Haug, E.J., 1987, "Design Sensitivity Analysis of Dynamic Systems," Computer-Aided Design: Structural and Mechanical Systems, C.A. Mota-Soares, Ed., Springer-Verlag, Berlin, pp. 705-756

Haug, E.J., 1989, Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume 1: Basic Methods, Allyn and Bacon, Needham Heights, Massachusetts

Haug, E.J., Luh, C.M., Adkins, E.A. and Wang, J.Y., 1996, "Numerical Algorithms for Mapping Boundaries of Manipulator Workspaces," *Journal of Mechanical Design*, Vol. 118, pp. 228-234

Haug, E.J., Negrut, D. and Iancu, M., 1997a, "An Implicit Numerical Integration Algorithm for Differential-Algebraic Equations of Multibody Dynamics," *Mechanics of Structures and Machines*, Vol. 25, No. 3, pp. 311-334

Haug, E.J., Iancu, M. and Negrut, D., 1997b, "Implicit Integration of the Equations of Multibody Dynamics in Descriptor Form," Proceedings, 1997 ASME Design Automation Conference

Hay, A.M. and Snyman, J.A., 1999, "The Determination of Non-Convex Workspaces of Generally Constrained Planar Stewart Platforms" Submitted to *Computers and Mathematics with Applications*

Jo, D.Y and Haug, E.J., 1989, "Workspace Analysis of Multibody Mechanical Systems Using Continuation Methods," *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 3, pp. 581-589

Krishnaswami, P., Wehage, R.A. and Haug, E.J., 1983, "Design Sensitivity Analysis of Constrained Dynamic Systems by Direct Differentiation," Technical Report No. 83-5, Center for Computer-Aided Design, The University of Iowa, Iowa City, Iowa

Litvin, F.L., 1980, "Application of Theorem of Implicit Function System Existence for Analysis and Synthesis of Linkages," *Mechanism and Machine Theory*, Vol. 15, pp. 115-125 and

Negrut, D., 1998, "On the Implicit Integration of Differential-Algebraic Equations of Multibody Dynamics", Ph.D. Dissertation, The University of Iowa

Serban, R., 1998, "Dynamic and Sensitivity Analysis of Multibody Systems", Ph.D. Dissertation, The University of Iowa

Snyman, J.A., du Plessis, L.J., and Duffy, J. 1998 "An Optimization Approach to the Determination of the Boundaries of Manipulator Workspaces" Technical Report, Department of Mechanical and Aeronautical Engineering, University of Pretoria

Tsai, Y.C. and Soni, A.H., 1981, "Accesible Region and Synthesis of Robot Arms," *Journal of Mechanical Design*, Vol. 103, pp. 803-811

Wang, J.Y. and Wu, J.K., 1993, "Dextrous Workspace of Manipulators, Part II: Computational MEthods," *Mechanics of Structu and Machines*, Vol 21(4), pp. 471-506

Yang, F.C. and Lee, T.W., 1983, "On the Workspace of Mechanical Manipulators," *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 105, pp. 62-69

A FEEDBACK CONTROLLER DESIGN METHODOLOGY FOR VEHICLE SUSPENSION SYSTEMS

M. M. ALI

**Department of Computational and Applied Mathematics,
University of the Witwatersrand, Johannesburg**

ABSTRACT

This paper describes a model of an active vehicle suspension system and shows how its performance may be improved by the use of a non-linear feedback control with a non-quadratic cost function. Feedback is optimized with and without the imposition of constraints. A global optimization algorithm is used to minimize the underlying cost function. Comparisons between the open loop, the unconstrained closed loop and the constrained systems are given.

1 INTRODUCTION

The modelling of vehicle suspension systems and the design of suspension control strategies for the purpose of giving a "smoother" ride is a problem that has attracted much interest over the years (Sharp and Crola, 1987; Frühauf et al., 1985). In the field of active (computer-controlled) suspension control strategy design, a number of suspension systems have been produced in recent years (Gordon et al., 1990; Gordon et al., 1991). To date most of the models (Wilson et al., 1986; Hac, 1985) have used linear optimal control theory to solve the optimization problem. This leads to a linear feedback law and a closed loop system, which has fixed eigenvalues. Although this method provides an analytical solution with relatively low computational time, it places unsatisfactory limits on system performance, because the cost function must be a quadratic function of the state and control variables (Hac 1987; Gordon et al. 1990). Hac (1987) introduced the idea of an adaptive linear strategy, but it is demonstrated in Gordon et al. (1990) that this approach could provide poor performance in some situations, such as when potholes are encountered. The purpose of this paper is to solve the problem for a non-linear feedback control in order to see how the performance characteristics might be improved by using a non-quadratic cost function. A model of a car suspension system with realistic constraints is also examined. In a practical control problem the open loop approach is very inefficient because the control is not an instantaneous function of the state variables. To determine $u(t)$ requires that a series of differential equations be solved over the time interval $(t_0, t_{\max})$ before the control $u(t)$ can be determined. Closed loop control overcomes the computational problems of solving for u by specifying the control $u(t)$ to be a function of state variables $u(t)=U(\mathbf{x}(t), \mathbf{k})$ which contains free parameters $\mathbf{k}$ which are chosen so that $u(t)$ gives optimal responses. Often u is chosen as a low order polynomial in the state variables $\mathbf{x}$. The unknown coefficients in the polynomial are the parameters $\mathbf{k}$. The problem here requires that the controller performs the same operation, irrespective of whether the system is displaced from its equilibrium condition through positive or negative values. We therefore choose $U(\mathbf{x}(t))$ as a 12 term

polynomial (an odd function of state variables) in the state. Finally a model is considered where the damping provided by the force generator is not altered instantaneously by the controller but is subject to a delay. This damping force is also constrained to lie within realistic fixed bounds. The problem reduces to that of the closed loop problem but with an extra state variable. This problem is also solved.

2 UNCONSTRAINED PROBLEM MODEL

A quarter vehicle (1/4 of the vehicle) model is the simplest that can represent the dynamics of a suspension system and possesses particular advantages over more complex models (Sharp and Crolla 1987). The system is shown schematically in figure (1), which also defines the variables used in the description of the problem

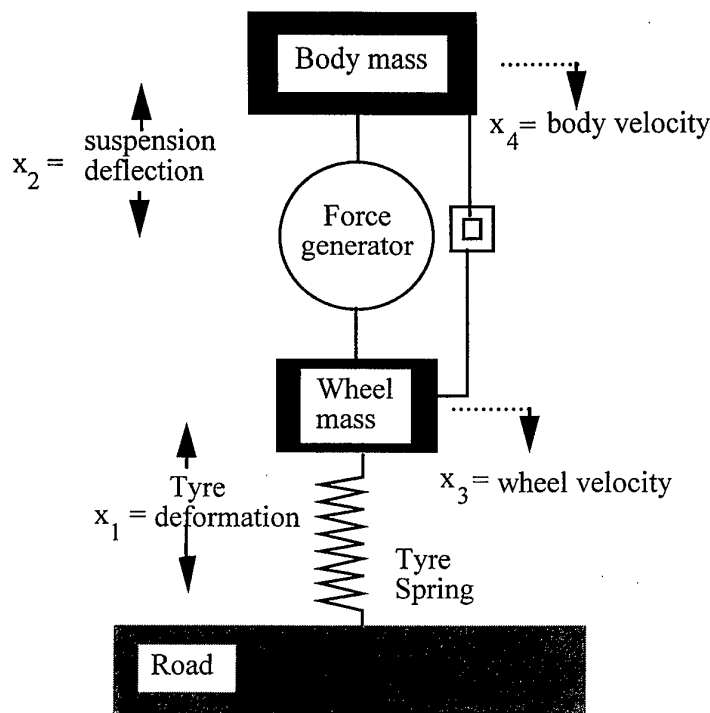


Figure (1): A schematic diagram of a vehicle suspension model.

This suspension system is controlled by the force (controller) generated by the generator situated between the wheel and the body. The state equation for the system are Newton's laws of motion, namely

$$\begin{aligned}\dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_3 - x_4 \\ \dot{x}_3 &= (k_t x_1 - u)/m \quad \dots\dots\dots(1) \\ \dot{x}_4 &= u/M\end{aligned}$$

Typical values for the constants might be $M(\text{Body mass})=320.0\text{kg}$, $m(\text{Wheel mass})=40.0\text{kg}$ and $k_t(\text{tyre stiffness})=2.0 \times 10^5 \text{N/m}$. The dynamic cost functional to be minimized is

$$I = \int_{t_0}^{t_{\max}} L(x, u) dt \quad \dots\dots\dots (2)$$

with $x(t_0)=x_0 \quad \dots\dots\dots (3)$

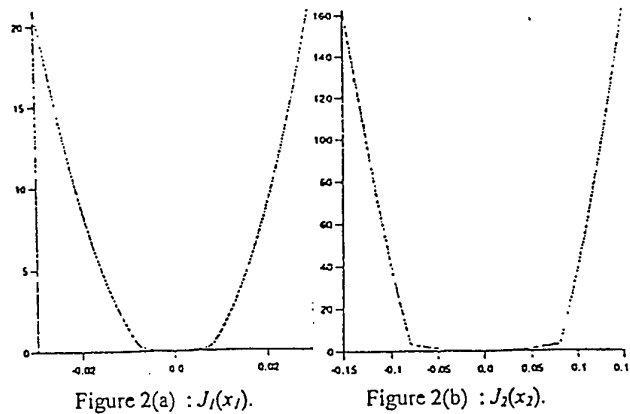
where $L(x, u)$ (a positive definite function of state variables and control inputs) is given below.

$$L(x, u) = J_1(x_1) + J_2(x_2) + \dot{x}_4^2$$

$$J_1(x_1) = \begin{cases} 4000x_1^2 & \text{if } |x_1| \leq 0.007 \\ 436000x_1^2 - 6048|x_1| + 21.168 & \text{if } 0.007 < |x_1| \leq 0.009 \\ 100000x_1^2 - 6.048 & \text{if } |x_1| > 0.009 \end{cases}$$

$$J_2(x_2) = \begin{cases} 500x_2^2 & \text{if } |x_2| \leq 0.079 \\ 2005250x_2^2 - 316750.5|x_2| + 12511.64475 & \text{if } 0.079 < |x_2| \leq 0.081 \\ 50000x_2^2 - 316.7505 & \text{if } |x_2| > 0.081 \end{cases}$$

The distances are in meters which will be used as our unit of length in the rest of the paper. The functions J_1 and J_2 are plotted in figure (2).



For the purpose of numerical calculation $t_{\max}$ is taken to be 2 secs. A typical workspace size is given by $-0.1 \leq x_2 \leq 0.1 \dots\dots (4)$. The tyre deformation is also limited and typically lies in the range $-0.025 \leq x_1 \leq 0.025 \dots\dots (5)$. The cost function $L(x, u)$ has been chosen with these design criteria in mind and has a very large value if x_1 and x_2 lie outside these limits. For the system (1) and (3) with performance index (2) the open loop control is determined using Pontryagin's maximum principle (Bryson and Ho, 1969). This gives the optimal control $u(t)$ that minimizes (2). The chosen feedback is a low order polynomial which is an odd function of state variables with 12 independent parameters.

$$U(\mathbf{x}(t)) = k_1 x_2 + k_2 x_3 + k_3 x_4 + k_4 x_1^2 x_3 + k_5 x_1 x_2 x_3 + k_6 x_2^3 + k_7 x_2^2 x_4 \\ + k_8 x_2 x_3 x_4 + k_9 x_4^2 x_2 + k_{10} x_3^3 + k_{11} x_3^2 x_4 + k_{12} x_4^2 x_3 \dots\dots\dots(6)$$

3 UNCONSTRAINED OPEN LOOP OPTIMIZATION

Unlike the closed loop model, the open loop optimization precalculates the force u that minimizes the cost. We apply Pontryagin's maximum principle to find the optimum open loop costs and controls for specified initial conditions. Introducing a 5th state variable, we can write

$$\dot{x}_5 = J_1(x_1) + J_2(x_2) + \dot{x}_4^2 \\ = J_1(x_1) + J_2(x_2) + u^2/M^2$$

which gives rise to the system

$$\dot{x}_1 = x_3 \\ \dot{x}_2 = x_3 - x_x \\ \dot{x}_3 = (k_1 x_1 - u)/m \dots\dots\dots(7) \\ \dot{x}_4 = u/M \\ \dot{x}_5 = J_1(x_1) + J_2(x_2) + u^2/M^2$$

Now, we redefine the problem as : minimize

$$I = \Phi(x_5(t_f)) = x_5(t_f) \dots\dots\dots(8)$$

subject to

$$x(t_0) = x_0.$$

The Hamiltonian H of the system becomes

$$H = \sum_{i=1}^5 \lambda_i f_i \text{ and } \dot{\lambda}_i(t) = -\frac{\partial H}{\partial x_i} \quad i=1, \dots, 5 \dots\dots\dots(9)$$

where f_i ($i=1,5$) are right hand sides of the system (7). The co-state equations are determined from Pontryagin's principle as follows

$$\begin{aligned}\dot{\lambda}_1(t) &= -\frac{\lambda_5 dJ_1(x_1)}{dx_1} - \frac{\lambda_3 k_t}{m} \\ \dot{\lambda}_2(t) &= -\lambda_5 \frac{dJ_2(x_2)}{dx_2} \dots\dots\dots(10) \\ \dot{\lambda}_3(t) &= \lambda_1 - \lambda_2 \\ \dot{\lambda}_4(t) &= \lambda_2 \\ \dot{\lambda}_5(t) &= 0.0\end{aligned}$$

with $\lambda(t_f) = (0, 0, 0, 0, 1)$, because

$$\lambda_i(t_f) = \frac{\partial \Phi(x(t_f))}{\partial x_i(t_f)} \quad i = 1, 5.$$

We define a gradient function $g(u)$ by

$$\begin{aligned}g(u) &= \frac{\partial H}{\partial u} \\ &= \frac{2u\lambda_5}{M^2} + \frac{\lambda_4}{M} - \frac{\lambda_3}{m} \dots\dots\dots(11)\end{aligned}$$

which is used in solving the system of equations (7) and (10). The numerical scheme proceeds as follows. Let u_i be the i th approximation to the open loop optimal control. The corresponding gradient $g(u_i)$ is computed by solving the state equations (7) forward with $u=u_i$, solving the co-state equations (10) backward and then computing $g(u_i)$ from (11). We solved the problem by the steepest descent gradient approach (Rosenbrock and Storey, 1966). This steepest descent algorithm also includes quadratic line searches (Rao, 1978). The algorithm is as follows. The step size ϵ was computed from the quadratic interpolation which minimizes $I(u_i - \epsilon \partial H / \partial u_i)$. Four representative points $x(t_0)$ are chosen in the space of initial conditions. More could be chosen if necessary at the expense of increased computing cost. Note that inclusion of the image point $-x(t_0)$ to distribute the initial conditions evenly in the state space will contribute the same cost as $x(t_0)$ and these are therefore not included here. The chosen initial conditions x_0 are

- (i) (0.025, 0.1, 0.0, 0.0, 0.0)
- (ii) (-0.025, 0.1, 0.0, 0.0, 0.0) (12)
- (iii) (0.01, 0.04, 0.0, 0.0, 0.0) and
- (iv) (0.01, -0.04, 0.0, 0.0, 0.0)

We took an integration step $h=0.005$ and solved the equations (7) and (10) using a Runge-Kutta 4th order method with $u_0=1.5$. The stopping criteria was $\|g(u_i)\| < 10^{-4}$.

Table (1)

Initial condition	number of iterations	cost
(i)	312	23.98
(ii)	258	4.75

(iii)	230	0.43
(iv)	120	0.25

Open loop results corresponding to initial conditions (12)

Closed loop optimisation : Closed loop optimization implements the control (6) with chosen parameter values $\mathbf{k}=(k_1, k_2, k_3, k_4, \dots, k_{12})$ as opposed to the open loop which pre-calculates $u(t)$ to minimize the cost. As a result the cost function depends implicitly on $\mathbf{k}$. Closed loop optimization does not involve the co-state equations (10) and the new cost function is defined as follows. Cost function = Sum of the dynamic costs incurred by the state from several initial conditions whilst implementing feedback law with parameter assigned.

$$C = \sum_{j=1}^4 \frac{\text{cost}_j}{F_j} \dots\dots\dots(13)$$

Here $\text{cost}_j = x_5(t_f)$, with initial condition (j) and for the closed loop control (6), $F_j = x_5(t_f)$ the cost incurred by the corresponding optimal open loop control. We add four weighted costs incurred by the initial conditions (i)-(iv) to fit the parameters in the feedback (6). The individual unweighted costs (cost_j) are found by solving the system (7) only. If, $\text{cost}_j = F_j$ for all j then we would infer that our optimal feedback law is exact. In order to determine the control $u(t)$ we must determine the optimal parameter set $(k_1, k_2, k_3, k_4, \dots, k_{12})$. This is done by using a controlled random search global optimization algorithm (Ali and Storey, 1994). We have found a region of parameter space (by numerical experiment using trial and error method) within which system is well defined and the global minimum occurs for the four sets of initial conditions given above. The function C attained its lowest value at the parameter set given in the table 2(a). The minimum cost function was calculated to be 4.23 but two local minima corresponding to different parameter sets were also found with values 4.26 and 4.29. The minimum value of 4.23 compares with the open loop cost of 4 a decrease in performance of about 6%. Note that the parameter values k_i are large quantities due to measuring the length in meters. If we had non-dimensionalised lengths so that typical displacements were $O(1)$, then the k_i would be scaled down accordingly.

Table (2a)

i	k_i	i	k_i
1	4.3542959×10^3	7	2.0217310×10^5
2	2.0056339×10^2	8	1.0612401×10^5
3	1.9045862×10^3	9	-2.8554219×10^4
4	4.1981636×10^6	10	6.0336024×10^2
5	5.7975217×10^5	11	3.2629846×10^3
6	1.2299534×10^6	12	-8.1452895×10^3

$C = 4.23$, cost_j : 25.83, 4.97, 0.45 and 0.26 and F_j : 23.98, 4.75, 0.43 and 0.25

The optimal parameter set $\mathbf{k}$ and the calculated cost functions for the initial conditions (12).

A further set of initial conditions were chosen at which to test the controller calculated above.

These results are shown in table 2(b).

Table 2(b)		
Initial conditions	Open loop cost	Closed loop cost
(-0.01, 0.04, 0, 0, 0)	0.25	0.26
(0.025, -0.1, 0, 0, 0)	4.75	4.97
(0.015, 0.05, 0, 0, 0)	0.89	1.06
(-0.015, 0.05, 0, 0, 0)	0.64	0.65
(0.025, -0.05, 0, 0, 0)	1.70	1.91
(-0.015, 0.1, 0, 0, 0)	4.44	4.74
(0.015, 0.1, 0, 0, 0)	15.95	19.23
(0.02, 0.09, 0, 0, 0)	12.24	14.41
(0.015, -0.08, 0, 0, 0)	1.17	1.36
(-0.025, 0.08, 0, 0, 0)	2.21	2.30

A comparison of the difference between open and closed loop costs for a variety of initial conditions.

The average decrease in performance for the results given in table 2(b) is less than 11% of the open loop costs. In Figure 3 we compare the open and closed loop control for initial condition (i) from (12) and it can be easily seen that the agreement is quite good.

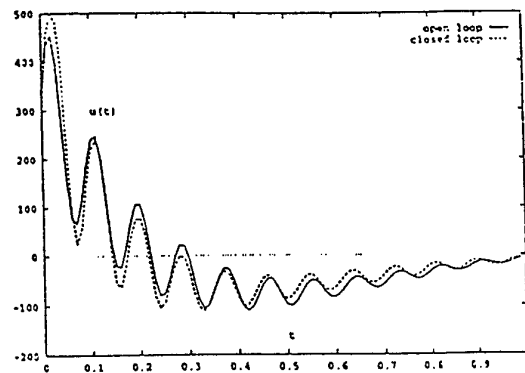


Figure 3 : Comparison of open and closed loop controls.

4 CONSTRAINED PROBLEM MODEL AND OPTIMIZATION

The optimal feedback given above has been calculated for a system which is not constrained by limits on the forces and in which the controller acts with immediate effect to counteract any disturbances to the system. In practice there are dissipative forces involved and these can be modelled by imagining that the controller operates a damper situated between the wheel and the body. The damper force F_d depends on the relative velocity ($x_3 - x_4$) and the operating current x_6 of the damper.

Table (3) : F_d

(x_3-x_4)	-1.5	-1.0	-0.5	-0.2	0.0	0.2	0.5	1.0
$x_6=0$	-2350	-1750	-1500	-800	0	900	1300	2400
$x_6=0.5$	-2000	-1450	-550	-250	0	250	650	1350
$x_6=1.0$	-1450	-900	-300	-200	0	200	350	800

A typical relationship between the damper force and the relative velocity across the damper for a realistic system.

For fixed values of (x_3-x_4) and x_6 , there is a unique value of the damper force F_d . A typical relationship between F_d , (x_3-x_4) and x_6 is shown in table (3) for a realistic system. The force F_d is a piecewise linear function of (x_3-x_4) for fixed x_6 . The force is bounded by a maximum value when $x_6=0$ and a minimum value when $x_6=1$. The relationship between F_d , (x_3-x_4) and x_6 is typical of a realistic automobile control problem and can be found for a real-life situation by system identification. For the constrained system the total force which takes the place of u on the right hand side of equations (7) is given by $u'=F_s+F_d$ where $F_s=k_s x_2$ is the spring force and F_d is the damper force. In a practical control situation the controller u passes a current I_0 to the damper. For the constrained system, u is determined from the state variables if $u - F_s$ lies within the damper force limits defined by table 4. If the damper force limits are exceeded then u is given by either $F_s + F_{\min}$ or $F_s + F_{\max}$ where $F_{\min}$ is the minimum damper force and $F_{\max}$ is the maximum damper force. Unfortunately the damper does not respond instantaneously to the change in the signal current I_0 but is subject to a delay. The operating current x_6 of the damper changes as a function of I_0-x_6 . For small differences between I_0 and x_6 , the rate of change is proportional to I_0-x_6 but for larger differences x_6 changes at a faster rate. We therefore write

$$\frac{dx_6}{dt} = 1/c_1(I_0-x_6)(1+c_1/c_2 |I_0-x_6|) \dots\dots\dots(14)$$

where c_1 and c_2 are constants depending on the system in question. Typical values for c_1 , c_2 and k_s are 3×10^{-2} s, 5×10^{-3} s² and 1.8×10^4 N/m respectively. These values are used in the numerical calculation described below. The full set of equations to be solved is equation (7) (with the exception that on the right hand sides u is replaced by u') and equation (14). These equations are integrated by a Runge-Kutta technique as before. At each timestep the controller determines a theoretical value for the damper force F_d^t which is either determined directly from the state variables or from the limits $F_s + F_{\min}$ or $F_s + F_{\max}$. The signal current I_0 is then determined from F_d^t . The values of x_6 and x_3-x_4 then determine F_d . Results from four sets of initial conditions were used in the optimization process which is as described above for the closed loop problem. The representative set of initial conditions were the same as those given by equations (12) for the unconstrained system with the exception that the value of $x_6(0)$ was chosen as 0.5, half way between the upper and lower bounds.

5 RESULTS

The global minimum was obtained at parameter values given below in table (4) but also a local minimum 5.63 was found at a different set of parameter values. Table (5) gives the calculated cost from a wider set of initial conditions and compares that cost to the unconstrained results.

Table (4)

i	k_i	i	k_i
1	7.4544835×10^3	7	3.5387222×10^6
2	-5.4871425×10^2	8	5.3122143×10^5
3	-2.913993×10^4	9	-2.8712891×10^4
4	1.5581296×10^6	10	1.9086715×10^3
5	2.878977×10^5	11	-3.719740×10^3
6	1.1408227×10^6	12	-4.769528×10^4

Total cost = 5.60, Individual cost : 28.61, 5.44, 0.72 and 0.39

Optimal parameter set and calculated individual cost for the unconstrained system with initial condition (12).

We compare the performance of the two feedbacks given in Tables (2a) and (4) on the constrained system. The following table represents the comparison which shows that the constrained controller provides lower cost all over the state space.

Table (5)

Initial conditions	cost(constrained)	cost(unconstrained)
(i)	28.42	28.90
(ii)	5.45	5.65
(iii)	0.71	0.72
(iv)	0.40	0.41
(-0.01, 0.04, 0, 0, 0, 0.5)	0.40	0.42
(-0.01, -0.04, 0, 0, 0, 0.5)	0.71	0.71
(-0.025, -0.1, 0, 0, 0, 0.5)	28.62	29.04
(0.025, -0.1, 0, 0, 0, 0.5)	5.45	5.49
(0.025, 0.05, 0, 0, 0, 0.5)	3.11	3.32
(-0.025, -0.05, 0, 0, 0, 0.5)	3.12	3.20
(0.0125, 0.1, 0, 0, 0, 0.5)	17.60	17.95
(-0.0125, -0.1, 0, 0, 0, 0.5)	17.74	18.02
(0.0125, -0.1, 0, 0, 0, 0.5)	5.06	5.34
(-0.0125, 0.1, 0, 0, 0, 0.5)	5.21	5.30

(0.0125, 0.05, 0, 0, 0, 0.5)	1.17	1.17
(-0.0125, -0.05, 0, 0, 0, 0.5)	1.16	1.16
(0.0125, -0.05, 0, 0, 0, 0.5)	0.70	0.72
(-0.0125, 0.05, 0, 0, 0, 0.5)	0.71	0.74
(0.0225, 0.09, 0, 0, 0, 0.5)	16.83	16.91
(-0.0225, -0.09, 0, 0, 0, 0.5)	17.00	17.05
(0.0225, -0.09, 0, 0, 0, 0.5)	3.46	3.48
(-0.0225, 0.09, 0, 0, 0, 0.5)	3.48	3.58
(0.005, 0.02, 0, 0, 0, 0.5)	0.17	0.17
(-0.005, -0.02, 0, 0, 0, 0.5)	0.17	0.17
(-0.005, 0.02, 0, 0, 0, 0.5)	0.094	0.096
(0.005, -0.02, 0, 0, 0, 0.5)	0.096	0.098
(0.015, 0.08, 0, 0, 0, 0.5)	5.13	5.19
(-0.015, -0.08, 0, 0, 0, 0.5)	5.15	5.19
(0.015, -0.08, 0, 0, 0, 0.5)	1.76	1.81
(-0.015, 0.08, 0, 0, 0, 0.5)	1.76	1.83
(0.0, 0.1, 0, 0, 0, 0.5)	9.06	9.14
(0.0, -0.1, 0, 0, 0, 0.5)	9.17	9.21
(0.025, 0.0, 0, 0, 0, 0.5)	1.96	2.01
(-0.025, 0.0, 0, 0, 0, 0.5)	1.85	1.86
(0.02, 0.09, 0, 0, 0, 0.5)	14.91	15.25

A comparison between the 'constrained' controller and the 'unconstrained' controller on the constrained system.

The results for the constrained controller show an average increase in performance >1.3% over the the unconstrained system.

6 DISCUSSION AND REMARKS

The philosophy adopted in this paper is to design a control system starting from initial $\mathbf{x}$ values which are displaced from their equilibrium values. This could be a good model of an isolated pothole on an otherwise uniform road. The alternative approach is to design the controller with a stochastic road input. However, we have not yet tested our controller on a system which includes a road input. This could be the basis of future work.

Acknowledgement : The author is indebted to Dr. T. J. Gordon of the Department of Transport Technology, Loughborough University of Technology, UK for providing the data in Table 3.

7 REFERENCES

- Sharp, R. S. and Crolla, D. A., "Road Vehicle Suspension System Design - a Review", *Vehicle System Dynamics*, 16, 1987, pp 167-192
- Frühauf, F., Kasper, R., and Lückel, J. L., "Design of an Active Suspension for a Passenger Vehicle Model Using Input Processes with Time Delays", *Vehicle System Dynamics*, 14, 1-3, 1985, pp 115-120.
- Gordon, T. J, Marsh, C., and Milsted, M. G., "Control Law Design for Active and Semi-active Automobile Suspension Systems", *VDI International Congress: Numerical Analysis in Automotive Engineering*, Wurzburg, 1990, VDI Berichte 816, 1990, 537-546.
- Gordon, T. J, Marsh, C., and Milsted, M. G., "A Comparison of Adaptive LQG and Nonlinear Controllers for Vehicle Suspension Systems", *Vehicle System Dynamics*, 20, 6, 1991, pp.321-340.
- Wilson, D. A., Sharp, R. S. and Hassan, S. A., "The Application of Linear Optimal Control Theory to the Design of Active Automotive Suspensions", *Vehicle System Dynamics*, 15, 1986, pp 105-118
- Hac, A., "Suspension Optimization of a 2-DOF Vehicle Model Using a Stochastic Optimal Control Technique", *J of Sound and Vibration*, 100, 3, 1985, pp 343-357
- Hac, A., "Adaptive Control for Vehicle Suspension", *Vehicle System Dynamic*, 16, 1987, pp 57-74.
- Bryson, A. E. and Ho, Y. C., (1969) *Applied Optimal Control*, Waltham Mass., Blaisdell .
- Rosenbrock H. H. and Storey, C., "Computational Techniques for Chemical Engineers", 1966, Pergaman Press, N. Y.
- Rao, S. S., 1978, "Optimization Theory and Applications", Wiley Eastern Limited, New Delhi India.
- Ali, M. M. and Storey, C., "Modified Controlled Random Search Algorithms", *International Journal of Computer mathematics*, 53, 1994, pp.229-235

The minimisation of noise diffraction over an irregularly shaped wall.

By: R. T. Bührmann

Research and Development department,
Centurion Systems, Kya Sands, Johannesburg,
South Africa

ABSTRACT

In this paper a mathematical model of the acoustic noise over a barrier wall is proposed, which is then used, in conjunction with mathematical optimisation, to determine the wall shape that yields the lowest possible noise levels at an observer.

From the results it is concluded that a flat wall is not the most optimum wall shape for minimum noise levels at the observer, when comparing wall geometries of identical area. The estimated noise level reduction of 0.9db compare to a flat wall could be of sufficient magnitude to justify the additional costs involved in the manufacturing of the irregular wall shape.

Keywords: Highway noise, Minimising urban noise, Minimising diffraction noise, Leapfrog, Barrier efficiency

1. INTRODUCTION

Walls in residential areas are build for many reasons ranging from security, privacy to the reduction of traffic noise. The requirement for lower noise levels especially in residential areas close to highways has led to an investigation into cost effective means for reducing noise levels in homes.

Traditionally these walls were simply build to the highest affordable height with a flat to edge. This article investigates the possibility of improving the barrier efficiency of a wall by changing the shape of top edge, and then also determines the optimal shape of the wall upper edge.

The article then concludes by numerically comparing the noise levels for both flat and irregular walls with equal net areas.

2. OVERVIEW

In order to do a comparison and optimisation study, a mathematical model describing the sound intensity response at the observer with respect to different wall configurations must be formulated. I.e., a method is required by which the average sound intensity at the observer can be computed as a car, for example passes by.

This was done by first formulating the noise intensity at several points as the car passed by, and then to average the noise intensity by taking the root mean square (RMS) over all the calculated points. This resulted in a single value which is indicative of the sound level experienced by the observer as a car passes by.

This approach of optimising for a single car significantly reduces the computational time over optimising for real highway traffic composed of several cars. The optimised result would in both cases be the same, since real highway traffic consists of several cars for which in this case the noise has been optimised for each of the cars separately.

By using the respective RMS values for different geometrical wall designs, and enforcing additional constraints such as, for example specifying a constant wall area, the wall shape can be optimised to achieve the lowest possible sound level at the observer.

3. THE ACOUSTIC MODEL

For computational efficiency, the analogy of the car passing the stationary observer was inverted to represent a stationary car or noise source, and a moving observer. In mathematical terms the position co-ordinates of the noise source (S_x , S_y , S_z) were fixed as well as the observer height (O_z) and distance from the wall (O_y). See Figure 1.

The spot noise levels were then calculated at short intervals in the (O_x) direction, which were then used to calculate the average RMS amplitude for each wall configuration.

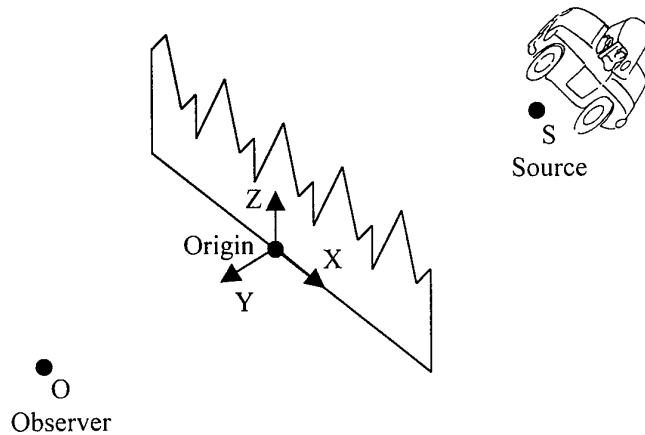


Figure 1. General layout of observer and source.

The wall itself was considered to be of the prefabricated type, being composed of a repeating top edge pattern between adjacent pillars, Photo 1 showing a very typical urban installation.



Photo 1. Typical prefabricated walling installation

This type of walling not only lends itself to a cost effective means of manufacture, but also significantly reduces the number of optimisation variables required to describe the wall shape. In this paper the top profile of each section were described by 5 points, of which one (z_5) were

constrained by the adjoining section too being equal to z_1 , giving only 4 real optimisation variables (z_1, z_2, z_3, z_4) which can be seen in figure 2.

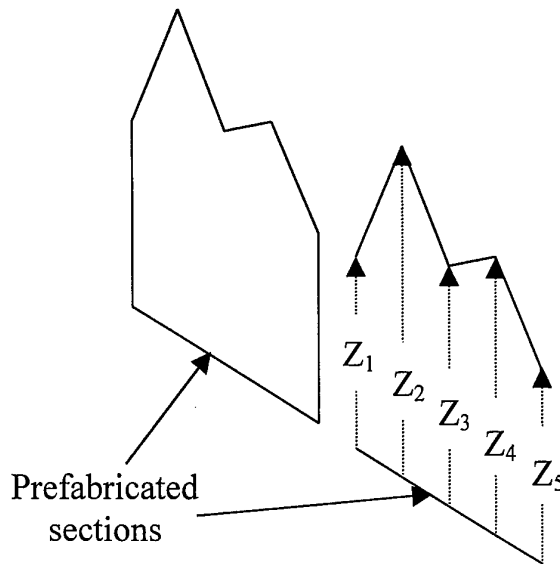


Figure 2. Wall upper edge shape definition.

The first assumption in solving for the sound intensity is that all the sound detected at the observer is only due to diffracted sound over the wall edges. [1, 2] (See figure 3.) This is a good assumption since the wall height is such that the observer is never in direct line of sight of the source, and the other effects, such as wall transmissibility and reflections are small in comparison. For the mathematical formulation, consider the wave analogy of water.

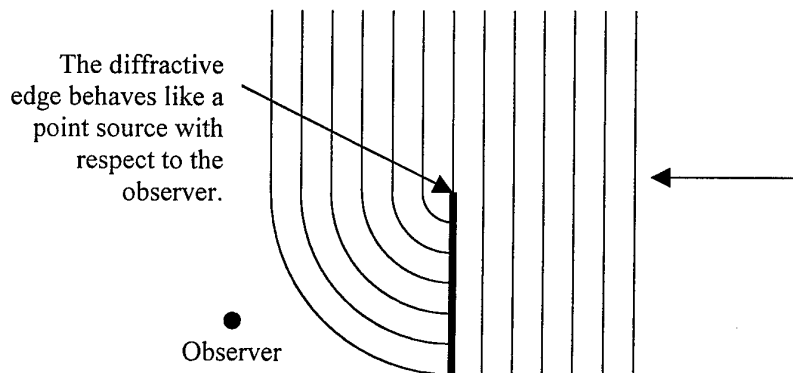


Figure 3. Wave analogy of water.

In order to approximate the sound amplitudes at the observer, the wall edges were approximated by a series of point sources or “speakers”, each with the correct sound phase and amplitude according to the respective speaker’s distance from the source. The sound magnitude at the observer is then obtained by vector summation of the effect of all the speakers on the wall, over all the evaluation frequencies. Mathematically it can be stated as follows:

For each observer location $O_i (O_{xi}, O_{yi}, O_{zi})$, $i=1,2,3\dots m$, speaker locations $P_j, (P_{xi}, P_{yi}, P_{zi})$, $j=1,2,3,\dots n$, frequencies F_k , $k=1,2,3\dots q$, and source location $S (S_x, S_y, S_z)$ the following expressions were used:

$$O_i P_j = \sqrt{(O_{xi} - P_{xj})^2 + (O_{yi} - P_{yj})^2 + (O_{zi} - P_{zj})^2}$$

$$P_j S = \sqrt{(P_{xj} - S_x)^2 + (P_{yj} - S_y)^2 + (P_{zj} - S_z)^2}$$

With $O_i P_j$ and $P_j S$ representing the respective distances from the observer to the wall, and the wall to the source, the corresponding amplitude is:

$$Amp\_O_{i,j} = \frac{1}{(O_i P_j^2 \cdot P_j S^2)} \quad \text{for all } i \text{ and } j. [1]$$

With the wavelength and phase as measured at the observer:

$$\lambda_k = \frac{343500}{F_k} \quad \text{being the wavelength for all frequencies } k,$$

$$Phase\_O_{i,j,k} = \text{remainder}\left(\frac{O_i P_j + P_j S}{\lambda_k}\right) \cdot \frac{2\pi}{\lambda_k} \quad \text{for all } i, j \text{ and } k.$$

Knowing the amplitude and phase at the observer for all the speakers, observer positions, and frequencies, the constructive and destructive interference can be computed over all the speakers as:

$$O_{i,k} = \sqrt{\left[\sum_j Amp\_O_{i,j} \cdot \cos(Phase\_O_{i,j,k}) \right]^2 + \left[\sum_j Amp\_O_{i,j} \cdot \sin(Phase\_O_{i,j,k}) \right]^2}$$

$O_{i,k}$ defines the spot sound amplitude level for at every observer location and for every frequency. From this, the RMS sound amplitude level is computed for every frequency which is then averaged over all the

frequencies to give the combined observer noise level for the whole frequency band.

$$RMS = \frac{1}{q} \sum_k \left[\sqrt{\sum_i (O_{i,k})^2} \right]$$

The combined phase is not calculated, since only the amplitudes determine the noise level at the observer when considering the random nature of traffic noise. It must be noted that the RMS signal amplitude is calculated for each frequency. The values for all the frequencies are then averaged to a single RMS value corresponding to the objective function to be minimised.

4. PARAMETER SENSITIVITY ANALYSIS

In preparation to any optimisation study, a suitable set of parameters like the number of prefabricated wall sections to use, the number of speakers to use on a wall section, etc. must be determined to ensure reliable operation of the objective function and also to improve the computational efficiency.

The following detail levels had to be determined, each being limited by the computational effort required when increasing the detail level:

- Frequency step size: A reasonable frequency step size had to be chosen between successive evaluations in the frequency band.
- Observer position step size: The distance between successive calculation points "as the car passes".
- Number of speakers: The number of speakers approximating the diffractive edge of the wall.
- Number of prefabricated sections: The number of wall sections making up the total length of the wall.

Firstly an acceptable set of detail levels were chosen, which for each parameter was such that on the next level of refinement only a 5% change in the objective function was observed for that parameter.

The convergence graphs shown in figure 4 resemble the change of only one parameter with the other parameters fixed to there respective acceptable limits.

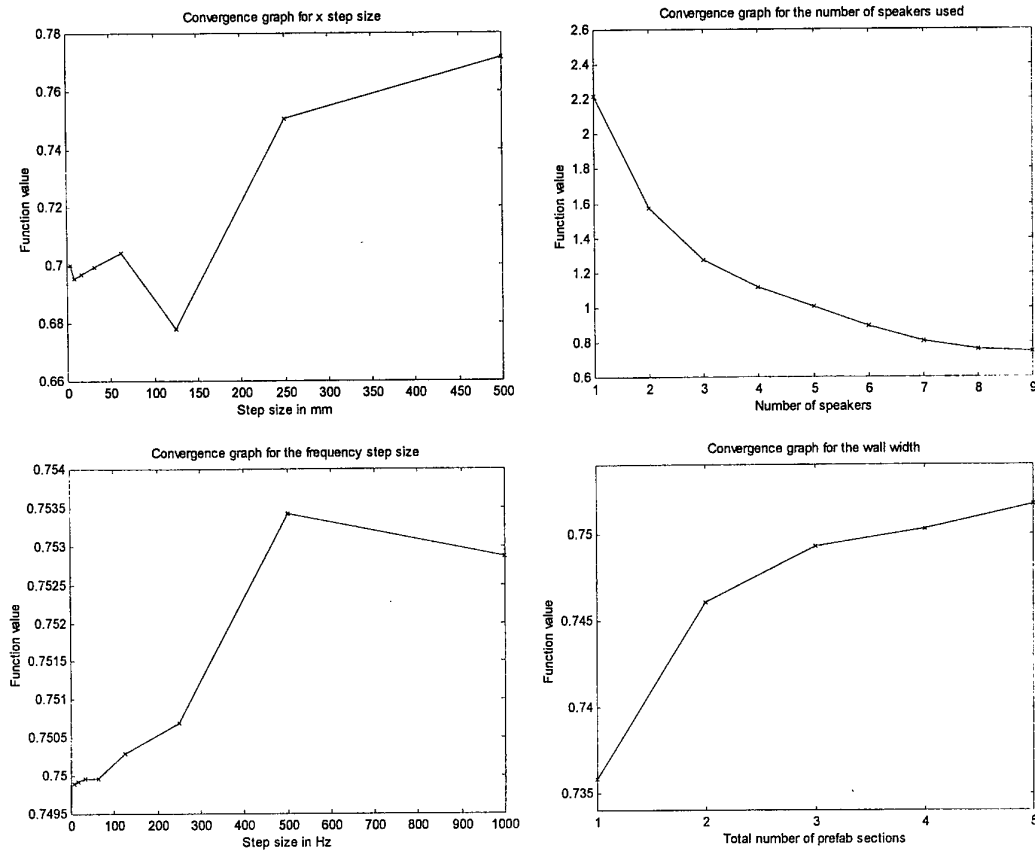


Figure 4. Plots showing convergence of the major parameters.

Another important factor to consider is the presence of “Aliasing Noise” which appears on the objective function if the observer step size is not smaller than the wavelength of the highest frequency.

Following from the graphs and limitations on “Aliasing Noise” the acceptable detail levels were found to be:

- Frequency step size: 30Hz
- Observer step size: 60mm
- Number of speakers: 9
- Number of prefabricated sections: 6

5. OBJECTIVE FUNCTION EVALUATIONS

Having defined the objective function, it is important to determine whether the computed observer response is an accurate reflection of what should be expected in practice. More explicitly, practical testing is required to verify the model.

Here, since the major aim is to demonstrate the proposed optimisation methodology, it is assumed that the model is sufficiently accurate.

In all the examples considered here, the following parameters are used:

- Wall parameters: The wall is composed of 6 prefabricated sections, each 2 meters wide, giving a total wall length of 12 m.
- Source parameters: The source is positioned 5 m from the wall, 0.25m above ground and on the wall centreline, i.e. $S_x=0\text{m}$, $S_y=5\text{m}$ and $S_z=0.25\text{m}$. The source amplitude were chosen to normalise the RMS results to around 1, and is kept constant for the remainder of the study.
- Observer parameters: The observer is also positioned 5m from the wall, but 1.8m above ground, i.e. $O_{y_i}=5\text{m}$ and $O_{z_i}=1.8\text{m}$ for all i , with amplitudes calculated at uniform intervals of 60mm along the length of the wall.

As a test of the model behaviour, a sequence of frequencies 10, 100, 1000Hz, with respective wavelengths of 34m, 3.4m, and 0.34m were applied and the RMS amplitude profile calculated over a flat wall.

The amplitude profiles can be seen in figure 5, for each of the 3 frequencies considered. These profiles resemble the actual noise amplitudes measured at the observer as the car emitting only the stated frequency passes by.

For the shorter wavelength, of higher frequency, a more complex amplitude profile is obtained, with peaks spaced approximately on the signal wavelength. From these plots it is interesting to note that the higher frequencies are attenuated much more than the lower frequencies. This explains to some extent why lower frequencies carry better over long distances. The sound intensity also slowly increases and then decreases as the car passes by, as would be expected.

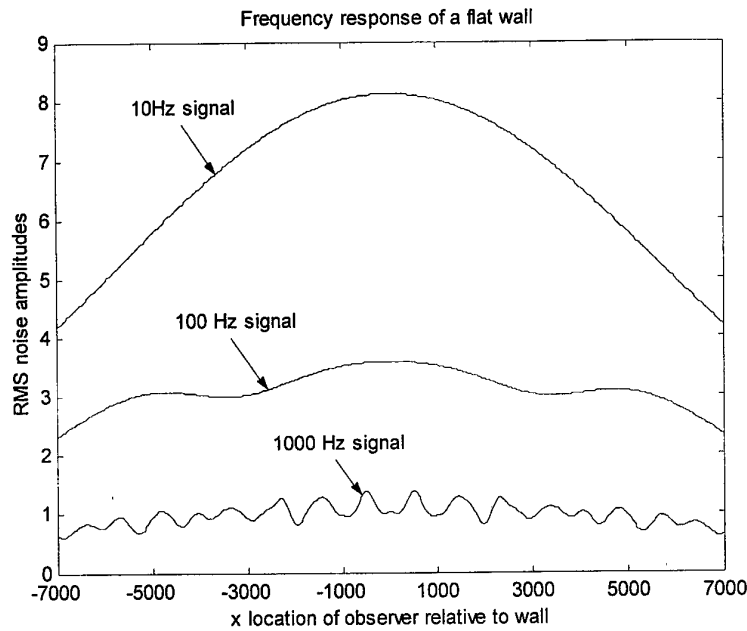


Figure 5. Flat wall response for 10, 100, and 1000Hz..

It is important to note that this represents only 3 distinct frequency bands and not a full frequency spectrum as is required for the optimisation.

6. THE OPTIMISATION PROBLEM

The goal is to minimise the noise level at the observer, by changing the upper edge shape of the wall (z_1, z_2, z_3, z_4), while maintaining a constant wall area.

Stated in general mathematical form:

Minimise $f(z_1, z_2, z_3, z_4)$, with respect to (z_1, z_2, z_3, z_4) .
Subject to the constraints:

$$\begin{aligned} g_1(z_1) &= -z_1 \leq 0, \\ g_2(z_2) &= -z_2 \leq 0, \\ g_3(z_3) &= -z_3 \leq 0, \\ g_4(z_4) &= -z_4 \leq 0, \end{aligned}$$

The wall area equality constraint compared to a flat 3m high wall:
$$h_1(z_1, z_2, z_3, z_4) = 2 \cdot (z_1 + z_2 + z_3 + z_4) / 4 - 6 = 0$$

Two gradient based routines were used to optimise this problem namely LFOPC [3, 4, 5] of Snyman et.al. and DYNAMIC-Q [6]. Both these

routines have been proven to be very robust on typical engineering problems, and always converges to an optimum point.

Both routines were used to solve the same problem, in order to illustrate and prove the benefits of the DYNAMIC-Q routine which uses successive spherical approximations to simplify and improve the convergence of the optimisation problem.

For both routines, all the input variables were normalised to values in the order of 1 to minimise any undue bias in the optimisation results.

7. OPTIMISED SOLUTIONS

Using as a basis the parameter values as determined in the sensitivity study and with the objective function evaluations, the following additional parameters were used for the optimisation:

Frequency range:	A constant amplitude source over the frequency range of 3000 to 4000Hz in steps of 30 Hz are used, giving a total of 33 frequency steps.
x-direction sampling:	Symmetry was assumed, with a scanning interval along the length of the wall ranging from 0 to 7000mm, in steps of 60mm, giving a total of 116 steps.
Speakers:	A total of 9 speakers per wall section were used in the evaluations, giving a total of 216 speakers over the 6 wall sections.

The above selections resulted in 826 000 phase and amplitude evaluations for every evaluation of the objective function.

The global optimum solution to the problem was obtained, by randomly selecting starting points, and then to select the lowest local optimum point.

Using both routines, the global optimum wall shape were found to be $\mathbf{z}^*=(3.078, 2.962, 2.916, 3.044)$, with the corresponding objective function value as $F(\mathbf{z}^*)=0.4956$ after 121 iterations with LFOCV and 30 iterations with DYNAMIC-Q.

A flat wall on the other hand defined as $\mathbf{z}^0=(3, 3, 3, 3)$, returned a corresponding objective function result of $F(\mathbf{z}^0)=0.55$. This represents a net noise reduction of 0.9db, without any additional area or material added to the wall.

This optimum point was achieved to an accuracy of better than 0.1mm with both optimisation routines returning the same optimum wall shape. See figures 5, and 6.

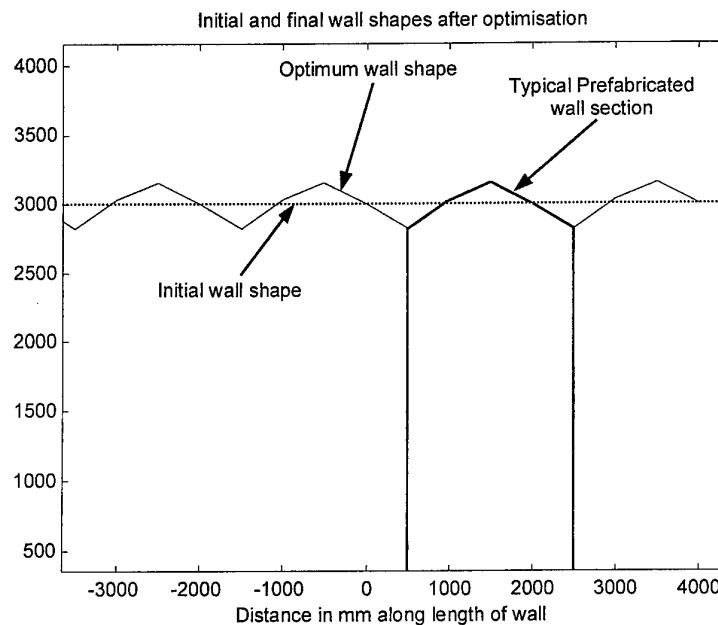


Figure 5. Initial and final wall shapes starting with a 3m high flat wall

The LFOPC algorithm has again been proven as a reliable and robust method that performs no line searches and uses only penalty function gradient information. It is however slow in converging (See figure 6). DYNAMIC-Q is a recently developed method that sets up successive

approximate subproblems where the objective and constraint functions are approximated by spherical quadratic functions. The LFOP algorithm is then used to solve each approximate sub-problem.

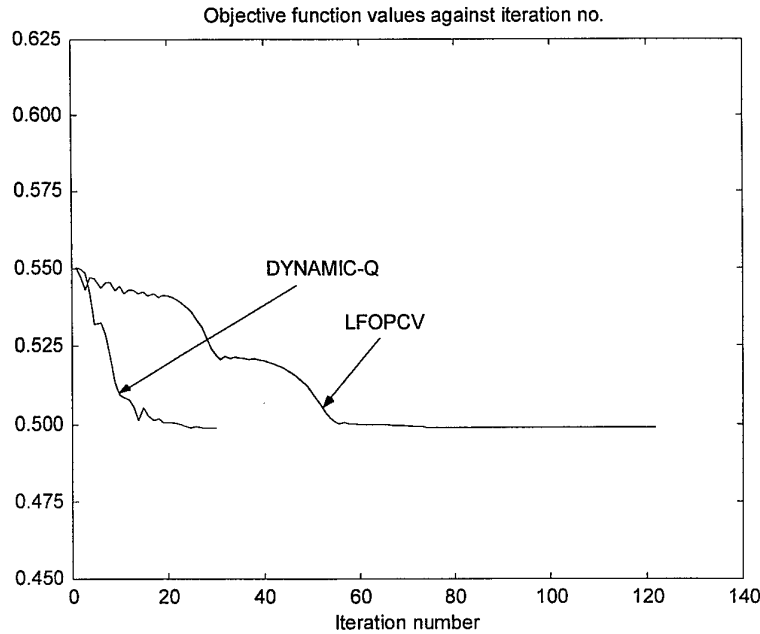


Figure 6. Objective function convergence to the optimum point

The convergence of the DYNAMIC-Q algorithm is significantly faster than LFOPC, again highlighting the value of using successive approximations for optimisation problems. The objective function convergence histories can be seen in figure 6.

8. CONCLUSIONS

It is possible to improve urban noise levels next to highways, by simply changing the top edge of the barrier wall to be irregular, without changing the net area of the wall. The specific shape of the wall edge is very important, since it is also possible to reduce the barrier efficiency by adding just any shape of edge.

In this optimisation study it has been found that the noise levels over a barrier a wall can theoretically be reduced by 0.9db. This improvement in noise levels could either be used to further reduce the noise levels in urban areas, or would allow for lower and more cost effective walls with the same noise reduction.

The optimisation routines (LFOPC and DYNAMIC-Q) used to optimise this highly non-linear problem has again proven its robustness and reliability and has also shown the significant improvement that can be made by using successive approximations to an optimisation problem.

The convergence speed advantage of DYNAMIC-Q having converged in 30 steps compared to 121 for LFOPC will allow for significantly more complex and accurate problems to be solved in the same time.

With regard to possible future work the following recommendations can be made:

1. The current analysis was done for a fixed and well determined observer height and distance from the wall. A more complete analysis should allow for the observer to be anywhere in the vicinity of the wall to ensure that the optimum wall shape is a general optimum, and not just for a specific observer location as is the case with this study.
2. The human ear amplitude response spectrum, as well as more accurate data on the frequency spectrum and amplitudes of high speed traffic noise is required and should be included in the analysis.
3. Practical measurements and tests must be conducted to confirm the accuracy of the model.

References:

1. B. J. Smith, R. J. Peters, Stephanie Owen, *Acoustics and noise control*. Longman New York, 1982, pp. 65-66.
2. D. Templeton, D. Saunders, *Acoustic Design*. The Alden Press, Oxford, 1987, pp. 114-116.
3. J. A. Snyman, A new and dynamic method for unconstrained minimisation. *Appl Math Modelling*, Vol 6 (1982) 449-462.
4. J. A. Snyman, An improved version of the original leap-frog dynamic method for unconstrained minimization LFOP1(b), *Appl Math Modelling*, Vol 7 (1983) 216-218.
5. J. A. Snyman, The LFOPC leap-frog method for constrained optimization. To appear in *Computers Math Applic* (1999).
6. J. A. Snyman, N. Stander & W. J. Roux, A dynamic penalty function method for the solution of structural optimization problems. *Appl Math Modelling*, Vol 18 (1994) 453-460.

Global Multidisciplinary Optimization of a High Speed Civil Transport

Steven E. Cox and Raphael T. Haftka

University of Florida, Gainesville, FL

Chuck A. Baker, Bernard Grossman, William H. Mason and Layne T. Watson

Virginia Polytechnic Institute & State University, Blacksburg, VA

ABSTRACT

The conceptual design of aircraft often entails a large number of nonlinear constraints that result in a non-convex feasible design space and multiple local optima. The design of the High Speed Civil Transport (HSCT) is used as an example of a highly complex conceptual design with 26 design variables and 68 constraints. This paper compares three global optimization techniques on the HSCT problem and two test problems containing thousands of local optima and noise: multistart local optimizations using either sequential quadratic programming (SQP) or Snyman's dynamic search method, and Jones' DIRECT global optimization algorithm on the HSCT problem. SQP is a local optimizer, while Snyman's algorithm is capable of moving through shallow local minima. The DIRECT algorithm is a global search method based on Lipschitzian optimization that locates small promising regions of design space and then uses a local optimizer to converge to the optimum. The DIRECT algorithm is found to be the most cost effective for locating the global optimum of functions with true local optima.

1. INTRODUCTION

The conceptual design of complex systems often involves optimization with a large number of design variables involving multiple disciplines. Often the feasible design space or the objective functions are nonconvex and may contain multiple local optima that can trap local optimizers and prevent them from locating the best design. To solve this problem, either multiple starting points or a global optimization algorithm may be used. Both of these methods will increase the cost of the optimization.

Discretization errors, round-off errors, and less than fully converged iterative calculations within analysis codes can result in noisy constraints and objective functions. This can create additional spurious optima. A distinction is made between these noise-generated numerical optima and genuine optima due to non-convexity. Designers are primarily concerned with locating the physical local optima modeled by the analysis functions and need a way to bypass the numerical noise.

Many global optimization codes have been developed and tested for use with different classes of problems [1]. However, most of these global optimization algorithms are specialized to a narrow class of problems. Similarly, different local optimizers have been compared in terms of their performance as multistart optimizers (e.g., [2]). One general purpose global optimization algorithm to draw attention lately is a Lipschitzian optimizer called DIRECT [3]. It has recently been applied to airfoil design [4] and modifications have been proposed to speed up its convergence [5].

This paper compares three global optimization methods for optimizing the configuration design of the High Speed Civil Transport (HSCT), a high dimensional design problem with a nonconvex feasible domain due to nonlinear constraints and numerical noise [6]. The first method uses multiple starting points with sequential quadratic programming (SQP) as implemented in DOT [7]. The second method, Snyman's dynamic search algorithm [8], can pass through shallow local minima to locate a better optimum but still requires multiple starting points. The third method is DIRECT, which is run only once.

In addition to the HSCT problem, two test functions were examined to explore the performance of the optimizers on different classes of problems. The comparisons help to show the strengths of each optimizer and suggest the types of problems each is best suited for.

2. OPTIMIZERS

Sequential Quadratic Programming

The commercial program *Design Optimization Tools* (DOT) [7] was used with SQP. SQP forms quadratic approximations of the objective function and a linear approximation for the constraints and moves towards the optimum within given move limits. Due to the use of approximations, DOT is relatively quick to perform a single optimization and will handle a limited amount of noise without becoming stuck in spurious local minima. DOT has been successfully used in the past with the HSCT problem, and compared favorably to other local optimizers [2]. For global search, DOT was started at 100 random initial designs for the HSCT problem and up to 20,000 starting points for the test problems.

Dynamic Search

Snyman's dynamic search method, Leap Frog Optimization Procedure with Constraints, Version 3 (LFOPCV3) [8], is a semiglobal optimization method that can move through shallow local minima. The method is based on the physical situation of a particle rolling down hill. As the particle moves down, it builds momentum, which carries it out of small dips in its path. When ascent occurs, a damping strategy is used to extract energy from the particle to prevent endless oscillation about a minimum. However, it still requires multiple starts to sample the entire design box. For this comparison, we used the same 100 initial designs to compare the performance with DOT for the HSCT problem and up to 20,000 starting points for the test problems. LFOPCV3 handles constraints with a standard quadratic penalty function approach.

DIRECT

The DIRECT algorithm [3] is a variation of Lipschitzian optimization that uses all values for the Lipschitz constant. Lipschitzian optimization requires the user to specify the Lipschitz constant, K , which is used as a prediction of the maximum possible slope of the objective function. Lipschitzian optimization uses the value of the objective function at the corners of each box and K to find the box with potentially the lowest objective function value. The function is evaluated at the predicted minimum possible value and the process is repeated for a set number of iterations. (See Figure 1.)

DIRECT does not require the user to predict the Lipschitz constant. It uses the function value at the center of each box and the box size to find the boxes which potentially contain the optimum. A box is selected if using some Lipschitz constant K , that box could contain the lowest function value. (See Figure 2.) It can be shown that this requires the box to lie on the bottom part of the convex hull of the set of boxes in a graph such as Figure 2. In Figure 2, each of the boxes is one of four sizes. The smallest, largest and next to largest boxes are potentially optimal because, for each box, there is some value of K for which that box could contain a better design than any other box. Graham's Scan routine is used to identify the potentially optimal boxes. [3] Boxes that were not previously potentially optimal can become potentially optimal in later iterations as the boxes are divided.

The DIRECT algorithm is as follows:

- 1) Normalize the search space to the unit hypercube. Let c_1 be the centerpoint of the hypercube and evaluate $f(c_1)$

- 2) Identify the set S of potentially optimal rectangles (boxes).
- 3) For each rectangle $j \in S$:
 - a) Identify the set of dimensions I with the longest side length. Let δ equal one-third of this length.
 - b) Sample the function at the points $c \pm \delta e_i$ for all $i \in I$, where c is the center of the rectangle and e_i is the i th unit vector.
 - c) Divide the rectangle containing c into thirds along the dimensions in I , starting with the dimension with the lowest value of $f(c \pm \delta e_i)$ and continuing to the dimension with the highest $f(c \pm \delta e_i)$.
- 4) Repeat 2. - 3. until stopping criterion is met.

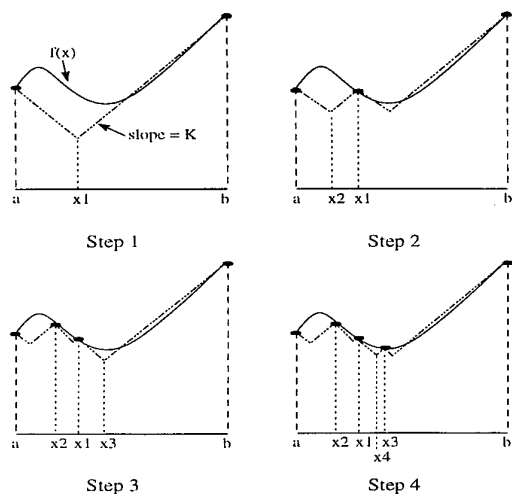


Figure 1. Lipschitzian Optimization.

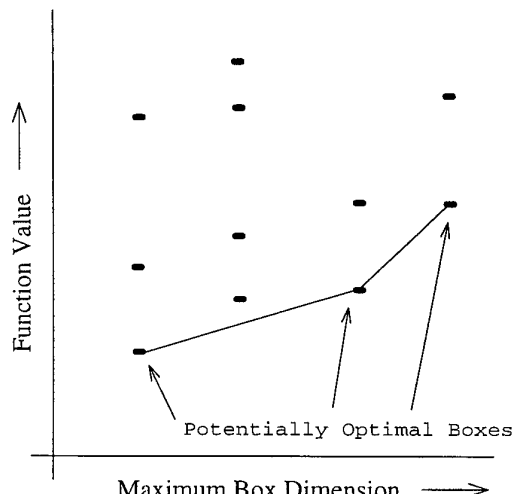


Figure 2. DIRECT Point Selection.

DIRECT was found to be quick to locate regions of local optima but slow to converge. To speed up the convergence, DIRECT is stopped once the smallest box reaches a specified percentage of the original box size and a local optimizer is used for the final optimization. The optimization was stopped when the smallest box reached 0.01% of the original box size for the HSCT comparisons and 0.001% for the test functions presented in the next section. DOT was then used starting from up to 15 of the best points analyzed by DIRECT, which were at least a certain percentage of the width of the design space away from the other starting points selected for local optimization. For the HSCT and Quartic functions 5% separation was used while 0.5% separation was used for the Griewank Function.

DIRECT uses a linear penalty function to handle constraints, $g \leq 0$. The penalty function is given as follows.

$$F = f + \sum_{i=1}^n g_i P_i, \quad (1)$$

where f is the objective function, g is the constraint vector, and P_i is a penalty parameter, which is zero for unviolated constraints and a small positive number for violated constraints. The penalty parameter should be as small as possible while still preventing the optimizer from selecting an infeasible design. Choosing a constant that is approximately twice the magnitude of the largest Lagrange multipliers will push the optimizer out of the infeasible region without over penalizing boxes with an infeasible center.

For the HSCT problem the range constraint had the largest effect on the objective. On average, the HSCT requires about 90 lbs. of fuel per mile of range deficiency. Therefore, the penalty constant was chosen to

increase the objective function by twice this much per mile of violation. The range constraint is given as:

$$g = 20 \left(\frac{R}{5500} - 1 \right), \quad (2)$$

where R is the range. The objective function, f , is the gross weight normalized by 700,000 lbs. This gives a value of 0.071 for the penalty multiplier. For our study we rounded this up to 0.1.

3. TEST FUNCTIONS

The optimizers were compared for two algebraic test functions in addition to the HSCT problems. The Griewank function was used as an example with one true optimum superimposed with noise. A simple quartic function optimized in a hypercube provides an example of large number of widely separated local optima. Each optimizer was run multiple times for the 5, 10 and 20 design variable (DV) cases.

The Griewank function is a quadratic function with trigonometric noise added. Without the noise, the function is convex with one optimum. This function tests how well the optimizers could move through the noise to locate the actual optimum. The one-dimensional Griewank function is given in Figure 3.

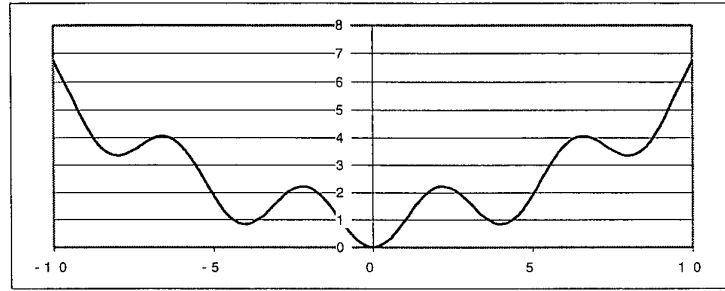


Figure 3. Griewank Function in One Dimension.

The n -dimensional Griewank function is defined as:

$$F(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{d} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right), \quad (3)$$

When d is small, the Griewank function is primarily a quadratic function with a small noise component from the cosine terms. As d is increased, the objective function becomes flatter and the cosine portion becomes more important. The function changes from a noisy function with one global optimum, to an almost flat surface with hundreds of nearly equal local optima.

The constant d is taken to be 200, 1000 and 20,000 for the 5, 10, and 20 DV cases, respectively. The design domain was $[-400,600]^n$ for DOT and LFOPCV3 with random starting points. For DIRECT, the box had edges of length 1000 with the upper limit randomly set between 100 and 900 to randomize the results for the different runs. The global optimum is at $\mathbf{x} = \mathbf{0}$. For this comparison, the optimizer was considered to have reached the global optimum if all of the design variables were in the range ± 0.1 . Each optimizer was run enough times to compute the average number of evaluations for each optimum found. The results are given in Table 1.

The 90% confidence column gives the number of function evaluations needed to be 90% certain of locating the global optimum at least once. If n is the total number of optimization runs per optima located, the probability of not locating the optimum in r runs is given as:

$$P = ((n-1)/n)^r \quad (4)$$

The 90% confidence column in Table 1 gives the number of optimization runs, r , times the average number of function evaluations for each case, needed for P to be less than 10%.

Table 1. Comparisons for 5, 10, and 20 DV Griewank Functions.

5 DV	# of optima located / # of runs	function evaluations per optimum	90% Confidence
DIRECT	87 / 100	3400	3335
DOT	5 / 300	10260	23416
LFOPCV3	60 / 400	9050	19235
10 DV			
DIRECT	56 / 100	11810	18550
DOT	96 / 500	1260	2604
LFOPCV3	136 / 500	32240	63598
20 DV			
DIRECT	8 / 200	102650	231593
DOT	16 / 2000	19740	45265
LFOPCV3	128 / 2000	7220	16084

The results in Table 1 indicate that DIRECT is not the most efficient optimizer in higher dimensional space. The Griewank function is smooth with an underlying quadratic shape. This is suitable for gradient based optimization if the algorithm is capable of moving past the weak local optima. LFOPCV3 does this by design while DOT is able to do this by using approximations based on widely separated points.

The lack of a clear trend in Table 1 may be due to the effect of d , which was different for each case. To investigate the effect of the noise on the different optimizers, the 10 variable case was repeated for different values of d . The results for this comparison are given in Table 2.

Table 2. Effect of Variation of d in 10-dimensional Griewank Function.

DIRECT – 100 runs	# of times optimum located	function evaluations per optimum	90% Confidence
$d = 4000$	19	39480	81958
$d = 1000$	56	11810	18550
$d = 200$	83	6990	7539
DOT – 500 runs			
$d = 4000$	37	3970	8801
$d = 1000$	96	1260	2604
$d = 200$	160	620	1188
LFOPCV – 500 runs			
$d = 4000$	25	193410	434117
$d = 1000$	136	32240	63598
$d = 200$	390	830	989

As expected, the optimization is easier for small values of d , where the global optimum is more distinct from other local optima. DOT appears to be the least sensitive to this parameter while LFOPCV3 was the

most sensitive to d . The approximations used by DOT appear to allow it to move past the noise and capture the underlying quadratic function. LFOPCV3, however, is slowed down as it passes through the noise and is not able to locate small improvements in the local optima for the large value of d .

The one-dimensional quartic function is given in Figure 4.

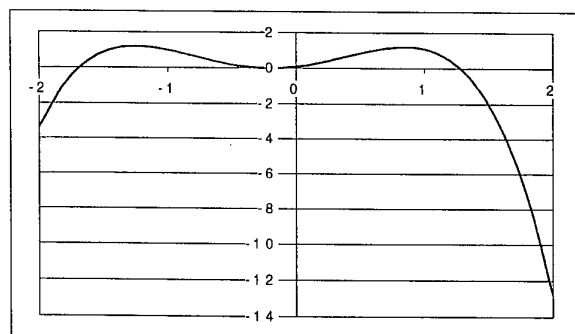


Figure 4. Quartic Test Function ($e=0.3$).

The n -dimensional quartic function is defined as

$$F(x) = \sum_{i=1}^n [2.2(x_i + e_i)^2 - (x_i + e_i)^4], \quad (5)$$

where e_i is a random number in the range $[0.2, 0.4]$. The design space was the hypercube $[-2, 2]^n$. The global optimum is at $\mathbf{x} = \mathbf{2}$. This problem contains 3^n widely separated local optima located at the constraint boundaries and close to the center of each variable. The optimizer was considered to have located the optimum if all of the design variables were greater than 1.9. DIRECT was run for 200 random values for e_i for each case. LFOPCV3 and DOT used 20,000 starting points in order to get a statistically meaningful average number of function evaluations per optimum found. The results are given in Table 3.

Table 3. Comparisons for 5, 10, and 20 DV Quartic Functions.

5 DV	# of times optimum located	function evaluations per optimum	90% Confidence
DIRECT	200	1025	1025
DOT	410	3397	7741
LFOPCV3	8	2581418	5942746
10 DV			
DIRECT	200	2192	2192
DOT	16	174189	400925
LFOPCV3	0	-	-
20 DV			
DIRECT	200	11266	11266
DOT	0	-	-
LFOPCV3	0	-	-

For the Griewank function, none of the optimizers were 100% effective in locating the optimum. For 90% confidence, equation 4 requires 2.3 times as many function evaluations as the average number of

function evaluations per optimum. For the Quartic function, DIRECT was 100% efficient at locating the optimum. This means that a single run of DIRECT more than satisfies the 90% requirement. For this reason the 90% confidence column equals the function evaluations per optimum column for DIRECT.

DIRECT is clearly better suited for this problem than either DOT or LFOPCV3. The space partitioning method employed by DIRECT allows it to examine a wider range of points, which increases the likelihood of locating the basin that contains the global optimum. DOT is unable to adequately capture the shape of the entire design space with the quadratic approximation. The quadratic approximation is only good for describing small portions of the design space, which prevents it from accurately extrapolating to the regions of other local optima.

LFOPCV3 is strictly a local optimizer for this problem. It is designed to move through noise and weak local minima, not the deep basins found in this problem. In order for this optimizer to find the global optimum, it must start in the basin that contains the global optimum. For $e = 0.3$, the odds of starting in this region for the 5, 10, and 20 DV cases are 1 in 333, 1 in 111,000 and 1 in 123×10^8 respectively. This makes multistart local optimization methods a poor choice for this type of problem. In many cases LFOPCV3 stopped at saddle points and some points where the slope was not zero. This suggests that some of the performance is due to false convergence instead of failure of the algorithm.

4. HSCT DESIGN PROBLEM

The design problem used to compare the three optimization methods is the configuration design of a HSCT. The HSCT design code uses up to 29 design variables and up to 68 constraints [9], with a design goal of minimizing the gross take off weight (GTOW).

The HSCT code uses simple structural and aerodynamic models to analyze a conceptual design of the aircraft. The current version of the HSCT code employs an aerodynamic drag response surface based on solutions of the Euler equations [10]. Once the variable designs are selected using design of experiments theory, the wing camber for each design is found using Carlson's modified linear theory optimization method WINGDES [11,12]. This design is then analyzed using the Euler equations. The viscous contribution to the drag C_{D0} is obtained from standard algebraic estimates [13] of the skin friction assuming turbulent flow.

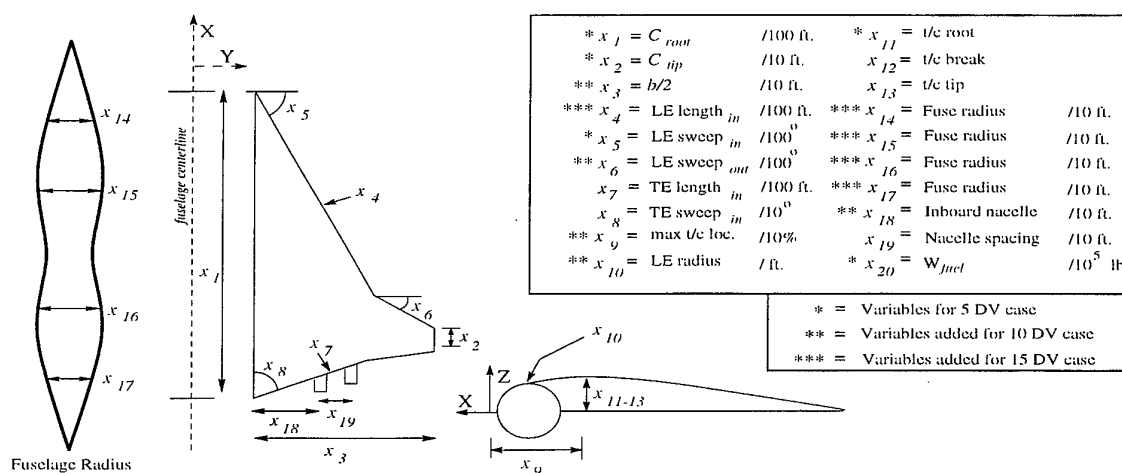


Figure 5. Definitions of Design Variables.

Previous work with the HSCT code has demonstrated the non-convexity of the feasible domain and the existence of multiple local minima [6]. In addition, several analyses, including range calculations, result in noisy performance functions. The HSCT optimization uses subsets of 5, 10, 15, or 20 of the design variables as defined in Figure 5. The variables designated with one asterisk are used in the 5 DV case. The 10, 15 and 20 DV cases add the variables designated with two, three and four asterisks respectively. These design cases were used to compare the performance of the optimizers. The five DV cases have only one optimum, while the higher dimensional designs contain many local optima and nonconvex feasible regions. This allowed us to compare the optimizers on problems of different complexities.

5. HSCT RESULTS

For each design case, the HSCT code was run for 100 starting points for DOT and LFOPCV3, and once for DIRECT. Table 4 shows the results from these runs. The 90% confidence column gives the number of function evaluations needed to find a design within 1000 lb of the best optimum design out of all 3 optimizers. In order to ensure that DIRECT will locate the same optimum with slightly different starting conditions, it was run a second time for each case with the design box perturbed by 1%. In each case, DIRECT located an optimum that was within 300 lb of the original optimum. DIRECT and DOT alternated as the fastest optimizer with LFOPCV3 performing the worst. LFOPCV3 was slowed down by the need to continuously calculate gradients by finite differences while DOT was able to construct an approximation to the problem, which reduced the number of gradient calculations it required.

Table 4. Comparison of 5, 10, 15, and 20 DV HSCT Problem

5 DV	Best GTOW Located	Function evaluations	# of Optima	90% Confidence
DIRECT	638238	2345	1	2345
DOT	638231	10870	94	89
LFOPCV3	638813	33234	5	14919
10 DV				
DIRECT	624751	9067	1	9067
DOT	624731	29791	10	13373
LFOPCV3	631300	137929	0	-
15 DV				
DIRECT	603127	40662	1	40662
DOT	602685	47440	12	8545
LFOPCV3	620538	1437312	0	-
20 DV				
DIRECT	588404	51147	1	51147
DOT	588586	57428	2	65453
LFOPCV3	620092	418315	0	-

For the 5 DV case, each optimizer found approximately the same global optimum. This was expected as earlier experiments showed that the 5 DV case had only one local optimum. Here DOT was the most efficient optimizer since it did not require more than a few starting points to locate the optimum. For the 10 DV case, DIRECT required fewer function evaluations than DOT while locating a design of the same weight. The 10 DV case has distinct local optima [6] and the multistart strategy appears to be less efficient than the global strategy used by DIRECT.

For the 15 DV case DOT found a slightly lower weight than DIRECT. However, this difference is less than 0.08% of the total weight of the HSCT. The designs found by DOT and DIRECT are shown in Table 5. They differ only slightly in most of the design variables. This indicates that the difference between the two is primarily due to noise preventing the optimizer from moving through a narrow region of similar weights to find the best optimum.

For the 20 DV case, DIRECT performed slightly better than DOT. The weights they found differed by only 200 lb, but the design variables differed by as much as 35% of their respective ranges. This case showed that DIRECT is still able to perform well for a 20 dimensional design space. The number of function evaluations increased rapidly for more than 10 dimensions, but it was still less expensive than the two multistart methods for the 20 DV case. Without prior knowledge of the design space, it is difficult to decide on the number of starting points to choose for the multistart methods.

Table 5: 15 DV Optima for DOT and DIRECT.

Design Variables*	DOT	DIRECT	% difference
Root chord	1.7306	1.7116	4.2
Tip chord	0.8425	0.8563	2.3
Semispan	6.9164	6.9336	1.1
LE length in.	1.2052	1.2074	1.0
LE sweep in.	0.7128	0.7121	1.2
LE sweep out.	0.1205	0.1300	4.8
Max t/c loc.	4.8400	4.8502	0.6
LE radius	3.0660	2.4353	33.2
t/c ratio	1.9872	2.0304	5.4
Fuse. Radius 1	0.5176	0.5144	2.1
Fuse. Radius 2	0.5669	0.5699	2.0
Fuse. Radius 3	0.5659	0.5699	2.7
Fuse. Radius 4	0.4966	0.4928	2.5
Inboard nacelle	2.8293	2.8519	0.9
Fuel wt.	3.0938	3.1005	1.1

*defined in Figure 5

The HSCT problem has both physical local optima and numerical noise. The good performance of DIRECT and DOT for this problem is consistent with the fact that DOT handled well low amplitude noise for the Griewank test function, and DIRECT handled well widely separated local optima for the quartic test function. The wide fluctuations in the number of function evaluations required for DOT is an indication of the layout of the various design spaces. Presumably, the 10 DV and 20 DV cases have more disjointed optima while the 5 and 15 DV cases are more nearly convex. The poor performance of LFOPCV3 compared to the Griewank problem may have to do with the fact that the noise for the HSCT problem is not differentiable.

6. Concluding Remarks

Three optimization procedures were tested for the global optimization of a HSCT. DOT-- a local optimizer used with sequential quadratic programming--and LFOPCV3--a semi-global optimizer--were applied with random multistarts. DIRECT--a global Lipschitzian optimizer--was applied in a single run.

The three optimization procedures were first tested on two simple algebraic problems. The Griewank function is a convex function superimposed with low amplitude noise from a trigonometric function. The quartic test function does not have any noise but displays a large number of widely separated local optima. These two functions revealed that the two multistart procedures, DOT and LFOPCV3, dealt well with trigonometric noise, while DIRECT was much more effective in finding the global minimum among widely separated local optima.

The HSCT problem presents a combination of numerical noise and multiple physical optima, and thus has elements from both test functions. DIRECT and DOT alternated as the most efficient optimizer, with the number of function evaluations required by DOT fluctuating the most based on the number of good local optima it found. LFOPCV3 did poorly for the HSCT problem, possibly because of the effect of the noise on the derivatives that have to be calculated by its search procedure more frequently than by DOT.

7. Acknowledgement

This research was supported in part by NASA grant NAG2-1179.

8. References

- [1] Floudas C.A., Pardalos P.M., *State of the Art in Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996
- [2] Haim D., Giunta A.A., Holzwarth M.M., Mason W.H., Watson L.T., Haftka R.T., 'Comparison of Optimization Software Packages for an Aircraft Multidisciplinary Design Optimization Problem', *Design Optimization*, 1999, Vol. 1, pp. 9-23.
- [3] Jones D.R., Perttunen, C.D., and Stuckman, B.E. 'Lipschitzan Optimization Without the Lipschitz Constant', *Journal of Optimization Theory and Application*, 1993, Vol.79, pp. 157-181.
- [4] Cramer E.J., 'Using Approximate Models for Engineering Design', 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St Louis, Missouri, AIAA Paper-98-4716, Sept., 1998
- [5] Nelson II S.A., Papalambros P.Y., 'A Modification to Jones' Global Optimization Algorithm for Fast Local Convergence', 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St Louis, Missouri, AIAA Paper-98-4751, Sept., 1998
- [6] Knill D.L., Giunta A.A., Baker C.A., Grossman B., Mason W.H., Haftka R.T., Watson L.T., 'Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design', *Journal of Aircraft*, 1999, Vol. 36, pp.75-86.
- [7] *DOT: Design Optimization Tools*, Vanderplaats Research & Development, Inc., 1995, Version 4.20
- [8] Snyman, J.A. 'An Improved version of the original Leap-Frog dynamic method for unconstrained minimization: LFOP1(b)', *Appl. Math. Modelling*, 1983, Vol.7, pp. 216-218.
- [9] MacMillin P.E., Golovidov, O.B., Mason, W.H., Grossman, B., and Haftka, R.T., 'An MDO Investigation of the Impact of Practical Constraints on an HSCT Configuration', AIAA 35th Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 97-0098, Jan., 1997
- [10] McGrory, W.D., Slack, D.C., Applebaum, M.P., and Walters, R.W., *GASP Version 2.2 Users Manual*, Aerosoft, Inc., Blacksburg, VA, 1993
- [11] Carlson, H. W., and Miller, D. S., 'Numerical Methods for the Design and Analysis of Wings at Supersonic Speeds', NASA TN D-7713, Dec., 1974
- [12] Carlson, H.W. , and Walkley, K.B., 'Numerical Methods and a Computer Program for Subsonic and Supersonic Aerodynamic Design and Analysis of Wings with Attainable Thrust Corrections', NASA CR-3808, 1984

SADDLE POINTS IN DESIGN OPTIMIZATION

K.J. Craig and D.J. de Kock

Multidisciplinary Design Optimization Group (MDOG)

Department of Mechanical and Aeronautical Engineering

University of Pretoria, Pretoria 0002 South Africa

Email: ken.craig@eng.up.ac.za

ABSTRACT

This paper describes the use of saddle points in engineering design optimization problems. With saddle points are meant points in design space where the objective function is maximized with respect to some design variables while it is minimized with respect to other design variables simultaneously. For saddle points to exist, the maximization variables have to be separable from the minimization variables, otherwise a transformation is necessary. The philosophy behind the saddle-point optimization approach is described first, after which the approach is illustrated through two diverse design optimization problems. Both case studies employ the coupling of Computational Fluid Dynamics (CFD) with a version of the DYNAMIC-Q optimization algorithm of Snyman, modified to locate saddle points instead of minima. The case studies illustrate the power of this approach in that the influence of environmental variables is taken into account automatically in the design of a plant.

INTRODUCTION

The general plant optimization process can be depicted as in Figure 1. Shown are the successive steps that are typically followed when a new plant is first designed using mathematical modelling techniques, then built, commissioned, and optimized during operation and upgrades. Also shown in Figure 1 is the influence of environmental, or random, variables. These variables represent those over which the engineer has no control. The influence of these variables is usually countered using feedback or even optimal control techniques, but the ability of the plant to reject this influence may be limited due to the fixed nature of the plant already in operation. In this paper, it is shown that it would be advantageous to consider these variables in a worst-case scenario framework during the mathematical modelling phase using so-called saddle points. The main advantage is that the plant is not yet built at this stage, implying that changes are cheaper to implement. Examples of these environmental variables are environmental temperature, plant operating parameters that are not measured or controlled, or unknown disturbances, etc. The influence of these variables would typically be rejected using an optimal control strategy when operating a plant, but when the plant is designed, and major decisions as to plant lay-out and geometrical parameters are still being made, their effect could be minimized in the optimal design process or synthesis described in this paper.

To the best of the authors' knowledge, no systematic algorithm exists that is specifically tailored to search for saddle points instead of the usual minima. As the optimization method used in this study is gradient-based, it is relatively straightforward to modify it to search for saddle points instead of minima. This modification has been implemented by other authors using this algorithm [1,2]. In the first instance, it was used during the optimization of chemical molecular structures, while in the second study, it was used in the optimal synthesis of planar mechanisms. In the latter case, the eigenvalues of the Hessian matrix (second-order derivatives of the objective function with respect

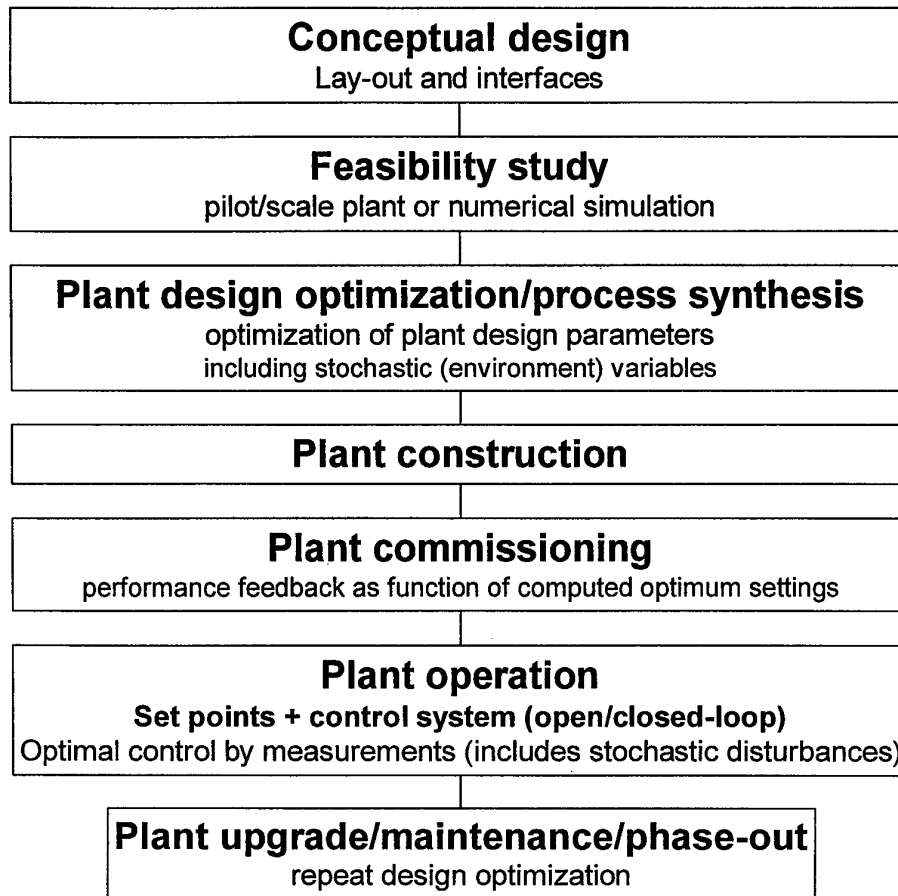


Figure 1 – General plant optimization process

to the design variables) was used to transform the problem by rotating the variable axes, and switching the sign of the gradients corresponding to the negative eigenvalues, such that a saddle point could be found.

The first case study presented in this paper, that of urban air pollution minimization, was the first implementation of saddle points performed by the authors [3,4,5]. In this study, the geometrical plant variables considered are those of the lay-out and configuration of an idealized urban design, as represented by street width and building height. The environmental (random) variables considered are the wind direction and speed.

The second case study presented extends the concept to the continuous casting field. The plant to be designed is that of the tundish of a continuous caster. The tundish acts as both a reservoir and a flow control device during the casting of liquid steel. The geometrical variables chosen for this study are related to the configuration and position of the so-called furniture in the tundish. The environmental (or saddle-point) variable is the steel inlet temperature to the tundish. This variable is chosen as it is not measured directly in the plant and typically has some unknown variation that could have a large impact on the flow patterns due to the density-driven buoyancy in the tundish.

The next section will define the saddle-point optimization problem and its solution methodology. This is followed by two case studies, each with its own formulation, theoretical modelling, and results section. Conclusions made from the results conclude the paper.

SADDLE-POINT OPTIMIZATION

The problem to be solved is as follows. Determine $(\mathbf{x}^*, \mathbf{y}^*)$ such that

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}^*) &\geq f(\mathbf{x}^*, \mathbf{y}^*) \geq f(\mathbf{x}^*, \mathbf{y}) \\ \text{subject to } x_j^{\min} &\leq x_j \leq x_j^{\max}, j=1,2 \text{ and } y_j^{\min} \leq y_j \leq y_j^{\max}, j=1,2 \end{aligned} \quad (1)$$

Assume that the solution lies at the saddle point $(\mathbf{x}^*, \mathbf{y}^*)$, and that f is convex with respect to $\mathbf{x}$, and concave with respect to $\mathbf{y}$. If these two sets of variables are separable (i.e., $f(\mathbf{z}) = f(\mathbf{x}, \mathbf{y}) = f_1(\mathbf{x}) + f_2(\mathbf{y})$, so that they do not have a significant cross-influence on each other), then this problem becomes tractable as outlined below.

The reader is referred to Refs. 4-15 for the detail implementation of the optimization method used in this study. In essence, the DYNAMIC-Q method of Snyman et al [9,15] involves the application of a dynamic trajectory method for unconstrained optimization [7,8], adapted to handle constrained problems through appropriate penalty function formulations. This DYNAMIC method is applied to successive approximate Quadratic subproblems that are constructed from sampling, at relative high computational expense, the behaviour of the objective function at successive approximate solution points in the design space. The subproblems, which are analytically simple, are solved quickly and economically using the adapted dynamic trajectory method.

To illustrate the implementation of DYNAMIC-Q in the determination of saddle points, consider the unconstrained problem:

$$\underset{\mathbf{w}, \mathbf{r}, \mathbf{t}, \mathbf{x}}{\text{minimize}} \left\{ \underset{\mathbf{w}, \mathbf{r}, \mathbf{t}, \mathbf{y}}{\text{maximum}} f(\mathbf{x}, \mathbf{y}) \right\}, \mathbf{x} \in R^p, \mathbf{y} \in R^q \quad (2)$$

If the values of $f(\mathbf{x}, \mathbf{y})$ and the associated gradient vectors are known at two successive design points $(\mathbf{x}^{(k-1)}, \mathbf{y}^{(k-1)})$ and $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$, then the spherical quadratic approximation to the objective function is taken as

$$\tilde{f}(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) + \nabla_{\mathbf{x}}^T f^k (\mathbf{x} - \mathbf{x}^{(k)}) + \nabla_{\mathbf{y}}^T f^k (\mathbf{y} - \mathbf{y}^{(k)}) + \frac{1}{2} a^{(k)} \|\mathbf{x} - \mathbf{x}^{(k)}\|^2 - \frac{1}{2} a^{(k)} \|\mathbf{y} - \mathbf{y}^{(k)}\|^2 \quad (3)$$

with $a^{(k)}$ given by

$$a^{(k)} = \frac{2\{f^{(k-1)} - f^k - \nabla_{\mathbf{x}}^T f^k (\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}) - \nabla_{\mathbf{y}}^T f^k (\mathbf{y}^{(k-1)} - \mathbf{y}^{(k)})\}}{\{\|\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}\|^2 - \|\mathbf{y}^{(k-1)} - \mathbf{y}^{(k)}\|^2\}}$$

where $f^k = f(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$

For application in DYNAMIC-Q, the saddle point of a fictitious function $fic(\mathbf{x}, \mathbf{y})$ is found if the gradients in the dynamic trajectory algorithm are chosen as follows:

$$\nabla_x fic := \nabla_x f^k + a^{(k)}(x - x^{(k)}) \quad (4)$$

and

$$\nabla_y fic := -\nabla_y f^k + a^{(k)}(y - y^{(k)}) \quad (5)$$

By switching the sign of the actual gradient with respect to y of $\tilde{f}$ in (3), as shown in the latter expression (5), the algorithm, *since it uses only gradient information*, effectively minimizes a 'fictitious' convex function $fic(x, y)$, the minimum of which $((x^{(k+1)}, y^{(k+1)}))$ corresponds to the saddle point of $\tilde{f}(x, y)$.

The computed function and gradient values at the point $z^{(k+1)} = (x^{(k+1)}, y^{(k+1)})$ may now be used to construct the next approximation $\tilde{f}(x, y)$ to $f(x, y)$, as given by (3) but for $(x^{(k+1)}, y^{(k+1)})$, and the approximation procedure is continued until convergence is obtained to (x^*, y^*) . It is further assumed that this procedure, as applied to the objective function above, is also valid in the presence of constraints with, of course, no such adjustment ((4) and (5)) applied to the gradients of the constraint functions.

In the next sections, this procedure is applied to two diverse test cases.

CASE 1 : OPTIMIZATION OF URBAN GEOMETRY TO MINIMIZE THE EFFECT OF AUTOMOTIVE POLLUTION

Problem definition and formulation

Refer to Refs. 3, 4 for a detailed motivation and background to this problem. In short, the idealized urban automotive air pollution optimization problem shown in Figure 2 is solved. Shown is a 3x3 array of buildings with two streets in each of the two main directions. The buildings are square-shaped when viewed from above, and the streets are assumed to have a constant width with uniform spacing. The geometrical design variables considered are street width and building height. The size of the urban section considered is 500m by 500m, while an up- and downwind fetch of 5km and 300m high is modelled.

The two urban geometry design variables are shown in Figure 2. $x_1 = w$ is the street width and $x_2 = h$ is the building height, i.e. $x = (x_1, x_2)$. Due to the symmetric lay-out of the geometry, the range of wind directions that needs to be considered is only one quarter of all wind directions, i.e. from 0° to 45° in Figure 2. The atmosphere is considered as neutrally stable in this study. The wind profile is assumed to be steady and to follow a power law distribution. In general, pollutant episodes with transient wind conditions can be considered if instantaneous pollutant levels are required. The current type of analysis is more suited for averaged or accumulative pollutant scenarios. The wind direction, α and wind speed V are variables that affect the pollutant level and are allowed to vary continuously between specified limits. The aim is to find values of the variables $y_1 = \alpha$ and $y_2 = V$, i.e., $y = (y_1, y_2)$, that would produce the worst-case scenario as far as CO pollutant concentration level is considered, and then minimizing this level by varying the geometrical variables, x . The objective function, $f(x, y)$ (refer to Eq. 2), is the CO level obtained from a Computational Fluid Dynamics (CFD) simulation given a specific geometry (defined by x) at a 2-m level in the streets for the specified wind speed and direction (defined by y). Refer to Refs.3 and 4 for a detailed description of the CFD modelling performed. The numerical grid contained approximately 300 000

cells, and the commercial CFD code, STAR-CD [16], solving for the Reynolds-average Navier-Stokes equations with turbulent closure provided by the $k-\epsilon$ turbulence model, was employed in the solution of, first, the steady-state wind field, and then the dispersion and convection of the pollutant, CO.

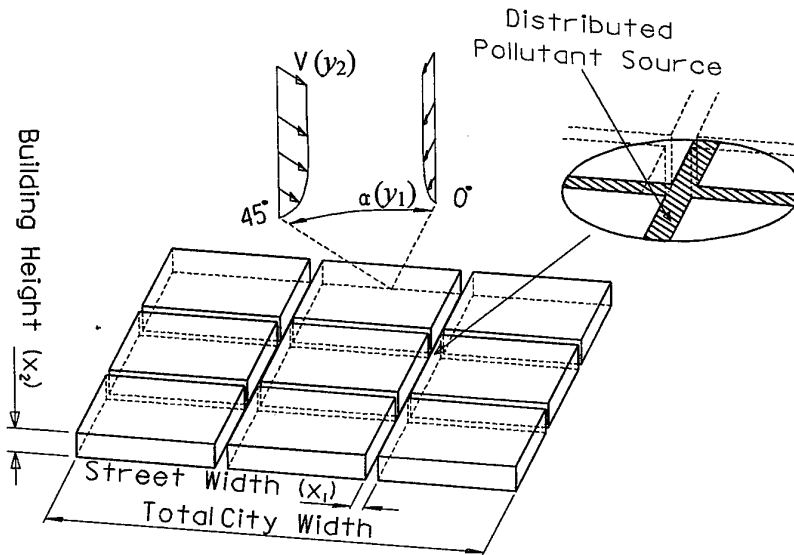


Figure 2 – Graphical representation of automotive urban air pollution problem

Results

The histories of the objective function as well as that of the design variables for Case 1 are given in Figure 3. The first design has the lowest CO pollution value, but it does not represent a worst-case scenario as far as wind spend and direction are concerned. The saddle point (where the objective function is minimized with respect to the geometrical variables, and maximized with respect to the meteorological (wind) variables), is essentially found in 6 iterations.

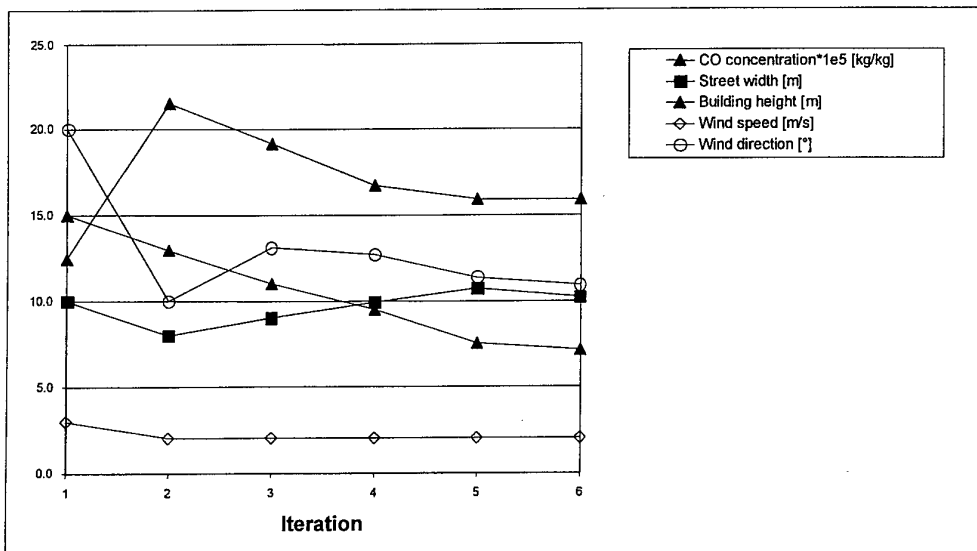


Figure 3 – History of CO concentration, wind speed and direction, and street width and building height. CO source: 15g/km/lane (Initial design: $\alpha = 20^\circ$; $V = 3.000 \text{ m.s}^{-1}$, $w = 10 \text{ m}$, $h = 15 \text{ m}$)

CASE 2 : MINIMIZATION OF TUNDISH DEAD VOLUME BY VARYING BAFFLE CONFIGURATION AND POSITION FOR RANDOM TEMPERATURE VARIATION

Problem definition and formulation

A diagrammatic view of the continuous casting process is given in Figure 4. The figure shows the position of the tundish relative to the other components of the caster. The molten steel is poured from the ladle into the tundish. The steel flows through the tundish and exits the tundish through the submerged entry nozzle (SEN) into the mould. The purpose of the tundish is to remove impurities and as well as to act as a reservoir during a ladle change.

Time is normalised by the theoretical average mean residence time ($\bar{t} = V/Q$ = tundish volume/flow rate through tundish) for easy comparison. For the tundish under consideration this time was calculated to be, $\bar{t} = 38.78$ sec. The tundish geometry considered is shown in Figure 5. This geometry corresponds to the single-strand stainless steel continuous caster in operation at the Middelburg plant of Columbus Stainless. Only a two-dimensional centre-line section of the tundish is considered here. Related work by the authors [18-20] considered three-dimensional effects. The flow rate is controlled by the position of a stopper in the outlet (not shown). When injecting a tracer element at the shroud (inlet), this tracer is detected at the SEN (outlet) after a certain amount of time. The concentration at the outlet increases to a maximum where after it decreases. This time history of the tracer concentration as measured at the outlet is called the residence time distribution (RTD). The rate of decay is of interest since it correlates with the ratio of plug flow versus mixed flow in the tundish.

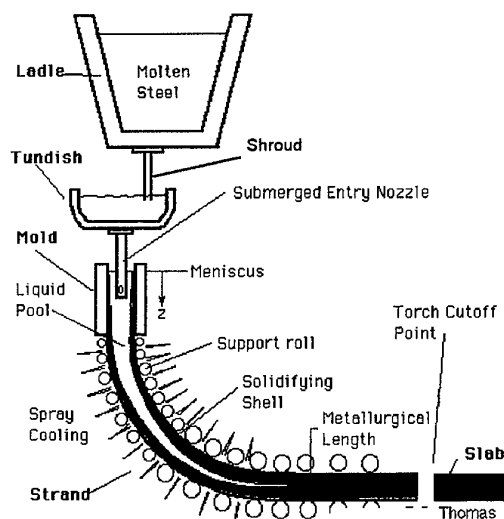


Figure 4 – Diagram of the continuous casting process [17]

The amount of tracer left in the tundish after $2\bar{t}$ (twice the mean residence time) is defined in this study as the dead volume, and is minimized in the optimization problem, i.e. it is the objective function, f = dead volume = (1 - concentration at $t = 2\bar{t}$), in Eq.2. The geometrical variables ($x = (x_1, x_2)$) are shown in Figure 5, together with the environmental or saddle-point variable ($y = (y_1)$). The objective function is minimized with respect to the former, while it is maximized with respect to the latter. The location of the baffle shown is the starting design, i.e. $x_1 = 0$.

The evaluation of the objective function is done through a CFD simulation with the commercial CFD code, Fluent and its pre-processor, Gambit [21,22]. The realizable $k-\epsilon$ turbulence model is used, while the influence of temperature is modelled through a Boussinesq approximation. A new 2-D mesh is obtained in each iteration for the tundish based on the geometry prescribed by the optimizer. The environmental variable is posed as part of the boundary condition specification in Fluent. The mesh is solution-adapted using y^+ and velocity gradients. A steady-state solution is first obtained, where after only the convection and dispersion due to a step input of a tracer scalar are solved for in a transient simulation. The result is a concentration history at the tundish outlet (SEN), of which a sample is shown in Figure 6.

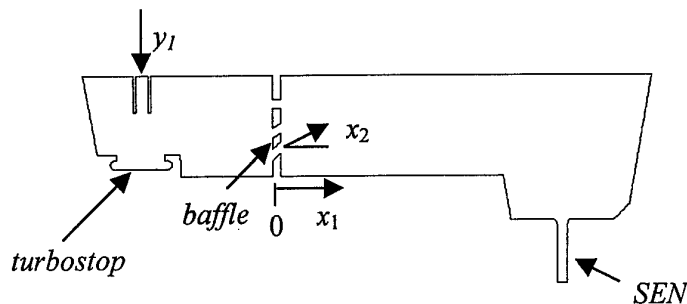


Figure 5 – Side view of tundish showing baffle position and configuration

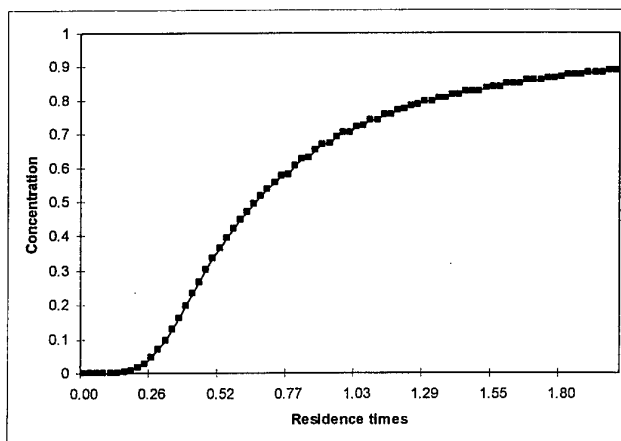


Figure 6 – Outlet concentration of scalar due to step input

Results

The optimization history for first a case excluding the saddle-point variable and then a case including it, are shown in Figures 7 and 8, respectively. For the two-variable minimization problem in Figure 7, the inlet temperature is taken at a constant value of 1850K, while the temperature of the surroundings is held at 310K. Note how the optimizer pushes the baffle towards the inlet, while a maximum baffle angle of around 20° appears to be the optimum. The variable ranges allowed were $x_1 = (-500\text{mm}, 1000\text{mm})$ and $x_2 = (0^\circ, 50^\circ)$. Note that the dead volume decreases by about 1% from over 11% to just over 10%.

When the saddle-point variable (inlet temperature) is included, the optimum geometrical design tends towards $x_1 = -500\text{mm}$ and $x_2 = 50^\circ$ (minimum and maximum bounds, respectively), while the inlet temperature tends towards an average value of 1775K.

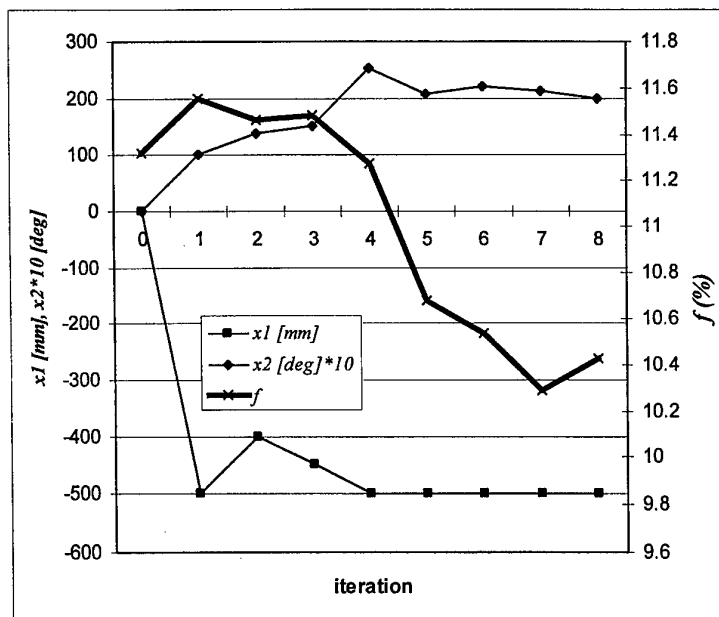


Figure 7 – History of dead volume (f), relative baffle position (x_1) and maximum baffle hole angle (x_2)

The higher baffle angle can be explained by the fact that the lower inlet temperature has reduced the buoyancy of the steel in the tundish. Previous research has shown that one of the main contributors to dead volume is the region in the corner above the outlet, and this region is only reached if a 'jet' is directed towards it. In the current design, the direction of this jet is determined both by the baffle position and the baffle hole angles, and this combination is influenced by the temperature patterns in the tundish.

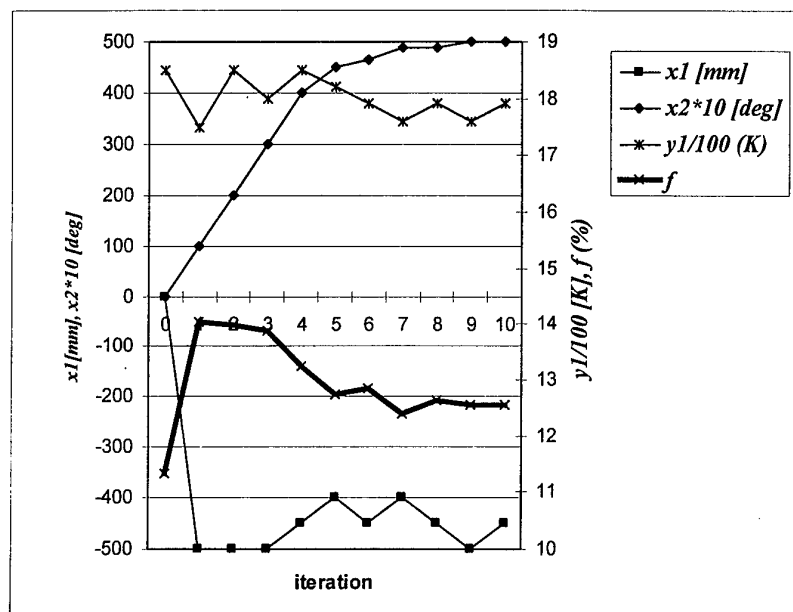


Figure 8 – History of dead volume (f), relative baffle position (x_1), maximum baffle hole angle (x_2), and inlet temperature (y_1)

With the addition of the saddle-point variable, the dead volume minimization shown in Figure 7 now becomes irrelevant, as a fixed inlet temperature of 1850K was used there. The aim is now to find the best design for a worst-case inlet temperature scenario. The difference in dead volume percentages for similar geometrical designs (x_1, x_2) between the two cases is due to the fact that the reduction in inlet temperature brought about by the simultaneous maximization (worst-case) process, changes the flow field to such an extent that the dead volume is increased to the value of 12.5% as shown in Figure 8. The temperature of the surroundings is held at 293K in the Figure 8 results. The oscillatory nature of the baffle distance and the inlet temperature history is indicative of numerical noise that exists in the design space of this problem. Both the cause and treatment of this noise are being researched further [23].

CONCLUSION

This study illustrates the combination of CFD and mathematical optimization to find saddle points for plants where the influence of random variables is to be taken into account. The approach is shown to provide interesting results for two diverse case studies in the fluid flow and heat transfer field and can easily be extended to study the influence of random variables in other fields.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of Columbus Stainless, Iscor and the Department of Trade and Industry under THRIP grant GUN2043307.

REFERENCES

1. Smith, C.M., How to find a saddle point, *Int. J Quantum Chemistry*, Vol. 37, p.773, 1990.
2. Minnaar, R.J., Optimal dimensional synthesis of planar mechanisms, MEng thesis, Department of Mechanical Engineering, University of Pretoria, Pretoria, South Africa, April 1999.
3. Craig, K.J. & de Kock, D.J., Optimization of urban geometry to minimize the effect of automotive pollution, *Proceedings of the 7th International Conference on Air Pollution, Air Pollution VII*, 27-29 July 1999, Palo Alto, USA. Editors: C.A. Brebbia, M. Jacobson & H. Power, WIT Press, pp.331-340.
4. Craig, K.J., de Kock, D.J. & Snyman, J.A., Optimization of urban geometry to minimize the effect of automotive pollution, in press, *Atmospheric Environment*, May 2000.
5. Craig, K.J., de Kock, D.J. & Snyman, J.A., Optimization of urban geometry to minimize the effect of automotive pollution using saddle points, *Proceedings of SACAM2000, International Conference on Applied Mechanics*, Durban, South Africa, January 11-13, 2000.
6. Craig, K.J., de Kock, D.J. & Snyman, J.A., Using CFD and mathematical optimization to investigate air pollution due to stacks, *Int. J. Num. Methods Engng.*, Vol. 44, pp. 551-565, 1999.
7. Snyman, J.A., A new dynamic method for unconstrained minimization, *Appl. Math. Modelling*, Vol. 6, pp. 449-462, 1982.
8. Snyman, J.A., An improved version of the original leap-frog dynamic method for unconstrained minimization LFOP1(b), *Appl. Math. Modelling*, Vol.7, pp. 216-218, 1983.
9. Snyman, J.A., Stander, N. & Roux, W.J., A dynamic penalty function method for the solution of structural optimization problems, *Appl. Math. Modelling*, Vol.18, pp. 453-460, 1994.
10. Craig, K.J., Venter, P.J., de Kock, D.J. & Snyman, J.A., Optimization of structured grid spacing parameters for separated flow using mathematical optimization, *J. Wind Engineering and Industrial Aerodynamics*, Vol. 80, pp.221-231, 1999.

11. Craig, K.J. & Venter, P.J., Optimization of the k- ϵ coefficients for separation on a high-lift airfoil, AIAA Paper 99-0151, 37th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 11-14, 1999.
12. Craig, K.J., de Kock, D.J. & Gauché, P., Minimization of heat sink mass using CFD and mathematical optimization, *ASME Journal of Electronic Packaging*, Vol.121 No.3, pp.143-147, September 1999.
13. De Kock, D. J., Craig, K. J. & Snyman, J. A., Using mathematical optimization in the CFD analysis of a continuous quenching process. *Int. J. Num. Methods Engng.*, Vol. 47, pp.985-999, 2000.
14. Craig, K.J., Snyman, J.A., Venter, P.J. & de Kock, D.J., The development of a Toolkit for Design Optimization (TDO) for applied mechanics applications, *Proceedings of SACAM2000, International Conference on Applied Mechanics*, Durban, South Africa, January 11-13, 2000.
15. Snyman, J.A. & Hay, A.M., The DYNAMIC-Q optimization method: An alternative to SQP?, *Proceedings of the International Workshop on Multidisciplinary Design Optimization*, Pretoria, South Africa, 8-10 August 2000.
16. STAR-CD, Version 3.05 manuals, Computational Dynamics Ltd., London, United Kingdom, 1998.
17. Thomas BG, 'Introduction to continuous casting', *Continuous Casting Consortium*, URL: <http://bgtibm1.me.uiuc.edu>, 1999.
18. De Kock, D.J., Craig, K.J. & Venter, P.J., Optimization of tundish configuration for a single-strand stainless steel continuous caster, *Proceedings of SACAM2000, International Conference on Applied Mechanics*, Durban, South Africa, January 11-13, 2000.
19. Craig, K.J., de Kock, D.J. & Snyman, J.A., Continuous casting optimization using CFD and DYNAMIC-Q, ICTAM2000, Chicago, USA, August 27 – September 1, 2000.
20. De Kock, D.J. & Craig, K.J., Optimal tundish design using CFD with inclusion modelling, *Proceedings of the International Workshop on Multidisciplinary Design Optimization*, Pretoria, South Africa, 8-10 August 2000.
21. FLUENT, Version 5.3, Fluent Inc., Lebanon, NH, USA, 2000.
22. GAMBIT, Version 1.2, Fluent Inc., Lebanon, NH, USA, 2000.
23. Craig, K.J., The use of saddle points in the optimal design of continuous casting tundishes, in preparation.

OPTIMAL TUNDISH DESIGN USING CFD WITH INCLUSION MODELLING

D.J. de Kock, K.J. Craig

*Multidisciplinary Design Optimization Group (MDOG)
Department of Mechanical and Aeronautical Engineering
University of Pretoria, Pretoria 0002, South Africa
Tel. +27(12) 420-2448 Fax. +27(12) 420-2451
email: djdekock@postino.up.ac.za*

ABSTRACT

The tundish in the continuous casting process plays an important role in determining the quality of the steel and the mixed-grade length during a grade change. The tundish configuration (i.e. the position and sizes of dams, weirs, baffles and/or striker pad) and the operating levels determine the flow patterns inside the tundish. These variables can be adjusted by the engineer to improve the flow through the tundish. This is usually done on-site in an experimental trial-and-error basis with little knowledge of the detail flow patterns inside the tundish. Water models are traditionally used to study the flow patterns inside the tundish. An alternative is to use Computational Fluid Dynamics (CFD) in the numerical investigation of the tundish flow. This paper combines CFD with mathematical optimisation to improve the flow inside the tundish. The Columbus single-strand continuous caster with one dam and one weir is investigated in this paper. Two optimisation cases are considered in this study. The position of the dam and weir are optimised, first to minimise the dead volume inside the tundish and secondly to maximise the inclusion removal at the slag layer. The optimisation is achieved by coupling the commercial CFD solver, FLUENT, with the DYNAMIC-Q optimisation algorithm of Snyman in an automated fashion.

INTRODUCTION

The tundish plays an important role in the quality and productivity of the caster. Extensive research has been performed to improve tundish flow [1,2,3]. The flow improvements are achieved using dams, weirs and striker pads (collectively called flow control devices). In these studies, water modelling and CFD have been used to investigate the flow and the influence of these flow control devices on the flow field. These kinds of studies mainly involve trial-and-error designs of flow control configurations. The current study uses CFD combined with mathematical optimisation to find the optimum position of one dam and one weir for two different cases. This is achieved in an automatic fashion in this study, by linking the commercial CFD code FLUENT [4] and its pre-processor GAMBIT [5] to the DYNAMIC-Q algorithm of Snyman et al. [6], in which a gradient method for constrained optimisation is applied to successive approximate quadratic subproblems [7,8].

The optimisation problem is defined in the next section, giving the necessary background on the flow inside the tundish. This is followed by a brief presentation of the theoretical model of the tundish flow as well as the optimisation method used here. The results of the CFD analysis and of the optimisation procedure are discussed next. The paper ends with conclusions drawn from the study and a brief discussion of future work.

PROBLEM DEFINITION AND FORMULATION

A diagrammatic view of the continuous casting process is given in Figure 1. The figure shows the position of the tundish relative to the other components of the caster. The molten steel is poured from the ladle into the tundish. The steel flows through the tundish and exits the tundish through the submerged entry nozzle (SEN) into the mould. The purpose of the tundish is to remove impurities and to act as a reservoir during a ladle change.

s

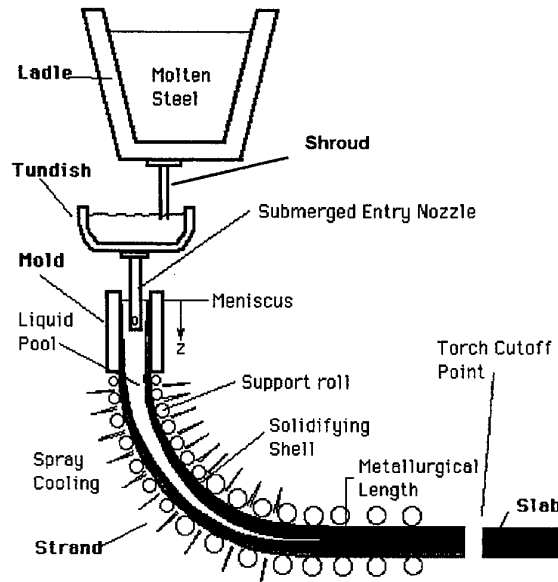


Figure 1: Diagram of the continuous casting process [9]

Time is normalised by the theoretical average mean residence time ($\bar{t}$) for the tundish for easy comparison [10]:

$$\bar{t} = V / Q \quad (1)$$

where V = tundish volume
 Q = flow rate through tundish

For the tundish under consideration this time was calculated to be, $\bar{t} = 588 \text{ sec}$.

The tundish geometry considered is shown in Figure 2. This geometry corresponds to the single-stand stainless steel continuous caster in operation at the Middelburg plant of Columbus Stainless. The flow rate is controlled by the position of the stopper. The design variables ($x_1; x_2$) are indicated in the figure. Both weir (x_1) and dam distance (x_2) are referenced to the left-most lower corner of the tundish.

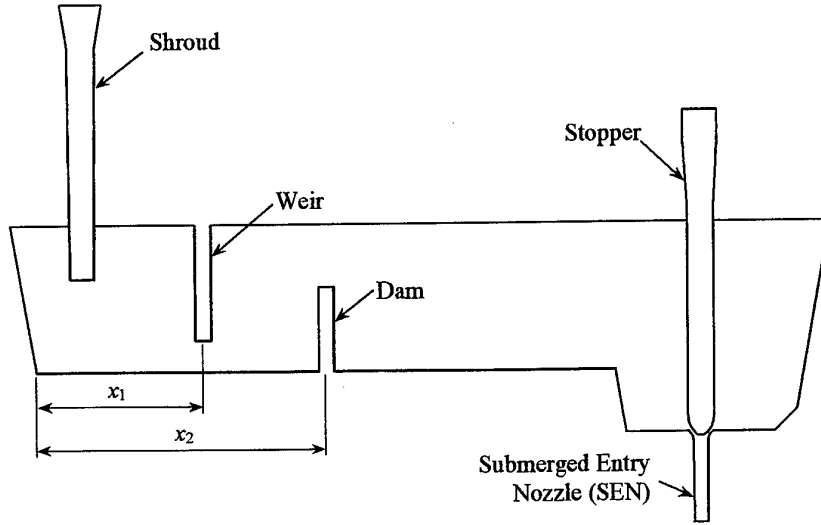


Figure 2: Side view of tundish showing dam, weir, stopper, SEN and shroud

Case 1: Dead Volume Minimisation

When injecting a tracer element at the shroud (inlet), this tracer is detected at the SEN (outlet) after a certain amount of time. The concentration at the outlet increases to a maximum where after it decreases. This time history of the tracer concentration as measured at the outlet is called the residence time distribution (RTD). The rate of decay is of interest since it correlates with the ratio of plug flow volumes versus mixed flow and dead flow volumes. It is desirable to have high a plug flow volume compared to the mixed flow and dead flow volumes. The amount of tracer left in the tundish after $2\bar{t}$ (twice the mean residence time) is defined in this study as the dead flow volume, and is minimised in the first optimisation problem.

The complete mathematical formulation of the first optimisation problem, in which the inequality constraints are written in the standard form $g_j(\mathbf{x}) \leq 0$, where $\mathbf{x}$ denotes the vector of the design variables $(x_1, x_2)^T$, is as follows:

$$\begin{aligned} &\text{minimise } f(\mathbf{x}) = \text{dead volume} \\ &\quad = 1 - (\text{concentration at } t = 2\bar{t}) \\ &\text{subject to: } \begin{cases} g_1 = -x_1 + x_1^{\min} \leq 0 \\ g_2 = x_1 - x_2 + \delta \leq 0 \\ g_3 = x_2 - x_2^{\max} \leq 0 \end{cases} \end{aligned} \quad (2)$$

where δ is the minimum allowable distance between the dam and the weir

Case 2: Inclusion Removal Optimisation

Another important role of the tundish is to remove any inclusions that may be present in the steel. In the second optimisation study, the inclusions that escape through the SEN are minimised. Inclusions are released at the inlet of the tundish and their trajectories were calculated to see if they escape through the SEN. The slag layer is assumed to remove an inclusion from the flow if the inclusion came into contact with the slag layer.

The complete mathematical formulation of the second optimisation problem, again writing it in the standard form, is as follows:

$$\begin{aligned} &\text{minimise } f(\mathbf{x}) = \frac{\text{number of escaped particles}(\mathbf{x})}{\text{number of injected particles}} \\ &\text{subject to: } \begin{cases} g_1 = -x_1 + x_1^{\min} \leq 0 \\ g_2 = x_1 - x_2 + \delta \leq 0 \\ g_3 = x_2 - x_2^{\max} \leq 0 \end{cases} \end{aligned} \quad (3)$$

where δ is the minimum distance between the dam and the weir

THEORETICAL MODELLING

CFD Modelling

As mentioned above, the commercial CFD code, FLUENT [4] and its pre-processor GAMBIT [5] are used in this study. This software is used to solve the Reynolds-Averaged Navier-Stokes equations with turbulent closure provided by the standard k- ϵ turbulence model. The liquid steel is modelled as water because of the similarity in terms of Reynolds and Froude numbers for steel and water. This is also done in order to verify the results with those obtained in a water model of the tundish [11].

Because of symmetry, only half the tundish is solved by applying a symmetry plane boundary condition on the centre plane of the tundish. A mass flow inlet is used at the shroud and an outflow boundary is used at the outlet of the SEN. A slip surface is used at the top boundary of the steel to model the slag layer on top of the steel.

The inclusions are modelled using a Lagrangian approach of tracking the inclusions through the flow domain. This is a built-in model in FLUENT and takes into account the drag and buoyancy forces on the inclusions. In this study the inclusions are assumed to be perfectly spherical and they are also assumed to bounce perfectly on all the walls except the slag layer where the inclusions are trapped when they come into contact with the slag layer. The stochastic model available in FLUENT is also used to take into account the effect of turbulent dispersion of the inclusions. The inlet size distribution of the inclusions was obtained through an optical process from sampled plant data of the inclusion at the shroud location.

The tundish grid is automatically generated for all the design iterations using GAMBIT, the pre-processor of FLUENT. A fully unstructured grid is generated using tetrahedron elements. The mesh of the starting design configuration is shown in Figure 3 and consisted of approximately 180 000 cells. The high-density grid near the expected high gradient areas (i.e. at the inlet and outlet of the tundish) can clearly be seen.

For each design iteration, the CFD model is first converged to a steady-state solution. An unsteady solution is then obtained by solving a step input of a tracer element at the inlet. This transient calculation provides the concentration values at the outlet from which the %dead volume is calculated.

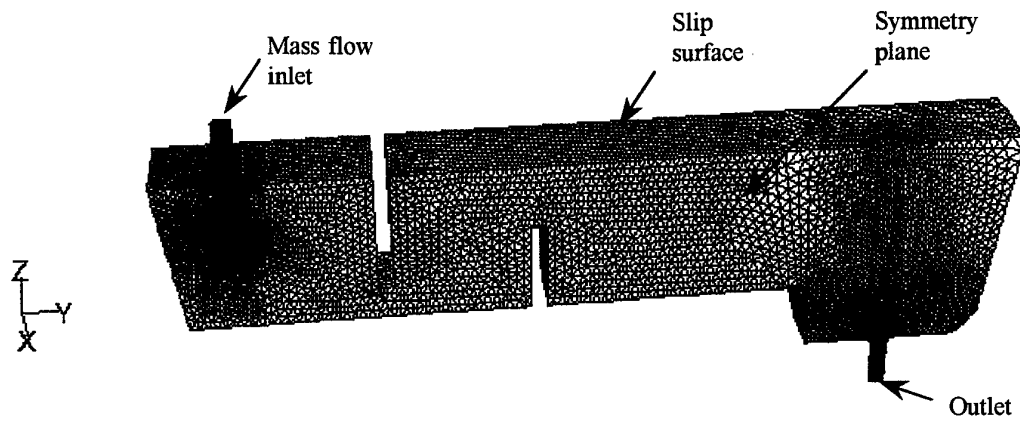


Figure 3: Three-dimensional view of unstructured mesh and boundary conditions
($x_1 = 0.9$; $x_2 = 1.5$)

Mathematical Optimisation

The optimisation method used in this study is the DYNAMIC-Q method of Snyman et al [6]. This approach involves the application of a dynamic trajectory method for unconstrained optimisation [12,13] adapted to handle constrained problems through appropriate penalty function formulations [6,14,15]. This DYNAMIC method is applied to successive approximate Quadratic subproblems [6,7,8] of the original problem (here problem (2) or (3)). The successive subproblems are constructed from sampling, at relative high computational expense, the behaviour of the objective function at successive approximate solution points in the design space. These subproblems, which are analytically simple, are solved quickly and economically using the adapted dynamic trajectory method giving a new approximate solution. This methodology is described in detail in [16,17,18].

RESULTS AND DISCUSSION

Case 1: Dead Volume Minimisation

Parameters

The parameters used in this study are given in Table 1. The limits on the design variables and move limits are shown in Table 2.

Table 1: Parameters used in the model

Mass Flow Rate [kg.s ⁻¹]	Tundish Volume [m ³]	Dead Volume Time [s]	Water density ρ_{water} [kg.m ⁻³]	Water Viscosity μ_{water} [kg.m ⁻¹ .s ⁻¹]	Minimum distance δ [m]
4.75	2.05	$2t = 1176$	998	0.00103	0.2

Table 2: Upper and lower limits and move limits on the design variables

	Weir Position (x_1) [m]	Dam Position (x_2) [m]
Minimum	0.4629	-
Maximum	$x_2 - 0.275$	2.7266
Move Limit	0.5	0.5

Flow Results

The outlet concentration as well as the RTD curve for the initial design iteration ($x_1 = 0.9, x_2 = 1.5$) as obtained with a transient simulation are shown in Figure 4. The curves end at $\bar{t} = 2.0$. The dead volume is obtained from 1 minus the concentration at $\bar{t} = 2.0$ in Figure 4a), since the integral of Figure 4b) is equal to 1.0.

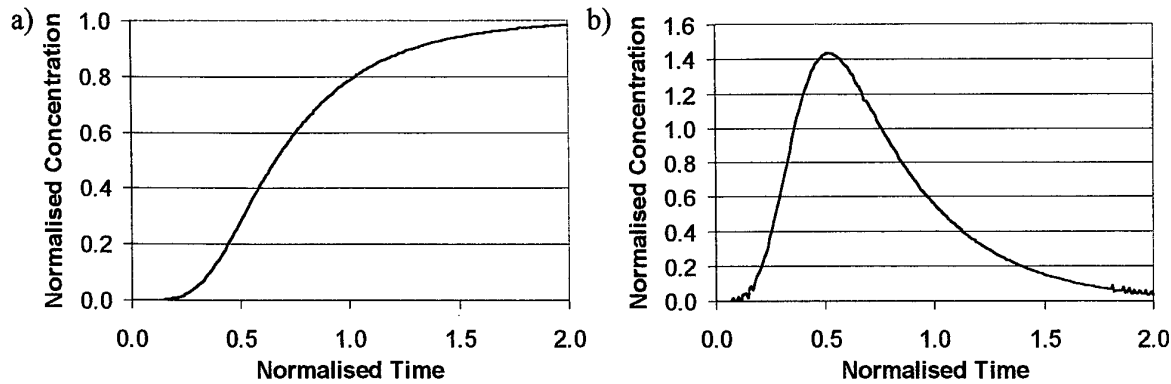


Figure 4: a) Concentration profile and b) RTD curve of starting configuration ($x_1 = 0.9, x_2 = 1.5$)

Optimisation Results

The results of the first optimisation run are shown in Figure 5. The history of the objective function is shown in Figure 5a) while the history of the design variables is shown in Figure 5b). A graphical representation of the optimum configuration is also shown in Figure 5b). It can be seen that the first potential optimum is found in iteration 4, where after the optimiser diverges, but then recovers to find a slightly better minimum. At iteration 4, the constraint on x_1 is only just not active, whereas from iteration 5 onwards, it remains active, with x_1 at its lowest allowable value. Physically, this means that the design pushes the weir as close to the inlet as possible, while it searches for the optimum position of the dam, relatively close to the outlet. The main deduction that can be made from the current optimisation results for this two-dimensional design problem, is that the dead volume is relatively insensitive to the position of the dam and weir. Although the dead volume has been reduced to nearly half its value in the initial design, the dead volume was very small to begin with.

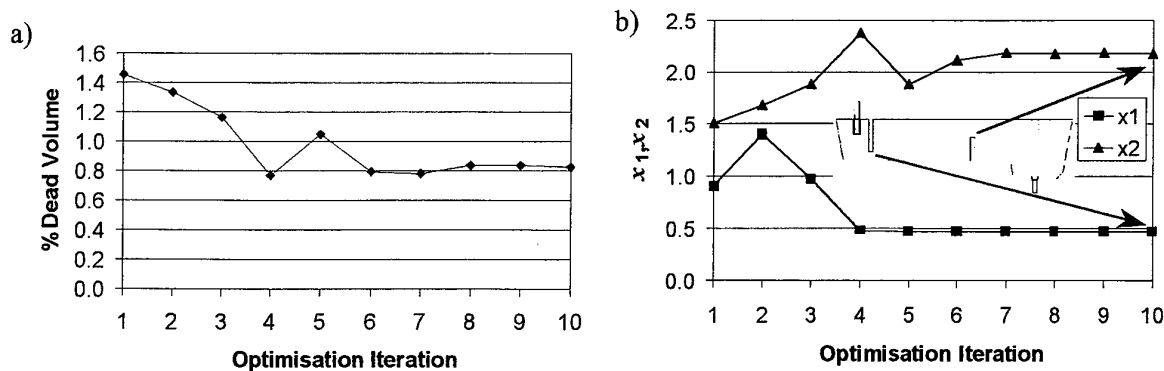


Figure 5: Optimisation history: a) Objective Function b) Design Variables

Case 2: Inclusion Removal Optimisation

Parameters

The same parameters and limits on the design variables are used as in Case 1. The only additional parameters considered is the inclusion size distribution, mass flow and density. The inclusion size distribution is given in Table 3. This data is acquired from taking a sample of the steel at the shroud on the plant and then using an image analyser to determine the size distribution. The mass flow is determined from the total percentage of inclusions in the sample and were calculated as 0.469g.s^{-1} for the whole tundish. The density of the inclusions was taken as 456kg.m^{-3} which is 2.2 times smaller than the density of the water. This represents the same density ratio of the alumina inclusions relative to the molten steel in the plant. A total of 2700 inclusions with the same size distribution as in Table 3 are released in the CFD simulation.

Table 3: Inclusions size distribution obtained from image analyser

Size [mm]	0.024	0.093	0.207	0.366	0.571
Number of particles	113	37.75	4.5	0.5	0.25

Inclusion Tracks

The inclusion tracks for the initial design are shown in Figure 6 for two of the inclusion sizes considered (i.e. 0.024mm and 0.207mm), released at one specific location for illustration purposes. It can be seen that all the larger inclusions are trapped at the slag layer before the weir due to their larger buoyancy forces. The smaller inclusions tend to follow the flow stream and exit at the SEN.

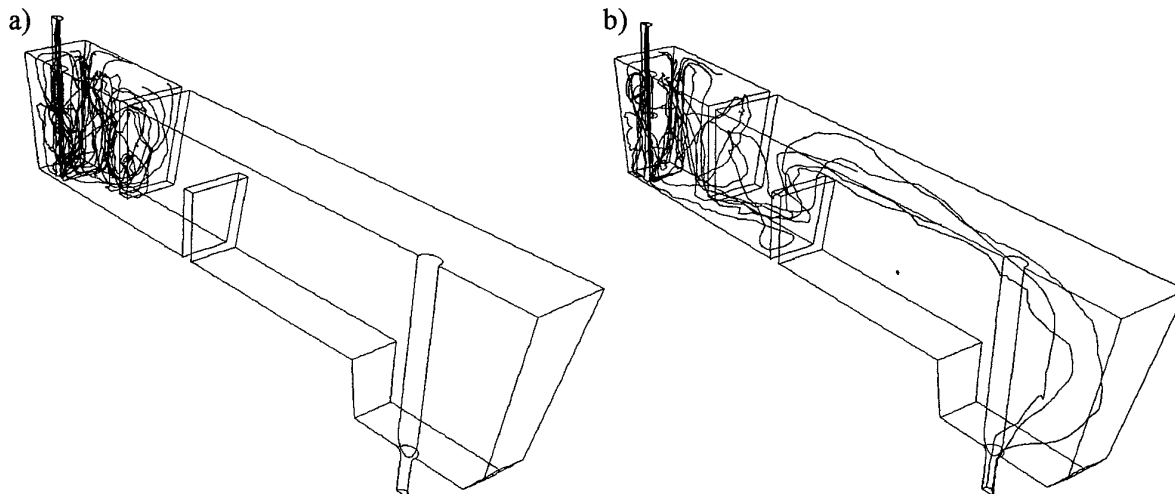


Figure 6: Particle tracks for a) 0.024mm b) 0.207mm size inclusions released at one specific location only (Starting configuration: $x_1 = 0.9$, $x_2 = 1.5$)

Optimisation Results

The history of the inclusion optimisation is shown in Figure 7. The history of the objective function is shown in Figure 7a), while the history of the design variables is shown in Figure 7b). It can be seen from Figure 7b) that the optimiser tends to hit the constraint of minimum distance between the dam and the weir, and then moves the dam and weir together as a pair. A graphical representation

of the optimum configuration is also shown in Figure 7b). This small distance between the dam and the weir creates a jet with a high velocity directed towards the surface providing a mechanism for a lot of particles to be trapped. This may however cause some entrainment of the slag layer, which was not modelled in this case. It can also be seen that this optimum differs completely from the optimum in the first case. The main reason for this is that the two objective functions represent two opposing design criteria.

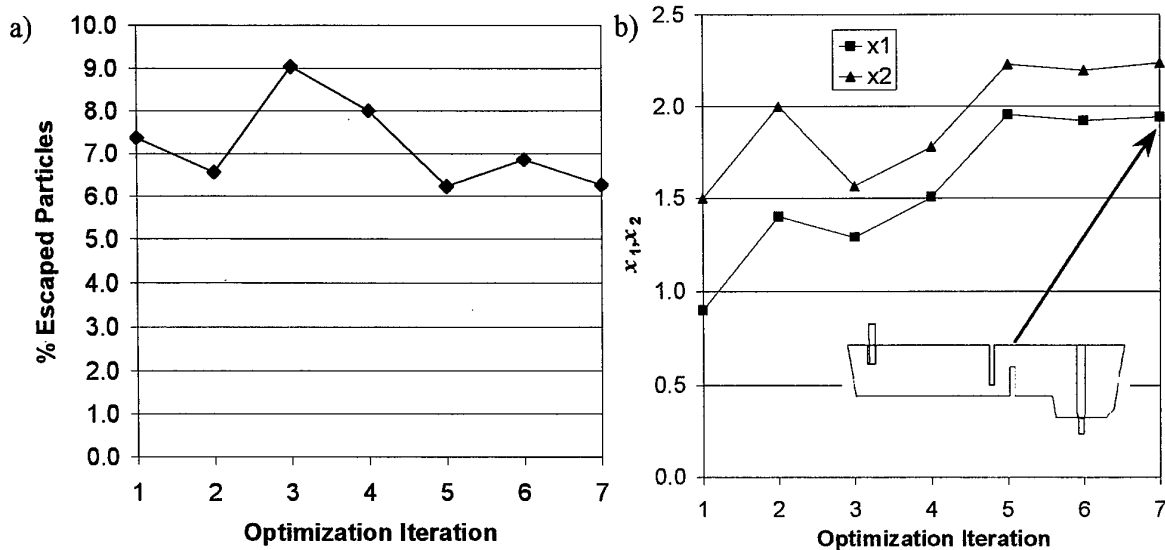


Figure 7: Optimisation history: a) Objective Function b) Design Variables

CONCLUSIONS AND FUTURE WORK

This paper illustrates the power of mathematical optimisation when combined with CFD. The DYNAMIC-Q algorithm of Snyman proved to be very robust. FLUENT was easily incorporated into an automatic loop for the optimisation. In both optimisation cases the optimiser predicted a configuration that is not obvious and one that is not used in industry. In the first case, the optimum design gave an improvement of 43% in the reduction of dead volume from the initial design. As the initial dead volume was already small (1.45%), this reduction represents only a decline of 0.63% in dead volume. In the second case, the percentage of inclusions that escaped through the SEN was only slightly reduced from 7.37% to 6.26%. This may indicate that the 1-dam-1-weir configuration is not suitable for the removal of inclusions.

Future work includes the investigation of additional constraints (e.g. on the maximum velocity at the top surface to reduce the amount of slag that is entrained into the molten steel). The effect of modelling liquid steel instead of water will be investigated, as well as the influence of non-symmetry in the inlet due to the flow control slider valve upstream of the shroud. The optimum configurations obtained by the optimiser will also be verified experimentally. The influence of temperature variation and the corresponding buoyancy forces can be also solved for in the model. Finally, the two different optima obtained for the two different optimisation cases investigated shows that there exists the need of a multi-criteria optimisation to optimise both the dead volume and the inclusion removal simultaneously.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of Columbus Stainless, Iscor (Pty) Ltd and the Department of Trade and Industry under THRIP grant GUN2043307.

REFERENCES

- [1] Sarah H, Pehlke RD, 'Mathematical Modeling of Tundish Operation and Flow Control to Reduce Transition Slabs', *Metallurgical and Materials Transactions B*, **27B**, pp. 745-756, October 1996.
- [2] Crowley RW, Lawson GD, 'Cleanliness Improvements using a Turbulence-Suppressing Tundish Impact Pad', *Steelmaking Conference Proceedings*, pp. 629-635, 1995.
- [3] Rehlaender EM, *Water Model Studies of Fluid Flow Phenomena and Inclusion Separation in Tundishes*, Mater Thesis, University of Toronto, 1983.
- [4] FLUENT, Version 5.3, Fluent Inc., Lebanon, NH, USA, 2000.
- [5] GAMBIT, Version 1.2, Fluent Inc., Lebanon, NH, USA, 2000.
- [6] Snyman JA, Stander N, Roux, WJ 'A dynamic penalty function method for the solution of structural optimization problems', *Appl. Math. Modelling*, **18**, pp. 453-460, 1994.
- [7] Snyman JA, Stander N, 'A new successive approximation method for optimum structural design', *AIAA J.* **32**, pp. 1310-1315, 1994.
- [8] Snyman JA, Stander N, 'Feasible descent cone methods for inequality constrained optimization methods in engineering', *Int. J. Numer. Methods Eng.* **38**, pp. 119-135, 1995.
- [9] Thomas BG, 'Introduction to Continuous Casting', *Continuous Casting Consortium*, URL: <http://bgtibm1.me.uiuc.edu>, 1999.
- [10] Levenspel O, *Chemical Reaction Engineering*, 2nd Edition, John Wiley and Sons, New York, 1972.
- [11] Venter PJ, Craig KJ, Hasse GW, De Kock DJ, Ackerman J, 'Comparison of water model and CFD results for a single-strand continuous caster', *Proceedings of SACAM2000*, Durban, South Africa, 11-13 January 2000.
- [12] Snyman JA, 'A new dynamic method for unconstrained minimization', *Appl. Math. Modelling* **6**, pp. 449-462, 1982.
- [13] Snyman JA, 'An improved version of the original leap-frog dynamic method for unconstrained minimization', *Appl. Math. Modelling* **7**, pp. 216-218, 1983.
- [14] Snyman JA, Frangos C, Yavin Y, 'Penalty function solutions to optimal control problems with general constraints via a dynamic optimization method', *Comput. Math. Applic.* **23**, pp. 46-47, 1992.
- [15] Snyman JA, 'The LFOPCV leap-frog algorithm for constrained optimization', In press, *Computers Math Applic.*, 1999.
- [16] Craig KJ, De Kock DJ, Snyman JA, 'Using CFD and Mathematical Optimization to Investigate Air Pollution due to Stacks', *International Journal for Numerical Methods in Engineering*, **44**, pp 551-565, 1999.
- [17] Craig KJ, De Kock DJ, Gauché P, 'Minimization of Heat Sink Mass Using CFD and Mathematical Optimization', *ASME Journal of Electronic Packaging*, **21**, no. 3, pp. 143-147, September 1999.
- [18] De Kock DJ, Craig KJ, Snyman JA, 'Using Mathematical Optimization in the CFD Analysis of a Continuous Quenching Process', *International Journal for Numerical Methods in Engineering*, **47**, pp 985-999, 2000.

A GENERAL MATHEMATICAL PROGRAMMING METHOD FOR MANIPULATOR WORKSPACE DETERMINATION

L.J. du Plessis and A.M. Hay

Multidisciplinary Design Optimization Group (MDOG)
Department of Mechanical and Aeronautical Engineering
University of Pretoria
Pretoria 0002
South Africa
Tel: +27-12-4203125 Fax: +27-12-3625087
email: lplessis@eng.up.ac.za

ABSTRACT

Different variations of an optimization approach to the determination of maximal and dextrous workspaces of manipulators are presented. The methods vary according to the specific type of workspace being determined and depending on whether the particular manipulator being considered is planar or spatial. A novel and extremely robust constrained optimization algorithm is used throughout which has the considerable advantage that the mapping of workspace boundaries may easily be automated.

1. Introduction

This paper presents an *overview* of the work done to date on the implementation of a novel mathematical programming or optimization approach to the determination of workspaces of manipulators. The original optimization approach was proposed by Snyman et al. (1998) as a possible alternative to the well-established *geometrical methods* (Gosselin and Angeles, 1988, Merlet et al., 1998) and *continuation methods* (Jo and Haug, 1988, 1989). The proposed optimization approach should not be confused with the more cumbersome and computationally intensive *discretization methods* (Fichter, 1986, Arai et al. 1990). The outstanding feature of the optimization approach to the determination of workspaces is that it provides an efficient technique that may easily be automated. In this respect the optimization method is probably more efficient than the previously proposed continuation methods. Furthermore the optimization approach allows for the easy and systematic inclusion of various physical constraints acting on manipulators.

Three specific methodologies, namely the ray method, modified ray method, and chord method, have been proposed and applied to find the *maximal* workspaces of different planar parallel manipulators and one specific serial manipulator. The *dextrous* workspaces of a planar and spatial Gough-Stewart platform have been determined by mapping different fixed orientation workspaces and isolating the corresponding intersections. Formal definitions for the different types of workspaces are given in the relevant sections of this paper. In essence, a *maximal* workspace may contain points on the boundary at which no rotation is possible, whereas a *dextrous* workspace contains a set of points at which the specified range of rotation is possible at all interior and boundary points.

In the first part of this paper, the basic general principles behind the optimization approach will be presented. In the second part, the different methods developed will be outlined and some of the results obtained will be presented.

2. Mapping the workspace boundary

2.1. Coordinates

As described by Haug et al. (1994a), *generalized coordinates* $\mathbf{q} = [\mathbf{q}_1, \dots, \mathbf{q}_{nq}]^T \in R^{nq}$ are defined that characterize the position and orientation of each body in the mechanism. In the vicinity of an *assembled configuration* of the mechanism, these generalized coordinates satisfy m independent holonomic kinematic constraint equations of the form

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (1)$$

where $\Phi : R^{nq} \rightarrow R^m$ is a smooth function.

The generalized coordinates are divided into *input coordinates* $\mathbf{v} = [v_1, \dots, v_{nv}]^T$ controlling the motion of the mechanism, *output coordinates* $\mathbf{u} = [u_1, \dots, u_{nu}]^T$ defining the useful functionality of the mechanism and *intermediate coordinates* $\mathbf{w} = [w_1, \dots, w_{nw}]^T$, where $nw = nq - nv - nu$.

2.2. Constraints and the accessible output set

Inequality constraints are often imposed on the input variables and may also apply to the intermediate variables. These respectively take the forms

$$\mathbf{v}^{\min} \leq \mathbf{v} \leq \mathbf{v}^{\max} \quad (2)$$

and

$$\mathbf{w}^{\min} \leq \mathbf{w} \leq \mathbf{w}^{\max} \quad (3)$$

There may be additional inequality constraints acting on the system, representing relationships between the input, output and intermediate coordinates, that must be satisfied and which take the general form

$$\mathbf{g}^{\min} \leq \mathbf{g}(\mathbf{u}, \mathbf{v}, \mathbf{w}) \leq \mathbf{g}^{\max} \quad (4)$$

The *accessible output set* of the manipulator is the collection of all possible output coordinates of the manipulator. To present this more precisely, the generalized coordinates are partitioned as follows:

$$\mathbf{q} = [\mathbf{u}^T, \mathbf{v}^T, \mathbf{w}^T]^T \quad (5)$$

The constraint equations (1) may be rewritten in terms of this partitioning of generalized coordinates:

$$\Phi(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathbf{0} \quad (6)$$

The accessible output set A is defined as:

$$A \equiv \{ \mathbf{u} \in R^{nu} : \Phi(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathbf{0}; \mathbf{v} \text{ satisfying (2); } \mathbf{w} \text{ satisfying (3); } \mathbf{g}(\mathbf{u}, \mathbf{v}, \mathbf{w}) \text{ satisfying (4)} \} \quad (7)$$

The boundary ∂A of the accessible output set may then be defined as:

$$\partial A \equiv \{ \mathbf{u} \in R^{nu} : \mathbf{u} \in A \text{ and } \exists \text{ an } \mathbf{s} \in R^{nu} \text{ such that for } \mathbf{u}' = \mathbf{u} + \lambda \mathbf{s}, \lambda \in R \text{ arbitrarily small and either positive or negative, no } \mathbf{v} \text{ and } \mathbf{w} \text{ exist that satisfy } \Phi(\mathbf{u}', \mathbf{v}, \mathbf{w}) = \mathbf{0} \text{ as well as inequalities (2), (3) and (4)} \} \quad (8)$$

2.3. Finding a point on ∂A

With respect to the system of equations (1) and (6), a distinction can be made between two possibilities:

Case (i): where $m = nv$ and, given $\mathbf{u}$ and $\mathbf{w}$, system (6) may easily be solved to give $\mathbf{v}$ in terms of $\mathbf{u}$ and $\mathbf{w}$:

$$\mathbf{v} = \mathbf{v}(\mathbf{u}, \mathbf{w}) \quad (9)$$

This is typically the situation with parallel manipulators, where the inverse kinematics can easily be solved.

Case (ii): where $m = nu$ and, given $\mathbf{v}$ and $\mathbf{w}$, system (6) may easily be solved to give $\mathbf{u}$ in terms of $\mathbf{v}$ and $\mathbf{w}$:

$$\mathbf{u} = \mathbf{u}(\mathbf{v}, \mathbf{w}) \quad (10)$$

This is typically the situation with serial manipulators, where the forward kinematics is relatively easy to solve.

Consider *Case(i)*. Assume that a radiating point $\mathbf{u}^0$ has been chosen and that it is interior to the accessible set, A . Consistent with the definition of ∂A in (8), a point $\mathbf{u}^b$ on the boundary in the direction $\mathbf{s} \in R^{nu}$ from $\mathbf{u}^0$ is determined by solving the following constrained optimization problem:

Problem (i):

$$\begin{aligned} & \underset{\mathbf{u}, \mathbf{w}}{\text{maximize}} \|\mathbf{u} - \mathbf{u}^0\| \\ & \text{such that } \mathbf{v}^{\min} \leq \mathbf{v}(\mathbf{u}, \mathbf{w}) \leq \mathbf{v}^{\max} \\ & \quad \mathbf{w}^{\min} \leq \mathbf{w} \leq \mathbf{w}^{\max} \\ & \text{and } \mathbf{g}^{\min} \leq \mathbf{g}(\mathbf{u}, \mathbf{v}(\mathbf{u}, \mathbf{w}), \mathbf{w}) \leq \mathbf{g}^{\max} \\ & \text{and subject to equality constraints} \\ & \quad \mathbf{h}(\mathbf{u}, \mathbf{s}) = \mathbf{0}, \mathbf{h} \in R^{nu-1} \end{aligned}$$

where $\|\cdot\|$ denotes the Euclidean norm. The equality constraints define a point on the parameterized straight line $\mathbf{u}(\lambda) = \mathbf{u}^0 + \lambda \mathbf{s}, \lambda \in R$. (For example, if $nu = 2$, $\mathbf{u} = (x, y)^T$, $\mathbf{u}^0 = (x^0, y^0)^T$ and $\mathbf{s} = (s_x, s_y)^T$, then $\mathbf{u} = \mathbf{u}^0 + \lambda \mathbf{s}$ has the components $x = x^0 + \lambda s_x$ and $y = y^0 + \lambda s_y$; it follows that $\mathbf{h}(\mathbf{u}, \mathbf{s}) = (x - x^0)/s_x + (y - y^0)/s_y = 0$). The solution of *Problem (i)* is illustrated in Figure 1.

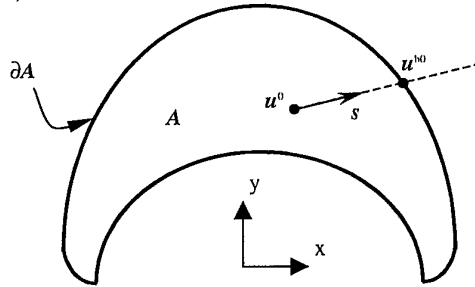


Figure 1: Finding an initial point on ∂A

For *Case (ii)* the associated constrained optimization problem is:

Problem (ii):

$$\begin{aligned} & \underset{\mathbf{v}, \mathbf{w}}{\text{maximize}} \|\mathbf{u}(\mathbf{v}, \mathbf{w}) - \mathbf{u}^0\| \\ & \text{such that } \mathbf{v}^{\min} \leq \mathbf{v} \leq \mathbf{v}^{\max} \\ & \quad \mathbf{w}^{\min} \leq \mathbf{w} \leq \mathbf{w}^{\max} \\ & \text{and } \mathbf{g}^{\min} \leq \mathbf{g}(\mathbf{u}(\mathbf{v}, \mathbf{w}), \mathbf{v}, \mathbf{w}) \leq \mathbf{g}^{\max} \\ & \text{and subject to equality constraints} \\ & \quad \mathbf{h}(\mathbf{u}(\mathbf{v}, \mathbf{w}), \mathbf{s}) = \mathbf{0}, \mathbf{h} \in R^{nu-1} \end{aligned}$$

where the equality constraints again define a point $\mathbf{u}$ on the straight line through $\mathbf{u}^0$ in the direction $\mathbf{s}$.

Note that should the radiating point u^0 be chosen exterior to A , then the above problems will become *minimization* problems.

A question which arises in connection with the above problems is how the radiating point may be obtained. For *Case (i)*, assume a planar manipulator with a two-dimensional accessible output set A . Depending on the particular geometry of the manipulator a suitable choice for a radiating point u^0 , inside the accessible output set, may be self-evident. If not, then u^0 may be obtained from Eq. (9) by solving for u in

$$\bar{v} = v(u, \bar{w}) \quad (11)$$

where

$$\begin{aligned} \bar{v} &= (v^{\min} + v^{\max})/2 \\ \bar{w} &= (w^{\min} + w^{\max})/2 \end{aligned}$$

In practice this can be done by solving the least squares optimization problem

$$\underset{u}{\text{minimize}} \|v(u, \bar{w}) - \bar{v}\|^2 \quad (12)$$

For *Case (ii)*, if an obvious choice for u^0 is not available, then an indication may be obtained from (10).

$$u^0 = u(\bar{v}, \bar{w}) \quad (13)$$

3. Optimization methodologies and examples of maximal workspaces

A *maximal* workspace is also referred to as an *accessible* or *reachable* workspace and is defined as the set of all points which a chosen reference point on the manipulator end-effector can reach with at least one orientation. The three optimization methodologies for determining maximal workspaces described here are: the *ray* method, the *modified ray* method and the *chord* method.

3.1. The ray method

The original optimization methodology proposed by Snyman et al. (1998) is called the *ray method*. Simply stated the ray method consists of finding a suitable initial radiating point, and then finding the points of intersection of a pencil of rays, emanating from this point with the boundary of the accessible output set. For each emanating ray, optimization problem (i) or (ii) (Section 2.3) is solved using the robust LFOPC leap-frog algorithm for constrained optimization (Snyman (1999)).

Using this methodology, Snyman et al. (1998) have determined *maximal workspaces* of a planar serial and planar parallel manipulator. The manipulators studied were taken from the foundation paper of Haug et al. (1994a), and the results obtained by the respective numerical methods are in exact agreement (see Figure 2 and Figure 3 and note that P indicates the position of the working point). For both these manipulators, the internal curves were also mapped using an optimization approach. These interior curves are of importance, since, according to Haug et al. (1994b), limits on controllability of the manipulators are associated with configurations lying on the interior curves.

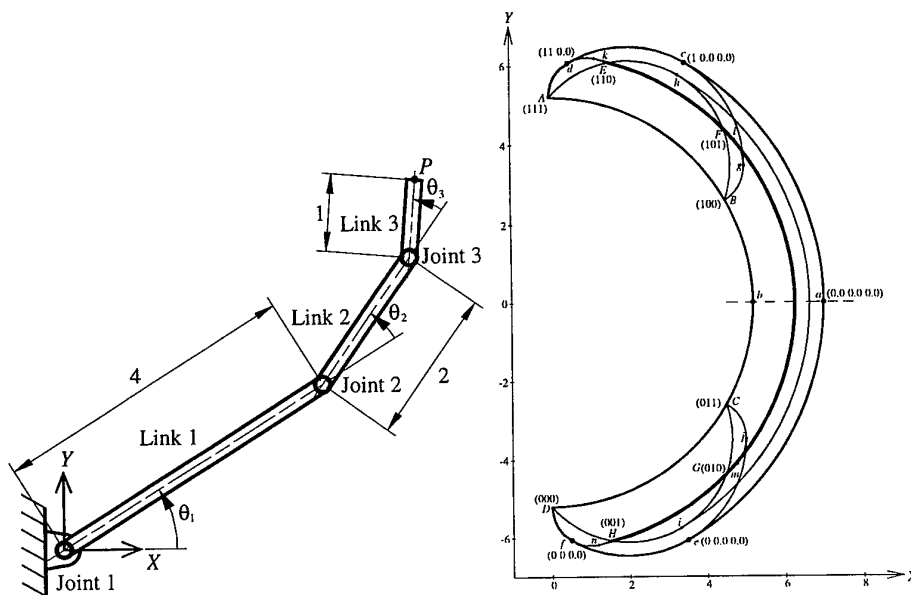


Figure 2: Planar serial manipulator and its associated maximal workspace

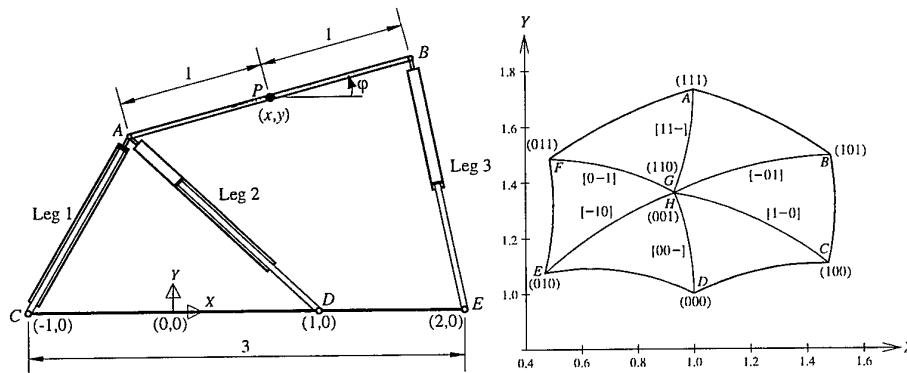


Figure 3: Planar Gough-Stewart platform and its associated maximal workspace

A new and concise notation for labeling the corners and curves of the boundaries of maximal workspaces of manipulators resulted from the optimization study of Snyman et al. (1998). Consider the maximal workspace of the planar Gough-Stewart platform shown in Figure 3. The boundary of the workspace clearly consists of six smooth curves intersecting at distinct corners A, B, C, D, E and F. These corners correspond to configurations where each actuator leg is either at its minimum or maximum allowable value. The precise state at each corner may be indicated by a triplet $(X_1 \ X_2 \ X_3)$, where $X_i := 0$ or $X_i := 1$ respectively denote that leg i is at its minimum or maximum value. The interior bifurcation points G and H are also labeled in the same way.

The boundary and interior curves may be labeled in a similar manner by a triplet enclosed in square brackets, i.e. $[X_1 \ X_2 \ X_3]$. Here $X_i := -$ indicates that the leg i varies from one extreme to the other as the working point advances along the specific boundary or interior curve.

The major drawback of the ray method is that it breaks down for some non-convex workspaces.

3.2. The modified ray method

To counteract this problem, a *modified ray method* has been proposed by Hay and Snyman (1999). In this modification, if due to non-convexity any sections of workspace boundary cannot be determined using the ray method, then the missing sections are mapped using the ray method with suitably chosen new radiating points. This approach has proven capable of mapping most non-convex workspaces.

The modified ray method has been applied to the same planar parallel manipulator considered in the previous section. (Figure 3) In Figure 4 the effects of varying the upper length limit of legs 1 and 2 are investigated. The central plot shows the workspace of the standard case. The plots to the right of this central case show the workspaces with the appropriate limit of Leg 1 increased by 25% and to the left decreased by 25%. The plots above the central plot show the workspaces with the limit of Leg 2 increased by 25% and below decreased by 25%.

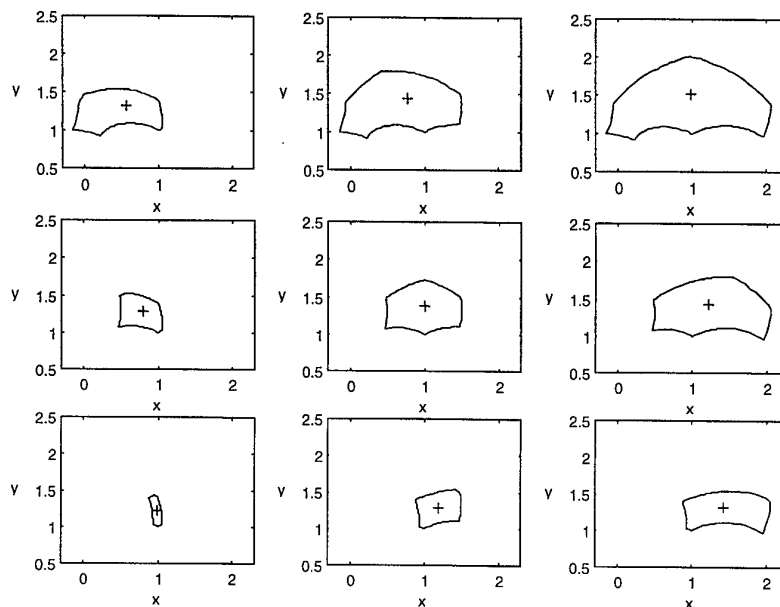


Figure 4: $l_i^{\max}$ variation, $i=1,2$

Since the modified ray method requires significant user interaction, the stated advantage of the optimization approach in allowing automated mapping of the workspace is counteracted.

3.3. The chord method

In order to improve the efficiency of the mapping of the workspace boundary, a new *chord method*, which allows for the automated mapping of most non-convex manipulator workspaces, has been proposed by Snyman and Hay (2000). This method is based on the optimization principles similar to those outlined above, but uses a search *arc* centered at successive points along the workspace boundary, instead of search *rays* emanating from a fixed point inside the workspace.

The chord method has successfully been applied to the determination of workspaces of general planar parallel manipulators. In the paper of Snyman and Hay maximal workspaces of 3-RPR manipulators of the type studied by Merlet et al. (1998) are determined. These manipulator workspaces were accurately determined with efficiency comparable to that of the geometric method.

In Figure 5(a) the basic general architecture of the manipulator studied and the workspaces obtained for different manipulator dimensions are given in Figure 5(b) and (c). C indicates the position of the working point.

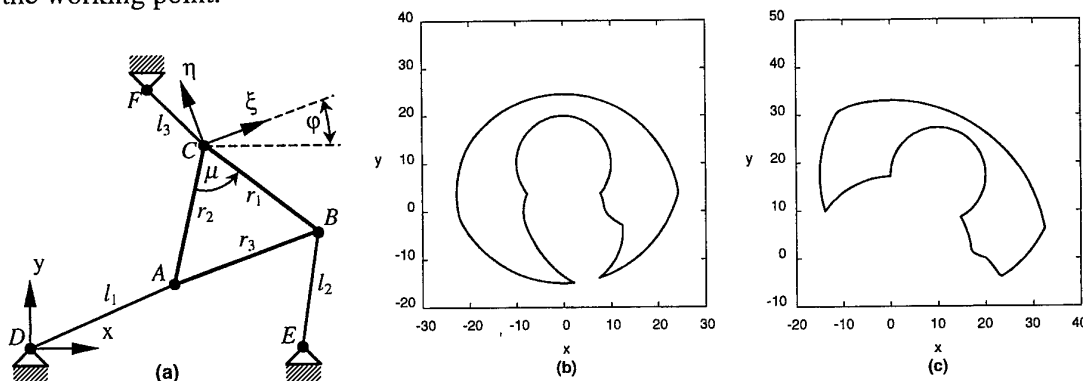


Figure 5: (a) 3-RPR planar manipulator; (b) and (c) Maximal workspaces for different manipulator dimensions

4. Optimization methodologies and for the determination of dextrous workspaces

A dextrous workspace is defined as the set of points that can be reached by a given point and at which specified ranges of rotation of the working body can be achieved (Haug et al. (1992)). According to Haug et al. (1992), this distinguishing definition is necessary since the literature often discusses the “workspace” of a manipulator, when in fact the manipulator cannot “work” while following a continuous path within this so-called workspace.

4.1. Using fixed orientation workspaces to find a dextrous workspace

For the planar Gough-Stewart platform shown in Figure 3, the following dexterity requirement may be specified: ϕ to assume *all* values in the range $[\phi_{\min} - \phi_{\max}]$ at *any* point in the dextrous workspace.

Du Plessis and Snyman (1999) determined the associated dextrous workspace by mapping different fixed orientation workspaces. The constrained optimization problem (i) in Section 2.3 may be modified, so that when it is solved, the optimum solution will correspond to a position where a specified fixed orientation requirement is achieved.

This *modified* optimization problem is:

$$\begin{aligned}
 & \underset{u, w}{\text{maximize}} \quad \|u - u^0\| \\
 & \text{such that } v^{\min} \leq v(u, w) \leq v^{\max} \\
 & \text{and subject to the equality constraints} \\
 & \quad h(u, s) = 0, \quad h \in R^{nu-1} \\
 & \quad w = w^{\text{fix}}
 \end{aligned} \tag{14}$$

It should be clear that the dextrous workspace of interest, and denoted by $A[\phi_{\min} - \phi_{\max}]$, is given by the intersection of all possible fixed orientation workspaces $A[\phi_{\text{fix}}]$, $\phi_{\text{fix}} \in [\phi_{\min}, \phi_{\max}]$. Since it is expected that $A[\phi_{\text{fix}}]$ varies in a “continuous” manner as ϕ_{fix} is varied, a good numerical approach to determining $A[\phi_{\min} - \phi_{\max}]$ would be to determine $A[\phi_{\text{fix}}]$ for a finite number of regularly spaced ϕ_{fix} values in the given range, and then computing the boundary of their intersection. It is now proposed that extreme economy may be achieved by simply determining $A[\phi_{\text{fix}}]$ for the two

extreme values $\varphi_{\text{fix}} = \varphi_{\text{min}}$ and $\varphi_{\text{fix}} = \varphi_{\text{max}}$ and then obtaining $A[\varphi_{\text{min}} - \varphi_{\text{max}}]$ by simply considering the intersection of the extreme fixed angle sets, i.e.

$$A[\varphi_{\text{min}} - \varphi_{\text{max}}] = A[\varphi_{\text{min}}] \cap A[\varphi_{\text{max}}] \quad (15)$$

The validity of assumption (15) must in practice be reinforced by checking whether at the intermediate central value, φ_i , the following additional necessary condition is also satisfied:

$$\{A[\varphi_{\text{min}}] \cap A[\varphi_{\text{max}}]\} \subset A[\varphi_i] \quad (16)$$

where $\varphi_i = (\varphi_{\text{min}} + \varphi_{\text{max}})/2$.

The respective fixed orientation workspaces involved in mapping the dextrous workspace $A[(-10^\circ) - (10^\circ)]$ are shown in Figure 6(a). These fixed orientation workspaces were traced using the *ray* method (Section 3.1), solving *modified* optimization problem (14) at each angular interval. The intermediate fixed orientation workspace $A[\varphi_i] = A[0^\circ]$ is also shown, indicating that the additional necessary condition (16) is clearly satisfied, seeing that the hashed area, corresponding to the intersection of $A[-10^\circ]$ and $A[10^\circ]$, is taken as the dextrous workspace $A[(-10^\circ) - (10^\circ)]$.

In accordance with the work of Haug et al.(1992), the boundaries of the dextrous output sets for which φ achieves the respective full ranges $[(-5^\circ) - (5^\circ)]$, $[(-10^\circ) - (10^\circ)]$ and $[(-15^\circ) - (15^\circ)]$ of rotatability, are also respectively shown in Figure 6(b).

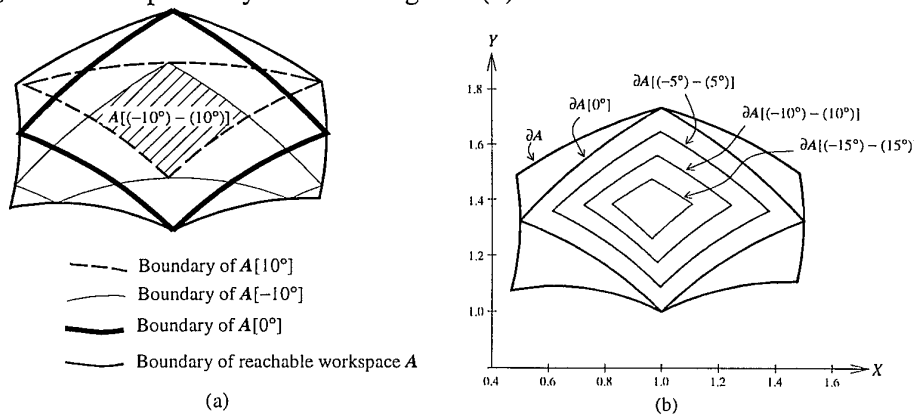


Figure 6: (a) Dextrous workspace $A[(-10^\circ) - (10^\circ)]$; (b) Dextrous workspaces for different full-range rotatability requirements.

Du Plessis and Snyman (1999) also extended this *fixed orientation workspace* method to determine a *simplified* dextrous workspace of the spatial 6-3 Gough-Stewart platform of Liu et al. (1993) (see Figure 7(a)). This was done by adapting the two-dimensional approach used for mapping planar workspaces.

For any vertical slice at an angle θ with the X-axis, the associated two-dimensional fixed orientation workspace of the working point P is mapped using the *ray* method (Section 2.3) in solving an optimization problem similar to (14). For each vertical plane i , $i = 1, 2, 3, \dots, N_p$, through OZ, there corresponds a unique angle θ_i that the plane makes with the OX-axis, resulting in an additional equality constraint $y/x = \tan \theta_i$ to be included in (14). A composite of a representative sample of such vertical workspaces may then yield a representation of the three-dimensional fixed orientation workspace.

The dexterity requirement for a spatial Gough-Stewart platform will in general be specified by a triplet indicating the three ranges of the three respective orientation angles involved, i.e. $[\alpha_{\min} - \alpha_{\max}, \beta_{\min} - \beta_{\max}, \gamma_{\min} - \gamma_{\max}]$. Du Plessis and Snyman (1999), however, used the *restricted* dexterity requirement $[\alpha_{\text{fix}}, \beta_{\text{fix}}, \gamma_{\min} - \gamma_{\max}]$, indicating that only γ must be able to assume *all* values in the range $[\gamma_{\min} - \gamma_{\max}]$, while α and β must respectively assume the fixed values α_{fix} and β_{fix} , at *any* point in the associated dextrous workspace denoted by $A[\alpha_{\text{fix}}, \beta_{\text{fix}}, \gamma_{\min} - \gamma_{\max}]$. Although simple, this requirement can be of practical importance.

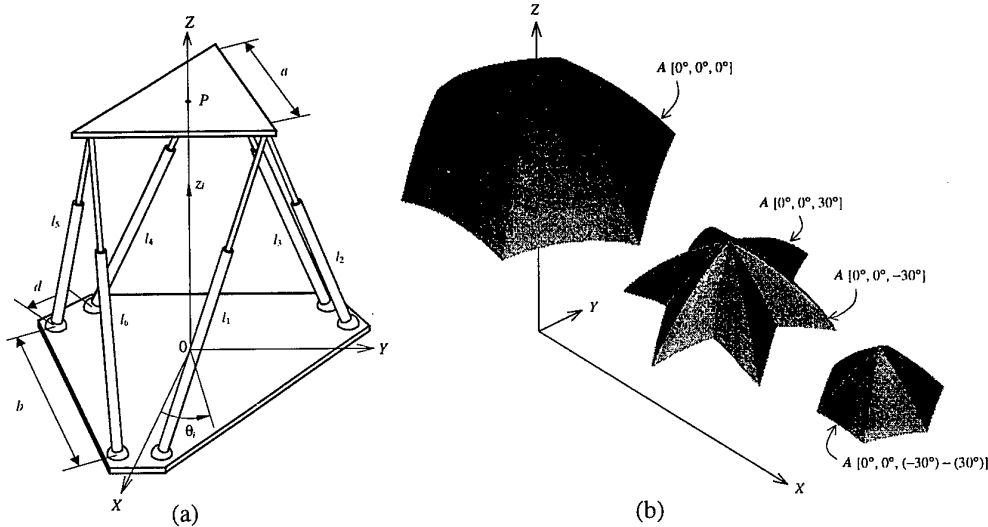


Figure 7: (a) Geometry of the 6-3 Gough-Stewart platform; (b) $A[0^\circ, 0^\circ, 0^\circ]$, the overlap of $A[0^\circ, 0^\circ, -30^\circ]$ and $A[0^\circ, 0^\circ, 30^\circ]$ and the final dextrous workspace $A[0^\circ, 0^\circ, (-30^\circ) - (30^\circ)]$.

The specific dextrous workspace of the 6-3 Gough-Stewart platform considered has the dexterity requirement

$$[0^\circ, 0^\circ, (-30^\circ) - (30^\circ)] \quad (17)$$

Here, in accordance with the treatment for the planar case (equation (15)), two spatially fixed orientation accessible workspaces, $A[0^\circ, 0^\circ, -30^\circ]$ and $A[0^\circ, 0^\circ, 30^\circ]$ are determined. The intersection or overlapping volume of these two workspaces is assumed to be the dextrous workspace $A[0^\circ, 0^\circ, (-30^\circ) - (30^\circ)]$, within which the full dexterity requirement (17) is met at each point provided again, in accordance with condition (16) for the planar case, that

$$\{A[0^\circ, 0^\circ, -30^\circ] \cap A[0^\circ, 0^\circ, 30^\circ]\} \subset A[0^\circ, 0^\circ, 0^\circ] \quad (18)$$

The validity of the generalization of assumption (15) to the spatial case, is borne out in Figure 7(b), where the different fixed orientation accessible workspaces are placed at an X-offset next to each other. The assumed dextrous workspace $A[0^\circ, 0^\circ, (-30^\circ) - (30^\circ)]$ is clearly contained in $A[0^\circ, 0^\circ, 0^\circ]$ and therefore satisfies condition (18), which is the generalizing of the necessary condition (16) for the planar case.

5. Conclusion

This overview shows that the mathematical optimization approach is a generally applicable method for manipulator workspace determination. Due to the fact that the formulation of the mathematical programming problem to be solved is highly flexible, the method easily determines different

maximal and dextrous workspaces. Since an extremely *robust* optimization algorithm is used in solving the differently formulated optimization problems, the methodology for workspace determination may easily be automated. The latter feature is probably the most significant factor to the successful implementation of the optimization approach reviewed here.

Efficient methods for dextrous workspace determination of parallel manipulators remain critical to the successful design and practical application of these manipulators. In this respect the optimization approach can make a valuable contribution. It is foreseen that the different variations of the optimization method will, in future, be combined and extended to enable the mapping of general spatial dextrous workspaces.

REFERENCES

- Arai, T. et al., 1990-07, "Design, analysis and construction of a prototype parallel link manipulator", *IEEE International Workshop on Intelligent Robots and Systems*, Vol.1, pp. 205-212, Ib Araki, Japan.
- Du Plessis, L. J. and Snyman, J.A., 1999, "A numerical method for the determination of dextrous workspaces of Gough-Stewart platforms", *Submitted for publication*.
- Fichter, E. F., 1986, "A Stewart platform-based manipulator: general theory and practical construction", *International Journal of Robotics Research*, Vol. 5, pp.157-182.
- Gosselin, C. M. and Angeles, J., 1988, "The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator", *Journal of Mechanisms, Transmissions, and Automation in Design*. Vol. 110, pp. 35-41.
- Haug, E.J., Luh, C.M., Adkins, F.A. and Wang, J.Y., 1994a, "Numerical algorithms for mapping boundaries of manipulator workspaces", *Concurrent Engineering Tools for Dynamic Analysis and Optimization, IUTAM Fifth Summer School on Mechanics*, Denmark.
- Haug, E.J., Adkins, F.A., Qiu, C. and Yen, J., 1994b, "Analysis of barriers to control of manipulators within accessible output sets", Technical Report R-174, University of Iowa, Center for Computer-Aided Design.
- Haug, E.J., Wang, J.-Y. and Wu, J.K., 1992, "Dextrous workspaces of manipulators. I. Analytical criteria", *Mechanics of Structures and Machines*. Vol. 20, pp. 321-361.
- Hay, A.M. and Snyman, J.A., 1999, "The determination of non-convex workspaces of generally constrained planar Stewart platforms", To appear in *Computers and Mathematics with Applications* (2000).
- Jo, D.-Y. and Haug, E.J., 1988, "Workspace analysis of closed-loop mechanisms with unilateral constraints", *Advances in Design Automation ASME DE* Vol. 19, No. 3, pp. 53-60.
- Jo, D.-Y. and Haug, E.J., 1989, "Workspace analysis of multibody mechanical systems using continuation methods", *The ASME Journal of Mechanisms, Transmissions, and Automation in Design*. Vol. 111, pp. 581-589.
- Liu, K., Fitzgerald, J.M. and Lewis, F.L., 1993, "Kinematic analysis of a Stewart platform manipulator", *IEEE Transactions on Industrial Electronics*, Vol. 40, pp. 282-293.
- Merlet J.-P., Gosselin, C.M. and Mouly, N., 1998, "Workspaces of planar parallel manipulators", *Mechanisms and Machine Theory*. Vol. 33, No. 1, pp. 7-20.
- Snyman, J.A., Du Plessis, L.J. and Duffy, J., 1998, "An optimization approach to the determination of the boundaries of manipulator workspaces", To appear in *The ASME Journal of Mechanical Design* (2000).
- Snyman, J.A., 1999, "The LFOPC leap-frog algorithm for constrained optimization", To appear in *Computers and Mathematics with Applications* (2000).
- Snyman, J.A. and Hay, A.M., 2000, "The chord method for the determination of non-convex workspaces of planar parallel platforms", *Proceedings of the 7th International Symposium on Advances in Robot Kinematics (ARK)*, 26-30 June 2000, Piran-Portoroz, Slovenia, pp. 285-294.

MINIMUM COST DESIGN OF WELDED STRUCTURES

József Farkas

professor emeritus, doctor of technical science

University of Miskolc, Hungary

ABSTRACT

In the optimum design of welded structures the cost function plays an important role, since the welding is an expensive technology. The developed cost function contains costs not only for materials, but also for fabrication. Since the cost is a function of main structural dimensions, it is possible to achieve cost savings in design stage by function minimization. To assure the quality of weldments the residual welding deformations due to shrinkage of welds should be limited. A relatively simple calculation method is worked out to predict these deformations. In an illustrative numerical example the cost function minimization and the constraint on residual welding deformation is applied to a welded structure, which consists of a horizontal plate stiffened by rectangular hollow section ribs. This simply supported beam is subjected to bending by a uniformly distributed normal static load. Depending on the measure of the deflection constraint the limitation of welding deformation can be active, so it influences the optimum structural dimensions and the cost.

1. INTRODUCTION

The aim of the present paper is to show our special results in the field of structural optimization. Our research field is the design of welded structures. Welding is an expensive technology, so it is worth to study the ways for cost savings. We have worked out a relatively simple cost function, containing costs not only for material, but also for fabrication. An advanced welded structure can be constructed with a cooperation of designers and manufacturers. Our aim is to incorporate into the optimum design process all the important engineering aspects relating to design and fabrication.

Welded structures have many thin-walled elements, since the best way to decrease the mass and cost is the decreasing the plate thicknesses. There are some phenomena dangerous for thin-walled structures: global and local buckling, vibration and noise, residual welding stresses and distortions, fatigue due to high stress concentrations, additional normal stresses due to warping torsion. These phenomena should be avoided by defining adequate design constraints. To guarantee the exact fabrication for quality assurance we define constraints limiting the residual distortions due to shrinkage of welds. For this purpose we have worked out a relatively simple calculation method of residual welding deformations.

Our cost function and design constraints are highly nonlinear, therefore we need efficient mathematical methods, the computer software of which can be used for constrained function minimization problems. We have successfully applied the Rosenbrock's hillclimb method, the backtrack method, the FSQP sequential quadratic programming method.

Optimum design is a process of searching better solutions. Structural optimization means that we understand the behaviour of a structure in various environments by using structural analytical methods. Better structural versions can be achieved by changing the structural characteristics. Structural characteristics are as follows: loads, materials, type of structure, geometry, topology,

profiles, fabrication technology, connections, environmental protection, transportation, erection, maintenance.

The optimum design process has three main phases as follows:

- (1) preparation phase: selection of candidate structural versions by defining their characteristics, formulation of the cost function and the design constraints;
- (2) constrained function minimization using computerized mathematical methods;
- (3) evaluation phase: comparison of optimized structural versions, formulation of design rules, working out expert systems.

The research results in the field of welded structures can be discussed in the commissions, subcommissions and working groups of the International Institute of Welding (IIW), e.g. Commission XIII (fatigue), XV (design), Subcommission XV-E (tubular structures), XV-F (interaction of design and fabrication), XV-G (seismic-resistant design), WG for residual stresses and distortions, WG for economy of design and fabrication.

In the present paper our cost function and calculation method for prediction of residual welding stresses and distortions are briefly treated and applied to an illustrative numerical example.

2. THE COST FUNCTION

In the cost function the material and fabrication costs are included

$$K = K_m + K_f = k_m \rho V + k_f \sum T_i \quad (1)$$

where ρ is the material density, V is the volume of structure, k_m and k_f are the material and fabrication cost factors, respectively, T_i are the production times. Equation (1) can be written in the form of

$$\frac{K}{k_m} = \rho V + \frac{k_f}{k_m} (T_1 + T_2 + T_3) \quad (2)$$

Time for preparation, assembly and tacking can be expressed as

$$T_1 = C_1 \Theta_d (\kappa \rho V)^{1/2} \quad (3)$$

where $C_1 = 1 \text{ min/kg}^{0.5}$, Θ_d is a difficulty factor expressing the complexity of the structure (planar or spatial, constructed from simple plate elements or profiles), κ is the number of structural elements to be assembled.

Welding time is

$$T_2 = \sum C_{2i} a_{wi}^n L_{wi} \quad (4)$$

where a_w is the weld size, L_w is the weld length. Formulae for $C_{2i} a_w^n$ are developed using the COSTCOMP database for different welding technologies and weld types (COSTCOMP 1990, Bodt 1990).

The additional time for electrode changing, deslagging and chipping can be calculated as

$$T_3 = 0.3 T_2 \quad (5)$$

The final form of the cost function is

$$\frac{K}{k_m} = \rho V + \frac{k_f}{k_m} \left(\Theta_d \sqrt{\kappa \rho V} + 1.3 T_2 \right) \quad (6)$$

The following data of cost factors are used: $k_m = 0.5\text{--}1.2 \text{ \$/kg}$, $k_f = 0\text{--}60 \text{ \$/manhour} = 0\text{--}1 \text{ \$/min}$. To give internationally usable results, values of $k_f/k_m = 0, 1$ and 2 kg/min can be considered, the value of 0 means minimum weight design.

This cost function is detailed in the book Farkas – Jármai (1997) and in Jármai – Farkas (1999b) and has been successfully applied to minimum cost design of welded silos (Farkas – Jármai 1996a),

Vierendeel tubular trusses (Farkas - Jármai 1996b), highway bridge decks (Jármai - Farkas - Horikawa 1998), hydrostatically loaded stiffened plates (Farkas - Jármai 1999), and longitudinally stiffened box beams (Jármai - Farkas 1999a)

3. CALCULATION OF RESIDUAL WELDING DEFORMATIONS

First the beam deformations due to a stationary thermal load are determined. The thermal distribution is nonlinear as shown in Fig. 1. The thermal strain would be different at different points of cross section if they were independent from each other: $\varepsilon = \alpha_o T_e(y)$. Because they are connected to each other, we assume that the cross section remains planar, only a linear strain can occur in the cross section. This linear strain is characterized by the strain of the gravity centre and the curvature of the beam: $\varepsilon = \varepsilon_G + Cy$. The differences between the theoretical thermal strain and the linear strain cause the stresses:

$$\sigma = E\varepsilon = E\{\varepsilon_G + Cy - \alpha_o T_e(y)\} \quad (7)$$

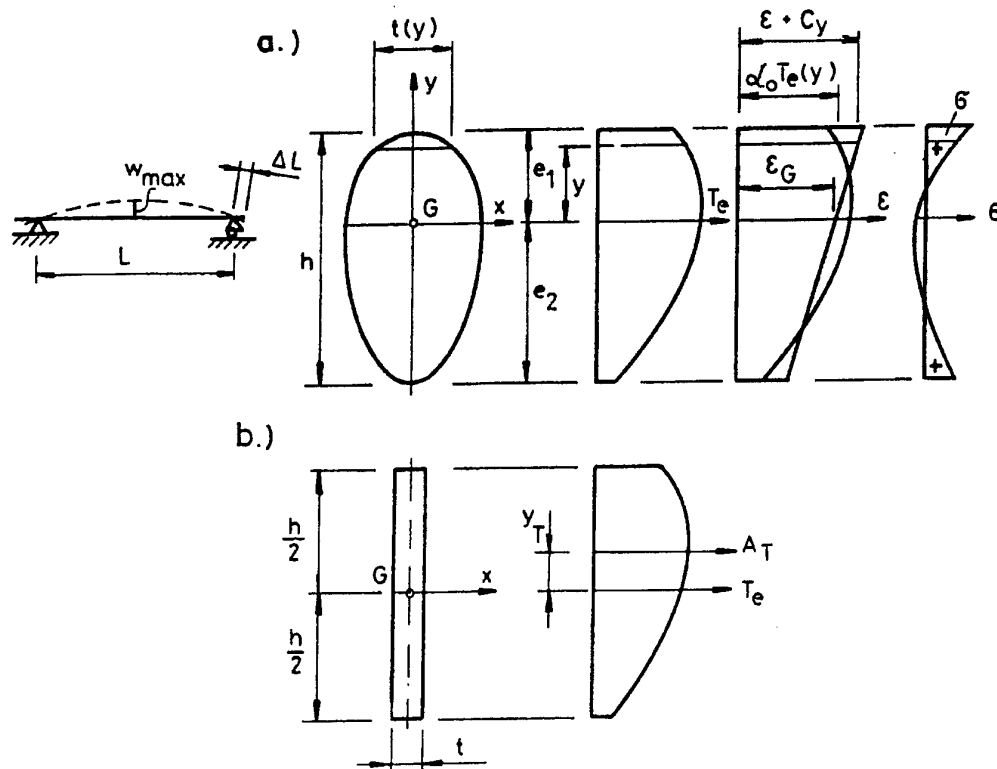


Fig.1. Strains, stresses and deformations in a simply supported beam with nonlinear temperature distribution

There is no external loading on the beam, so the internal stresses caused by thermal difference are in equilibrium,

$$\int_A \sigma dA = 0 \quad \text{and} \quad \int_A \sigma y dA = 0 \quad (8)$$

By inserting Eq. (7) to (8), we get

$$\varepsilon_G = \frac{1}{A} \int_{e_1}^{e_2} \alpha_o T_e(y) t(y) dy \quad \text{and} \quad C = \frac{1}{I_x} \int_{e_1}^{e_2} \alpha_o T_e(y) y t(y) dy \quad (9)$$

If the thickness is constant, i.e. $t(y)=t$ we can define the thermal shrinkage impulse A_T as

$$A_T = \int_{e_1}^{e_2} \alpha_o T_e(y) dy \quad (10)$$

The thermal impulsive moment is defined as follows

$$A_T y_T = \int_{e_1}^{e_2} \alpha_o T_e(y) y dy \quad (11)$$

Using these definitions the strain at the centre of gravity and the curvature are as follows

$$\varepsilon_G = \frac{A_T t}{A} \quad (12)$$

$$C = \frac{A_T t y_T}{I_x} \quad (13)$$

The thermal shrinkage impulse due to a single pass longitudinal weld has been determined for welding by Okerblom (1963) with the following formula

$$A_T = \frac{0.3355 \alpha_o Q_T}{c_o \rho t} \quad (14)$$

where $Q_T = \eta_o \frac{UI_w}{v_w} = q_o A_w$, U arc voltage, I_w arc current, v_w speed of welding, c_o specific heat,

η_o coefficient of efficiency, q_o is the specific heat for the unit cross-sectional area of a weld (1 mm²), A_w is the cross-sectional area of a weld. Formula (14) contains all the important material and welding parameters, thus it is valid not only for steels, but also for other materials, e.g. for aluminium alloys.

For a mild or low alloy steels, where $\alpha_o = 12 \times 10^{-6}$ [1/C°], $c_o \rho = 4.77 \times 10^{-3}$ [J/mm³/C°], the thermal impulse is

$$A_T t \text{ [mm}^2\text{]} = 0.844 \times 10^{-3} Q_T \text{ [J/mm]}$$

Inserting this into Eqs. (12) and (13), we get the basic Okerblom formulae

$$\varepsilon_G = \frac{A_T t}{A} = -0.844 \times 10^{-3} \frac{Q_T}{A} \quad (15)$$

The minus sign means shrinkage.

$$C = \frac{A_T t y_T}{I_x} = -0.844 \times 10^{-3} \frac{Q_T y_T}{I_x} \quad (16)$$

Note that the distorted form can be determined by view. y_T and C have opposite signs (Fig. 1).

The formulae above are valid for symmetrically arranged welds $y_A = 0$ when

$$\frac{Q_T}{A} \leq 2.50 \left[\frac{\text{J}}{\text{mm}^3} \right], \quad (17)$$

for eccentric welds ($y_A \neq 0$) when

$$\frac{Q_T}{A} \leq 0.63 \left[\frac{\text{J}}{\text{mm}^3} \right]. \quad (18)$$

Knowing ε_G and C the residual deformations of the beam can be calculated in the following manner. Assuming that A_T and the cross-section is constant along the beam, the resulting shrinkage of the whole beam of length L is

$$\Delta L = \varepsilon_G L \quad (19)$$

and the residual deflection, using the relation between the curvature C and the bending moment $M = CEI_x$, is

$$w_{max} = CL^2/8 \quad (20)$$

For the optimization purposes a formula is needed expressing the relation between Q_T and the cross-sectional area of the weld (the coefficient q_0). This coefficient characterizes the welding technology and can be determined defining the coefficient of penetration α_N [kg/Ah] (Amper-hour) in following way: the mass of penetrated metal during a weld pass lasting $t[s]$ time can be calculated as

$$m[kg] = \frac{\alpha_N I_w [A] t [s]}{3600}, \quad (21)$$

This mass can be expressed also by

$$m = A_w L \rho \quad (22)$$

L is the weld length. Combining (21) and (22) the welding speed can be calculated as

$$v_w [mm/s] = \frac{L [mm]}{t [s]} = \frac{\alpha_N I_w}{3600 \rho A_w} \quad (23)$$

(23) can be substituted in the original formula for Q_T

$$Q_T = \frac{\eta_0 U I_w}{v_w} = \eta_0 \frac{3600 U [V] \rho}{\alpha_N} A_w = q_0 A_w \quad (24)$$

Thus, for the constant we need the values of η_0, U, α_N for different welding technologies.

Numerical example

for hand welding of butt welds, material: structural steel

$$\alpha_N = 8.8 \times 10^{-3} \text{ kg/Ah}; \eta_0 = 0.7; \rho = 7.85 \times 10^{-6} \text{ kg/mm}^3; U = 27V$$

$$Q_T (J/mm) = 0.7 \frac{3600 \times 27 \times 7.85 \times 10^{-6}}{8.8 \times 10^{-3}} A_w = 60.7 A_w (mm^2)$$

We use for hand welding of fillet welds of size a_w $Q_T = 78.8 a_w^2$ (25)

and for automatic welding $Q_T = 59.5 a_w^2$. (26)

The value of $q_0 = 78.8$ has been verified by measurements carried out in a Japanese shipyard Shinkurushima (near Matsuyama). GMAW-C technology (gas metal arc welding with CO_2) has been used for fillet welds with data of $I_w = 260$ A, $U = 32$ V, $v_w = 550$ mm/min, the leg size of the fillet weld was $b = 4$ mm. The cross-sectional area of the weld is $A_w = b^2/2 = 8 \text{ mm}^2$. Using (24)

$$\alpha_N = \frac{\rho v_w A_w}{I_w} = \frac{7.85 \times 10^{-6} \times 550 \times 8 \times 60}{260} = 7.97 \times 10^{-3} \text{ (kg/Ah)}$$

$$\text{and } q_0 = \eta_0 \frac{3600 U \rho}{\alpha_N} = 0.7 \frac{3600 \times 32 \times 7.85 \times 10^{-6}}{7.97 \times 10^{-3}} = 79.4 \text{ (J/mm}^3\text{)}.$$

It should be noted that the calculation method described above is worked out also for beams containing two or more longitudinal welds. Using this method the correct welding sequence can be predicted and methods for decreasing or eliminating the residual distortions can be analysed (Farkas – Jármai 1997, Farkas – Jármai 1998a).

4. ILLUSTRATIVE NUMERICAL EXAMPLE

The investigated structure consists of a deck plate stiffened by longitudinal ribs of rectangular hollow section (RHS) connected to the deck plate by longitudinal fillet welds (Fig.2). The simply supported stiffened plate is subjected to uniform normal load (Farkas – Jármai 1998a, Farkas 1999).

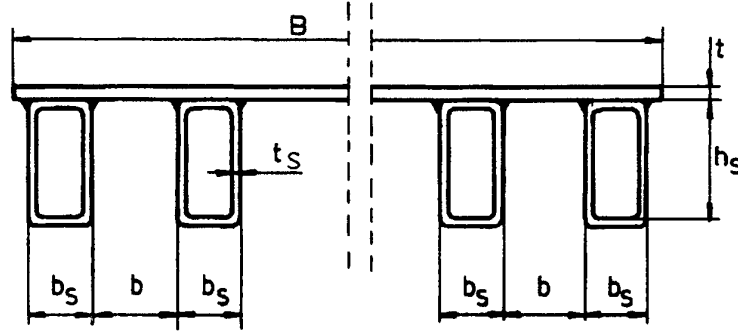


Fig.2. Cross-section of the optimized beam structure

In the optimization procedure the dimensions of RHS ribs and the thickness of the deck plate are sought, which minimize the cost and fulfil the design constraints. The cost function contains the material and fabrication costs. The design constraints relate to the maximum stress due to bending of the whole structure and the local bending of deck plate strips as well as to the limitation of residual welding deflection due to shrinkage of eccentric welds.

For submerged arc welding (SAW) we use in the cost function

$$C_2 a_w^n = 0.2349 \cdot 10^{-3} a_w^2 \quad (27)$$

and the weld length is $L_w = 2nL$ (L in mm), n is the number of longitudinal ribs, A is the cross-sectional area

$$A = nA_{RHS} + Bt \quad (28)$$

where A_{RHS} is the cross-sectional area of a RHS.

RHS are used according to prEN 10219-2 (1992). We select RHS with $b_s = h_s/2$ only. The corner radius is taken as $2t_s$. For the calculation of cross-sectional area and moment of inertia the approximate formulae proposed by DAST Richtlinie 016 (1986) are used as follows:

$$A_{RHS} = 2t_s(1.5h_s - 2t_s) \left(1 - 0.43 \frac{4t_s}{1.5h_s - 2t_s} \right) \quad (29a)$$

$$I_{RHS} = \left[\frac{(h_s - t_s)^3 t_s}{6} + \frac{t_s}{2} \left(\frac{h_s}{2} - t_s \right) (h_s - t_s)^2 \right] \left(1 - 0.86 \frac{4t_s}{1.5h_s - 2t_s} \right) \quad (29b)$$

Constraint on maximum stress due to bending of the whole structure with n ribs of RHS can be formulated as

$$\sigma_{max} = M_{max} / W_x \leq f_{y1} = f_y / \gamma_{M1} \quad (30)$$

where f_y is the yield stress, $\gamma_{M1} = 1.1$ is a partial safety factor according to Eurocode 3 (EC3) (1992), $M_{max} = pL^2/8$, $p = Bp_0$, p_0 is the factored intensity of the uniform normal load, the section modulus and the moment of inertia are given by

$$W_x = I_x / (h_s - y_G) \quad (31)$$

$$I_x = nI_{RHS} + nA_{RHS} \left(\frac{h_s}{2} - y_G \right)^2 + Bty_G^2 \quad y_G = \frac{nA_{RHS}h_s}{2(nA_{RHS} + Bt)} \quad (32)$$

Constraint on reduced stress due to bending of the whole structure and local bending of deck plate strips between ribs can be formulated as follows:

stress in deck plate due to bending of the whole structure is

$$\sigma_L = \sigma_{\max} y_G / (h_s - y_G) \quad (33)$$

and the stress in deck plate due to local bending is

$$\sigma_T = \frac{pb^2}{2t^2} \quad b = \frac{B - 50 - nb_s}{n - 1} \quad (34)$$

$$\text{the reduced stress is} \quad (\sigma_L^2 + \sigma_T^2 + \sigma_L \sigma_T)^{1/2} \leq f_{y1} \quad (35)$$

Local buckling constraint of the deck plate strips can be formulated according to EC3 as follows:

$$b/t \leq 42(235/\sigma_L)^{1/2} \quad (36)$$

Constraint on maximum deflection due to shrinkage of longitudinal welds is given by

$$w_{\max} = CL^2/8 \leq w_{\text{allow}} = L/\theta \quad (37)$$

According to (16)

$$C = 0.844 \times 10^{-3} \times 2nQ_T y_T / I_x \quad (38)$$

where $a_W = 0.5t_s$ and $y_T = y_G - t/2$, Q_T is given by (26).

Data: $n = 6$, $p_0 = 500 \text{ N/m}^2 = 0.5 \times 10^{-3} \text{ N/mm}^2$, $B = 2000$, $L = 12000 \text{ mm}$, $f_y = 355 \text{ MPa}$, $k_f/k_m = 2 \text{ kg/min}$; $\Theta_d = 3$, $\kappa = n + 1$, $\rho = 7.85 \times 10^{-6} \text{ kg/mm}^3$, $b_s = h_s/2$.

Variables: h_s , t_s , t . The optimum design is performed using the Rosenbrock's Hillclimb mathematical programming method complemented by a search for discrete values. Optimum values of variables are calculated for various values of θ to show the effect of distortion limitation. The computational results are given in Table 1.

Table 1. Optimum dimensions in mm, K/k_m values for cost in kg and checks of fulfilling the design constraints (35) and (37) in function of $\theta = L/w_{\text{allow}}$

θ	h_s	t_s	t	K/k_m	(35)(MPa)	(37)(mm)
400	200	5	11	5025	303<323	13.6<30
600	200	5	11	5025	303<323	13.6<20
800	200	5	11	5025	303<323	13.6<15
900	200	5	12	5234	255<323	12.4<13.3
1000	200	5	13	5442	217<323	11.8<12

It can be seen that the deflection constraint is active for $\theta > 800$ and the reduced stress constraint is active for $\theta < 800$.

5. CONCLUSIONS

It is shown how to incorporate the constraint on residual deformation due to shrinkage of welds into the minimum cost design process. To assure the prescribed maximum distortion the structural

dimensions should be increased which increases the cost. For the objective function a special cost function is used containing the material and fabrication costs.

REFERENCES

- Bodt, H.J.M. 1990. The global approach to welding costs. The Netherlands Institute of Welding.
- COSTCOMP, 1990. Programm zur Berechnung der Schweisskosten. Deutscher Verlag für Schweißtechnik, Düsseldorf.
- DAST (Deutscher Ausschuss für Stahlbau) Richtlinie 016. 1986. Bemessung und konstruktive Gestaltung von Tragwerken aus dünnwandigen kaltgeformten Bauteilen. Köln.
- Eurocode 3. 1992. Design of steel structures. Part 1.1. CEN European Committee for Standardization, Brussels.
- Farkas, J., Jármai, K. 1996a. Fabrication cost calculation and optimum design of welded steel silos. *Welding in the World* 37: No.5. 225-232.
- Farkas, J., Jármai, K. 1996b. Minimum cost design of SHS Vierendeel trusses. *Tubular Structures VII. Proc. 7th Int. Symposium on Tubular Structures, Miskolc, 1996.* Eds Farkas, J. and Jármai, K. Balkema, Rotterdam-Brookfield, 463-468.
- Farkas, J., Jármai, K. 1997. Analysis and optimum design of metal structures. Balkema, Rotterdam-Brookfield.
- Farkas, J., Jármai, K. 1998a. Analysis of some methods for reducing residual beam curvatures due to weld shrinkage. *Welding in the World* Vol.41. No.4. 385-398.
- Farkas, J., Jármai, K. 1998b. Optimum design of a welded beam considering the constraint on residual welding deformation. *Proc. Int. Symposium Design and Reconstruction of Steel Structures, Bratislava, Techn. Univ. Bratislava, 1998.* 52-57.
- Farkas, J. 1999. Optimum design of tubular structures. In *Mechanics and design of tubular structures.* Eds. Jármai, K., Farkas, J. Springer, Wien-New York, pp.285-337.
- Farkas, J., Jármai, K. 1999. Optimum design of welded stiffened plates loaded by hydrostatic pressure. *3rd WCSMO World Congress of Structural and Multidisciplinary Optimization, Buffalo, New York, 1999. Short paper proceedings Vol.2.* 493-495.
- Jármai, K., Farkas, J., Horikawa, K. 1998. Economic design of steel bridge decks. *Welding in the World* Vol.41. No.1. 49-59.
- Jármai, K., Farkas, J. 1999a. Optimum cost design of welded box beams with longitudinal stiffeners using advanced backtrack method. *3rd WCSMO World Congress of Structural and Multidisciplinary Optimization, Buffalo, New York, 1999. Short paper proceedings Vol.2.* 363-365.
- Jármai, K., Farkas, J. 1999b. Cost calculation and optimization of welded steel structures. *Journal of Constructional Steel Research* Vol.50. 115-135.
- Okerblom, N.O., Demyantsevich, V.P., Baikova, I.P. 1963. Design of fabrication technology of welded structures. Leningrad, Sudpromgiz (in Russian).
- prEN 1029-2: 1992. Cold formed structural hollow sections of non-alloy and fine grain structural steels. Part 2. European Committee for Standardization, Brussels.

Acknowledgements

This work has been supported by the Hungarian Fund for Scientific Research grant OTKA 22846 and by the Fund of Higher Education 8/2000.

PARTICLE SWARMS IN SIZE AND SHAPE OPTIMIZATION

P.C. Fourie¹ and Albert A. Groenwold²

¹*Department of Mechanical Engineering, Technikon Pretoria, Pretoria, South Africa*

²*Department of Mechanical Engineering, University of Pretoria, Pretoria, South Africa*

ABSTRACT

The particle swarm optimization algorithm (PSOA) is applied to the optimal design of structures with sizing and shape variables. In our implementation of the PSO, the social behavior of birds is mimicked. Individual birds exchange information about their position, velocity and fitness, and the behavior of the flock is then influenced to increase the probability of migration to regions of high fitness.

The parameters in the PSO, namely inertia, craziness and velocity matching, are studied and adapted for structural optimization. Standard benchmark problems in optimal sizing and shape design are used to evaluate the performance of the PSO. The PSO is compared with the genetic algorithm (GA) and also the gradient based recursive quadratic programming (RQP) algorithm. Our implementations suggest that the PSO is superior to the GA, and comparable to the RQP algorithm.

1 INTRODUCTION

In recent years, a number of efficient optimization algorithms mimicking natural phenomena and physical processes have been formulated. Amongst others, notable formulations are the genetic algorithm (GA), simulated biological growth (SBG), simulated annealing (SA) and particle swarm optimization (PSO).

In all probability, the best known of these methods is the GA [1, 2], which mimics natural selection and survival of the fittest. Simulated biological growth [3] mimics phenomena that have been observed in animal and human bone tissue. This involves the addition of bone material in regions of high stress and conversely, the reduction of material in regions of low stress. Simulated annealing [4] is based on statistical thermodynamics and is used to simulate the behavior of the atomic arrangements in solid material during an annealing process.

As opposed to the well established methods mentioned above, PSO is still in its infancy. Proposed by Kennedy and Eberhart [5], the method is based on the simulation of a simplified social model. In particular, *bird flocking* and *fish schooling* are some of the behavioral patterns which are mimicked. Kennedy and Eberhart noted that the social sharing of information among members offers an evolutionary advantage. This observation is fundamental to the development of the PSO. Previously, the PSO has been applied to analytical test functions (mostly unconstrained univariate or bivariate) [6, 7], and multimodal problem generators [6].

In structural optimization, the genetic algorithm [8, 9], simulated biological growth [10] and simulated annealing [11, 12] have been applied successfully¹. Applications of the the GA in structural optimization in particular have been numerous. We are however unaware of an application of the PSO in structural optimization.

¹These selected references are by no means exhaustive.

In this paper, we apply the PSOA to optimal sizing and shape design problems. The originally proposed PSOA is adapted for these problems. The changes relate to the continual reduction in the inertia and maximum velocity of each particle as the optimal solution is approached, the choice of the confidence parameters as well as the use of a craziness function. We also introduce constraints to the PSOA. An important motivation for the current line of research is that the PSOA is easily parallelized, as opposed to conventional gradient based local search algorithms.

The development of our paper is as follows: Firstly, the structural design problem is formulated, where after the PSOA is introduced, studied and modified. Finally, the PSOA is applied to four benchmark problems in size and shape optimization. The performance of the PSOA is compared with that of the GA, and also with the gradient based recursive quadratic programming (RQP) algorithm [13, 14].

2 PROBLEM FORMULATION

We consider two distinct problem classes in structural optimization, namely optimal sizing design and optimal shape design. In both cases, minimum weight is selected as the objective function f . The general optimal design problem is formulated as follows: Find the minimum weight f^* such that

$$f^* = f(\mathbf{x}^*) = \min f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} \quad (1)$$

subject to the general inequality constraints

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \quad (2)$$

where $\mathbf{a}$ and $\mathbf{x}$ are column vectors in $\mathbb{R}^n$, and f and g_j are scalar functions of the design variables $\mathbf{x}$. $\mathbf{x}$ represents the member cross-sections and the geometry of the structure. The inequality constraints g_j resemble stress, strain, displacement or linear buckling constraints. The finite element method (FEM) is used to approximate the objective function f and the constraint functions g_j .

To facilitate inclusion of the constraints (2) in the GA and the PSOA, (1) is modified to become

$$\bar{f} = f(\mathbf{x}) + \sum_{j=1}^m \lambda_j [g_j(\mathbf{x})]^2 \mu_j(g_j) \quad (3)$$

with

$$\mu_j(g_j) = \begin{cases} 0 & \text{if } g_j(\mathbf{x}) \leq 0 \\ 1 & \text{if } g_j(\mathbf{x}) > 0 \end{cases} \quad (4)$$

and $\lambda_j > 0$, prescribed.

3 AN OUTLINE OF PARTICLE SWARM OPTIMIZATION

For reasons of brevity, we restrict ourselves to bird flocking². In the following, the terms *particle* and *birds* are used interchangeably. In flight, each bird in a flock is considered to continuously process information about its current *position* and *velocity*. In addition, information regarding its position with respect to the flock is also processed. In (structural) optimization, the position of each bird is represented by the design variables $\mathbf{x}$, while the velocity of each bird $\mathbf{v}$ influences the incremental change in the position of each bird, and hence the design variables. The value of the objective function

²Promising results based on bee swarms have recently been proposed. In particular, the swarm-and-queen approach [16] seems worthy of future perusal.

Problem		PLBA	RQP	GA [15]	GA	PSO	Analytical
RS	x_1	0.	0.	0.0196	0.0096	-0.0166	0.
	x_2	1.	1.	1.0026	0.9271	0.9874	1.
	x_3	2.	2.	1.9840	2.0130	2.0152	2.
	x_4	-1.	-1.	-1.0190	-0.9951	-0.9841	-1.
	f^*	56.00	56.00	56.01	56.03	56.00	56.00
	N_{fe}	13	123	12000	35520	1304	—
SPA	x_1	0.0517	0.0524	0.0646	0.0523	0.0509	0.0517
	x_2	0.3570	0.3746	0.6532	0.3712	0.3387	0.3570
	x_3	11.3000	10.33	4.5001	10.630	12.446	11.3000
	f^*	0.01268	0.0127	0.0177	0.0128	0.0127	0.01268
	N_{fe}	58	21	11000	6480	927	—

Table 1: Numerical results for two analytical test functions, namely the Spring Design problem (SPA), and the Rosen-Suzuki problem (RS). The results obtained using the various algorithms may be compared with the analytical solution given in the last column.

$f(x)$ has a direct influence on the general direction of travel of the flock: The higher the fitness of bird d , the more likely it is to influence the flock to fly in its general direction. (Here, ‘fitness’ is used in the same sense as in the genetic algorithm.)

Let us now consider a flock of p particles or birds. For particle d , Shi and Eberhart [7] propose that the position $\mathbf{x}^d$ is updated as

$$\mathbf{x}_{k+1}^d = \mathbf{x}_k^d + \mathbf{v}_{k+1}^d, \quad (5)$$

while the velocity $\mathbf{v}^d$ is updated as

$$\mathbf{v}_{k+1}^d = w\mathbf{v}_k^d + c_1r_1(\mathbf{p}_k^d - \mathbf{x}_k^d) + c_2r_2(\mathbf{p}_k^g - \mathbf{x}_k^d). \quad (6)$$

Here, the scalar w represents the inertia of each particle, subscript k indicates pseudo-time, and $\mathbf{p}_k^d$ represents the best position of particle d to date, while $\mathbf{p}_k^g$ represents the best position of the swarm at time k . r_1 and r_2 represent uniform random numbers between 0 and 1. The ‘attraction’ between particles is controlled by the parameters c_1 and c_2 . This mimics the level of *trust* or *confidence* between individuals, and prevents separation of an individual bird from the flock. In particle swarm jargon, the updating of the velocity is referred to as ‘velocity matching’.

Kennedy and Eberhart [5] propose that $c_1 = c_2 = 2$, to allow a mean of 1 (when multiplied by the random numbers r_1 and r_2). This results in birds *overflying* the target half the time. Shi and Eberhart [7] propose that the inertia w is selected such that $0.8 < w < 1.4$. In addition, they report improved convergence rates when w is decreased linearly during the optimization.

3.1 On inertia, maximum velocity and the best position to date

Noting that the problems we consider are mostly convex in nature, we propose that $\mathbf{p}_g$ (indicating the best ever position in the swarm) replaces the best position of the swarm $\mathbf{p}_k^g$ at time k when updating the velocity. Hence, (6) is modified to become

$$\mathbf{v}_{k+1}^d = w\mathbf{v}_k^d + c_1r_1(\mathbf{p}_k^d - \mathbf{x}_k^d) + c_2r_2(\mathbf{p}_g - \mathbf{x}_k^d). \quad (7)$$

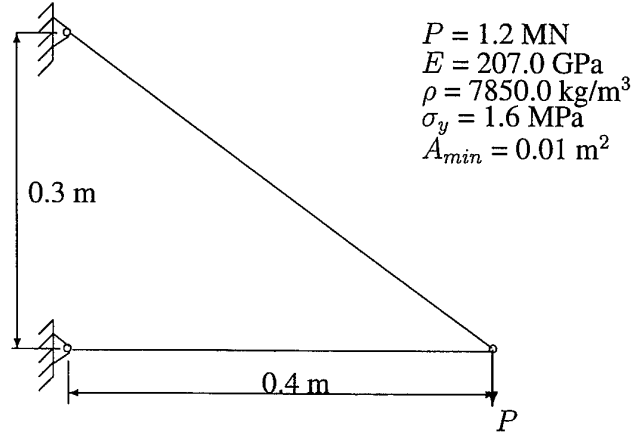


Figure 1: 2-bar truss.

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
RQP	10	58	9	76	8.046×10^3
GA	390	2340	0	2340	8.083×10^3
PSO	111	333	0	333	8.046×10^3
Analytical [17]	-	-	-	-	8.046×10^3

Table 2: Numerical results for the 2-bar truss

The value of w is problem dependent [7]. Based on numerical experimentation, we select a fixed starting value w^0 and decrease this value by the fraction α if no improved solution is obtained within h consecutive time steps, i.e.

$$\text{if } f(\mathbf{p}_g)|_k \geq f(\mathbf{p}_g)|_{k-h} \text{ then } w_{k+1} = \alpha w_k, \quad 0 < \alpha < 1. \quad (8)$$

Simultaneously, we reduce the maximum allowed velocity when the inertia is reduced, i.e.

$$\text{if } f(\mathbf{p}_g)|_k \geq f(\mathbf{p}_g)|_{k-h} \text{ then } \mathbf{v}_{k+1}^{max} = \beta \mathbf{v}_k^{max}, \quad 0 < \beta < 1. \quad (9)$$

Through (8) and (9) we attempt to reduce the sensitivity of the PSOA to the values of w and $\mathbf{v}$. Numerical experimentation suggests that this approach improves the convergence rate of the algorithm, as opposed to linearly decreasing w during the optimization.

3.2 On craziness

Kennedy and Eberhart [5] introduced a craziness operator to mimic the random (temporary) departure of birds from the flock. However, this operator was superseded by the introduction of a cornfield vector, initially included for demonstration purposes. We reintroduce the concept of craziness, with the particles having a predetermined probability of craziness Pr_c . The direction and magnitude of the velocity of influenced particles is then changed randomly, i.e.

$$\text{if } r < \text{Pr}_c, \text{ then randomly assign } \mathbf{v}_{k+1}, \quad 0 < \mathbf{v}_{k+1} \leq \mathbf{v}^{max} \quad (10)$$

for all particles d , where r again represents a uniform random number between 0 and 1.

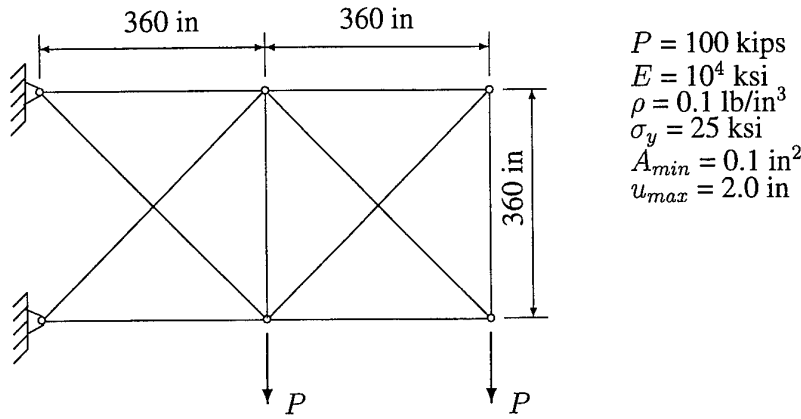


Figure 2: 10-bar truss.

Craziness seems to us to have some similarity with the mutation operator in the genetic algorithm, and increases directional diversity in the flock. ‘Crazy’ birds explore previously uncovered ground, which in general increases the probability of finding the optimum, albeit at additional computational expense, since the optimal craziness in the flock cannot be predetermined.

4 ALGORITHM

We now present an outline of our implementation of the PSOA. We introduce s , the number of steps in which no improvement in the objective function occurs within a prescribed tolerance ε .

1. **Initialization step:** Set k_{max} , c_1 , c_2 , Pr_c , h , α , β , v_0^{max} , s and ε .
 Randomly generate $x_0^d \in D$ in $\mathbb{R}^n$, and $0 < v_0^d < v_0^{max}$, for $d = 1, 2, \dots, p$.
 Set $k := 1$.
2. **Optimization steps:**
 - (a) Evaluate the function $\bar{f}$ according to (3) for each particle. Record the best function value $\bar{f}_k$ for time step k .
 - (b) If $k > s$ and $(|\bar{f}_k - \bar{f}_{k-s}|) / |\bar{f}_k| < \varepsilon$, go to 3.
 - (c) If $k > h$, conditionally reduce the inertia w and velocity v^{max} , using (8) and (9).
 - (d) Update the best position p_k^d for particle d and the best ever position p_g .
 - (e) Set $k := k + 1$. If $k = k_{max}$, go to 3, else go to 2a.
 - (f) Update the velocity v , according to (7).
 - (g) Stochastically implement craziness, using (10).
 - (h) Update the position x , according to (5).
 - (i) Go to 2a.
3. **Termination:** STOP.

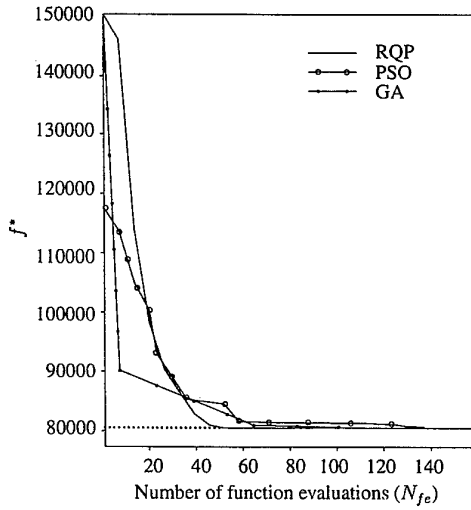


Figure 3: Convergence history for the 2-bar truss.

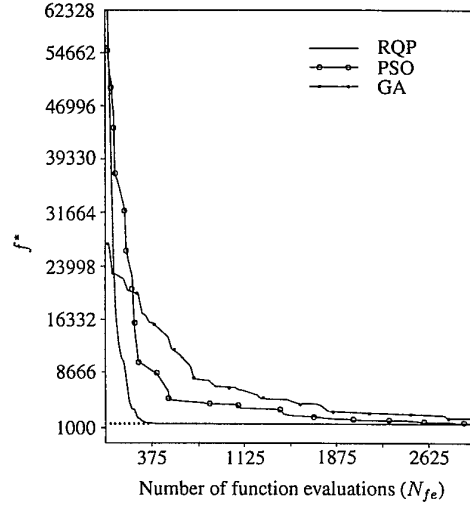


Figure 4: Convergence history for the 10-bar truss. (Subproblem 1.)

5 NUMERICAL RESULTS

In this section, we compare our implementation of the PSOA with our implementation of the gradient based recursive quadratic programming (RQP) algorithm [13, 14], the GA implementation by Carroll [15], and our implementation of the GA³. These algorithms were all coded in FORTRAN77, and numerical results were obtained using a 500 MHz personal computer. Where possible, we also compare our results with published results for the NPSOL algorithm [19] and the PLBA algorithm [14].

We report the number of function evaluations N_{fe} required to find f^* to a tolerance of 10^{-3} . We also report the *compound cost* N_c , which we define as $N_c = N_{fe} + nN_{ge}$, where N_{ge} indicates the number of gradient evaluations. N_c represents a pseudo cost which allows direct comparison between the efficiency of the gradient based algorithms and the derivative free algorithms.

To illustrate the robustness of the PSOA, all the examples were performed using $c_1 = c_2 = 0.9$, $Pr_c = 0.4$, $w_0 = 1.4$, $\alpha = 0.99$, and $\beta = 0.8$. For the PSOA and the GA, $k_{max} = 700$ iterations (or generations) was used. Unless otherwise stated, the population size in both implementations of the GA was always set to $3n$, while for the PSOA, the number of particles p was always set to $p = n$.

5.1 Analytical Test Functions

We apply our implementation of the PSOA to two simple, well-known analytical test functions, namely the Rosen-Suzuki problem (RS) [14, 21], and the so-called Spring Design problem (SPA) [21]. Numerical results are tabulated in Table 1.

As can be expected for smooth constrained functions, the gradient based algorithms require fewer

³Our GA is based on a binary representation, and elitism is used to ensure that the current best individual never vanishes. Selection pressure is applied similar to that in the canonical GA of Whitley [18]. A uniform crossover strategy is used, while jump mutation is employed to protect against loss of genetic diversity. In addition, the probability of mutation is increased as the generations start to converge.

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
NPSOL [20]	10	13	180	1813	1.665×10^3
RQP	23	497	84	1337	1.665×10^3
GA	492	14760	0	14760	2.439×10^3
PSO	665	6650	0	6650	1.665×10^3

Table 3: Numerical results for the 10-bar truss. (Subproblem 1.)

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
NPSOL [20]	32	50	608	6130	5.061×10^3
RQP	32	683	93	1613	5.077×10^3
GA	569	17070	0	17070	5.837×10^3
PSO	159	1590	0	1590	5.066×10^3

Table 4: Numerical results for the 10-bar truss. (Subproblem 2.)

function analyses to converge than the derivative free implementations of the GA and the PSOA. Notwithstanding the fact that very few particles were used in the PSOA, this algorithm is notably more efficient than both implementations of the GA. (For the GA and the PSOA, the reported results are the average of 10 independent runs of the respective algorithms.)

5.2 2-bar truss

The geometry and loading conditions of this simple structure are depicted in Figure 1. The member cross-sections represent the design variables. Ref [17] gives an analytical solution $f^* = 8.046 \times 10^3$. Only 3 particles were used to solve this problem using the PSOA. Numerical results are tabulated in Table 2, while the convergence history is depicted in Figure 3.

The PSOA is more expensive than the RQP algorithm, but notably less expensive than the GA. In addition, the quality of the solution found using the PSOA is superior to the solution found using the GA. The convergence history (depicted in Figure 3) reveals that most of the function values in the PSOA and GA are associated with refinement of the minimum to the (reasonably stringent) tolerance of 10^{-3} . The neighborhood of the the minimum is however found quite quickly.

5.3 10-bar truss

The geometrical data of the non-convex 10-bar truss structure is depicted in Figure 2. The member cross-sectional areas represents the design variables. Two distinct cases are considered, namely stress constraints only (Subproblem 1), and stress and displacement constraints (Subproblem 2). For the two subproblems, the results are tabulated in Tables 3 and 4 respectively, while the convergence histories are plotted in Figures 4 and 5 respectively (for $\beta = 0.92$).

Once again, most of the computational effort of the PSOA is associated with refinement of the minimum to the prescribed tolerance, with the PSOA yielding superior results to the GA.

5.4 Torque arm

This problem (Figure 7) is an adaption of the problem studied by Bennett and Botkin [22]. The outer boundaries of the torque arm are represented by a spline function, described by the 7 design

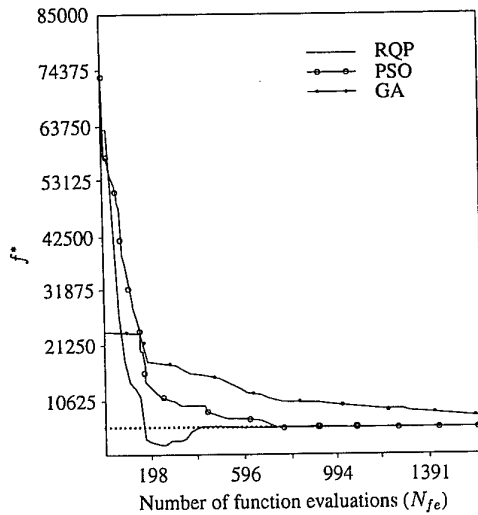


Figure 5: Convergence history for the 10-bar truss. (Subproblem 2.)

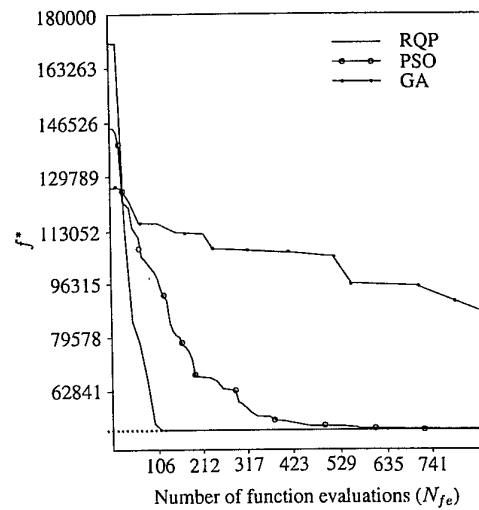


Figure 6: Convergence history for the torque arm.

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
RQP	9	144	92	788	4.61×10^4
GA	215	4515	0	4515	4.86×10^4
PSO	121	847	0	847	4.62×10^4

Table 5: Numerical results for the torque arm.

variables. No topological changes are allowed, and geometric symmetry is enforced. To prevent excessive distortion of the finite element mesh, move limits were included. The allowable stress limit is 0.8 MPa in both tension and compression.

The quality of the solutions found with the RQP algorithm and the PSOA are superior to the solution found using the GA. In addition, the GA is notably more expensive than the PSOA. The compound cost N_c of the PSOA and the RQP algorithm are comparable.

6 CONCLUSIONS

We have applied the particle swarm optimization algorithm (PSOA) to the optimal design of structures with sizing and shape variables. Our PSOA mimics the social behavior of birds in flight, and the parameters in the algorithm, namely inertia, craziness and velocity matching, are modified for structural optimization.

Using comparative studies and benchmark problems, the suitability of the PSOA for problems in structural optimization is demonstrated. Although implemented in a simple form, the PSOA outperforms the GA for our implementation. The computational effort is, for some problems, comparable to that of gradient based recursive quadratic programming algorithm. An important motivation for this

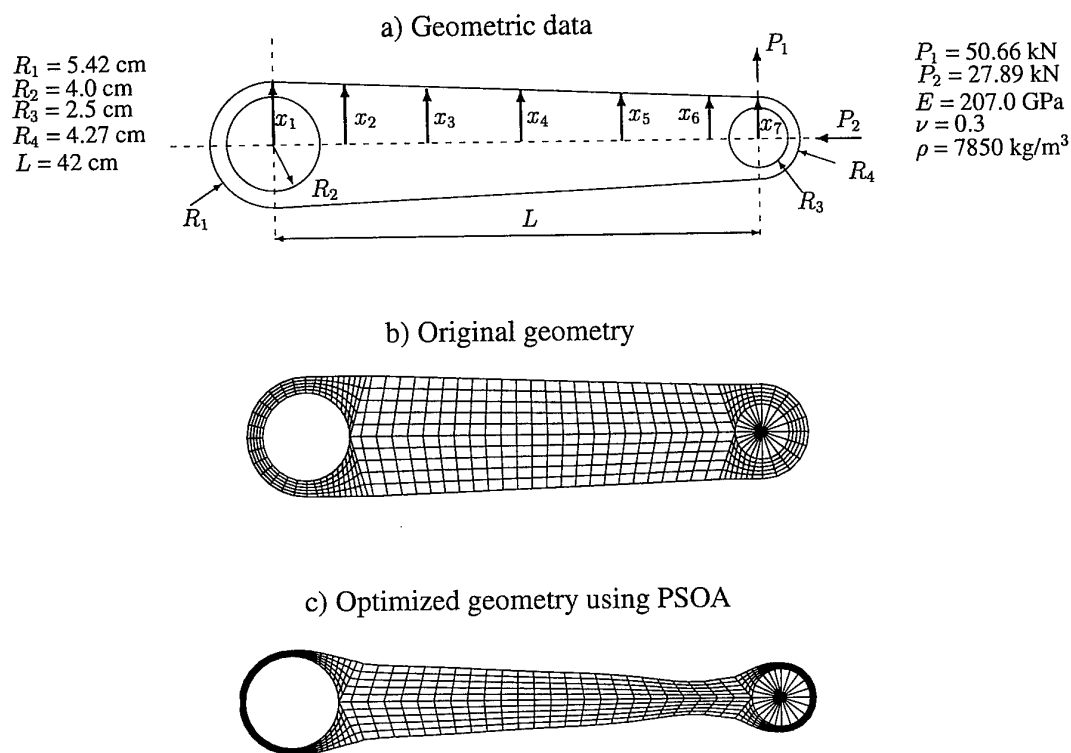


Figure 7: Definition and discretization of the torque arm.

line of research is that the PSOA can easily be parallelized on massive parallel processing machines.

REFERENCES

- [1] D. Beasley, D.R. Bull, and R.R. Martin. An overview of genetic algorithms: Part 1, Fundamentals. *University Computing*, 15:58–69, 1993.
- [2] D. Beasley, D.R. Bull, and R.R. Martin. An overview of genetic algorithms: Part 2, Research topics. *University Computing*, 15:170–181, 1993.
- [3] C. Mattheck and S. Burkhardt. A new method of structural shape optimization based on biological growth. *International Journal of Fatigue*, 12:185–190, 1990.
- [4] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1084–1092, 1953.
- [5] J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.
- [6] J. Kennedy and W.M. Spears. Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. *Proceedings of the 1998 International Conference on Evolutionary Computation*, pages 78–83, 1998.

- [7] Y. Shi and R.C. Eberhart. Parameter selection in particle swarm optimization. *The Seventh Annual Conference on Evolutionary Programming*, 1998.
- [8] P. Hajela. Genetic search - An approach to the nonconvex optimization problem. *AIAA Journal*, 28:1205–1210, 1990.
- [9] A.A. Groenwold, N. Stander, and J.A. Snyman. A regional genetic algorithm for the discrete optimal design of truss structures. *International Journal for Numerical Methods in Engineering*, 44:749–766, 1999.
- [10] J.L. Chen and W.C. Tsai. Shape optimization by using simulated biological growth approaches. *AIAA Journal*, 31:2143–2147, 1993.
- [11] P.Y. Shim and S. Manoochehri. Generating optimal configurations in structural design using simulated annealing. *International Journal for Numerical Methods in Engineering*, 40:1053–1069, 1997.
- [12] R.P. Shimpi and S. Joshi. Simulated annealing optimization using features of genetic algorithm. *3rd World Congress of Structural and Multidisciplinary Optimization*, 1999. Paper No. 50-GAM2-4.
- [13] A.D. Belegundu and J.S. Arora. A recursive quadratic programming method with active set strategy for optimal design. *International Journal for Numerical Methods in Engineering*, 20:803–816, 1984.
- [14] O.K. Lim and J.S. Arora. An active set RQP algorithm for engineering design optimization. *Computer Methods in Applied Mechanics and Engineering*, 57:51–65, 1986.
- [15] D. L. Carroll. Chemical laser modeling with genetic algorithms. *AIAA J.*, 34:338–346, 1996.
- [16] M. Clerc. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. *Congress on Evolutionary Computation, Washington D.C., IEEE*, 1999.
- [17] J. Arora. *Introduction to Optimum Design*. McGraw-Hill Book Company, 1989.
- [18] D. Whitley. A genetic algorithm tutorial. *Statics and Computing*, 4:65–85, 1994.
- [19] P.B. Thanedar, J.S. Arora, C.H. Tseng, O.K. Lim, and G.J. Park. Performance of some SQP algorithms on structural design problems. *International Journal for Numerical Methods in Engineering*, 23:2187–2203, 1986.
- [20] P.B. Thanedar, G.J. Park, and J.S. Arora. Performance of two superlinearly convergent RQP optimization algorithms on some structural design problems. Technical report, Optimal Design Laboratory, College of Engineering, The University of Iowa, Iowa City, Iowa 52242, 1985.
- [21] W. Hock and K. Schittkowski. *Test examples for nonlinear programming codes*. Lecture Notes in Economics and Mathematical Systems 187 (Springer, New York), 1980.
- [22] J.A. Bennett and M.E. Botkin. Structural shape optimization with geometric description and adaptive mesh refinement. *AIAA Journal*, 23:458–464, 1984.

Control of a Three - Link Manipulator Subject to Inequality Constraints

C. FRANGOS¹ and Y. YAVIN²

¹Department of Statistics , The Rand Afrikaans University , P O Box 524 ,
Auckland Park 2006 , Johannesburg , South Arica .

²Department of Electrical and Electronic Engineering , University of Pretoria,
Pretoria 0002 , South Africa.

ABSTRACT This work deals with the control of a three - link planar manipulator whose motion is subjected to inequality constraints on the trajectories of its joints , and to other constraints on the trajectory of its end - effector .

KEYWORDS Three - link planar manipulator , inequality constraints on trajectories .

1 Introduction

This work deals with the control of a three - link planar manipulator . It is assumed here that the motion of all parts of the manipulator is confined to a vertical (X, Z) - plane . It is further assumed that the motion of the manipulator is driven by three motors . The first motor is located at the base of the first link , the second motor is located at the joint r_1 between the first and second links , and the third motor is located at the joint r_2 between the second and third links (see Fig. 1). Let r_{EF} denote the location of the manipulator's end - effector . Given a (solid) plane $\mathcal{B} = \{(x, y, z) : z - (ax + c) = 0\}$ in the (X, Y, Z) - space . Let r_A and r_B denote two given points , $r_A \neq r_B$, each of them located in a narrow strip that is situated in the (X, Z) plane below $\mathcal{B}$, and such that $r_{EF}(0) = r_A$. Also , let $t_f > 0$ be a given number . In this work we consider the motion of the manipulator in which its end - effector is confined to move in the above - mentioned strip and where all the parts of the manipulator are confined to move strictly below $\mathcal{B}$. Thus the following

control problem is considered here : Find control laws for the torques applied on the links ¹ such that : (i) $\mathbf{r}_{EF}(t_f)$ will be in the above - mentioned strip and in a small neighbourhood of $\mathbf{r}_B$. (ii) During the time interval $[0, t_f]$ the system's motion will be subjected to the following constraints :

$$-\epsilon_1 \leq z_{EF}(t) - (a x_{EF}(t) + c) \leq -\epsilon_2 \quad , \quad \text{for all } t \in [0, t_f] , \quad (1)$$

$$z_k(t) - (a x_k(t) + c) \leq -\epsilon_2 \quad , \quad k = 1, 2 \quad , \quad \text{for all } t \in [0, t_f] , \quad (2)$$

where $\mathbf{r}_{EF} = x_{EF}\mathbf{I} + z_{EF}\mathbf{K}$, $\mathbf{r}_k = x_k\mathbf{I} + z_k\mathbf{K}$, $k = 1, 2$; and ϵ_i , $i = 1, 2$ are given positive numbers , $\epsilon_2 < \epsilon_1$. If equations (1) and (2) are satisfied then all the parts of the manipulator will move strictly below B .

2 Dynamical Model

In this work , we consider the control of the motion of a three - link planar manipulator . Let $\mathbf{I}$, $\mathbf{J}$ and $\mathbf{K}$ be unit vectors along an inertial (X, Y, Z) - coordinate system . Denote by $\mathbf{i}_k$

$$\mathbf{i}_k = \cos \theta_k \mathbf{I} + \sin \theta_k \mathbf{K} , \quad (3)$$

a unit vector along the k -th link , $k = 1, 2, 3$ (see Fig. 1) , and let

$$\mathbf{j}_k = -\sin \theta_k \mathbf{I} + \cos \theta_k \mathbf{K} , \quad (4)$$

¹These torques are generated by the motors . However , the inclusion of the motors' dynamics in the dynamical model of the system and the computation of the corresponding control laws for the inputs to the motors will be dealt with elsewhere . Here , only the contribution of the motors' masses to the dynamical model of the system is considered .

be a unit vector perpendicular to $\mathbf{i}_k$, $k = 1, 2, 3$ respectively. Note that

$$\frac{d\mathbf{i}_k}{dt} = \frac{d\theta_k}{dt} \mathbf{j}_k, \quad k = 1, 2, 3. \quad (5)$$

The motion of the manipulator is driven by three motors. The first motor is located at the origin (see Fig. 1), the second motor is located at the point $\mathbf{r}_1$, $\mathbf{r}_1 = l_1 \mathbf{i}_1$; and the third motor is located at $\mathbf{r}_2$, $\mathbf{r}_2 = l_1 \mathbf{i}_1 + l_2 \mathbf{i}_2$.

Here l_k denotes the length of link k , $k = 1, 2, 3$. Also, $\mathbf{r}_{Ck}$, the location of the center of mass of link k , $k = 1, 2, 3$, is given by: $\mathbf{r}_{C1} = l_{C1} \mathbf{i}_1$, $\mathbf{r}_{C2} = l_1 \mathbf{i}_1 + l_{C2} \mathbf{i}_2$ and $\mathbf{r}_{C3} = l_1 \mathbf{i}_1 + l_2 \mathbf{i}_2 + l_{C3} \mathbf{i}_3$, $0 < l_{Ck} < l_k$, $k = 1, 2, 3$. In addition, the location of the end-effector is given by $\mathbf{r}_{EF} = l_1 \mathbf{i}_1 + l_2 \mathbf{i}_2 + l_3 \mathbf{i}_3$.

Denote by m_k the mass of link k and by m_{Rk} the mass of motor k , $k = 1, 2, 3$, respectively. Thus, using these notations, the Lagrangian function for the motion of the manipulator is given by

$$\begin{aligned} \mathcal{L} = & \frac{1}{2} A_{11} \left(\frac{d\theta_1}{dt} \right)^2 + \frac{1}{2} A_{22} \left(\frac{d\theta_2}{dt} \right)^2 + \frac{1}{2} A_{33} \left(\frac{d\theta_3}{dt} \right)^2 + A_{12} \frac{d\theta_1}{dt} \frac{d\theta_2}{dt} \cos(\theta_2 - \theta_1) \\ & + A_{13} \frac{d\theta_1}{dt} \frac{d\theta_3}{dt} \cos(\theta_3 - \theta_1) + A_{23} \frac{d\theta_2}{dt} \frac{d\theta_3}{dt} \cos(\theta_3 - \theta_2) \\ & - [V_{O1} g \sin \theta_1 + V_{O2} g \sin \theta_2 + V_{O3} g \sin \theta_3], \end{aligned} \quad (6)$$

where

$$A_{11} = m_1 l_{C1}^2 + I_1 + (m_2 + m_{R2} + m_3 + m_{R3}) l_1^2,$$

$$A_{22} = m_2 l_{C2}^2 + I_2 + (m_3 + m_{R3}) l_2^2, \quad A_{33} = m_3 l_{C3}^2 + I_3,$$

$$A_{12} = m_2 l_1 l_{C2} + (m_3 + m_{R3}) l_1 l_2, \quad A_{13} = m_3 l_1 l_{C3},$$

$$A_{23} = m_3 l_2 l_{C3}, \quad V_{O1} = m_1 l_{C1} + (m_2 + m_{R2} + m_3 + m_{R3}) l_1,$$

$$V_{O2} = m_2 l_{C2} + (m_3 + m_{R3}) l_2 , \quad V_{O3} = m_3 l_{C3} .$$

In the expressions above , I_k denotes the moment of inertia of link k about a vector in the direction of J located at r_{Ck} , $k = 1, 2, 3$, respectively .

Denote

$$\mathbf{q} = (\theta_1, \theta_2, \theta_3)^T , \quad \mathbf{p} = \left(\frac{d\theta_1}{dt}, \frac{d\theta_2}{dt}, \frac{d\theta_3}{dt} \right)^T .$$

Using the expression for r_{EF} and (5) , the velocity v_{EF} of the end - effector is given by

$$\begin{aligned} v_{EF}^2 = & \sum_{k=1}^3 l_k^2 \left(\frac{d\theta_k}{dt} \right)^2 + 2 l_1 l_2 \frac{d\theta_1}{dt} \frac{d\theta_2}{dt} \cos(\theta_2 - \theta_1) \\ & + 2 l_1 l_3 \frac{d\theta_1}{dt} \frac{d\theta_3}{dt} \cos(\theta_3 - \theta_1) + 2 l_2 l_3 \frac{d\theta_2}{dt} \frac{d\theta_3}{dt} \cos(\theta_3 - \theta_2) . \end{aligned} \quad (7)$$

In this work the motion of the system is subjected to the following constraints

$$-\epsilon_1 \leq z_{EF}(t) - (a x_{EF}(t) + c) \leq -\epsilon_2 , \quad \text{for all } t \in [0, t_f] , \quad (8)$$

and

$$z_k(t) - (a x_k(t) + c) \leq -\epsilon_2 , \quad k = 1, 2 , \quad \text{for all } t \in [0, t_f] , \quad (9)$$

or , equivalently , using the expressions for r_1 , r_2 and r_{EF} , equations (8) and (9) can be written as

$$-\epsilon_1 \leq \sum_{i=1}^3 l_i \sin \theta_i(t) - a \sum_{i=1}^3 l_i \cos \theta_i(t) - c \leq -\epsilon_2 , \quad t \in [0, t_f] , \quad (10)$$

$$\sum_{i=1}^k l_i \sin \theta_i(t) - a \sum_{i=1}^k l_i \cos \theta_i(t) - c \leq -\epsilon_2 , \quad t \in [0, t_f] , \quad k = 1, 2 . \quad (11)$$

Assume that equations (10) , (11) are replaced by

$$f_k(\mathbf{q}(t)) = \sum_{i=1}^k l_i \sin \theta_i(t) - a \sum_{i=1}^k l_i \cos \theta_i(t) - c + \epsilon_2 \leq 0, \quad t \in [0, t_f], \quad (12)$$

$k = 1, 2, 3$. Then, it can be shown, by using the Calculus of Variations [1] and introducing the Valentine variables $\psi_k(t)$, $k = 1, 2, 3$, [2], that the equations of motion of the system are determined by

$$\tau_j(t) = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial p_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} - \sum_{i=1}^3 \lambda_i(t) \frac{\partial f_i}{\partial q_j}, \quad j = 1, 2, 3, \quad (13)$$

$$\lambda_i(t) \psi_i(t) = 0, \quad \psi_i(t) \geq 0, \quad i = 1, 2, 3, \quad (14)$$

which have to be solved together with

$$f_i(\mathbf{q}(t)) + \psi_i^2(t) = 0, \quad i = 1, 2, 3. \quad (15)$$

In (13), τ_j is the torque applied by motor j on link j , $j = 1, 2, 3$, respectively.

Remark 1 Assume that for the time interval $[0, t_f]$, $f_i(\mathbf{q}(t)) < 0$, $i = 1, 2, 3$, then, (15) implies: $\psi_i(t) > 0$, $i = 1, 2, 3$, $t \in [0, t_f]$. Consequently, (14) implies: $\lambda_i(t) = 0$, $i = 1, 2, 3$, $t \in [0, t_f]$. Thus, in this case, equations (13) - (15) reduce to

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial p_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} = \tau_j(t), \quad f_j(\mathbf{q}(t)) < 0, \quad j = 1, 2, 3, \quad t \in [0, t_f].$$

It can be shown that equations (16)

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial p_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} = \tau_j(t), \quad j = 1, 2, 3, \quad t \in [0, t_f] \quad (16)$$

which are the classical Lagrange equations [3], are also valid in the domain $\mathcal{D}$

$$\begin{aligned} \mathcal{D} &= \{ \mathbf{q} \in \mathbb{R}^3 : -\epsilon_1 + \epsilon_2 < \sum_{i=1}^3 l_i \sin \theta_i - a \sum_{i=1}^3 l_i \cos \theta_i - c + \epsilon_2 < 0 \\ &\text{and} \quad \sum_{i=1}^k l_i \sin \theta_i - a \sum_{i=1}^k l_i \cos \theta_i - c + \epsilon_2 < 0, \quad k = 1, 2 \}. \end{aligned} \quad (17)$$

Henceforward, the solution of the control problem dealt with here will be confined to $\mathbf{q}(t)$ - trajectories which are in the set $\mathcal{D}$ for all $t \in [0, t_f]$.

Thus , using the Lagrangian function (6) and the Lagrange equations , (16), the following equations are obtained

$$\mathbf{M}(\mathbf{q}) \frac{d^2 \mathbf{q}}{dt^2} + \mathbf{h}(\mathbf{q}, \mathbf{p}) = \boldsymbol{\tau} \quad , \quad \mathbf{q} \in \mathcal{D} , \quad (18)$$

where , using the notations $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)^T$, $\mathbf{M}_{ij} = m_{ij}$, $i, j = 1, 2, 3$ and $\mathbf{h} = (h_1, h_2, h_3)^T$

$$m_{11} = A_{11} \quad , \quad m_{12} = A_{12} \cos(\theta_2 - \theta_1) \quad , \quad m_{13} = A_{13} \cos(\theta_3 - \theta_1) , \quad (19)$$

$$m_{21} = m_{12} \quad , \quad m_{22} = A_{22} \quad , \quad m_{23} = A_{23} \cos(\theta_3 - \theta_2) , \quad (20)$$

$$m_{31} = m_{13} \quad , \quad m_{32} = m_{23} \quad , \quad m_{33} = A_{33} , \quad (21)$$

$$h_1 = -A_{12} \left(\frac{d\theta_2}{dt} \right)^2 \sin(\theta_2 - \theta_1) - A_{13} \left(\frac{d\theta_3}{dt} \right)^2 \sin(\theta_3 - \theta_1) + V_{O1} g \cos \theta_1 , \quad (22)$$

$$h_2 = A_{12} \left(\frac{d\theta_1}{dt} \right)^2 \sin(\theta_2 - \theta_1) - A_{23} \left(\frac{d\theta_3}{dt} \right)^2 \sin(\theta_3 - \theta_2) + V_{O2} g \cos \theta_2 , \quad (23)$$

$$h_3 = A_{13} \left(\frac{d\theta_1}{dt} \right)^2 \sin(\theta_3 - \theta_1) + A_{23} \left(\frac{d\theta_2}{dt} \right)^2 \sin(\theta_3 - \theta_2) + V_{O3} g \cos \theta_3 . \quad (24)$$

It can be shown , for a proper set of parameters , that $\det \mathbf{M}(\mathbf{q}) > 0$, for all θ_k , $k = 1, 2, 3$. Eqs. (18) - (24) constitute the dynamical model for the system dealt with here .

3 Inverse Dynamics Control

Denote $\mathbf{r}_{EF} = x_{EF}\mathbf{I} + z_{EF}\mathbf{K}$ and let $\mathbf{r}_B$ be given by $\mathbf{r}_B = x_D\mathbf{I} + z_D\mathbf{K}$.

In this section a procedure is described for the derivation of control laws, for the torques τ_k , $k = 1, 2, 3$ such that the motion of the manipulator will satisfy the following specifications:

$$|x_D - x_{EF}(t_f)| < \epsilon_{EF}, \quad |z_D - z_{EF}(t_f)| < \epsilon_{EF}, \quad (25)$$

and during $[0, t_f]$

$$-\epsilon_1 + \epsilon_2 < \sum_{i=1}^3 l_i \sin \theta_i(t) - a \sum_{i=1}^3 l_i \cos \theta_i(t) - c + \epsilon_2 < 0, \quad (26)$$

$$\sum_{i=1}^k l_i \sin \theta_i(t) - a \sum_{i=1}^k l_i \cos \theta_i(t) - c + \epsilon_2 < 0, \quad k = 1, 2, \quad (27)$$

where ϵ_{EF} is a given positive number. Thus, the proposed procedure is as follows: First, by introducing the following control law

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\mathbf{v} + \mathbf{h}(\mathbf{q}, \mathbf{p}), \quad \mathbf{q} \in \mathcal{D}, \quad (28)$$

where $\mathbf{v} = (v_1, v_2, v_3)^T$, then, for cases for which $\det \mathbf{M}(\mathbf{q}) \neq 0$, for all $\mathbf{q}$, equations (18) and (28) imply

$$\frac{d^2 \theta_1}{dt^2} = v_1, \quad \frac{d^2 \theta_2}{dt^2} = v_2, \quad \frac{d^2 \theta_3}{dt^2} = v_3, \quad \mathbf{q} \in \mathcal{D}. \quad (29)$$

The control law given by (28) is called the *inverse dynamics control* (see for example, [4]).

Second, equations (29) are solved by choosing the auxiliary control functions v_1, v_2 and v_3 such that the specifications given by (25) - (27) will be satisfied. This control problem will be called here *the auxiliary control problem*.

Hence, once v_k , $k = 1, 2, 3$ are computed, by solving the auxiliary control problem, then, the required torques τ_k , $k = 1, 2, 3$ are computed by using (28).

The auxiliary control functions will be computed here by using the method of feasible command functions, see for example [5].

4 The Auxiliary Control Problem

As mentioned above, this work deals with the motion of the system only during the time interval $[0, t_f]$, where $t_f > 0$ is a given number.

Let $\tau_0 = 0 < \tau_1 < \tau_2 < \dots < \tau_{N-1} = t_f$ be a partition of $[0, t_f]$ such that $\tau_{i+1} - \tau_i = \Delta_c$, $i = 0, \dots, N-2$.

In this work the following class of auxiliary control functions $\mathbf{v} = (v_1, v_2, v_3)^T$ is dealt with. Consider the class of all functions $\mathbf{v} = (v_1, v_2, v_3)^T : [0, t_f] \rightarrow \mathbb{R}^3$ such that

$$v_1(t) = A_i(t) c_i + B_i(t) c_{i+1}, \quad t \in [\tau_i, \tau_{i+1}], \quad i = 0, \dots, N-2, \quad (30)$$

$$v_2(t) = A_i(t) c_{N+i} + B_i(t) c_{N+i+1}, \quad t \in [\tau_i, \tau_{i+1}], \quad i = 0, \dots, N-2, \quad (31)$$

and

$$v_3(t) = A_i(t) c_{2N+i} + B_i(t) c_{2N+i+1}, \quad t \in [\tau_i, \tau_{i+1}], \quad i = 0, \dots, N-2, \quad (32)$$

where

$$A_i(t) = \frac{\tau_{i+1} - t}{\Delta_c}, \quad B_i(t) = \frac{t - \tau_i}{\Delta_c}, \quad i = 0, \dots, N-2. \quad (33)$$

Define the following functions

$$G_O(z, e) = [\max(z + e, 0)]^2, \quad e > 0, \quad (34)$$

$$G(z, e) = [\min(z + e, 0) + \max(z - e, 0)]^2, \quad e > 0, \quad (35)$$

$$G_{12}(z, e_1, e_2) = [\min(z - e_1, 0) + \max(z - e_2, 0)]^2, \quad e_1 < e_2, \quad (36)$$

and

$$\begin{aligned} J(\mathbf{c}) &= G(x_D - x_{EF}(t_f), \epsilon_{EF}) + G(z_D - z_{EF}(t_f), \epsilon_{EF}) \\ &+ \int_0^{t_f} [G_{12}(z_{EF}(t) - (ax_{EF}(t) + c), -\epsilon_1, -\epsilon_2)] dt \\ &+ \int_0^{t_f} \sum_{k=1}^2 G_O(z_k(t) - (ax_k(t) + c) + \epsilon_2, 0) dt. \end{aligned} \quad (37)$$

The functional $J(\mathbf{c})$ is a sum of penalty functions , incorporating the state constraints , and the required goals . An element $\mathbf{c}^o = (c_0^o, \dots, c_{3N-1}^o) \in \mathbb{R}^{3N}$ for which $J(\mathbf{c}^o) = 0$ will be called here a *feasible command vector* , and the control vector $\mathbf{v}^o$ induced by $\mathbf{c}^o$ via (30) - (32) will be called here a *feasible command strategy* .

Thus , once a feasible command strategy $\mathbf{v}^o(t)$ is applied to Eqs. (29) , then , all the specifications and goals of the auxiliary control problem posed in the last section are satisfied . Consequently , using (28) , it follows that

$$\boldsymbol{\tau}^o = \mathbf{M}(\mathbf{q}) \mathbf{v}^o + \mathbf{h}(\mathbf{q}, \mathbf{p}) , \mathbf{q} \in \mathcal{D} .$$

Thus , $\boldsymbol{\tau}^o(t)$ is the vector of the required torques .

The computation of $\mathbf{c}^o$ was conducted by solving an unconstrained minimization problem on $\mathbb{R}^{3N}$. This was done by using a gradient method described in [6] . However , any other gradient method or search method may be applied . At each stage , during the minimization process , the functional $J(\mathbf{c})$ (which is a function of $\mathbf{c}$) was computed by solving Eqs. (29) on $[0, t_f]$. Eqs. (29) (after writing Eqs. (29) as a set of first order ODE's) were solved by using a fourth - order Runge - Kutta method with a time step Δ_t .

The question of the existence of solutions to $J(\mathbf{c}) = 0$ in $\mathbb{R}^{3N}$ is out of the scope of this work . The mapping from $\mathbf{c}$ to $J(\mathbf{c})$ is too complicated for guaranteeing the existence of $\mathbf{c}^o$.

4.1 Example

In the example dealt with here the following set of parameters has been used :

$l_k = 1.$ meter and $l_{Ck} = 0.5$ meter , $k = 1, 2, 3$; $m_1 = 10$ Kgm , $m_2 = 8$ Kgm , $m_3 = 6$ Kgm ; $m_{R1} = m_{R2} = 2$ Kgm , $m_{R3} = 1.5$ Kgm , $I_k = (1/12) m_k l_k^2$, $k = 1, 2, 3$. It can be shown , by caculating the values of $\det \mathbf{M}(\mathbf{q})$, that , for the above - mentioned values of parameters , $\det \mathbf{M}(\mathbf{q}) > 87.111$ for all $-\pi < \theta_k \leq \pi$, $k = 1, 2, 3$. The rest of the parameters are given below :

$t_f = 4.5$ sec , $\Delta_t = 4.5/1800$, $N = 10$, $\Delta_c = 0.5$ sec . The values for the plane $\mathcal{B}$ are given by $a = -1$ and $c = 1.845$. In order to find feasible values for $\theta_k(0)$, $k = 1, 2, 3$ (that is , to insure $\mathbf{q}(0) \in \mathcal{D}$) , the following choice has been made

$\theta_1(0) = c_{30}$, $\theta_2(0) = c_{31}$ and $\theta_3(0) = c_{32}$. In addition we choose $\dot{\theta}_k(0) = 0$, $k = 1, 2, 3$. By making this choice for $\{\theta_k(0)\}_{k=1}^3$ the vector $\mathbf{c}$ is given now by $\mathbf{c} = (c_0, c_1, \dots, c_{29}, c_{30}, c_{31}, c_{32})$ and the optimization of $J(\mathbf{c})$ is performed now on $\mathfrak{R}^{33}$.

In addition we choose $\mathbf{r}_B = -1.01042\mathbf{I} + 2.80053\mathbf{K}$, $\epsilon_2 = 10^{-4}$, $\epsilon_1 = 0.06$, $\epsilon_{EF} = 10^{-6}$.

Computation were carried on until the functional $J(\mathbf{c})$ reached the value of zero . Some of the results are shown in Figs. 2 - 12 . In addition , the results showed that $x_{EF}(0) = 1.56836120$ and $z_{EF}(0) = 0.2758999$. Thus , using these values we have $\mathbf{r}_A = x_{EF}(0)\mathbf{I} + z_{EF}(0)\mathbf{K} \in \mathcal{D}$.

5 References

1. L.E. Elsgolc , Calculus of Variations , Pergamon Press , Oxford , UK , 1963 .
2. M.R. Hestenes , Variational theory and optimal control theory , in Computing Methods in Optimization Problems , Edited by A.V. Balakrishnan and L.W. Neustadt , Academic Press , New York , 1964 .
3. E.T. Whittaker , A Treatise on the Analytical Dynamics of Particles and Rigid bodies , Cambridge University Press, Cambridge, UK, 1917 .
4. M.W. Spong and M. Vidyasagar , Robot Dynamics and Control , John Wiley & Sons , New York , 1989 .
5. Y. Yavin and C. Frangos , On a horizontal version of the inverse pendulum problem , Computer Methods in Applied Mechanics and Engineering , Vol. 141 , pp. 297 - 309 , 1997 .
6. Y. Yavin and C. Frangos , Computation of feasible control trajectories for the navigation of a big ship around an obstacle in the presence of a sea current , Math. Comput. Model. , Vol. 21 , pp. 99 - 117 , 1995 .

7. H. Nijmeijer and A.J. van der Schaft , Nonlinear Dynamical Control Systems ,
Springer - Verlag , New York , 1990 .

List of Captions

Fig. 1 : The three - link planar manipulator and the set $\mathcal{B}$.

Fig. 2 : $z_{EF}(t)$ as function of $x_{EF}(t)$, $t \in [0, t_f]$.

Fig. 3 : $D_{EF}(t) = z_{EF}(t) - (a x_{EF}(t) + c)$, $t \in [0, t_f]$. Here $\max_{t \in [0, t_f]} D_{EF}(t) = -2.64 \cdot 10^{-4}$.

Fig. 4 : $D_{R1}(t) = z_1(t) - (a x_1(t) + c)$, $t \in [0, t_f]$.

Fig. 5 : $D_{R2}(t) = z_2(t) - (a x_2(t) + c)$, $t \in [0, t_f]$. Here $\max_{t \in [0, t_f]} D_{R2}(t) = -7.93 \cdot 10^{-4}$.

Fig. 6 : $v_{EF}(t) = \|\mathbf{v}_{EF}(t)\|$, $t \in [0, t_f]$.

Fig. 7 : $v_1(t)$, $t \in [0, t_f]$.

Fig. 8 : $v_2(t)$, $t \in [0, t_f]$.

Fig. 9 : $v_3(t)$, $t \in [0, t_f]$.

Fig. 10 : $\tau_1(t)$, $t \in [0, t_f]$.

Fig. 11 : $\tau_2(t)$, $t \in [0, t_f]$.

Fig. 12 : $\tau_3(t)$, $t \in [0, t_f]$.

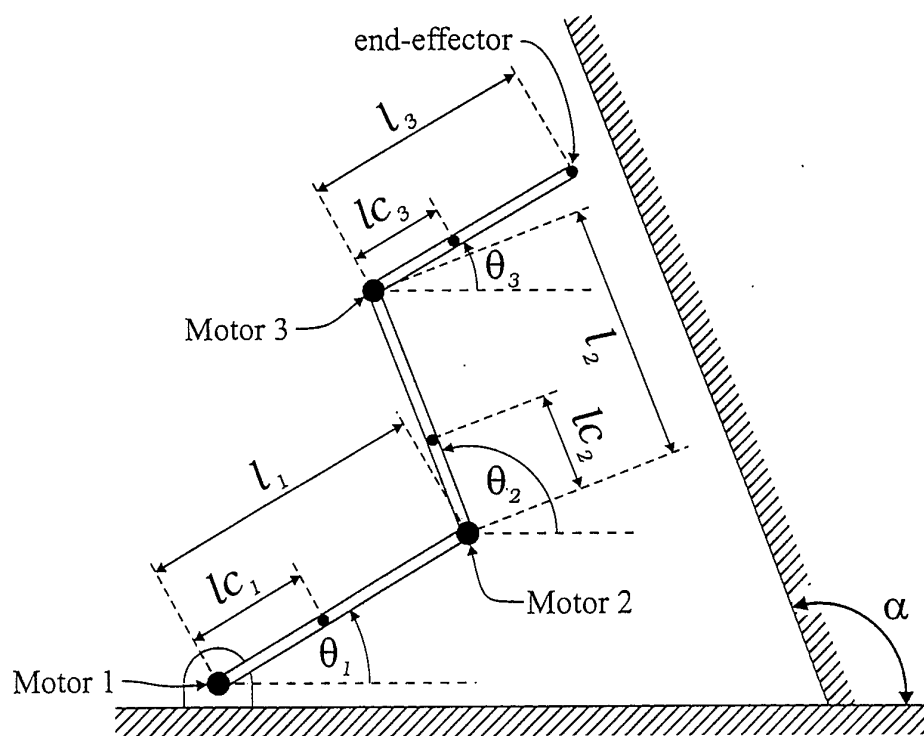


Figure 1

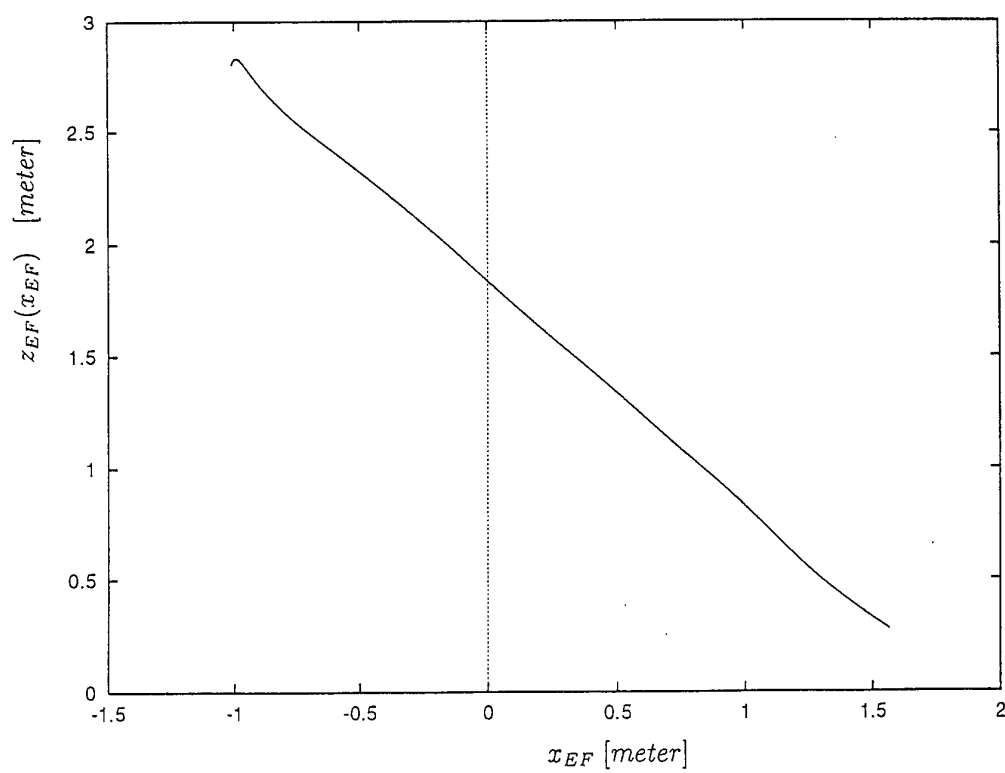


Figure 2

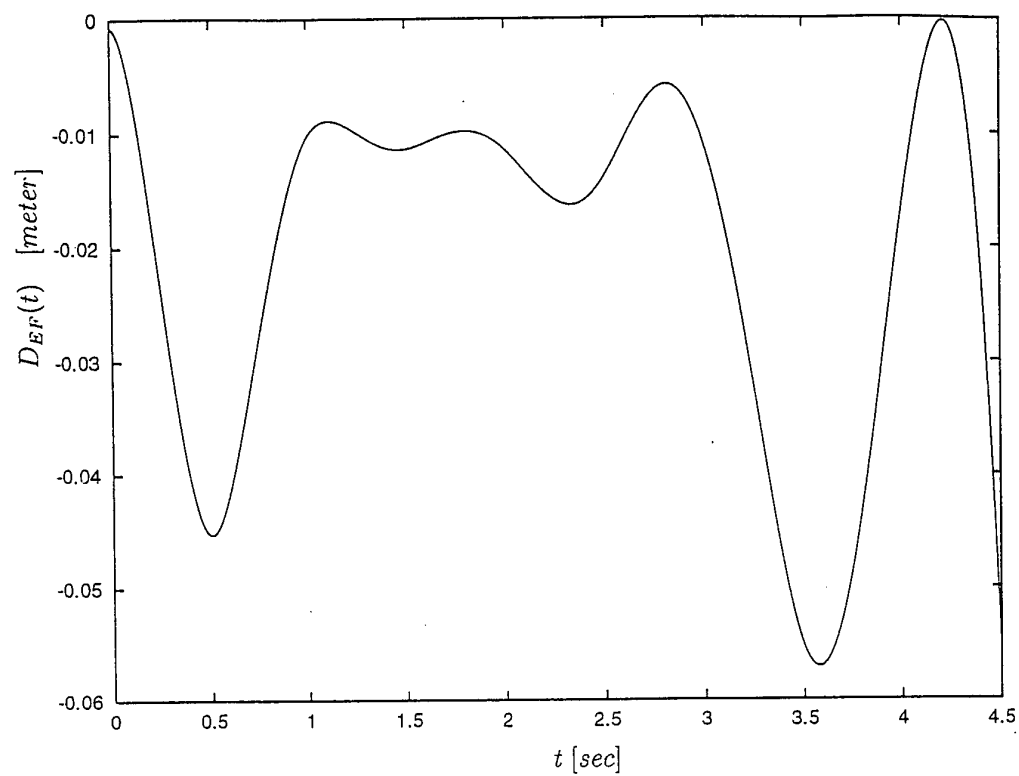


Figure 3

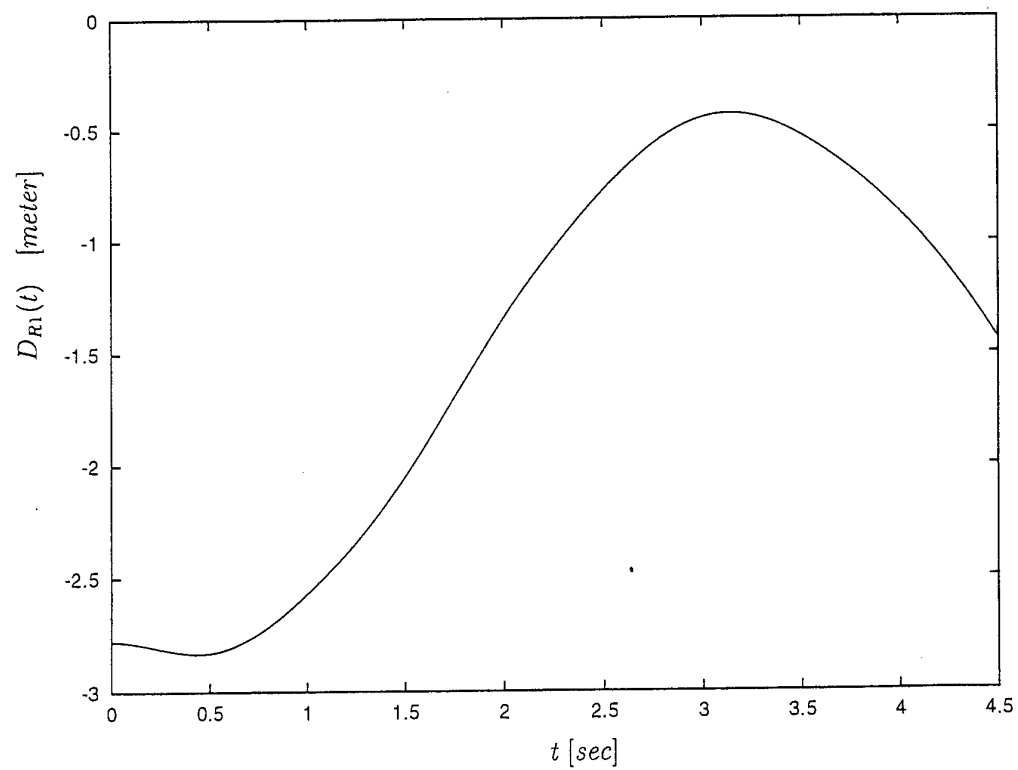


Figure 4

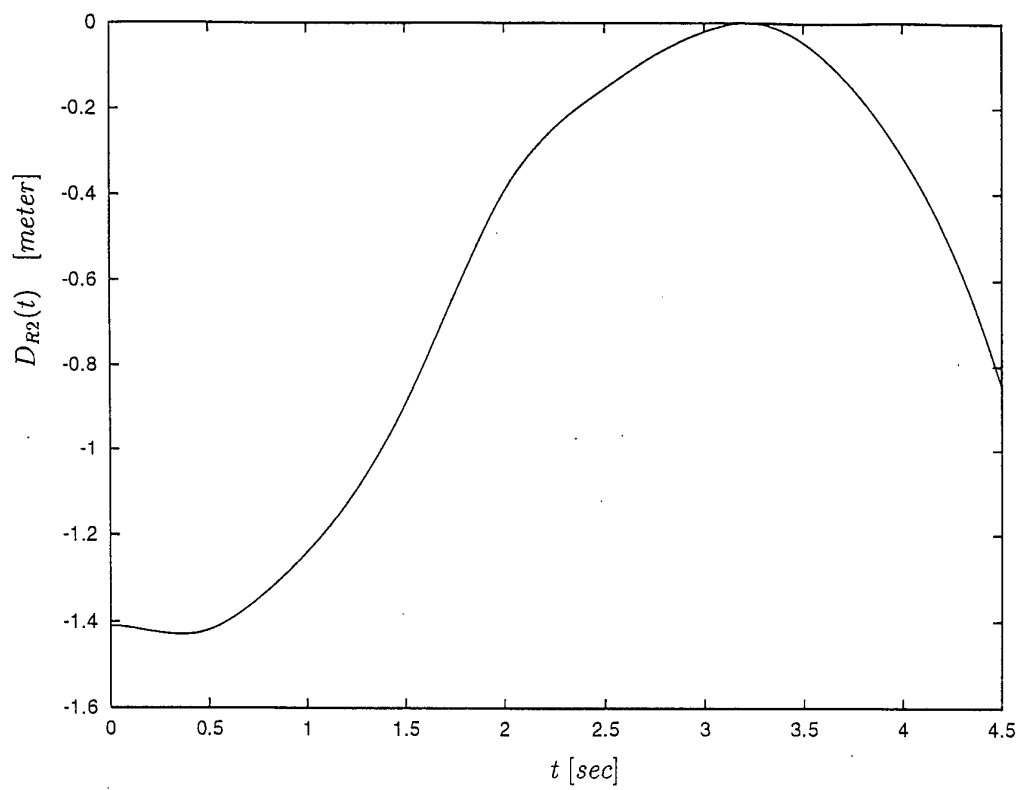


Figure 5

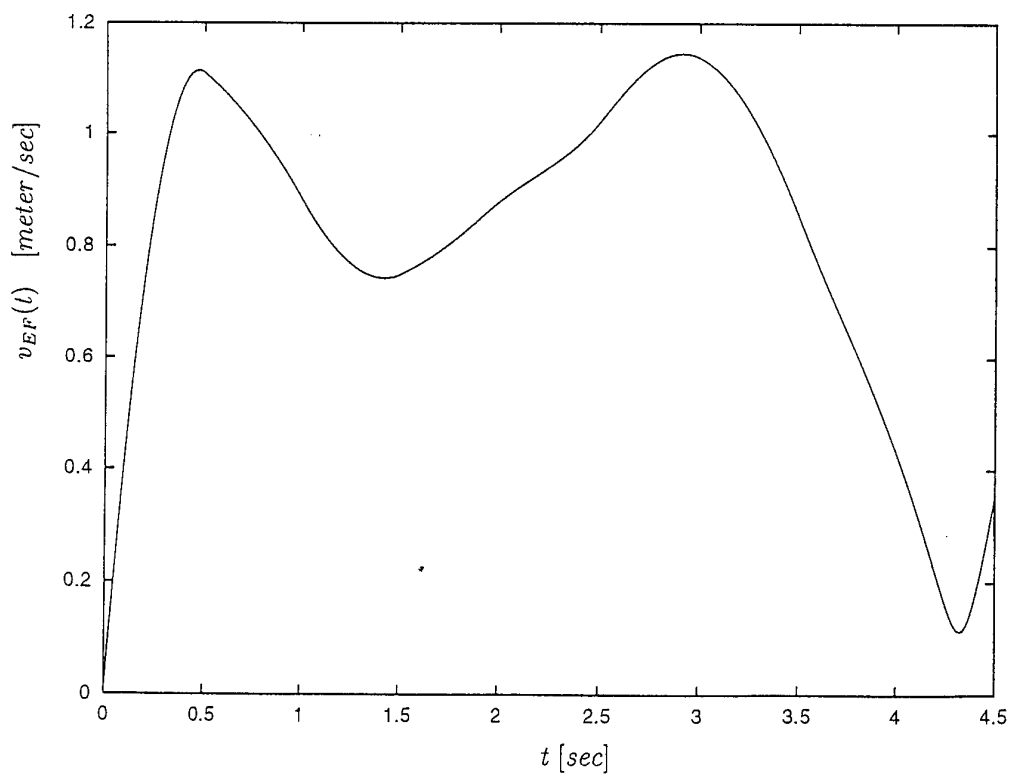


Figure 6

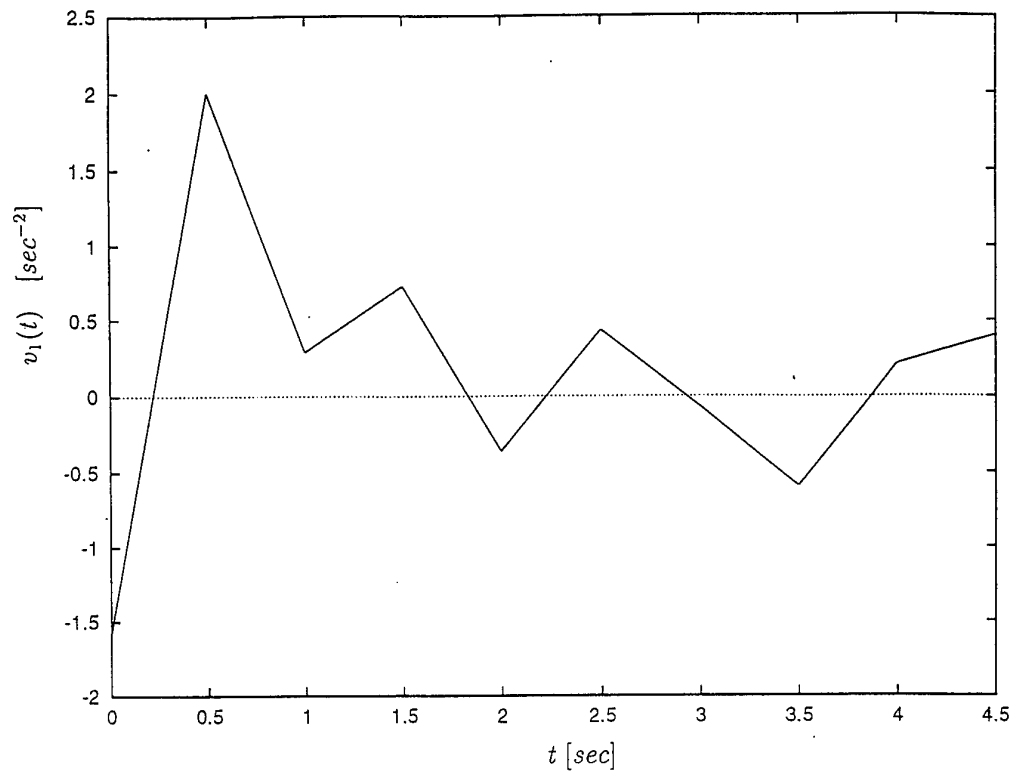


Figure 7

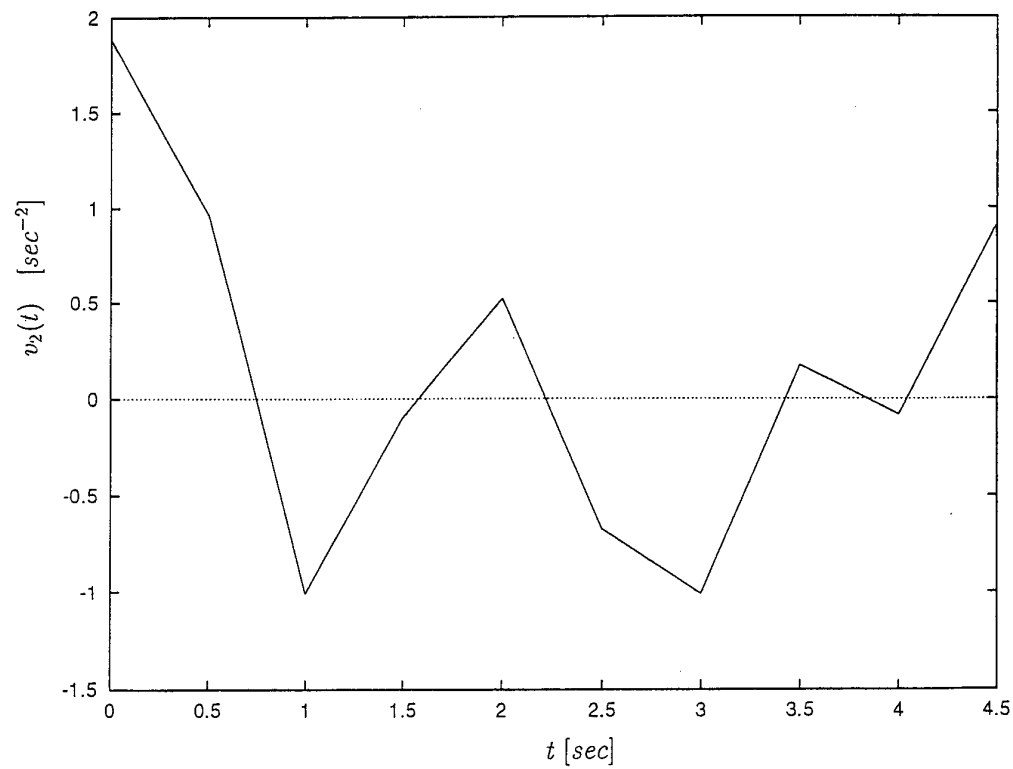


Figure 8

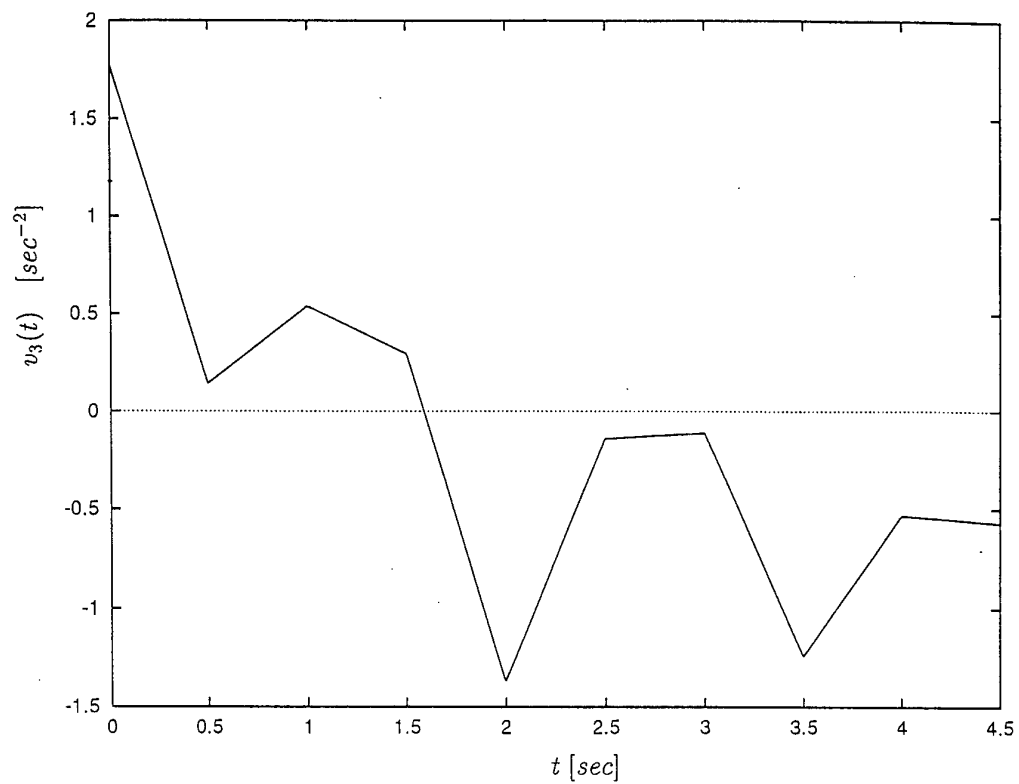


Figure 9

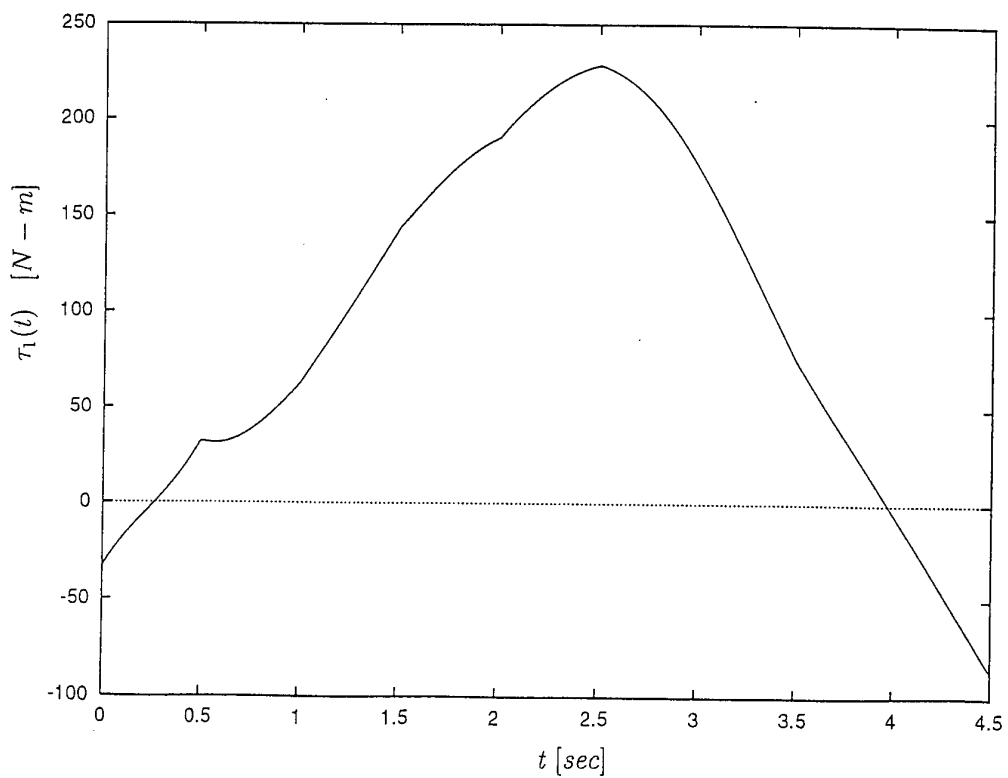


Figure 10

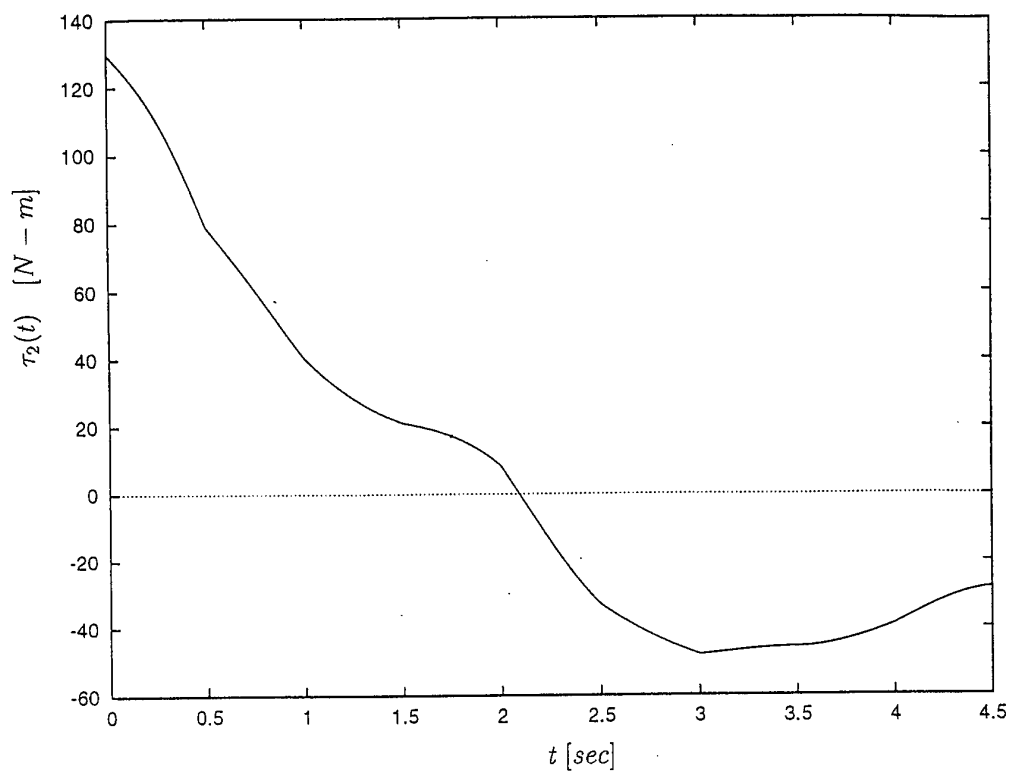


Figure 11

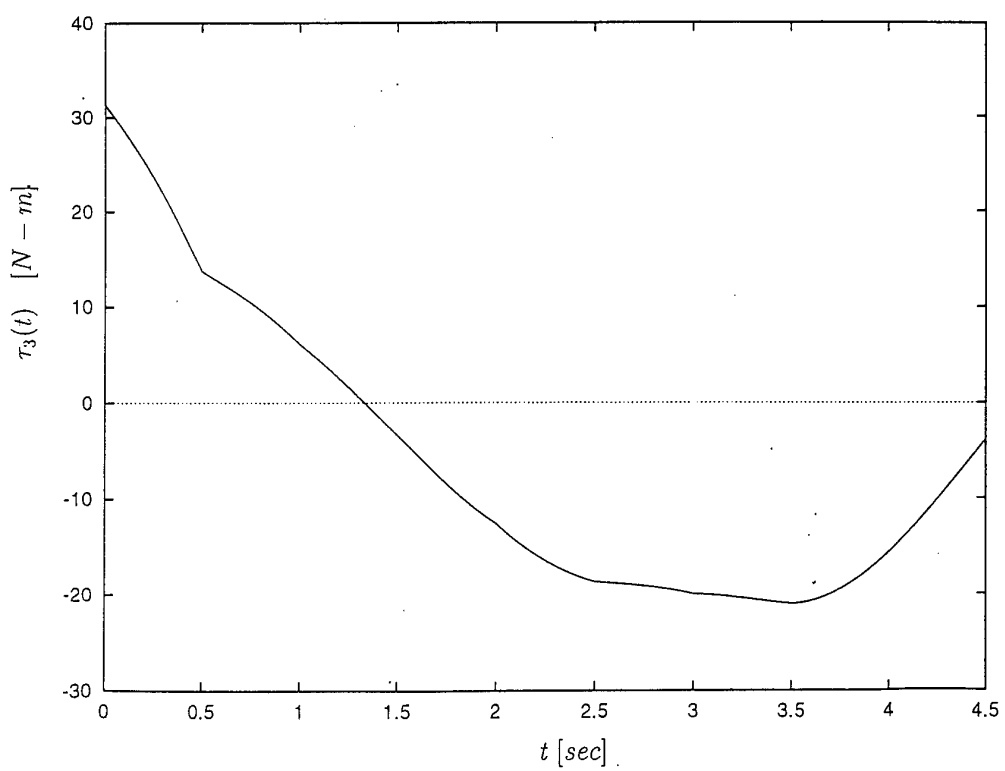


Figure 12

ROTOR DYNAMIC ANALYSIS IN THE DESIGN OF ROTATING MACHINERY

G. Genta, E. Brusa

Mechanics Department, Politecnico di Torino, Torino, Italy

ABSTRACT

Rotordynamic analysis is an important step in the design of any rotating machine. To go beyond the very simple models yielding a good qualitative insight but cannot predict the details of the dynamic behaviour of rotors, it is necessary to resort to numerical methods and among them the Finite Element Method is without doubt the most suited for implementation in the context of computer aided engineering. Instead of resorting to general purpose codes, the particular characteristics of rotordynamic analysis make it expedient to use specialised tools like DYNROT, a FEM code that allows to perform a complete study of the dynamic behaviour of rotors. Although initially designed to solve the basic linear rotordynamic problems (Campbell diagram for damped or undamped systems, unbalance response, critical speeds, static loading), it has been extended to the study of nonstationary motions of nonlinear rotating systems [1] and the torsional and axial analysis of rotors and reciprocating machines. Its distinctive features of resorting to Guyan reduction and of extensively using complex co-ordinates both for isotropic and non symmetric systems, allow to reduce the computer time and to perform a large number of computations at a reasonable cost. The code can thus be used as a routine to be called by optimisation procedures aimed at including rotordynamic performances into the definition of an optimum design of the machine.

INTRODUCTION

Rotordynamic considerations play an important role in the design of a number of rotating machine elements, particularly when rotational speeds are high. Dynamic analysis, in the past mostly aimed only to the computation of the critical speeds, but currently in many instances used to obtain a whole picture of the rotordynamic behaviour of the machine (e.g., the computation of the Campbell diagram and the unbalance response), must accompany stress analysis and all the other computations related to the working conditions of the machines (e.g. fluid dynamic study for turbomachinery, electrical analysis for generators and motors, etc.).

There are cases in which rotordynamic analysis deeply influences the design of the rotating parts of the machine: an example is the case of composite material transmission shafts for vehicular applications in which the diameter of the shaft and the orientation of the reinforcing fibres are determined by the need of avoiding the presence of a critical speed within the working range [2], or of high speed turbomolecular pumps on magnetic bearing in which the need of rising the third critical speed, the one related to the deformations of the rotor, dictates many design details. In these cases, if an optimisation of the design is attempted, rotordynamic analysis enters deeply the optimisation process.

Traditionally, rotordynamic analysis of complex machines was performed either using much simplified models or through procedures similar to Myklestat-Prohl method, based on the transfer matrices approach. The finite element method (FEM) is presently gaining popularity also in the field of rotordynamics, mainly for its ability of modelling intricate geometry in a simple way, at least from the point of view of the user, and its possibility of writing general purpose codes. However, although the use of standard commercial codes for structural dynamic analysis also for the dynamic study of rotors is sometimes possible, it compels to resort to some sort of "tricks" to take into account the effects of the rotation of the system, which can have a strong influence on its flexural vibration, affecting the natural frequencies and coupling the flexural motions in such a way that it is more correct to speak of whirling motion than of vibrations.

To take into account this instance, it is possible to force a gyroscopic matrix, which affects mainly the mass matrix of the system, into any code devised for dynamic analysis, but this is not a handy procedure and it has the disadvantage of allowing only the study of synchronous whirling. It is then possible only the computation of the critical speeds and the unbalance response, but not that of the Campbell diagram, i.e. the plot of the whirl natural frequencies against the spin speed, which is the basic tool for understanding the dynamic behaviour of any rotor. Even more problematic is the study of the behaviour of the accelerating rotor. Only a purposely written code, which takes correctly into account the presence of the gyroscopic matrix and of the circulatory matrix linked with rotating damping and perhaps of the centrifugal stiffening effect, due to the presence of bladed disks, can fulfil adequately the task.

Starting from the end of the seventies, the development of a FEM code specifically suited for rotordynamic computations was undertaken at the Department of Mechanics of Politecnico di Torino. It has been evolved in various versions through more than twenty years [3]. The original code was written using HPL and then HP-BASIC language for desktop HP 9800 computers but the present versions are based on the MATLAB (MATLAB is a trademark of The MathWorks, Inc.) interactive software package and can be used on any PC, workstation or mainframe system.

THEORETICAL BACKGROUND

Usually rotors are modelled as beam-like structures and some form of beam theory, with either the Euler-Bernoulli or the Timoshenko approach [4, 5] is used. Numerical solutions based on some form of the Myklestad-Prohl transfer matrix method [6, 7] were very widespread. Also most of the procedures based on the FEM consider the structural parts as composed by beam elements [8, 9]; although models based on other types of elements can be found the literature [10]. Also for the study of the torsional behaviour of shafts the structural parts are mainly modelled as beams or even simple torsional springs.

Under the above mentioned assumptions, if the beams which model the rotor are straight, their axes are all aligned with the spin axis and if the centre of gravity and the shear centre of all the cross sections lay on the spin axis, the axial, flexural and torsional behaviours are uncoupled. The presence of a small static or couple unbalance does not modify substantially this feature. Clearly the assumptions leading to uncoupling do not hold in the case of crankshafts, but in the study of the torsional vibrations of reciprocating machines it is customary to resort to a so-called equivalent model which is essentially a straight beam-like structure and, as a consequence, the uncoupling is restored [11].

The further assumptions of linearity, small unbalance and small displacements allow to obtain a linear equation of motion; however, even in the case of the discretised model of a linear rotor which is axially symmetrical about its spin axis and rotates at a constant spin speed ω , the linearised equation of motion is of the type

$$\mathbf{M}\ddot{\mathbf{x}} + (\mathbf{C} + \mathbf{G})\dot{\mathbf{x}} + (\mathbf{K} + \mathbf{H})\mathbf{x} = \mathbf{f}(t), \quad (1)$$

where $\mathbf{x}$ is a vector containing the generalised co-ordinates, referred to an inertial frame, $\mathbf{M}$ is the symmetric mass matrix, $\mathbf{C}$ is the symmetric damping matrix, $\mathbf{G}$ is the skew-symmetric gyroscopic matrix (it is usually linearly dependent on the spin speed ω , $\mathbf{K}$ is the symmetric stiffness matrix (it may contain a part which is proportional to ω^2), $\mathbf{H}$ is the skew-symmetric circulatory matrix, usually proportional to ω and $\mathbf{f}$ is a time-dependant vector in which all forcing functions are listed. One of these forcing functions is usually that due to the residual unbalance which, although small, cannot be neglected. Unbalance forces are harmonic functions of time, with amplitude proportional to ω^2 and frequency equal to ω . Equation (1) is that of a non-natural, circulatory system and hence differs from the typical equations encountered in dynamics of structures, where all matrices are

symmetric. It must be noted that, when ω tends to zero, the skew-symmetric terms vanish and the rotor reduces to a structure.

Equation has been obtained assuming that the rotor is axially symmetrical, but it runs on a stator, which can be without any particular symmetry properties. If, on the contrary, the rotor cannot be considered axially symmetrical, the study becomes very complicated, unless an axial symmetry assumption can be made on the nonrotating parts of the system. In the latter case, a rotor-fixed reference frame, i.e. one that rotates at the angular velocity of the rotor, can be used and an equation similar to equation (1), although written with reference to a non-inertial frame is obtained. If both stator and rotor are non-isotropic with respect to the rotation axis, the equation of motion which models its behaviour has coefficients which are periodic in time, with a frequency equal to 2ω . No closed form solution of such equation is available and even to reach approximated solution is far more complicated than in the case in which either the rotor or the stator is axially symmetrical.

When the flexural behaviour can be uncoupled from the axial and torsional ones, equation (1) holds for the first one, while the torsional and axial equations of motion are usually those of a natural, non-circulatory system.

If both stator and rotor are isotropic with respect to the rotation axis, the best choice for what the generalised co-ordinates for the study of the flexural behaviour are concerned is the use of complex co-ordinates. As the structure is assumed to be beam-like, each node has four real degrees of freedom, namely two lateral displacements and two rotations. Assuming that the axis of rotation of the system coincides with the z -axis of an orthogonal reference system xyz and using the assumptions which are customary in the beam theory, the flexural displacement of any point of the rotor axis and the rotation of the cross sections can be expressed by the complex displacement

$$\begin{cases} z = x + iy \\ \varphi = \varphi_y - i\varphi_x \end{cases} \quad (2)$$

When using complex co-ordinates for a system, which is axially symmetrical, Equation (1) yields:

$$\mathbf{M}\ddot{\mathbf{q}} + (\mathbf{C} + i\mathbf{G})\dot{\mathbf{q}} + (\mathbf{K} + i\mathbf{H})\mathbf{q} = \mathbf{f}(t), \quad (3)$$

where $\mathbf{q}$ is the vector containing the complex generalised co-ordinates. Note that when using complex co-ordinates all matrices are symmetrical: those, which in the standard approach are skew-symmetric, in the present case, are imaginary.

It is possible to demonstrate that the complex-coordinate approach is suitable also when the rotor or the stator lack of axial symmetry [12]. By releasing also the linearity and constant speed assumptions, the general equation of motion of a nonlinear rotor which is performing acceleration with a stated law $\omega(t)$ is [13]:

$$\mathbf{M}_m \ddot{\mathbf{q}} + (\mathbf{C}_m - i\omega\mathbf{G})\dot{\mathbf{q}} + (\mathbf{K}_m + \mathbf{K}_\omega\omega^2 - i\omega\mathbf{C}_r)\mathbf{q} + \mathbf{M}_{nd} \ddot{\mathbf{q}} + \mathbf{M}_{rd} e^{2i\theta} \ddot{\mathbf{q}} + \mathbf{C}_{nd} \dot{\mathbf{q}} + (\mathbf{C}_{rd} + 2i\omega\mathbf{M}_{rd})e^{2i\theta} \dot{\mathbf{q}} + \mathbf{K}_{nd} \bar{\mathbf{q}} + (\mathbf{K}_{rd} - i\omega\mathbf{C}_{rd})e^{2i\theta} \bar{\mathbf{q}} + g(q_i, \dot{q}_i, \mathcal{A}(t)) = (\omega^2 - ia)\mathbf{F}_r e^{i\theta} + \mathbf{F}_n, \quad (4)$$

where:

- $\mathbf{M}$, $\mathbf{G}$, $\mathbf{C}$, $\mathbf{K}$ and $\mathbf{K}_\omega$ are respectively the mass, gyroscopic, viscous damping stiffness and centrifugal stiffening matrices. They are all symmetrical and have been written in such a way to evidence the dependence of the various matrices (or parts of matrices) from the spin speed ω ;
- subscripts m , d , r and n refer respectively to the mean and deviatoric matrices (for their definition, see [11]) and to the matrices referred to the rotating and the non-rotating parts of the machine. Mean matrices without either subscript r or n refer to the whole model;
- $\mathbf{q}$ and $\mathbf{F}$ are respectively the vector of the complex generalised co-ordinates and that of the nodal forces. In this case subscript r designates forces due to unbalance and subscript n the non rotating forces as rotor weight. All forces can be prescribed functions of time;

- the generic vector function $g(q_i, \dot{q}_i, \mathcal{G}(t))$ is introduced to take into account the behaviour of the nonlinear part of the system;
- $\mathcal{G}$, $\omega = \dot{\mathcal{G}}$ and $a = \dot{\omega}$ are respectively the rotation about the spin axis, the spin speed and the angular acceleration. They are all known functions of time and are the same for the whole rotor; however with simple modifications to equation (4) to study the behaviour of multi-shaft systems, the model can be subdivided into different substructures each one having a different angular velocity and hence different laws $\mathcal{G}(t)$, $\omega(t)$ and $a(t)$.

The assumption that the angular acceleration is the same at all sections comes directly from the uncoupling between the torsional and the flexural behaviour and is usually referred to as torsionally stiff rotor assumption. However, an equation allowing to link the rotational degree of freedom of the system with the driving torque can be easily obtained. It is a single scalar equation, owing to the assumption of torsionally rigid rotor, except in the case of multi-shaft systems

$$M_z = \Im(\bar{\mathbf{F}}^T \ddot{\mathbf{q}} e^{i\theta}) + J_p, \quad (5)$$

where the total polar moment of inertia of the rotor can take also into account the effect of the different unbalances which are at any rate very small.

Equation (4) is the basic formulation for the study of the flexural behaviour of the system. Clearly its complete formulation cannot be solved in closed form and the only possible approach is the numerical integration in the time domain.

When the forcing functions are periodic, the system is linear, the spin speed ω is constant and either the stator or the rotor (or both) can be considered as axially-symmetrical bodies with respect to the spin axis, general solutions in the frequency domain can be obtained in closed form.

By assuming that a solution of the type of $\mathbf{q} = \mathbf{q}_0 e^{i\lambda t}$ or $\mathbf{q} = \mathbf{q}_0 e^{i\omega t}$ respectively for free whirling and synchronous forced whirling (i.e. a forced whirling due to a forcing function whose frequency is equal to the spin speed, as in the case of unbalance), simple frequency domain equations allowing to solve the typical problems of rotordynamics are readily obtained. The critical speeds can thus be computed, of the Campbell diagram, the stability maps and the roots loci can thus be obtained through an eigenanalysis, and the steady state response to unbalance or other periodic forcing functions can be obtained.

While in frequency-domain equations damping can be modelled using both the viscous and hysteretic models, when solving equation (4) in the time domain only viscous damping can be introduced. Viscous damping can be introduced in form of damper elements and hysteretic damping by stating the loss factor of the materials. When hysteretic damping can be used directly, the equation of motion in the frequency domain can be accordingly modified [11].

When on the contrary this is impossible the user has two alternatives: neglect hysteretic damping altogether or resorting to a form of equivalent viscous damping. A way to compute the latter is that of performing the modal transformation based on the modes of the linearised, natural, isotropic system (i.e. based only on the mass and stiffness mean matrices) of all hysteretic damping matrices and reducing them to their generalised proportional component [11] by cancelling all elements outside the main diagonal. The equivalent viscous damping matrices can then be obtained by dividing the elements on the main diagonal by the corresponding natural frequency and then performing the inverse modal transformation. This procedure is clearly approximated, but, if the system is lightly damped, leads to acceptable results. A better procedure, which takes into account also the non-natural nature of the system is being developed and will be published soon. There is no theoretical way to assess the precision of such approximated procedures, but a check can be done by

comparing the steady-state unbalance responses of the models with hysteretic and with viscous equivalent damping. Note that the equivalent damping so computed is just a mathematical approximation and the structure of the equivalent damping matrices has nothing to do with that of the starting hysteretic matrices, nor it retains any physical meaning.

In the study of the unbalance response, both at constant speed or during an acceleration, equation (5) may be used to compute the driving torque needed to maintain the spin speed constant or to follow the stated law $\omega(t)$, after the lateral behaviour of the system has been obtained.

The main advantages of the use of complex co-ordinates can be summarised as follows:

- in the case of isotropic systems, the equations are all real and their number is halved with respect to the approach based on real co-ordinates. A substantial reduction of computation time is so achieved;
- when the system is not isotropic, elliptical or polyharmonic whirling is expressed as the sum of forward and backward circular whirling components. When more complicated whirl patterns exist, the various harmonic components are well separated and this allows to obtain a good insight of the motion of the system in a simpler way;
- the model is built in terms of mean and deviatoric properties. Different solutions can then be obtained one after the other: by neglecting the deviatoric matrices the dynamic behaviour of a mean system, i.e. an equivalent symmetric system which retains many of the features of the original one, is obtained. Later more realistic simulations which take into account deviatoric matrices can be obtained without the need of building new models. The possibility of refining the solution without the need of re-modelling the system is a very interesting feature of this approach.

The study of the dynamic behaviour of reciprocating machines is performed using the equivalent system approach described in detail in [11]. The so-called inertia torques are directly introduced into the model, together with the forcing functions defined by the user. Also here hysteretic damping can be directly introduced into the model only in some of cases. When this is impossible, an equivalent viscous damping matrix can be computed following the lines seen for flexural behaviour.

DYNROT CODE

The code is implemented in the form of function and script M-files, all written in MATLAB language. However, user needs only a basic knowledge of the language to run DYNROT.

The user may define the model either through an interactive input routine or by preparing a data m-file, but the most interesting feature, particularly when performing an optimisation, is the possibility of using the code in parametric form. All characteristics of the elements can be introduced in form of parameters, which can be defined in the calling instruction of an external code, of which DYNROT can be thought as a subroutine. There is no limit to the number of parameters, which can be used, allowing to prepare a general model for a class of rotating machines more than for a particular rotating system.

For simple non-parametric computations the user can choose to run the code in an interactive way, but in general it is possible to prepare a driver file to run the code in a batch way or, as already stated, to call DYNROT from a code which defines the parameters and performs the optimisation.

As an example, if a genetic algorithm is used to optimise the shape of a rotor and some rotordynamic feature is included into the fitness function, the code in which the genetic algorithm is implemented can simply call DYNROT passing to it all the parameters required to define the rotor (which are defined by the genome of the particular individual machine whose fitness function must

be computed) and getting back the relevant rotordynamic features. The ability of DYNROT of performing the required analysis in a very short time is an essential feature in the context of genetic algorithms, as the number of individual configurations studied can be very high.

Apart from the results related to the dynamic analysis, while building the model, the inertial properties (mass, moments of inertia and co-ordinates of the centre of mass) of the whole model and of the various substructures are computed and a drawing of the model can be supplied.

In the present version of the code time-domain numerical integration is performed using 4-th order Runge-Kutta adaptive algorithm and nonlinear algebraic equations are solved using Newton-Raphson algorithm. Equations with time-depending coefficients are solved by resorting to an approach similar to Hill's infinite determinant, obviously truncated at a certain harmonic at the choice of the user, while linear sets of equations and eigenproblems are solved using the facilities which are standard in MATLAB package. In all cases, when complex arithmetic is needed, the ability of MATLAB to deal with complex quantities is exploited, the only exception being equation in which both a complex unknown and its conjugate are present which need to be treated by separation of the real from the imaginary part.

The element library of DYNROT code is particularly tailored for rotordynamics applications; in the present version 23 mechanical elements and 2 control elements are included. All beam elements are based on a formulation of the type usually referred to as "simple Timoshenko beam" with consistent formulation for mass and gyroscopic matrices. Spring and damper elements can connect two nodes of the structure (e.g. to simulate joints, bearings or dampers between two different shafts or between a shaft and the stator) or one node to a fixed point (e.g. to simulate an elastic support). If two nodes are present their co-ordinates must be the same. Also a cubic spring element and a linear spring with clearance, useful to model respectively ball bearings or elastomeric springs and roller bearings, are present.

Bladed discs can be modelled using six different elements for discs and rows of blades, including different types of transition elements to connect discs with shafts, modelled as beams. They are basically annular elements in which the displacement field is approximated by algebraic polynomials in radial direction and trigonometrical polynomials along the angle. As only the first two terms of the polynomial affect the dynamics of the rotor as a whole (the following ones affect the local dynamics of the bladed disc), the polynomials are truncated after two terms [14], [15].

Hydrodynamic bearings are modelled using the conventional 8-coefficient model, with either constant or speed-dependent parameters, following the short, fully cavitated, bearing assumption [11] or by entering the coefficients as functions of the Sommerfeld number. All tables reported in [16] are included, the user may enter other types of bearing.

The magnetic bearing elements can be connected at both ends at two nodes of the structure for the electromagnetic actuator and at two nodes for the sensor or at one node only, the other end being fixed, i.e. connected with the ground. The characteristics of the bearing can be isotropic in xy plane or anisotropic, which allows to study the cases of bearings which are geometrically and electrically isotropic but work in different conditions in xz and yz planes due to the presence of static forces. The element includes the actuators, two electromagnets in x and y directions, which supply a force proportional to the square of the current i_c and inversely proportional to the square of the radial displacement u :

$$F = -k \left(\frac{i_c}{u} \right)^2 \quad (6)$$

and two displacement sensors measuring the displacements in the same directions. The sensors may be located in nodes different from those in which the actuators are located (non-colocated bearings). A bias current can be given to linearize the characteristics of the actuators.

As active magnetic bearings need a suitable control system, two controller elements are included. The first is a PID controller on error feedback with an additional filter on sensor output. Its frequency-domain transfer function takes into account the pole necessary to make the derivative filter feasible

$$PID(s) = k_c \left(1 + \frac{1}{sT_i} + \frac{T_d s}{N} \right), \quad (7)$$

where k_c is the overall stationary gain, T_i the reset time, T_d the prediction or derivative time and N the ratio between zero and pole of the PD section. The filter on sensor output has the following transfer function

$$F(s) = \frac{1}{s\tau + 1}, \quad (8)$$

where τ is its time constant.

Also a general controller is included. Its characteristics are expressed in the form of the coefficients of the polynomials at numerator and denominator of the transfer function. The general controller transfer function $C(s)$ has the following structure

$$C(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + b_{n-2} s^{n-2} + \dots + b_0}{s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_0}. \quad (9)$$

An element which can be used to model the inertial properties of a crank mechanism for the study of crankshafts is included [17]; it has only one node, as a mass element. The elastic properties of the crank can be introduced in the form of a spring element, having the correct equivalent stiffness, or of a beam element with a suitable equivalent length.

DYNROT code works for most of the computations using directly physical generalised co-ordinates. In particular, the torsional response to a polyharmonic forcing function do not rely on any modal approximation, as the usual procedure which takes into account only the response related to the first or the few first modes can lead to unacceptable approximations. However, there are cases in which the modal approach allows strong reduction in computation time and consequently is worth while considering. In the flexural behaviour two different modal approaches are possible; the first is that of using as transformation matrix the matrix of the eigenvectors of the natural, undamped, linearized, isotropic system while the second uses the right and left eigenvectors of the non-natural system [18]. The modal approach (first of the above mentioned procedures) is compulsorily used for the computation of the acceleration response. To avoid high frequency modes, which would cause problems in the numerical integration procedure, only the modes with natural frequency not higher than the maximum spin speed are considered.

Another case in which the modal approach is used is the study of rotors including magnetic bearings. Here the user can decide the number of modes to be considered, being possible to chose all modes if the approximation due to the modal approach is considered unacceptable.

Guyan reduction is extensively used to reduce the size of the problem. It allows to obtain results, which are very close to the correct ones even with a small number of master degrees of freedom, if their choice is performed correctly. The choice of the degrees of freedom to be considered as slave is mainly a fact of experience and physical insight of the problem; different reduction schemes can be evaluated to obtain best results.

EXAMPLE 1

As a first example, consider a typical problem in rotordynamics: the influence of the bearing stiffness on the critical speeds of a rotor. In the case of a rigid rotor, the problem is easily solved by using a simple model with 4 degrees of freedom, but this approach does not allow to study the effect of modes linked with the deflection of the shaft, i.e. while being adequate for compliant bearings, does not allow to study the effect of very stiff supports. Using DYNROT it is possible to model also the compliance of the rotor and to perform a complete study.

Consider as an example the small gas turbine studied in [11] as Example 4-3 and modelled using 11 nodes and 14 elements (including 2 mass elements and 2 spring elements to model the bearings). The model has a total of 22 degrees of freedom reduced to 8 through Guyan reduction. A plot of the first 4 critical speeds as functions of the stiffness of the bearings (from a minimum of 10^5 to a maximum of 10^{10} N/m) is reported in Fig. 1a. The plot is a pretty much standard result; here the interesting point is the fact that it has been produced in a completely automatic way through a simple MATLAB program of about 25 lines (including all graphic commands) which uses DYNROT code as a subroutine. From the plot is clear that the first two critical speeds and, to a lesser extent, the fourth one, are strongly influenced by the stiffness of the bearings, while the third one is not.

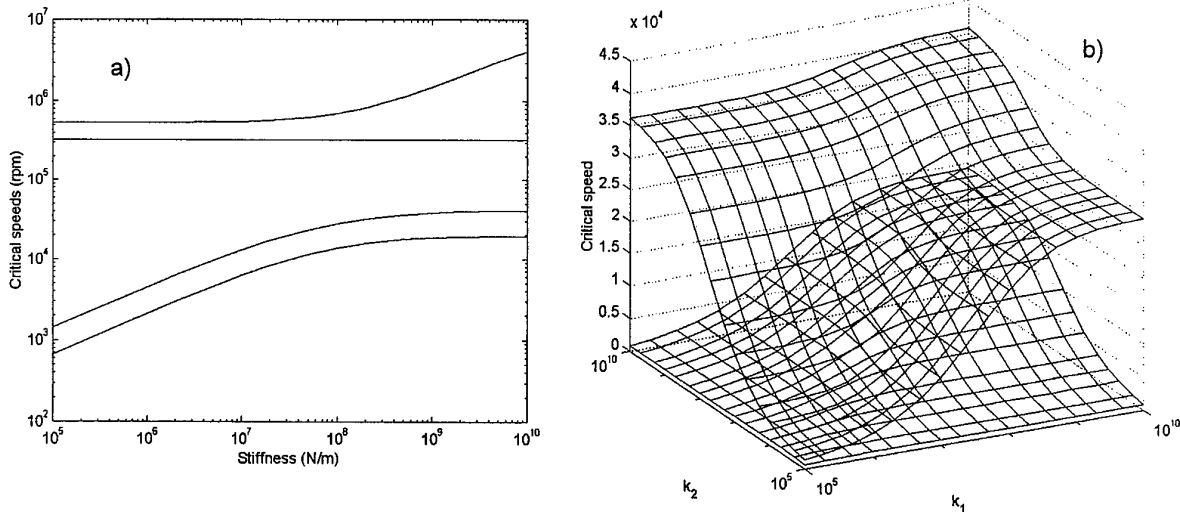


Fig.1. a): First 4 critical speed of the rotor of a small gas turbine as functions of the stiffness of the bearings. b): First 2 critical speeds for the same system of Fig. 1a), but varying independently the stiffness of the two bearings.

Using a slightly more complicated program it is possible to investigate separately the effect of the stiffness of the two bearings. The result shown in Fig. 1b in terms of a tri-dimensional plot of the first two critical speeds as functions of the stiffnesses of the two bearings k_1 and k_2 , has been obtained by analysing 256 automatically generated FEM models and then by summarising the results in a single plot. The total computation time was 94 s on a Pentium 2 desktop computer.

EXAMPLE 2

As a second example consider a canned pump on magnetic bearings. The machine is intended to be used in the subcritical range with respect to the critical speed related to the first deformation mode, while being supercritical with respect to the rigid body modes. To reduce the mass of the rotor the central shaft is bored, but the diameter of the bore is critical in that initially its increase raises slightly the critical speed, owing to a reduction of the mass, while then it reduces it owing to a

reduction of the stiffness. A plot of the mass of the rotor and of the critical speed as functions of the inner diameter can be obtained using a detailed model of the machine (Fig. 2).

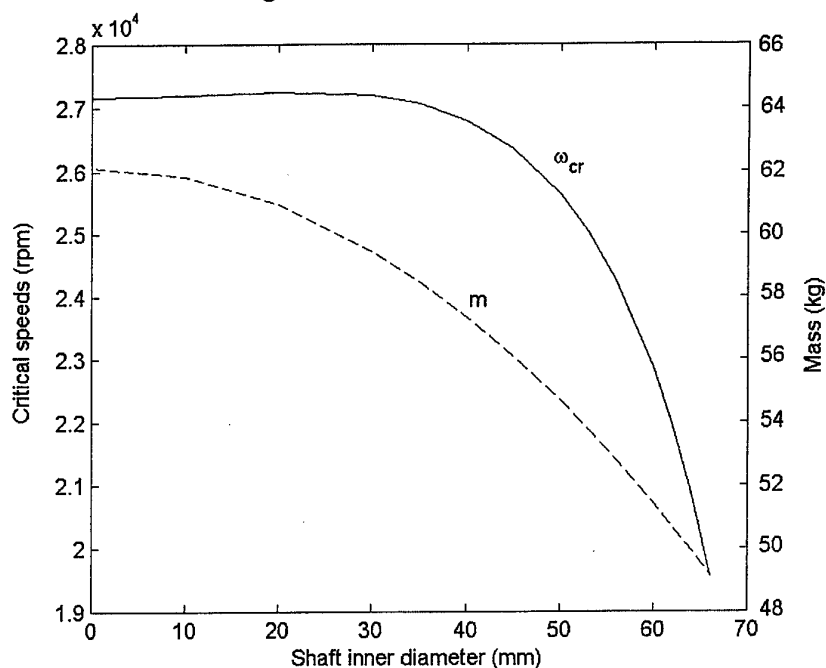


Fig.2. Critical speed related to the deformation of the rotor and mass of the rotating system of a canned pump as functions of the inner diameter of the rotor.

The plot can be useful to reach a compromise between the contrasting requirements of reducing the mass of the rotor and increasing the critical speed.

CONCLUSIONS

Rotordynamic analysis is an essential step in the design of machines containing rotating elements, particularly when high rotational velocities are involved. Current trends in technology towards faster and lighter machines, made using materials with higher performances working at higher stress levels, lead to an ever-increasing need of a detailed knowledge of the dynamic behaviour of the machine as a system. The simple computation of the first critical speed of the rotor is often not enough and design must include a more detailed analysis. Some of the classical assumptions of rotordynamics, like axial symmetry or linearity, have to be dropped, and more complete models are needed.

Finite element codes are flexible and powerful enough to answer the mentioned needs, but standard FEM codes do not include some of the features which are typical of rotordynamics, like gyroscopic effect, rotating damping and centrifugal stiffening. There is a need for purposely-written codes for rotordynamic analysis, which combine the possibility of performing an analysis of the required completeness with low computer time. The last feature is important particularly when the code is used in the design or optimisation process, when a large number of configurations must be considered.

DYNROT code allows to study the lateral, axial and torsional dynamic behaviour of rotating systems. When the assumptions of linearity, small displacements and rotations and constant spin speed and unbalance are possible it works preferentially in the frequency domain, allowing to reach general results, while in other cases it performs the time-domain numerical integration of the equations of motion.

The code is essentially a parametric code and can be used as a routine, which is called by optimisation procedures, being immaterial the type of the latter. Extensive use of Guyan reduction allows to obtain models which are simple enough to perform the dynamic analysis a large number of times in a quick and economical way. When dealing with controlled rotors, as in the case of rotors running on active magnetic bearings, it allows to deal also with the parameters of the control loop. Reduced-order models to be used in the design of the control system can thus be readily obtained.

REFERENCES

- [1] G. Genta, C. Delprete, E. Brusa, *Some considerations on the basic assumptions of the rotordynamics*, Journal of Sound and Vibration, **227**(3), p.611-645.
- [2] G. Belingardi, G. Genta, P. Campagna, *Hybrid Composite Technology for Automotive Drive Shaft: a Computer Aided Optimization*, ISATA 85, Graz, Sept. 1985
- [3] G. Genta, D. Bassani, C. Delprete, *DYNROT: A finite element code for rotordynamic analysis based on complex co-ordinates*, Engineering Computations, Vol. 13, n. 6, 1966, 86-19.
- [4] F.M. Dimentberg, *Flexural Vibrations of Rotating Shafts*, London, Butterworth, 1961.
- [5] R.G. Loevi, V.J. Piarulli, *Dynamics of Rotating Shafts*, The Shock and Vibration Inf. Center Naval Res. lab., Washington, 1969.
- [6] M.A. Prohl, *A General Method for Calculating Critical Speeds of Flexible Rotors*, ASME Journal of Applied Mechanics **67**, A142-A148, 1945.
- [7] J.W. Lund, *Stability and Damped Critical Speed of a Flexible Rotor in Fluid-Film Bearings*, Journal of Eng. for Industry, 509-517, 1974.
- [8] H.D. Nelson, J.M. Mc. Vaugh, *The Dynamics of Rotor-Bearing Systems Using Finite Elements*, Journal of Eng. for Industry, 593-600, 1976.
- [9] G. Genta, *Consistent matrices in rotordynamics*, Meccanica, **20**, 235-248, 1985.
- [10] M. Gerardin, N. Kill, *A New Approach to Finite Element Modeling of Flexible Rotors*, Eng. Computation, **1**, 52-64, 1984.
- [11] G. Genta, *Vibration of Structures and Machines*, 3rd ed., New York, Springer, 1999.
- [12] G. Genta, *Whirling of unsymmetrical rotors: a finite element approach based on complex co-ordinates*, Journal of Sound and Vibration, **124**(1), 27-53, 1988.
- [13] G. Genta, C. Delprete, *Acceleration Through Critical Speeds of an Anisotropic, Nonlinear, Torsionally Stiff Rotor with Many Degrees of Freedom*, Journal of Sound and Vibration, **180**(3), 1995.
- [14] G. Genta, A. Tonoli, *A Harmonic Finite Element for the Analysis of Flexural, Torsional and Axial Rotordynamic Behaviour of Discs*, Journal of Sound and Vibration, **196**(1), 1996, pp. 19-43.
- [15] G. Genta, A. Tonoli, *A Harmonic Finite Element for the Analysis of Flexural, Torsional and Axial Rotordynamic Behaviour of Bladed Arrays*, Journal of Sound and Vibration, **207**(5), 1997, pp. 693-720.
- [16] T. Someya (editor), *Journal Bearing Databook*, Springer, Tokyo, 1988.
- [17] E. Brusa, C. Delprete, G. Genta, *Torsional vibration of crankshafts: effects of non-constant moments of inertia*, Journal of Sound and Vibration, **205**(2), pp.135-150.
- [18] L. Meirovitch, *A new method of solution of the eigenvalue problem for gyroscopic systems*, AIAA Journal, **12**, 1974, 1337-1342.

COMPETING PARALLEL ALGORITHMS AND MULTIPLE LOCAL SEARCHES IN GLOBAL OPTIMIZATION¹

Albert A. Groenwold, Jaco F. Schutte and Hermanus P.J. Bolton

Department of Mechanical Engineering, University of Pretoria, Pretoria, South Africa

ABSTRACT

The unconstrained global programming problem is addressed using, firstly, an efficient multi-start algorithm, in which multiple parallel local searches contribute towards a Bayesian global stopping criterion, denoted the unified Bayesian global stopping criterion.

Secondly, a multi-start, multi-algorithm parallel infrastructure is presented, in which different algorithms, ranging from stochastic to deterministic, compete in parallel for a contribution towards the unified Bayesian global stopping criterion.

The combination of the simple multi-start local search strategy, the competing algorithms and the unified Bayesian global stopping criterion outperforms a number of leading global optimization algorithms, for both serial and parallel implementations. Results for parallel clusters of up to 128 machines are presented.

1 INTRODUCTION

Consider the unconstrained (or bounds constrained) mathematical programming problem represented by the following: Given a real valued objective function $f(\mathbf{x})$ defined on the set $\mathbf{x} \in D$ in $\mathbb{R}^n$, find the point $\mathbf{x}^*$ and the corresponding function value f^* such that

$$f^* = f(\mathbf{x}^*) = \min \{f(\mathbf{x}) | \mathbf{x} \in D\} , \quad (1)$$

if $\mathbf{x}^*$ exists and is unique. Alternatively, find a low approximation $\tilde{f}$ to f^* .

If the objective function and/or the feasible domain D are non-convex, then there may be many local minima which are not optimal. Hence, from a *mathematical* point of view, Problem (1) is essentially unsolvable, due to a lack of mathematical conditions characterizing the global optimum, as opposed to a strictly convex continuous function, which is characterized by the Karush-Kuhn-Tucker conditions at the minimum.

Optimization algorithms aimed at solving Problem (1) are divided in two classes, namely *deterministic* and *stochastic*. The first class being those algorithms which implicitly search all of the function domain and thus are guaranteed to find the global optimum. The algorithms within this class are forced to deal with restricted classes of functions (e.g. Lipschitz continuous functions with known Lipschitz constants). Even with these restrictions it is often computationally infeasible to apply deterministic algorithms to search for the guaranteed global optimum as the number of computations required increases exponentially with the dimension of the feasible space. To overcome the inherent

¹The multiple local searches presented herein are based on the paper 'Multiple parallel local searches in global optimization', by Bolton, Schutte and Groenwold, to be read at the *Euro PVM/MPI 2000 Conference*, Balatonfüred, Hungary, Sept. 2000.

difficulties of the guaranteed-accuracy algorithms, much research effort has been devoted to algorithms in which a stochastic element is introduced, this way the deterministic guarantee is relaxed into a *confidence measure*. A number of successful algorithms belong to the latter class.

A general stochastic algorithm for global optimization consists of three major steps [1]: a sampling step, an optimization step, and a check of some *global stopping criterion*. The availability of a suitable global stopping criterion is probably the most important aspect of global optimization. It is also the most problematic, due to the very fact that characterization of the global optimum is in general not possible.

Global optimization algorithms and their associated global stopping criteria should ultimately be judged on performance. However, when evaluating global optimization algorithms, the use of *a priori* known information about the objective function under consideration should be refrained from. For example, the termination of algorithms once the known global optimum has been attained within a prescribed tolerance complicates the use of these algorithms, and makes comparisons with other algorithms very difficult.

In this paper a number of very simple heuristic algorithms based on multiple local searches are constructed. A Bayesian stopping condition is presented, based on a criterion previously presented by Snyman and Fatti for their algorithm based on dynamic search trajectories [2]. The criterion is shown to be quite general, and can be applied in combination with any multi-start global search strategy. Since the local searches are independent of each other, they are ideally suited for implementation on a massively parallel processing machine.

In addition, we observe that no single global optimization algorithm can consistently outperform all other algorithms when large sets of problems in different classes are considered. This observation leads to the development of an infrastructure in which different algorithms, ranging from deterministic to stochastic, compete in parallel for a contribution towards the same global stopping criterion.

2 A GLOBAL STOPPING CRITERION

It is required to calculate $\tilde{f}$, i.e.

$$\tilde{f} = \min \{ \tilde{f}^j, \text{ over all } j \text{ to date } \} , \quad (2)$$

as the approximation to the global minimum f^* . In finding $\tilde{f}$, over-sampling of f should be prevented as far as possible. In addition, an indication of the probability of convergence to the f^* is desirable. A Bayesian argument seems to us the proper framework for the formulation of a such a criterion. Previously, two such criteria have been presented, respectively by Boender and Rinnooy Kan [3], and Snyman and Fatti [2].

The former criterion, denoted the optimal sequential Bayesian stopping rule, is based on an estimate of the number of local minima in D and the relative size of the region of attraction of each local minimum. While apparently effective, computational expense prohibits using this rule for functions with a large number of local minima in D .

The latter criterion is not dependent on an estimate of the number of local minima in D or the regions of attraction of the different local minima. Instead, a simple assumption about the probability of convergence to the global optimum x^* in relation to the probability of convergence to any local minimum $\tilde{x}_j$ is made. In addition, the probability of convergence to f^* can be calculated.

The rule presented by Snyman and Fatti is derived specifically for their dynamic search method, but

No.	Acronym	Name	No.	Acronym	Function
1	G1	Griewank G1	7	BR	Branin
2	G2	Griewank G2	8	H3	Hartman 3
3	GP	Goldstein-Price	9	H6	Hartman 6
4	C6	Six-hump camelback	10	S5	Shekel 5
5	SH	Shubert, Levi No. 4	11	S7	Shekel 7
6	RA	Rastrigin	12	S10	Shekel 10

Table 1: The extended Dixon-Szegö test set.

is in all probability of greater importance and more generally applicable than hitherto realized. In the following, we will show that this rule can be used as a general stopping criterion in multi-start algorithms, albeit for a restricted class of functions. In doing so, we do not consider the region of attraction R_k of local minimum k . Instead, for a given starting point, we simply refer to the probability of convergence α_k to local minimum k .² Henceforth, we will denote the rule of Snyman and Fatti the unified Bayesian stopping rule.

2.1 The unified Bayesian stopping rule

Let α_k denote the probability that a random starting point will converge to local minimum $\tilde{x}^k$. Also, the probability of convergence to the global minimum x^* is denoted α^* . The following mild assumption, which is probably true for many functions of practical interest, is now made:

$$\alpha^* \geq \alpha_k \text{ for all local minima } \tilde{x}^k. \quad (3)$$

Furthermore, let r be the number of starting points from which convergence to the current best minimum $\tilde{f}$ occurs after $\tilde{n}$ random searches have been started. Then, under assumption (3), the probability that $\tilde{f}$ is equal to f^* is given by

$$Pr [\tilde{f} = f^*] \geq q(\tilde{n}, r) = 1 - \frac{(\tilde{n} + \bar{a})! (2\tilde{n} + \bar{b})!}{(2\tilde{n} + \bar{a})! (\tilde{n} + \bar{b})!}, \quad (4)$$

with $\bar{a} = a + b - 1$, $\bar{b} = b - r - 1$, and a, b suitable parameters of the Beta distribution $\beta(a, b)$. On the basis of (4) the adopted *stopping rule* becomes

$$\text{STOP when } Pr [\tilde{f} = f^*] \geq q^*, \quad (5)$$

where q^* is some prescribed desired confidence level, typically chosen as 0.99 - 0.999.

3 A SIMPLE GLOBAL SEARCH HEURISTIC

In all probability, the simplest global optimization algorithm is the combination of multiple local searches, combined with some probabilistic stopping criterion. Here, we present such a formulation, and utilize (5). We also provide for a global minimization step. Various sequential algorithms may be constructed using the following framework:

²Studying simple 1-D search trajectories, we observe that the definition of region of attraction of a local minimum is problematic. Strictly speaking, the region of attraction can only be defined when non-discrete search trajectories (line search or other) are employed.

Algorithm	Function	Number of Failures			
		$q^* = .95$	$q^* = .99$	$q^* = .999$	$q^* = .9999$
GLS1	G1	27	18	6	5
	G2	21	11	4	3
	RA	20	18	6	2
LLS1	G1	39	17	8	4
	G2	12	7	3	2
	RA	54	33	15	4
GLS2	G1	16	12	1	0
	RA	12	8	7	4
LLS2	G1	22	18	7	4
	RA	15	12	7	2
SF [2]	G1	6	2	1	1
	G2	52	29	12	12
	SH	54	43	20	18
	RA	38	18	6	6

Table 2: Number of failures of convergence to the global optimum for 100 (random) restarts of each algorithm for the complete test set. For the problems not listed, the number of failures is 0 for all tabulated values of the prescribed confidence q^* . (Less than 3 failures at $q^* = 0.95$, combined with none at higher values of q^* are not reported).

1. **Initialization:** Set the trajectory counter $j := 1$, and prescribe the desired confidence level q^* .
2. **Sampling steps:** Randomly generate $\mathbf{x}_0^j \in D$ in $\mathbb{R}^n$.
3. **Global minimization steps:** Starting at $\mathbf{x}_0^j$, attempt to minimize f in a global sense by some preliminary search procedure, viz. find and record some low point $\bar{f}^j \leftrightarrow \bar{\mathbf{x}}^j$.
4. **Local minimization steps:** $\bar{\mathbf{x}}^j$ is used as the starting point for a robust gradient based convex minimization algorithm, with stopping criteria defined in terms of the Karush-Kuhn-Tucker conditions. Record the lowest function value $\tilde{f}^j \leftrightarrow \tilde{\mathbf{x}}^j$.
5. **Global termination:** Assess the global convergence after j searches to date (yielding $\mathbf{x}^k$, $k = 1, 2, \dots, j$) using (5). If (5) is satisfied, STOP, else, $j := j + 1$ and goto 2.

Pure multiple local searches are obtained if Step 3 is excluded, with $\bar{\mathbf{x}}^j = \mathbf{x}_0^j$. We now construct 2 such simple algorithms, namely

1. LLS1: multiple local searches using the bound-constrained BFGS algorithm [9, 10], and
2. LLS2: multiple local searches using the unconstrained Polak-Ribiere algorithm [11].

In addition, for both LLS1 and LLS2 we add a global minimization phase (step 3), and denote the respective algorithms GLS1 and GLS2. In the global phase we simulate the trajectories of a bouncing ball (the MBB algorithm, [12]), which is attractive due to its simplicity. The ball's elasticity coefficient is chosen such that the ball's energy is dissipated very quickly.

Problem	GLS1	LLS1	GLS2	LLS2	SF [2]	[4, 5]	[6, 7]	[8]
G1	2644	10678	2992	7215	5063	1822	396147	3623
G2	1882	1675	2398	1510	86672	10786	828441	16121
GP	454	229	403	471	2069	6775	94587	7450
C6	238	108	275	225	602	579	76293	3711
SH	1715	1626	1363	1485	93204	1443	139087	3788
RA	2893	1487	3119	3161	45273	3420	445711	251
BR	211	240	552	724	9553	594	71688	4769
H3	289	199	478	462	1695	915	103466	1698
H6	346	266	521	588	3550	3516	106812	9933
S5	315	479	353	607	6563	1772	234654	1915
S7	273	473	417	555	1848	1923	212299	4235
S10	382	508	449	564	1604	2631	330486	4226

Table 3: Comparison with some other algorithms. For the problems listed, the number of function values N_{fe} for the different algorithms are reported.

4 MULTIPLE COMPETING ALGORITHMS

We observe that no single global optimization algorithm can consistently outperform all other algorithms when large sets of problems in different classes are considered. A cursory glance at numerical results in the literature suffices to impress this observation.

Hence a sensible, if somewhat unconventional, approach is to attempt to solve global programming problems using a number of different algorithms simultaneously. The results of all the different algorithms combined may then be used to study the quality of local minima found. Obviously, this approach is senseless if the different algorithms are incorporated in a sequential algorithm. On the other hand, multiple algorithms in parallel are very viable.

A complication when using different algorithms simultaneously once again relates to the selection of a global stopping criterion. However, if assumption (3) holds for a given algorithm and objective function, then stopping criterion (5) may be used³. Hence, we implement multiple algorithms, ranging from deterministic to stochastic, which compete for a contribution towards the unified Bayesian stopping criterion.

Our motivation for including ‘deterministic’ algorithms is as follows: The cost of finding the global optimum in a deterministic sense may simply be prohibitive. For example, our numerical experiments have shown that when using the clustering algorithm [4, 5], it is advantageous to decrease the number of sampling points used, and to rather restart the algorithm a number of times using different random starting points. Hence we introduce a stochastic element into the ‘deterministic’ algorithms. A similar approach can even be followed when using the genetic algorithm (GA), which is frequently run a number of times using relatively small population sizes anyway, as opposed to a single run with a large initial population.

Currently, the various algorithms competing for a contribution to the unified Bayesian global stopping criterion in our infrastructure are GLS1, GLS2, LLS1, a genetic algorithm (GA), the Snyman-Fatti

³We have also combined stopping criterion (5) with local search algorithms when studying difficult (e.g. ill-posed) convex problems, since assumption (3) obviously holds.

Prob.	GLS1			32-node pvm			128-node pvm		
	N_{fe}	$r/\tilde{n}$	$q(\tilde{n}, r)$	N_{vc}	$r/\tilde{n}$	$q(\tilde{n}, r)$	N_{vc}	$r/\tilde{n}$	$q(\tilde{n}, r)$
G1	1599	6 / 76	0.9929	90	6 / 96	0.9929	30	7 / 128	0.9965
G2	2122	6 / 50	0.9933	189	6 / 96	0.9928	74	7 / 128	0.9965
GP	341	5 / 12	0.9903	40	18 / 32	1.0000	39	59 / 128	1.0000
C6	163	5 / 9	0.9923	22	19 / 32	1.0000	22	75 / 128	1.0000
SH	1290	6 / 49	0.9933	89	9 / 64	0.9993	50	17 / 128	1.0000
RA	817	6 / 41	0.9935	96	8 / 128	0.9982	26	9 / 128	0.9992
BR	107	4 / 4	0.9921	78	31 / 32	1.0000	76	120 / 128	1.0000
H3	207	5 / 8	0.9932	32	18 / 32	1.0000	32	77 / 128	1.0000
H6	288	5 / 8	0.9932	60	21 / 32	1.0000	59	79 / 128	1.0000
S5	132	5 / 8	0.9932	22	6 / 32	0.9939	52	52 / 128	1.0000
S7	293	6 / 17	0.9953	25	14 / 32	1.0000	37	56 / 128	1.0000
S10	336	6 / 17	0.9953	32	11 / 32	0.9999	39	48 / 128	1.0000

Table 4: Apparent visual cost N_{vc} for a 32-node parallel virtual machine and a 128 node parallel virtual machine. N_{vc} may be compared with the number of function evaluations N_{fe} of the sequential GLS1 algorithm. r represents the number of starting points from which convergence to the current best minimum $\tilde{f}$ occurs after $\tilde{n}$ random searches have been started. The probability that $\tilde{f}$ is equal to f^* is given by $q(\tilde{n}, r)$.

algorithm [2], Boender's algorithm [4], clustering [6, 7], and the algorithm presented by Mockus [8].

5 PARALLEL IMPLEMENTATION

The search trajectories generated in our algorithms are completely independent of each other. Hence the sequential algorithm presented in section 3, and the competing algorithms outlined in section 4, may easily be parallelized. To this extent, we utilize the freely available pvm3 [13] code for FORTRAN, running under the Linux operating system. Currently, the massive parallel processing virtual machine (MPPVM) consists of up to 128 Pentium III 450 MHz machines in an existing undergraduate computer lab.

The distributed computing model represents a master-slave configuration where the master program assigns tasks and interprets results, while the slaves compute the search trajectories. The workload is statically assigned, and no inter-slave communication occurs. The master program informs each slave task of the problem parameters by a single broadcast and awaits individual results from each slave.

5.1 Multiple competing algorithms

Many strategies for implementing the multiple competing algorithms are possible. For example, the algorithm assigned to a particular slave can be determined randomly with uniform probability, or according to a predetermined probability (based on, for instance, the performance of individual algorithms for a large set of test problems). In this initial study we have opted for the former.

5.2 A measure of computational effort

We will assume that our algorithm will ultimately be used in problems for which the CPU require-

n	LLS1		SF [2]		Clustering [4, 5]		Mockus [8]	
	N_{fe}	F	N_{fe}	F	N_{fe}	F	N_{fe}	F
2	1487	1	45273	0	3420	1	251	0
3	4247	4	348937	2	2451	7	292	0
5	6548	8	165287	7	2556	10	482	0
10	24281	8	> 500000	10	6141	10	964	0
20	—	—	—	—	—	—	1928	0

Table 5: Numerical results for the Rastrigin (RA) function. Influence of the number of variables n on the cost N_{fe} and the number of failures 'F' to converge to f^* . (Each algorithm was started 10 times at a random starting point.)

ments of evaluating f is orders of magnitudes larger than the time required for message passing and algorithm internals. (In structural optimization, for example, each function evaluation typically involves a complete finite element or boundary element analysis.) Hence we define a somewhat unconventional measure for the cost of our parallelized algorithm which we denote *apparent visible cost* (N_{vc}). This cost represents the number of function evaluations associated with the random starting point x_0^j which results in the most expensive search trajectory. The time window (in CPU seconds) associated with this search trajectory is denoted the *virtual CPU time*. The virtual CPU time includes the time window associated with initialization and evaluation of stopping criterion (5).

6 NUMERICAL RESULTS

The algorithms are tested using an extended Dixon-Szegö test set, presented in Table 1. The 12 well known functions used are given in, for instance, [14].

Firstly, Table 2 shows the effect of the prescribed confidence level q^* in stopping criterion (5). The decreasing number of failures of convergence to f^* as q^* increases illustrates the general applicability of the unified Bayesian global stopping rule. All of the new algorithms outperform the SF algorithm, for which algorithm the stopping criterion was originally derived.

Table 3 reveals that the simple sequential algorithms presented herein compare very favorably with a number of leading contenders, namely the Snyman-Fatti algorithm [2], clustering [4, 5], algorithm 'sigma' [6, 7] and the algorithm presented by Mockus [8]. (All the algorithms were started from different random starting points, and the reported cost is the average of 10 independent runs. The parameters in the clustering algorithm, and the algorithm presented by Mockus were selected such that convergence to f^* in at least 8 out of the 10 random starts were obtained.) For the new algorithms, the results for two very difficult test functions, namely Griewank G1 and Griewank G2 [15], in particular are encouraging: Few algorithms find the solution to G2, (which has a few thousand local minima in the region of interest), in less than some 20000 function evaluations.

Table 4 shows the effect of parallel implementation. For relatively 'simple' problems (viz. problems with few design variables or few local minima in the design space), the probability of convergence to the global optimum becomes very high when the number of nodes is increased. This is illustrated by, for example, the results for the C6 problem. For more difficult problems (e.g. the G1 and G2 problems), the probability of convergence to the global optimum f^* is increased.

Simultaneously, the total computational time, (as compared to the sequential GLS1 algorithm), decreases notably. For the 32-node parallel virtual machine, the virtual CPU time to evaluate all the test functions on average decreases by a factor of 1.93 (not shown in tabulated form). The time associated with message passing is negligible compared to the time associated with the global searches.

When the time associated with a single function evaluations become much larger than the time required for algorithm internals, the fraction N_{fe}/N_{vc} based on Table 4 may be used as a direct indication of the decrease in virtual computational time obtainable as a result of parallelization. For the G2 problem, this would imply a reduction in computational time by a factor of 28.68 for the 128-node parallel virtual machine.

6.1 Multiple competing algorithms

We now illustrate the benefits of combining the competing algorithms in one infrastructure: In general, the algorithm presented by Mockus [8] is outperformed by the simple line searches presented in section 3 (e.g. see Table 3). However, the algorithm of Mockus is extremely effective when applied to the difficult Rastrigin problem when the number of variables n increases (see Table 5).

For the sake of brevity, we do not present numerical results for our competing algorithm infrastructure. The results are very similar to those presented for the 32-node pvm in Table 4, since we terminate slave programs that have not converged once the probability $q(\tilde{n}, r)$ of convergence to f^* exceeds 0.999. Hence the only marked influence is that $q(\tilde{n}, r)$ in general decreases slightly. However, difficult problems with many design variables (e.g. the Rastrigin problem) can be solved more efficiently.

7 CONCLUSIONS

We have addressed the unconstrained global programming problem using an efficient multi-start algorithm, in which multiple parallel local searches contribute towards a Bayesian global stopping criterion. The stopping criterion, denoted the unified Bayesian global stopping criterion, is based on the mild assumption that the probability of convergence to the global optimum $\mathbf{x}^*$ is comparable to the probability of convergence to any local minimum $\tilde{\mathbf{x}}_j$.

In addition, a multi-start, multi-algorithm parallel infrastructure is presented, in which different algorithms, ranging from stochastic to deterministic, compete for a contribution towards the unified Bayesian global stopping criterion. The competing algorithms are motivated by our observation that no single (global) optimization algorithm can consistently outperform all other algorithms when large sets of problems in different classes are considered.

The combination of the simple multi-start local search strategy, the competing algorithms and the unified Bayesian global stopping criterion outperforms a number of leading global optimization algorithms, for both serial and parallel implementations.

Parallelization is shown to be an effective method to reduce the computational time associated with the solution of expensive global programming problems. While the apparent computational effort is reduced, the probability of convergence to the global optimum is simultaneously increased.

REFERENCES

- [1] F. Schoen. Stochastic techniques for global optimization: A survey of recent advances. *J. Global Optim.*, 1:207–228, 1991.

- [2] J.A. Snymann and L.P. Fatti. A multi-start global minimization algorithm with dynamic search trajectories. *J. Optim. Theory Appl.*, 54:121–141, 1987.
- [3] C.G.E. Boender and A.H.G. Rinnooy Kan. A Bayesian analysis of the number of cells of a multinomial distribution,. *Statistician*, 32:240–248, 1983.
- [4] C.G.E. Boender, A.H.G. Rinnooy Kan, G.T. Timmer, and L. Stougie. A stochastic method for global optimization. *Math. Program.*, 22:125–140, 1982.
- [5] A.H.G. Rinnooy Kan and G.T. Timmer. Stochastic global optimization methods, Part I: Clustering methods. *Mathemat. Program.*, 39:27–56, 1987.
- [6] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. Global optimization and stochastic differential equations. *J. Optim. Theory Appl.*, 47:1–16, 1985.
- [7] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. SIGMA - a stochastic-integration global minimization algorithm. *ACM Trans. Math. Softw.*, 14:366–380, 1988.
- [8] J. Mockus. *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1989.
- [9] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scient. Comput.*, 16:1190–1208, 1995.
- [10] C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: FORTRAN subroutines for large scale bound constrained optimization. Technical Report NAM-11, Northwestern University, EECS Department, 1994.
- [11] J.C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods. *SIAM J. Optim.*, 2, 1992.
- [12] A.A. Groenwold and J.A. Snymann. Global optimization using dynamic search trajectories. In *Proc. Conference Discrete and Global Optimization*, Chania, Crete, May 1998.
- [13] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. PVM: Parallel Virtual Machine - A users guide and tutorial for networked parallel computing. <ftp://netlib2.cs.utk.edu/pvm3/>, 1997. (ver. 3.4).
- [14] A. Törn and A. Zilinskas. *Global optimization*, volume 350 of *Lecture notes in computer science*. Springer-Verlag, Berlin, Heidelberg, 1989.
- [15] A.O. Griewank. Generalized descent for global optimization. *J. Optim. Theory Appl.*, 34:11–39, 1981.

APPENDIX: PROOF OF STOPPING CRITERION

We present here an outline of the proof of (4), and follow closely the proof in [2].

Given n and α^* , the probability that at least one point, $n^* \geq 1$, has converged to f^* is

$$\Pr[n^* \geq 1 | n, r] = 1 - (1 - \alpha^*)^n. \quad (6)$$

In the Bayesian approach, we characterize our uncertainty about the value of α^* by specifying a prior probability distribution for it. This distribution is modified using the sample information (namely, n and r) to form a posterior probability distribution. Let $p_*(\alpha^*|n, r)$ be the posterior probability distribution of α^* . Then,

$$\begin{aligned}\Pr[n^* \geq 1|n, r] &= \int_0^1 [1 - (1 - \alpha^*)^n] p_*(\alpha^*|n, r) d\alpha^* \\ &= 1 - \int_0^1 (1 - \alpha^*)^n p_*(\alpha^*|n, r) d\alpha^*.\end{aligned}\quad (7)$$

Now, although the r sample points converge to the current overall minimum, we do not know whether this minimum corresponds to the global minimum of f^* . Utilizing (3), and noting that $(1 - \alpha)^n$ is a decreasing function of α , the replacement of α^* in the above integral by α yields

$$\Pr[n^* \geq 1|n, r] \geq \int_0^1 [1 - (1 - \alpha)^n] p(\alpha|n, r) d\alpha. \quad (8)$$

Now, using Bayes theorem we obtain

$$p(\alpha|n, r) = \frac{p(r|\alpha, n)p(\alpha)}{\int_0^1 p(r|\alpha, n)p(\alpha) d\alpha}. \quad (9)$$

Since the n points are sampled at random and each point has a probability α of converging to the current overall minimum, r has a binomial distribution with parameters α and n . Therefore

$$p(r|\alpha, n) = \binom{n}{r} \alpha^r (1 - \alpha)^{n-r}. \quad (10)$$

Substituting (10) and (9) into (8) gives:

$$\Pr[n^* \geq 1|n, r] \geq 1 - \frac{\int_0^1 \alpha^r (1 - \alpha)^{2n-r} p(\alpha) d\alpha}{\int_0^1 \alpha^r (1 - \alpha)^{n-r} p(\alpha) d\alpha}. \quad (11)$$

A suitable flexible prior distribution $p(\alpha)$ for α is the beta distribution⁴ with parameters a and b :

$$p(\alpha) = [1/\beta(a, b)] \alpha^{a-1} (1 - \alpha)^{b-1}, \quad 0 \leq \alpha \leq 1 \quad (12)$$

Using this prior distribution gives:

$$\begin{aligned}\Pr[n^* \geq 1|n, r] &\geq 1 - \frac{\Gamma(n + a + b) \Gamma(2n - r + b)}{\Gamma(2n + a + b) \Gamma(n - r + b)} \\ &= 1 - \frac{(n + a + b - 1)! (2n - r + b - 1)!}{(2n + a + b - 1)! (n - r + b - 1)!},\end{aligned}$$

which is the required result.

⁴In [2], a statistically non-informative distribution ($a = b = 1$) was used, implying a prior expectation for α of 0.5. Noting that the criterion mostly fails when the algorithms used terminate very quickly, we utilize $\beta(1, 5)$.

Choosing Optimal Control Policies using the Attainable Region Approach

Sven Godorr, Diane Hildebrandt*, David Glasser, Craig McGregor and Brendon Hausberger

School of Process and Materials Engineering, University of the Witwatersrand, P. Bag 3, WITS 2050, South Africa.

E-Mail: dihil@chemeng.chmt.wits.ac.za

Fax: +27 11- 716 2482

*- Author to whom correspondence should be addressed

ABSTRACT

Previously the Attainable Region has been constructed for systems where the rate vector is uniquely defined. In this paper we extend the Attainable Region approach to situations where the rate vector depends on a control parameter, such as temperature. In these cases, the rate vector can take on a range of values, depending on the value of the control parameter. Arguments based on the geometry of the boundary of the Attainable Region are used to derive equations that describe the optimal control policies. These conditions are applied to various examples and both the optimal reactor structures as well as optimal operating and control policies are derived by looking at the structures that make up the boundary of the Attainable Region. In particular, an example is given where the optimal reactor structure has a reactor with simultaneous side stream addition and temperature control.

Introduction

The optimisation of reactor structures plays a central role in the development of efficient chemical processes. The behaviour of the reaction step in an industrial operation is often pivotal in determining its overall performance, since it will define the upstream and downstream processing steps that are required, either for preparing the feed or upgrading the reactor products, to meet the final product and/or effluent specifications. Reactor optimisation focuses on issues such as obtaining the best possible conversion and selectivity, or minimising the catalyst costs, equipment size and utility consumption of the overall process. This relies on controlling variables such as operating temperature, catalyst combinations and mixing within the reactor structure, often subject to constraints on pressure, temperature and concentrations of certain species in the reaction mixture.

Optimal control problems of this type are traditionally cast in the same formulation as the calculus of variations. They, however, allow for the state variables to be subject to some dynamic constraints, which depend both on the values of the state variables, $\mathbf{x}(t)$ and on a set of control variables, $\mathbf{u}(t)$:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad [1]$$

(An important property of this type of problem is that the dynamics $\mathbf{f}$, must depend only on the current state of the system and must thus not exhibit any memory, i.e. the dynamics must be Markovian.)

For optimal control problems the optimisation can be written as:

$$\underset{\mathbf{u}(t)}{\text{opt}}(I = \int_{t_1}^{t_2} L(\mathbf{x}(t), \mathbf{u}(t), t) dt) \quad [2]$$

subject to the boundary conditions:

$$\mathbf{x}(t_1) = \mathbf{x}_{\text{initial}}$$

$$\mathbf{x}(t_2) = \mathbf{x}_{\text{final}}$$

For these problems, the curves which are available for the optimisation are restricted to those obeying the dynamics defined by equation [1], so that the structure of the optimisation problem is influenced by both L and $\mathbf{f}$. The task then becomes that of choosing the control variables, $\mathbf{u}(t)$ in such a way that the path followed by the system between the initial and final conditions optimises I as required. When I is defined as in equation [2], it may easily be interpreted as some type of cost associated with changing the system from the initial to the final state. By suitable definition of

additional state variables a range of different optimisation problems may, however, be transformed into this standard form.

The three classic techniques for solving optimal control problems are the calculus of variations, dynamic programming and Pontryagin's Maximum Principle. These methods all involve finding a single path between the initial and final states of the system, along which the conditions for optimality are satisfied. In this way a direct solution of the optimisation problem (contained in [2]) is found subject to whatever dynamic behaviour governs the state variables. Thus, in principle, the optimal control policy and hence the path associated with it, are found directly.

It may however be possible to achieve the desired final state more efficiently by using several different paths to reach other states, from which the required final state may then be realised. For example, if a certain concentration is to be achieved in a chemical reactor it might be more efficient to produce other compositions by means of reaction and then to blend these together to create a mixture with the desired properties. Mixing such as this results in discontinuities in the state variables so that the classic techniques have difficulty in finding solutions of this type. Further, the concept of using several different paths which themselves may not lead directly to the required final state, implies a more global exploration of the solution space than is implicit in the classic techniques. Indeed, only an exhaustive exploration of all possible states that can be achieved in a system, will guarantee that no further improvements may be achieved by taking indirect advantage of other states that may be realised in the system.

The attainable region approach to optimisation differs fundamentally from the classical methods in that it separates the dynamic behaviour of the system from the optimisation itself. In this way the totality of states that can be achieved by the system is determined, which is equivalent to considering all possible curves that obey the dynamic constraints. This has the advantage that all possible states that can be achieved are considered simultaneously. Any improvements that occur due to states that have been achieved elsewhere in the system are also considered. Once the absolute bounds on system performance have been determined, the optimisation is easily performed as a subsequent exercise. This may seem inefficient when compared to the current approaches, which find a single optimal curve directly. However, by considering the dynamic behaviour of the system separately, important information is obtained about the control variable settings associated with the limits of system performance. These insights into the overall structure ensure that the global optimum is ultimately found and avoid some of the pitfalls associated with a simple application of the necessary conditions for optimality. Further, the attainable region approach allows for composite paths between the feed point and the desired final state, which have often been found to correspond to solutions that are globally optimal.

The Attainable Region

The concept of an attainable region (AR) was introduced by Horn (1964) as a way of representing all possible states that can be reached by a chemical reactor system, given a number of permitted fundamental processes that are operated according to a set of control variables. In particular, he used the attainable region to describe all possible compositions that can be achieved in a reactor by varying conditions such as residence time, temperatures and feed policies. Once the set of all achievable compositions had been found, he showed how it became trivial to optimise an objective function that depends on the variables defining the attainable region space. Horn suggested that the optimal operating policies would generally correspond to those required to reach points on the border of the attainable region. Although he did not present any general method for determining such a region, he pointed out several of its important and useful properties. Horn stated that this region would not necessarily include all stoichiometrically possible compositions and as such the attainable region would be the more restricted set of compositions that can actually be realised in all possible reactors.

Glasser et al. (1987) and Hildebrandt et al. (1990) developed a geometric approach for the construction of the attainable region for steady flow chemical reactors, in which the fundamental processes of reaction and mixing are allowed to occur. From this geometric interpretation, they developed some powerful necessary conditions for the attainable region. Hildebrandt and Glasser (1990) studied a number of three dimensional problems, providing some guidelines on how the attainable region can be constructed in $\mathbb{R}^3$ space. Based on the geometric interpretation of reaction and mixing in conjunction with the properties of convex sets, they started drawing some important conclusions about the general structures that can be expected to make up the boundary of the attainable region. They suggested that series-parallel arrangements of plug flow reactors (PFRs) and continuous flow stirred tank reactors (CSTRs) with bypassing would in general be the optimal structures that define the AR boundary. Conditions were proposed for situations where the fundamental processes of reaction and mixing would occur simultaneously in the boundary of the attainable region.

The concept of an attainable region has recently been applied to some simple, industrially relevant problems. Omtveit et al. (1993) used these ideas to study a process synthesis problem involving reactor-recycle loops, for steam reforming reactions. Glasser et al. (1992) examined the classical problem of a single exothermic reversible reaction, exploring reactor systems in which only reaction, mixing and preheating of a portion of the feed are allowed. Hopley et al. (1996)

found the optimal reactor structures for more complex kinetic expressions describing exothermic reversible reactions. They used the tangency conditions for reaction and mixing vectors in the boundary of the AR to identify situations where a certain amount of cross-mixing between parallel PFRs was required to achieve the full AR.

Feinberg & Hildebrandt (1992 and 1997) have provided a rigorous mathematical framework and formal proofs of various results that have been used in the attainable region literature dealing with the fundamental processes of reaction and mixing. These results will be dealt with in more detail in the next section and will form the basis for much of the work in this paper.

Theoretical Development

The theoretical development presented in this paper differs slightly in notation and approach from that used by Glasser, Hildebrandt and their co-workers and is somewhat less rigorous than that used by Feinberg & Hildebrandt (1997). Much of the development work was however conducted in collaboration with Feinberg who provided suggestions as well as a fundamental approach for developing further useful results. In this work the attainable region is applied to solving some more general optimisation problems in the design and operation of chemical reactors. In developing these ideas it became obvious that the concept of an attainable region had applications in other fields of optimisation. Although the fundamental ideas and most examples are still based in the field of chemical engineering, the notation and language is slightly more general with a view to applying the concepts to other problems.

Basic Concepts of the Attainable Region

If a system may be fully described by n variables, then every state of that system may be uniquely described by a point, $\mathbf{x}$, in $\mathcal{R}^n$ space. Fundamental processes that are allowed to operate within the system, transforming it from one state to another, may then be represented by vector fields. These move the system from one point, representing a given state, to another point or state, according to the change effected by the vector process.

A system starting in some initial state (described by the point $\mathbf{x}_0$) may be transformed into other states (corresponding to new points $\mathbf{x}$) as a result of the action of the permitted fundamental processes. The attainable region corresponds to all points (states) that can be achieved by means of the permitted fundamental processes. In this way a chemical stream may be described by a vector of concentrations, so that its composition maps to a point in concentration space. Starting at a feed composition that is described by the point $\mathbf{x}_0$, a chemical reaction, described by the vector $\mathbf{r}$, may be allowed to occur. This changes the composition of the stream so that its new state is described by a different point in the concentration space. The attainable region is the totality of all possible compositions (points) that can be achieved from the feed composition, by means of reaction and any other fundamental processes that are allowed. In the design of chemical reactor systems, the other main fundamental process available to the design engineer is that of mixing between various streams.

The fundamental processes permitted within a system may depend both on the state variables $\mathbf{x}$ as well as the values of the control variables $\mathbf{u}$. The net effect of all the fundamental processes operating in a system may be expressed as the resultant process:

$$\mathbf{p} = \mathbf{p}(\mathbf{x}, \mathbf{u}) \quad [3]$$

which changes the system state according to:

$$\frac{d\mathbf{x}}{d\tau} = \mathbf{p}(\mathbf{x}, \mathbf{u}) \quad [4]$$

with $\mathbf{x} = \mathbf{x}_0$ at $\tau = 0$.

By changing the settings of the control variables it may be possible to alter not only the fundamental processes but also the manner in which the various individual processes are combined.

Consider the example of a chemical system in which a single reaction, represented by the reaction vector $\mathbf{r}$, and mixing between achievable points is permitted. In a constant density system, the concentrations making up the composition vector, $\mathbf{x}$ obey linear mixing rules and so the mixing vector may be defined as:

$$\mathbf{v} = \mathbf{x}_m - \mathbf{x} \quad [5]$$

with $\mathbf{x}_m$ being some other attainable composition.

Generally the reaction rate depends both on the concentrations of the reacting species and the temperature, T of the mixture. The overall process vector, which allows any combination of reaction and mixing to occur, is given by:

$$\mathbf{p}(\mathbf{x}, \mathbf{u}) = \mathbf{r}(\mathbf{x}, T) + \alpha \mathbf{v}(\mathbf{x}, \mathbf{x}_m) \quad \text{where } \alpha \geq 0 \quad [6]$$

and the available control variables are:

$$\mathbf{u} = (T, \alpha, \mathbf{x}_m) \quad [7]$$

The temperature, T , and mixing point, $\mathbf{x}_m$, are examples of control variables that cause one of the available fundamental process to assume a range of values at a given point, $\mathbf{x}$. Control variables of this nature must be considered differently to variables such as α which determine how two (or more) fundamental processes are combined at a given point, to give the resultant vector, $\mathbf{p}$. In order to differentiate between the two types of variables, we will call variables which determine the combination of fundamental process vectors, such as α , **policy variables** and variables which affect the fundamental process vectors, such as temperature, T , and mixing point, $\mathbf{x}_m$, **control variables**. The attainable region then consists of the totality of all points (states) that can be reached from the initial state(s) by optimal choice of the policy and control variables.

Optimisation with the Attainable Region

Once the attainable region has been found, it represents the set of all possible states that can be achieved by a given system. The boundary of the region has a special significance since it will contain all the extremal values that the state variables may achieve. If an objective function that is an algebraic function of the state variables is to be optimised, then most often this will be extremalised on the boundary which represents some limit in terms of system performance. Thus a search of the attainable region boundary will generally result in the optimal operating conditions. Points on the boundary of the attainable region can usually only be achieved in a limited number of ways. The values of the policy variables along the boundary, indicate what combination of fundamental processes are required to reach this boundary point as well as the values of the control variables that are specific to the individual fundamental processes. Details of the fundamental processes required to reach a particular boundary point are determined by tracing the vectors back to the initial state.

Points within the attainable region can generally be reached in a variety of ways. If mixing is allowed for instance, any interior point can be achieved in infinitely many ways by mixing between appropriately chosen achievable points. Thus, if the optimum value of the objective function lies within the attainable region, the optimisation is unconstrained with reference to the combination of fundamental processes occurring.

Choice of State Variables

The state variables that define the space within which the attainable region is to be determined, must be chosen so that for a particular setting of the control variables each individual fundamental process is uniquely defined at every point in the stoichiometric subspace space. It thus follows that, for given control settings, the net (resultant) process will be represented by a well-defined vector field in the space. For the example of a chemical stream, sufficient compositions must be included so that the chemical reaction vector at a given temperature is uniquely defined at every point in the composition space.

Any objective function that is to be optimised once the attainable region has been determined, must also be fully defined by the state variables and this may place additional requirements on their choice. For instance, residence time may have to be included as an additional variable to describe the state of a chemical stream, if the ultimate optimisation is to take account of equipment size.

Properties of the Attainable Region

Necessary Conditions for the Attainable Region

Glasser et al. (1987) developed a set of necessary conditions for the attainable region and these have been more formally addressed by Feinberg & Hildebrandt (1997). These conditions provide a set of important properties that can be used to test whether a proposed region is in fact a candidate for the complete attainable region.

1. The attainable region includes the feedpoint(s) (initial state(s) of the system).
2. No fundamental process (rate) vector in the boundary of the attainable region points out of the region, i.e. the fundamental process vectors in the boundary must either point inwards, be tangent to the boundary or be zero.
3. If the state variables obey linear mixing rules and if the fundamental process of mixing is permitted, then the attainable region will always be convex. When working in a space where mixing is not a meaningful process or one has chosen not to allow it, the requirement that the attainable region be convex falls away.
4. If mixing is permitted as a fundamental process, it must not be possible to extend a negative rate vector at any point lying outside of the region so that it intersects the region. (A CSTR with feedpoint within the region could be used to achieve that point).

The following remarks are pertinent to the necessary conditions set out above:

1. The necessary conditions 1 and 2 are quite general in nature and will apply to the construction of any attainable region.
2. Condition 3 may be viewed as a result arising from the more general condition 2.
3. Condition 4 relies strongly on the geometric interpretation of reaction and mixing, and will not necessarily apply to systems in general in which arbitrary fundamental processes are allowed.

General Structure for Reaction and Mixing in three Dimensions

An AR in $\mathcal{R}^3$ space is a three dimensional volume and its boundary is a two dimensional surface. Feinberg and Hildebrandt (1997) have developed a number of powerful results to describe the properties of this two dimensional boundary when only reaction and mixing are allowed. These results have been rigorously proven and as such form a sound theoretical foundation for further development of AR theory.

1. It has been shown that the boundary of the AR is the union of straight line sections, corresponding to regions where only mixing is occurring ($\alpha=\infty$), and PFR trajectories, where only the fundamental process of chemical reaction occurs ($\alpha=0$). This means that the boundary is made up of surfaces where only a single process, either reaction or mixing, is occurring.
2. The one dimensional curves along which these single-process surfaces intersect, are called intersectors. These intersectors are the major process pathways along which access is gained to the single process surfaces lying between them.
3. If the PFR and mixing surfaces do not intersect smoothly, the intersector forms a ridge in the boundary of the AR. This ridge is itself a PFR trajectory, which means that mixing plays no part in determining the shape of this intersector. Along such a "ridge intersector" it is not possible to define a unique tangent plane to the boundary surface.
4. If there is a smooth transition from a (two dimensional) region in which only mixing occurs, to one where only reaction occurs, then a unique tangent plane may be defined. If the reaction vector points away from the intersector, then both the fundamental processes of mixing ($\mathbf{v}$) and reaction ($\mathbf{r}$) occur simultaneously along such a intersector, as described by equation [6]. The relative magnitudes of $\mathbf{r}$ and $\mathbf{v}$ may be controlled by varying α along the connector, which then represents an optimal combination of these two fundamental processes. Feinberg and Hildebrandt called this type of intersector a connector.

The connector is either a differential side stream reactor (DSR) or a locus of CSTR operating points. The mixing lines lie on one side of the connector, whilst on the other side of the connector a family of PFR trajectories define a two dimensional surface, where only reaction occurs. As a result of the smooth transition from mixing to reaction along such a connector, both the reaction and mixing vectors lie in the tangent plane (support hyperplane) to the boundary.

5. The condition for the occurrence of a DSR in the boundary of the AR has been shown to be:

$$\varphi(\mathbf{x}) = (\mathbf{r}(\mathbf{x}) \times \mathbf{v}(\mathbf{x})) \cdot \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \mathbf{v}(\mathbf{x}) \right) = 0 \quad [8]$$

where $\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian (derivative) matrix. The points where the condition given by equation [8] holds,

define a two dimensional manifold in which the connector must lie. This equation is often referred to as the "VdelR condition".

6. A functional form for the policy variable α has been developed. This ensures that the connector moves in the manifold defined by equation [8], by setting the extent of mixing that is permitted. The control variable is found simply by differentiation of equation [8] and is given explicitly by:

$$\alpha = \frac{-\nabla \varphi(\mathbf{x}) \cdot \mathbf{r}(\mathbf{x})}{\nabla \varphi(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x})} \quad \text{provided } \nabla \varphi(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) \neq 0 \quad [9]$$

Formal derivations for conditions [8] and [9] are given by Feinberg and Hildebrandt (1997).

As mentioned previously it is important to distinguish between variables that determine how different fundamental processes are combined to give the net resultant process at a certain point, called policy variables, and those which cause a single fundamental process to operate in different ways for a given set of state variables, which we call control variables. This paper deals with control variables which cause the vector describing a fundamental process to vary at every point.

Control Variables

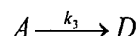
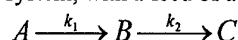
In this paper we consider finding the AR when one of the fundamental processes depends on a control variable in such a way that at every point x , that fundamental process will be represented by a range of vectors corresponding to different values of the control variable.

In this way, for example, temperature will cause the rate vector at a given point in concentration space to vary across a certain range. By choosing an optimal value of the temperature control variable it is possible to ensure that the fundamental process moves in such a way so as to extend the AR the furthest.

This will initially be illustrated by means of a simple two dimensional example, which will serve to introduce the underlying concepts that will be used in the theoretical development and subsequent examples.

Introduction to Control Variables - Two Dimensional van de Vusse Example

For the purposes of this example we consider the van de Vusse reaction scheme taking place in a steady flow reactor system, with a feed of a given flowrate and composition. The reactions for this system are:



The rate constants k_1 , k_2 and k_3 depend on temperature according to the Arrhenius form:

$$k_i = k_i^0 e^{\left(\frac{-E_i}{RT}\right)}$$

$$i = 1, 2, 3$$

The rate of formation of species D is second order with respect to the reactant A , whilst the rate of formation of B and C are both first order with respect to their reactants. The values of the constants used in the calculations are given in Table 1.

i	k_i^0	$\frac{E_i}{R}$ (K)
1	4	500
2	1.5	800
3	6	0

Table 1: Rate Constants for Example using Van de Vusse Kinetics

For a feed of pure A , it is of interest to determine all possible concentrations of A and B that are achievable using the fundamental processes of mixing and reaction at temperatures between 300 K and 1000 K. The attainable region will be found in the $\mathcal{R}^2$ space (x_1, x_2) , where x_1 and x_2 are the normalised concentrations of A and B respectively. The feedpoint, which corresponds to pure A , is thus given by $(1, 0)$.

The van de Vusse kinetics, written for this two dimensional formulation, give the reaction vector:

$$\mathbf{r} = \begin{pmatrix} -k_1^0 e^{\left(\frac{-E_1}{RT}\right)} x_1 - k_3^0 e^{\left(\frac{-E_3}{RT}\right)} x_1^2, & k_1^0 e^{\left(\frac{-E_1}{RT}\right)} x_1 - k_2^0 e^{\left(\frac{-E_2}{RT}\right)} x_2 \end{pmatrix} \quad [10]$$

Starting from the feedpoint and allowing only the fundamental processes of reaction and mixing between achievable compositions to occur, one can set about constructing the AR. The procedure to be followed is similar to that applied to the equivalent isothermal problem (Glasser et al. 1987), except that in this case the reaction vector can take on a range of values that at every point in the concentration space, depending on the temperature at which the reaction is carried out. Given the definition of the AR as the totality of all points (states) that can be achieved for a set of permitted fundamental processes and a range of allowed control variables which may be applied either to the fundamental

processes individually or to controlling the manner in which they are combined, it is interesting to choose that temperature which will result in the biggest possible AR.

For this two dimensional example it is required to find the AR of all possible concentrations of components A and B that can be achieved. Given that one is starting with pure A , it is obvious that (at least locally) the best temperature to choose would appear to be the one causing the reactor vector to point as much as possible in the direction of increasing B . In other words, the temperature is chosen so that the magnitude of the slope of the reaction vector is as large as possible. Using this guideline, it is possible to construct the PFR from the fresh feedpoint F as shown in Figure 1. At every point along the reactor the temperature is chosen so that the rate vector points "furthest outwards" with respect to the AR. This is achieved by operating the reactor at the maximum permissible temperature ($T_{max}=1000$ K) for the first portion, before the reaction temperature reduces gradually to the minimum permissible temperature ($T_{min}=300$ K), which is maintained for the last portion of the reactor. This trajectory is shown as curve FCO on Figure 1 and exhibits a clear concavity. This indicates that it does not fulfil the necessary conditions of the AR and that mixing would allow the region to be extended.

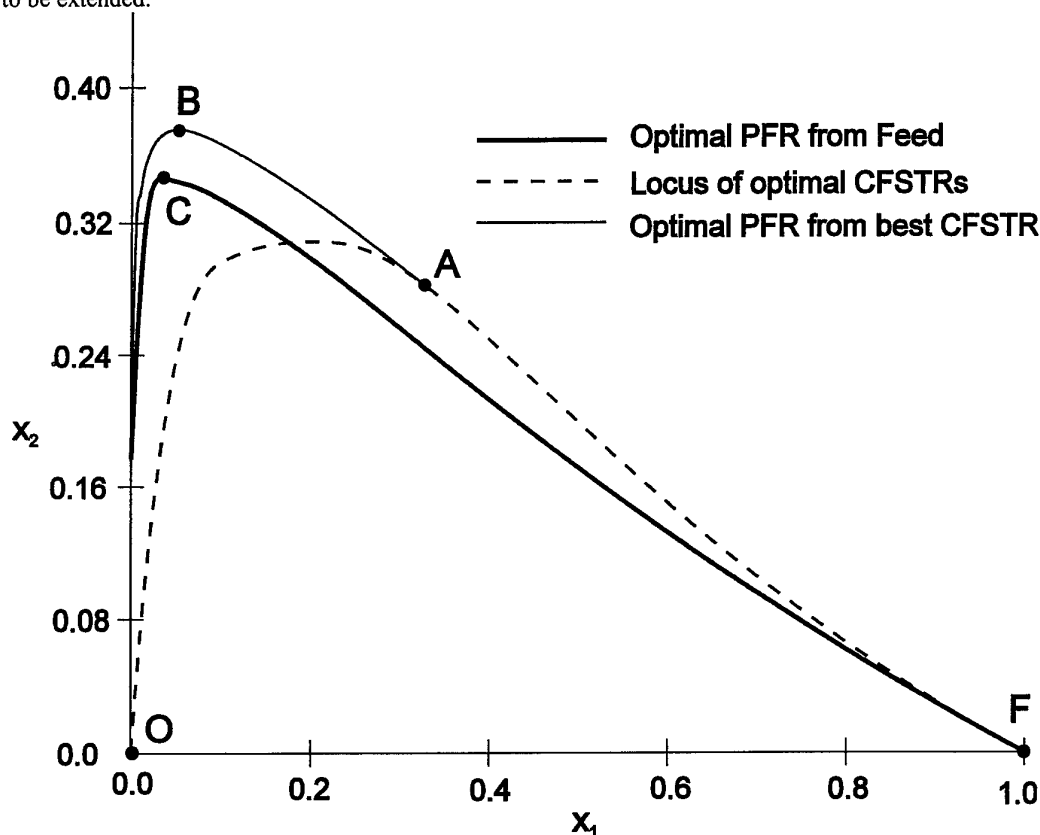


Figure 1: Construction of AR for van de Vusse Kinetics with variable Temperature

The locus of all optimal CSTR operating points, utilising pure A as a feed is thus plotted as FAO. Again the temperature is chosen so that the locus of operating points lies the furthest out. The optimal CSTR operates at point A , where the mixing vector is tangent to the CSTR locus and the temperature is $T=T_{max}=1000$ K. From this operating point a PFR reactor (trajectory) ABO is used which is again constructed by choosing the temperature so that the reaction vector always points the furthest outwards.

The final AR is shown in Figure 2. It consists of a CSTR operating at $T_{max}=1000$ K at point A . Concentrations along mixing line FA are achieved by bypassing feed around this reactor. The reactor product from the CSTR is fed to a PFR which achieves the concentrations along ABO . The temperature profile along the PFR is plotted in Figure 2, showing that the PFR is first operated at the maximum temperature, before the temperature is reduced progressively to the minimum temperature $T_{min}=300$ K.

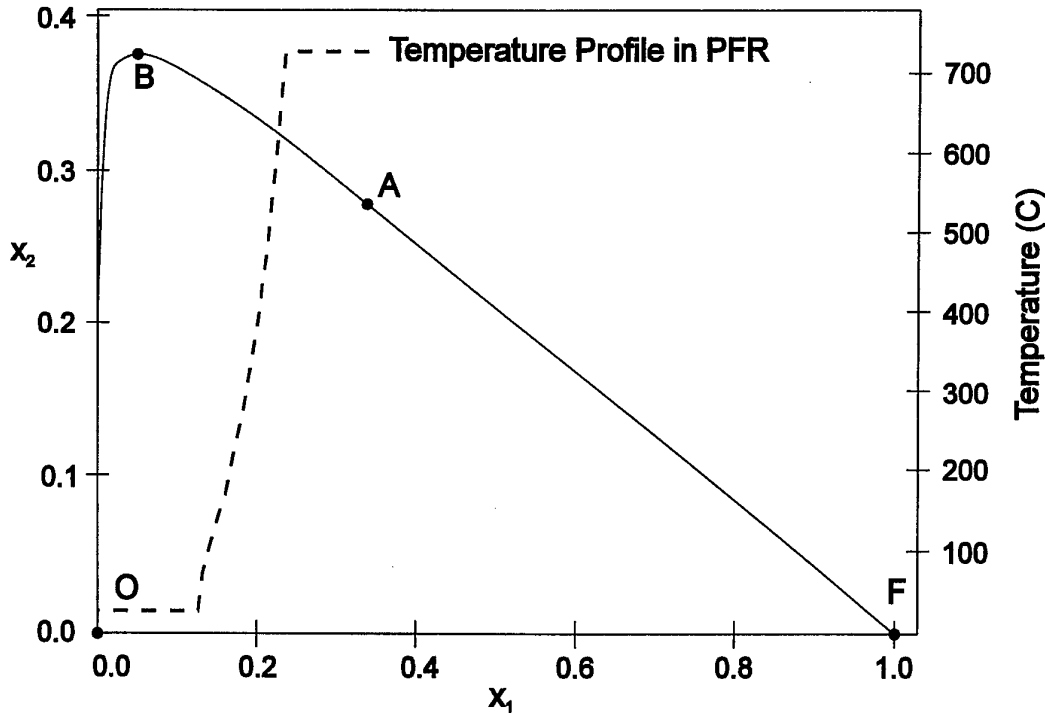


Figure 2: Attainable Region Two Dimensional van de Vusse Kinetics with Temperature Control

This example illustrates some of the underlying thought processes used in constructing the AR for a situation where the fundamental process vector may take on a range of values for a given set of state variable values. This requires that the control variable be chosen so that at every point in the boundary of the AR, the vector process that is operating cannot be made to point further outwards. We thus require that at every point on the boundary of the AR, that the rate vectors are either tangent, zero or point inwards. There cannot be a rate vector of non zero magnitude on the boundary that points outwards for any value of the control variable T . It would appear that two situations can be distinguished:

1. As the control variable is varied in the allowed range, the vector process may point further and further outwards relative to the local surface, but at some point may begin to point back into the region, i.e. point back towards points that have already been attained, instead of extending the region further. The control variable value for which the vector points the furthest outwards is characterised by a turning point condition relative to the local boundary surface. (This will be defined later.)
2. It may be found that the control variable value that causes the vector to point the furthest outward lies beyond the range of permitted values. In this case the permissible control variable value that extends the region the furthest must be chosen. This will in general correspond to one of the limits of the permitted control variable range.

Theoretical Development of General Control Policy

Consider an attainable region in $\mathcal{R}^3$ space:

$$\mathbf{x} \Rightarrow (x_1, x_2, x_3) \quad [11]$$

where the net process occurring along a given part of the boundary to the AR is given by $\mathbf{r}(\mathbf{x}, \theta)$. The boundary of the AR is a two dimensional surface that may be parameterised in terms of two parameters τ and s as shown in Figure 3. Here τ is the arclength of any single process trajectory, such as PQ. This follows from the definition:

$$\frac{\partial \mathbf{x}}{\partial \tau} = \mathbf{r}(\mathbf{x}, \theta) \quad [12]$$

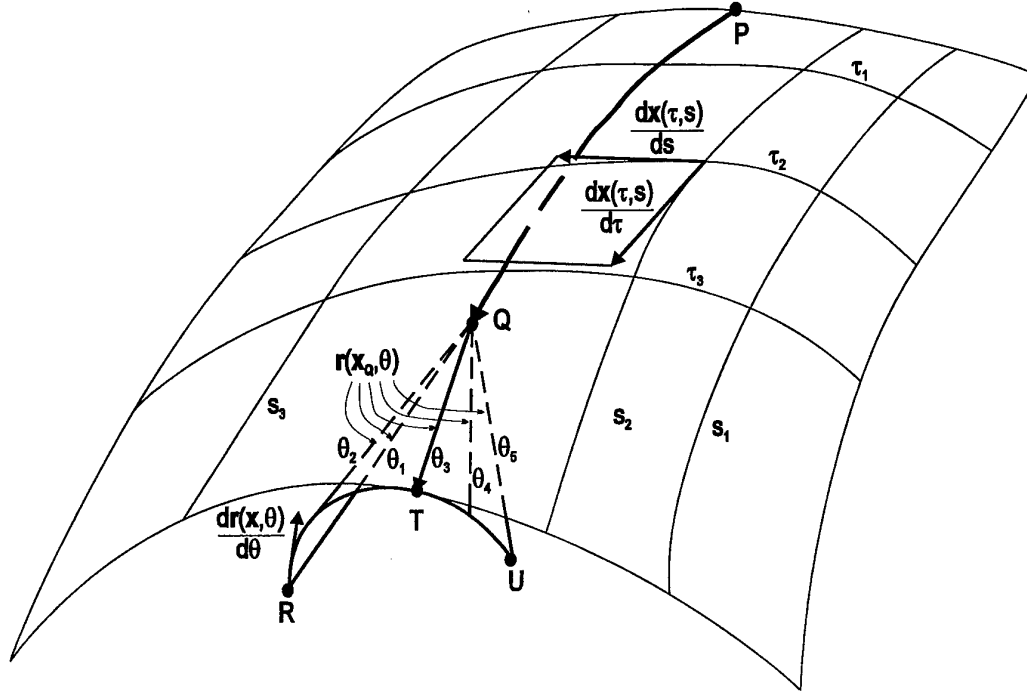


Figure 3: Schematic of AR Boundary showing Process depending on Control Variable

The second parameter, s , remains constant along any given trajectory and varies smoothly from one trajectory to the next. The tangent plane to the surface may be uniquely defined in terms of the derivatives with respect to these two parameters.

Consider the trajectory PQ lying in the boundary of the AR as shown in Figure 3. At point Q it is required to choose a value of the control variable θ that will extend the AR as far as possible, thus ensuring that the trajectory continues to lie in the boundary. As θ is allowed to vary in the permitted range for that control variable, the vector representing the net fundamental process at Q , $r(x_Q, \theta)$, varies as shown. Thus in a small time step any of the points along the locus RTU may be reached, depending on which value of the control variable is chosen. This locus is tangential to the boundary of the AR at point T . This tangency (which assumes that $\theta_{\min} < \theta < \theta_{\max}$) is described by:

$$\xi_1 = \left(r(x, \theta) \times \frac{\partial x(\tau, s)}{\partial s} \right) \cdot \frac{\partial r(x, \theta)}{\partial \theta} = 0 \quad [13]$$

This condition for tangency must hold at every point and thus for this condition to continue to remain true along the trajectory, we may differentiate :

$$\xi_2 = \frac{\partial \xi_1}{\partial \tau} = 0 \quad [14]$$

Using these conditions we can show (Godorr (1998)) that the general control policy is given by:

$$\frac{\partial \theta}{\partial \tau} = - \frac{\left(r(x, \theta) \times \frac{\partial r(x, \theta)}{\partial x} \frac{\partial x(\tau, \theta)}{\partial \theta} \right) \cdot \frac{\partial r(x, \theta)}{\partial \theta} + \left(r(x, \theta) \times \frac{\partial r(x, \theta)}{\partial \theta} \right) \cdot \frac{\partial}{\partial x} \left(\frac{\partial r(x, \theta)}{\partial \theta} \right) r(x, \theta)}{\left(r(x, \theta) \times \frac{\partial r(x, \theta)}{\partial \theta} \right) \cdot \frac{\partial^2 r(x, \theta)}{\partial \theta^2}} \quad [15]$$

It is important to note that this control policy does not give the actual value of the control variable as the process proceeds, but merely indicates how this value should change. This control policy is only relevant when the tangency

condition defined by [13] is satisfied. Once the bound on the control variable ($\theta_{\min} < \theta < \theta_{\max}$) becomes active, condition [13] is no longer satisfied and the control policy [15] may no longer be applied, as will become evident in the examples presented.

The initial value of the control variable for a particular trajectory is thus a critically important parameter that is determined by the overall structure of the AR. The following general situations may arise and will be illustrated in the ensuing examples:

1. At a given point on the boundary of the AR the control variable may be allowed to assume all values in the permissible range, $\theta_{\min} \leq \theta \leq \theta_{\max}$. For each initial value a unique process trajectory is defined with the subsequent control being governed by [15]. These trajectories will then form a surface, all or part of which may lie in the boundary of the AR.

As will be seen later, this situation is common when a new part of the boundary is to be generated from a single point. Referring back to Figure 3, one could imagine the boundary being generated from Q with the first part of the boundary consisting of the surface of adjacent vectors along RTU.

2. Depending on the shape of the AR, situations may arise where the control variable is maintained at the minimum or maximum permitted value for an extended distance along the trajectory, before the control policy defined by [15] is allowed to "take over". Such a situation may arise if the tangency condition [13] is not satisfied, initially. Alternatively, keeping the control variable at one of its bounds may allow the AR to be extended further in a particular direction.
3. Other structures in the AR may define what the initial value of the control variable should be.

Comparison to Horn's Control Policy

Horn (1961) developed a general expression for optimisation of chemical reactors in a space that can be described by

$$\mathbf{x} = (x_1, x_2, \tau)$$

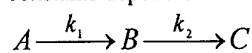
where τ represents the residence time in the reactor. This space can be used to solve an important set of three-dimensional problems that deal with minimising time or volume as part of the optimisation. This expression is identical to that obtained when [15] is expanded for the fundamental process vector $\mathbf{r} = (r_1, r_2, 1)$ in $\mathbf{x} = (x_1, x_2, \tau)$ space to give:

$$\frac{\partial}{\partial \tau} = \frac{\left\{ \frac{\partial^2}{\partial x_1^2} \left(\frac{\partial r_1}{\partial \theta} \right)^2 + \frac{\partial^2}{\partial \theta^2} \frac{\partial r_1}{\partial x_2} \frac{\partial r_2}{\partial \theta} - \frac{\partial^2}{\partial \theta^2} \frac{\partial r_1}{\partial x_1} \frac{\partial r_2}{\partial \theta} - \frac{\partial^2}{\partial x_2^2} \left(\frac{\partial r_2}{\partial \theta} \right)^2 + \right.}{\left. \frac{\partial^2 r_1}{\partial \theta \partial x_1} r_1 \frac{\partial r_2}{\partial \theta} + \frac{\partial^2 r_1}{\partial \theta \partial x_2} r_2 \frac{\partial r_2}{\partial \theta} - \frac{\partial^2 r_2}{\partial \theta \partial x_1} r_1 \frac{\partial r_1}{\partial \theta} - \frac{\partial^2 r_2}{\partial \theta \partial x_2} r_2 \frac{\partial r_1}{\partial \theta} \right\}}{\frac{\partial r_1}{\partial \theta} \frac{\partial^2 r_2}{\partial \theta^2} - \frac{\partial r_2}{\partial \theta} \frac{\partial^2 r_1}{\partial \theta^2}} \quad [16]$$

The usefulness of expression [16] in combination with the geometric approach to finding the AR will be illustrated by means of the well known example of successive reactions, which has been comprehensively studied by Coward & Jackson (1965). The more general control policy [15] has not been developed before and it will be applied in a subsequent example.

Optimising Temperature Profiles in Plug flow reactors

The reaction system studied to illustrate this example involves two successive first order reactions, where the rate constants depend on the temperature according to the Arrhenius form.



$$k_1 = k_1^0 e^{-\frac{E_1}{RT}}$$

$$k_2 = k_2^0 e^{-\frac{E_2}{RT}}$$

The constants used in this example are given in Table 2 and it is of interest to find the set of minimum reactor sizes to achieve all possible concentrations arising in this reaction system.

k_1^0	10 s^{-1}
k_2^0	800 s^{-1}
$\frac{E_1}{R}$	1000 K
$\frac{E_2}{R}$	4000 K

Table 2: Kinetic Constants for Successive First Order Reactions

Defining the normalised concentrations of components *A* and *B* as:

$$x_1 = C_A$$

$$x_2 = C_B$$

it follows that the concentration of *C* may always be found by mass balance. Because the optimisation is to involve a minimisation of residence time, the AR will be constructed in $\mathbf{x} = (x_1, x_2, \tau)$ space, where the reaction vector may be written as:

$$\mathbf{r} = \left(-k_1^0 e^{\left(\frac{-E_1}{RT}\right)} x_1, k_1^0 e^{\left(\frac{-E_1}{RT}\right)} x_1 - k_2^0 e^{\left(\frac{-E_2}{RT}\right)} x_2, 1 \right)$$

The permissible temperatures in the reactor system are limited to the range 400-1000 K.

The projection of the boundary of the AR for a feed composition given by:

$$x_1 = C_A = 0.9$$

$$x_2 = C_B = 0.1$$

onto (x_1, x_2) plane is shown in Figure 4. From the feedpoint P, the achievable compositions are bounded on one side by the trajectory PQO which corresponds to an isothermal PFR operated at the minimum allowable temperature of 400 K. Whilst this in general corresponds to the maximum amount of *B* that can be formed, the trajectory PUVWO gives the minimum amount of *B* and represents the compositions that are achieved in an isothermal PFR operated at the maximum allowable temperature of 1000 K. All compositions between these extremes are achieved by using a PFR with a temperature profile that can be calculated directly from condition [15]. Thus trajectory PMRO corresponds to a PFR with a feed temperature $T_0=600$ K and a temperature profile as shown in Figure 5.

Given a reactor with a residence of time of $\tau=1$, the maximum concentration of *B* that can be achieved is given by point M, which will be reached if the feed is heated to a temperature of 600 K and the temperature profile governed by condition [16] is followed. Once the AR has been constructed for a problem such as this, a number of different optimisations can be trivially performed. Thus it is clear that for a PFR whose feed is heated to 600 K, the maximum concentration of *B* will correspond to point R, if the smallest possible reactor is to be used. Similar composition trajectories and temperature profiles are illustrated for feed temperatures of $T_0=800$ K (PSO) and $T_0=1000$ K (PTO). In both these cases the optimal temperature profile (that leads to the smallest possible reactor) decreases smoothly from the feedpoint.

Compositions in the region PTOWVUP are achieved in the smallest reactor if there exists an initial segment where the reactor is operated at the maximum temperature ($T=1000$ K), after which the temperature is smoothly decreased as shown for PUVWYO. Temperature profiles such as that illustrated by PUVWYO in Figure 5, do not follow from a blind application of [16], but result from the actual construction of the entire AR. During this process condition [16] merely acts as a tool to assist in the appropriate the choice of continuous control policy.

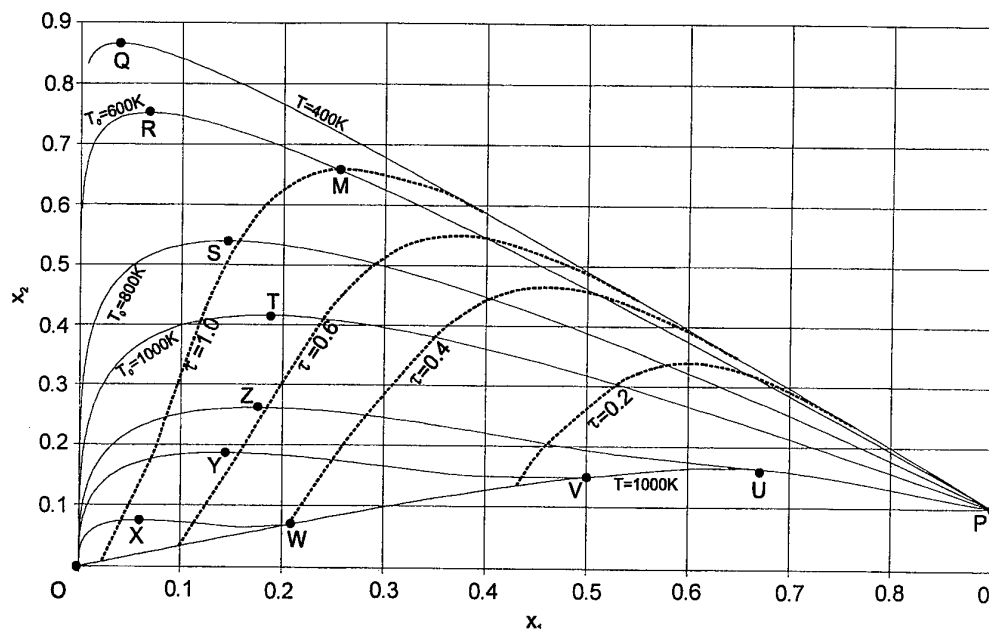


Figure 4: Attainable Region for Two Successive Reactions with Temperature Control

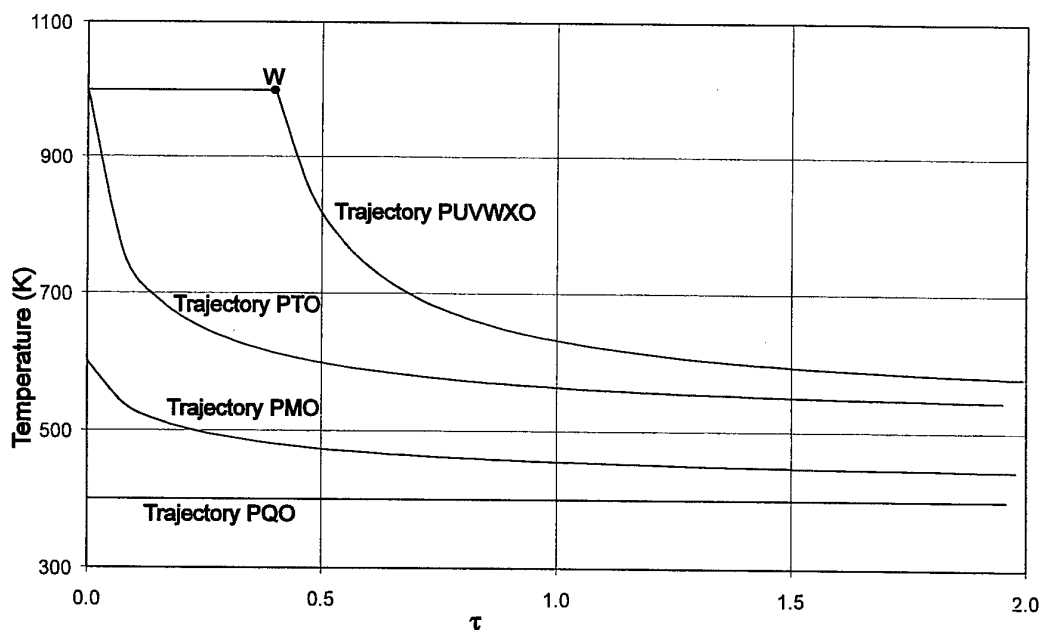


Figure 5: Axial Temperature Profiles in PFRs for Successive Reaction Problem

The trajectory PUVWO is concave and this part of the boundary is filled in with mixing lines. These have not been shown on Figure 4 as this region of the diagram would correspond to low selectivity to B and is thus not particularly interesting. This problem has been solved using other optimisation techniques and these techniques would not have been able to find the mixing region in PUVWO. The advantage offered by the AR approach as presented here is that this constructive approach allows the overall structure of the problem to be determined (including the mixing lines in region PUVWO), so that optimisations can be easily performed subsequently. In order to explore other chemical reaction examples that utilise the control policy [15], it is useful to integrate it with the existing AR conditions governing simultaneous reaction and mixing.

Simultaneous Mixing and Reaction with Temperature Control

The classic VdelR condition [8], holds at all points that are candidates for the simultaneous occurrence of the fundamental processes of reaction and mixing in the boundary of the AR. This condition may be rewritten to allow the reaction process to depend on some control variable (generally temperature):

$$\begin{aligned} \varphi_1(\mathbf{x}, \theta) &= (\mathbf{r}(\mathbf{x}) \times \mathbf{v}(\mathbf{x}, \mathbf{x}_m)) \cdot \frac{\partial \mathbf{r}(\mathbf{x}, \theta)}{\partial \mathbf{x}} \mathbf{v}(\mathbf{x}, \mathbf{x}_m) \\ &= 0 \end{aligned} \quad [17]$$

This condition will hold along any DSR lying in the boundary of the AR and will govern how the fundamental processes of reaction and mixing are combined. Since the reaction vector is not uniquely defined at every point, $\mathbf{x}$, condition [17] no longer defines a unique surface, but a volume of space. For a given temperature, the reaction vector is single valued at each point and condition [17] would again represent a two dimensional surface. Thus the volume may be viewed as consisting of all single temperature surfaces in the permitted range of the temperature control variable, $\theta_{\min} \leq \theta \leq \theta_{\max}$. If we are free to choose the temperature in the permissible range, a second condition is required to define that single surface lying within this volume. This intersects the boundary of the AR along a unique line that then becomes a candidate for an optimal DSR trajectory lying in the boundary.

Since the reaction vector depends on temperature, it becomes important that the optimal axial temperature profile be followed along the DSR. As in the derivation of condition [15], we require that the vector $\frac{\partial \mathbf{r}(\mathbf{x}, \theta)}{\partial \theta}$ lies in the tangent plane to the AR so that any change of the temperature causes the reaction vector to point into and not out of the AR. Along the DSR trajectory, this tangent plane is spanned by the reaction and mixing vectors, so that we may write the second condition:

$$\begin{aligned} \varphi_2(\mathbf{x}, \theta) &= (\mathbf{r}(\mathbf{x}, \theta) \times \mathbf{v}) \cdot \frac{\partial \mathbf{r}(\mathbf{x}, \theta)}{\partial \theta} \\ &= 0 \end{aligned} \quad [18]$$

(This is the analogous condition to that described by equation [13] in the previous derivation and again assumes that the control variable θ lies in the permitted range $\theta_{\min} < \theta < \theta_{\max}$.)

Simultaneous solution of [17] and [18] yields a two dimensional surface consisting of all DSR trajectories with optimal temperature profiles that are candidates for lying in the boundary of the AR. These conditions must remain true along the DSR trajectory, so that we may differentiate with respect to the residence time (length) parameter, τ , and set this equal to zero in each case:

$$\begin{aligned} d\varphi_1 &= \frac{\partial \varphi_1}{\partial x_1} dx_1 + \frac{\partial \varphi_1}{\partial x_2} dx_2 + \frac{\partial \varphi_1}{\partial x_3} dx_3 + \frac{\partial \varphi_1}{\partial \theta} d\theta \\ \frac{\partial \varphi_1}{\partial \tau} &= \left(\frac{\partial \varphi_1}{\partial x_1}, \frac{\partial \varphi_1}{\partial x_2}, \frac{\partial \varphi_1}{\partial x_3} \right) \cdot \left(\frac{\partial x_1}{\partial \tau}, \frac{\partial x_2}{\partial \tau}, \frac{\partial x_3}{\partial \tau} \right) + \frac{\partial \varphi_1}{\partial \theta} \frac{\partial \theta}{\partial \tau} \\ &= \nabla \varphi_1 \cdot (\mathbf{r}(\mathbf{x}, \theta) + \alpha \mathbf{v}) + \frac{\partial \varphi_1}{\partial \theta} \frac{\partial \theta}{\partial \tau} \\ &= 0 \end{aligned} \quad [19]$$

$$\begin{aligned} \frac{\partial \varphi_2}{\partial \tau} &= \nabla \varphi_2 \cdot (\mathbf{r}(\mathbf{x}, \theta) + \alpha \mathbf{v}) + \frac{\partial \varphi_2}{\partial \theta} \frac{\partial \theta}{\partial \tau} \\ &= 0 \end{aligned} \quad [20]$$

Equations [19] and [20] may then be used to solve for the control variables along the DSR trajectory, i.e. the mixing policy and temperature profile along the DSR trajectory may then be found:

$$\alpha = - \frac{\left(\frac{\partial \varphi_2}{\partial \theta} \nabla \varphi_1 - \frac{\partial \varphi_1}{\partial \theta} \nabla \varphi_2 \right) \bullet \mathbf{r}(\mathbf{x}, \theta)}{\left(\frac{\partial \varphi_2}{\partial \theta} \nabla \varphi_1 - \frac{\partial \varphi_1}{\partial \theta} \nabla \varphi_2 \right) \bullet \mathbf{v}} \quad [21]$$

$$\frac{\partial \theta}{\partial \tau} = -\nabla \varphi_2 \bullet \frac{\left[\left(\frac{\partial \varphi_2}{\partial \theta} \nabla \varphi_1 - \frac{\partial \varphi_1}{\partial \theta} \nabla \varphi_2 \right) \bullet \mathbf{v} \right] \mathbf{r} - \left[\left(\frac{\partial \varphi_2}{\partial \theta} \nabla \varphi_1 - \frac{\partial \varphi_1}{\partial \theta} \nabla \varphi_2 \right) \bullet \mathbf{r} \right] \mathbf{v}}{\frac{\partial \varphi_2}{\partial \theta} \left(\frac{\partial \varphi_2}{\partial \theta} \nabla \varphi_1 - \frac{\partial \varphi_1}{\partial \theta} \nabla \varphi_2 \right) \bullet \mathbf{v}} \quad [22]$$

As for the control policy [15], policies [21] and [22] hold only for $\theta_{\min} < \theta < \theta_{\max}$, so that when a bound on θ becomes active, the tangency condition [18] is no longer satisfied.

It is important to note that CSTR locus points, where the mixing and reaction vectors are collinear, represent a degenerate solution to [17] and [18] because the cross product gives the null vector. CSTR operating points in the permitted range of operating temperatures, will form a surface (or set of surfaces in the case of multiple solutions) that are automatically found when solving these conditions. These points represent the stationary points in which DSR trajectories terminate.

However we must also consider CSTR points as initial points for DSR or PFR trajectories. In this situation we cannot use equations [17] and [18], and we need additional conditions. In this situation we require that the tangent to the CSTR

locus $\mathbf{t}(\mathbf{x}, \theta)$ must be coplanar with the mixing (or reaction) vector and the vector $\frac{\partial \mathbf{r}(\mathbf{x}, \theta)}{\partial \theta}$ i.e.:

$$\varphi_3(\mathbf{x}, \theta) = (\mathbf{r}(\mathbf{x}, \theta) \times \mathbf{t}(\mathbf{x}, \theta)) \bullet \frac{\partial \mathbf{r}(\mathbf{x}, \theta)}{\partial \theta} = 0 \quad [23]$$

where $\mathbf{t}(\mathbf{x}, \theta)$ follows from the differentiation of the equation governing the existence of a CSTR, with respect to residence time, τ :

$$\mathbf{t}(\mathbf{x}, \theta) = \left[I - \tau \frac{\partial \mathbf{r}(\mathbf{x}, \theta)}{\partial \mathbf{x}} \right]^{-1} \mathbf{r}(\mathbf{x}, \theta) \quad [24]$$

Three Dimensional - van de Vusse Kinetics

If we consider the van de Vusse kinetics that were used in the introductory example and, using the same kinetic constants, construct the AR in $\mathcal{R}^3$ space:

$$\mathbf{x} = (x_1 \quad x_2 \quad x_3)$$

where we define the normalised concentrations of the species as:

$$C_A = x_1$$

$$C_B = x_2$$

$$C_D = x_3$$

The reaction vector in this $\mathcal{R}^3$ space is then written as:

$$\mathbf{r} = \left(-k_1^0 e^{\left(\frac{-E_1}{RT}\right)} x_1 - k_3^0 e^{\left(\frac{-E_3}{RT}\right)} x_1^2, \quad k_1^0 e^{\left(\frac{-E_1}{RT}\right)} x_1 - k_2^0 e^{\left(\frac{-E_2}{RT}\right)} x_2, \quad \frac{1}{2} k_3^0 e^{\left(\frac{-E_3}{RT}\right)} x_1^2 \right) \quad [25]$$

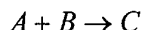
The general properties of the AR for isothermal van de Vusse kinetics has been rigorously studied (Feinberg and Hildebrandt, 1997) and the boundary is found to contain two connectors corresponding to a CSTR locus and a DSR trajectory.

If we now consider temperature to be a control variable, we find that the only solutions that conditions [17] and [18] have within the stoichiometrically permissible subspace (using the reaction vector [25] and mixing with the feedpoint) is the degenerate solution corresponding to CSTR operating points. The fact that both conditions could not be satisfied simultaneously for non-collinear reaction and mixing vectors indicates that there will be no DSR trajectory in the boundary of the AR where the temperature varies along the length of the reactor. This is an important result, since it means that the possibility of such a structure can be immediately eliminated. Thus any DSR trajectory in the AR boundary will correspond to an isothermal reactor. It is also found that the CSTR locus that extends the AR the most is in fact the locus operating at the maximum temperature, $T = 1000\text{K}$.

In the next example we find a DSR trajectory that lies in the boundary of the AR along which the temperature has to be varied to obtain the optimal performance.

Example Illustrating a DSR with Temperature Profile

We consider a system of chemical reactions whereby substance A reacts to form B and these two species in turn combine to form C as shown by:



The state of the chemical mixture is fully described, either by the concentrations of A and B or by B and C , so that we choose to define the normalised concentration as:

$$C_B = x_1$$

$$C_C = x_2$$

so that by mass balance :

$$C_A = 1 - x_1 - x_2$$

The rate of formation of B is first order with respect to the concentration of A , whilst the rate of formation of C is proportional to the concentrations of A and B . Species A however inhibits the rate of formation of C when present in higher concentrations, so that the rate of formation of B is written as:

$$r_B = k_1 C_A - \frac{k_2 C_A C_B}{1 + k_3 C_A^2} \quad [26]$$

In this example it will be of interest to determine the smallest possible reactors to achieve various outlet compositions from the reactor system. The AR will therefore be constructed in (x_1, x_2, τ) space, where the reaction vector may be written as:

$$\mathbf{r} = \left(k_1 e^{\frac{-E_1}{RT}} (1 - x_1 - x_2) - \frac{k_2 e^{\frac{-E_2}{RT}} (1 - x_1 - x_2) x_1}{1 + k_3 e^{\frac{-E_3}{RT}} (1 - x_1 - x_2)^2}, \quad \frac{k_2 e^{\frac{-E_2}{RT}} (1 - x_1 - x_2) x_1}{1 + k_3 e^{\frac{-E_3}{RT}} (1 - x_1 - x_2)^2}, \quad 1 \right) \quad [27]$$

The kinetic parameters used in this example are given in Table 3 and the reactor temperatures are limited to the range 298-973 K.

k_1^0	2 s^{-1}
k_2^0	10 s^{-1}
k_3^0	10 s^{-1}
$\frac{E_1}{R}$	30 K
$\frac{E_2}{R}$	60 K
$\frac{E_3}{R}$	1200 K

Table 3: Kinetic Constants for Example illustrating DSR with Temperature Profile

In this example we consider a process where all the material *B* is added initially and a stream containing pure *A* may be mixed into the reaction mixture at any point. The mixing vector which describes the addition of fresh feed consisting of pure *A* is:

$$\mathbf{v} = (-x_1, -x_2, -\tau) \quad [28]$$

For such a system it is obvious that any mixture of *A* and *B* that contains no *C*, can be achieved immediately simply by mixing the available streams. This anchors the AR at zero residence time along the $x_2=0$ axis.

In order to test for the possibility of mixing (described by [28]) and reaction (described by [27]) with temperature control occurring simultaneously in the boundary of the AR, conditions equations [17] and [18] are solved simultaneously. This yields a surface of DSR trajectories with axial temperature variation that are candidates for lying in the boundary of the AR and which lie within the stoichiometrically feasible subspace. This surface is found to intersect the x_1 axis ($x_2=0, \tau=0$) at $x_1=0.52631$, where the temperature may be calculated to be 326.45 K. (This composition and temperature correspond to the maximum rate of production of component *C*, as may be determined with ordinary differential calculus.) As this point is immediately achievable by appropriate mixing of the available feed material and therefore lies in the boundary of the AR, it represents a strong candidate as a feedpoint to a DSR trajectory.

Since the various tangency conditions are satisfied at the feedpoint, the remaining DSR trajectory can be constructed by integrating the appropriate differential equations, using the control policies defined by [21] and [22]. These ensure that the tangency conditions remain true along the length of the reactor. Figure 6 shows the two dimensional projection of the AR constructed for this example, where the feedpoint to the DSR trajectory is given by P and the actual compositions achieved in this reactor lie along PR. The temperature increases monotonically along the trajectory and until it reaches the maximum allowable temperature of 973 K. The temperatures in the boundary of the AR are illustrated in Figure 7. Once the maximum temperature has been reached, the problem reduces to a simple isothermal problem, where the classic VdelR condition applies again and the second tangency condition [18] no longer holds. It is important to check however that the isothermal DSR trajectory does not re-intersect the surface where both conditions [17] and [18] hold (at a permissible temperature), since this would suggest that the temperature should be varied again to extend the AR further.

The fan of dashed mixing lines with the feedpoint O represent the mixing lines with pure A. From the DSR trajectory

PR, a family of PFR trajectories is generated. Since the $\frac{\partial \mathbf{x}(\mathbf{x}, \theta)}{\partial \theta} \left(= \frac{\partial \mathbf{x}(\mathbf{x}, T)}{\partial T} \right)$ vector lies in the tangent to the AR

along the DSR trajectory (condition [18] is satisfied) it means that tangency condition [13] is satisfied at the feed to PFR trajectories and that the control policy [15] may be applied immediately. Along all the PFR trajectories the temperatures increase monotonically, until the maximum temperature is reached. At this point the temperature is held constant at this value and tangency condition [13] is no longer satisfied. By using adjacent PFR trajectories to find a numerical approximation to the tangent plane to the AR boundary, it is possible to confirm that temperature variation is not required again, since the tangency condition is never achieved again in the allowable temperature range.

The PFR trajectories originating from the last section of the DSR trajectory, where the temperature is at its maximum are best operated isothermally ($T = 973 \text{ K}$).

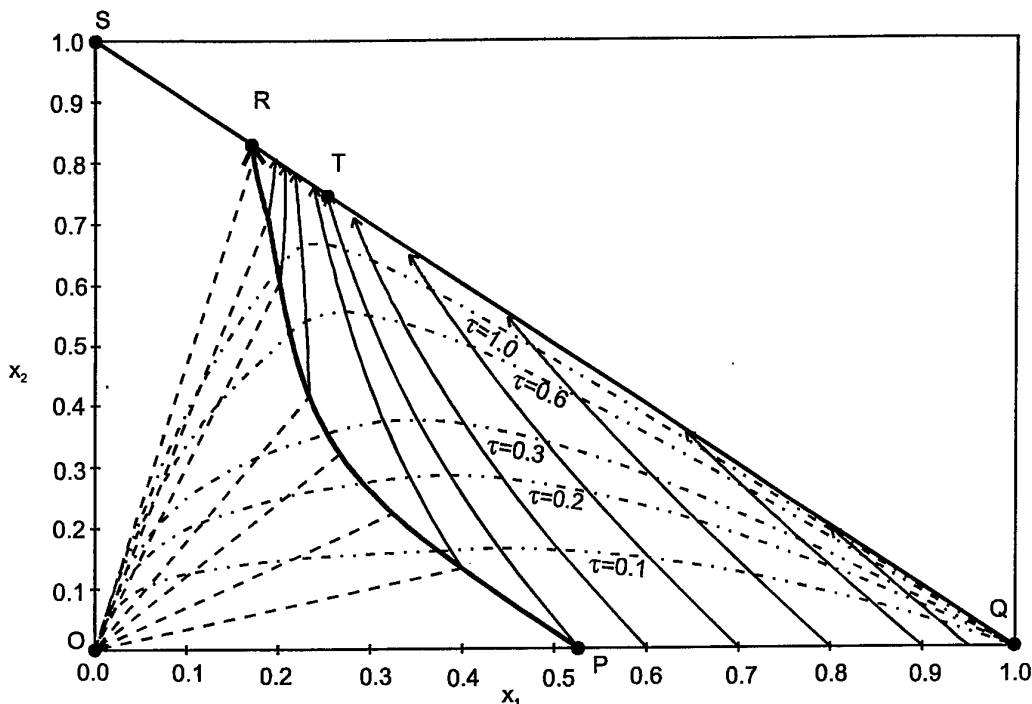


Figure 6: Contour Plot of Residence Time in AR Boundary for Example illustrating DSR with Temperature Profile

PFR trajectories that make up the remaining part of the AR boundary QPTQ, originate along the x_1 axis in the region PQ. In order to determine the optimal feed temperature to these reactors it is intuitively obvious that the rate of formation of component C should be maximised, by varying the temperature in the appropriate range. Whilst this may be trivially accomplished by differentiating the second term of the vector expression [27] with respect to temperature, the problem can also be recast in AR parlance. The feed temperatures to the PFRs originating along PQ is determined recognising that along this line we may write:

$$\frac{\partial \mathbf{x}(\tau, s)}{\partial s} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

At each point along PQ it is then possible to write tangency condition [13] as a function of temperature only, so that the optimal PFR feed temperature may be determined. The temperature control policy along the trajectory is then given by equation [15], since the tangency condition holds at the feedpoint of the trajectory.

For a small section of PQ, the optimal feed temperature lies above the upper bound of allowable values as shown in Figure 7. The maximum permissible temperature was found to extend the AR the furthest in this region. This means however that tangency condition [13] is not satisfied at the feedpoints of the PFR trajectories and that the control policy [15] cannot be applied at these points. It is found that operating these PFR trajectories at the maximum temperature places them in the boundary of the AR.

Conclusions

The AR approach to optimisation has been expanded to allow for the optimal selection of a control variable that affects how a single fundamental process operates at every point in the boundary. In this situation it is no longer possible to define a unique fundamental process vector for a given set of state variables because the vector magnitude and/or direction for a given point are intrinsically dependent on the control variable. The development of the control policy is based on local tangency conditions which ensure that any change in the fundamental process vector due to a change in the control variable causes this vector to move in the tangent plane to the boundary of the AR. It is possible to define a tangency condition which ensures that this control variable is always chosen to give the biggest possible AR. This condition has allowed the AR approach to easily incorporate the important field of temperature control in chemical reactors and shows great potential for application of the AR to other optimisation problems. The control policy developed via the AR approach is more general than any previously developed policy for the optimal choice of temperature profiles in chemical reactors.

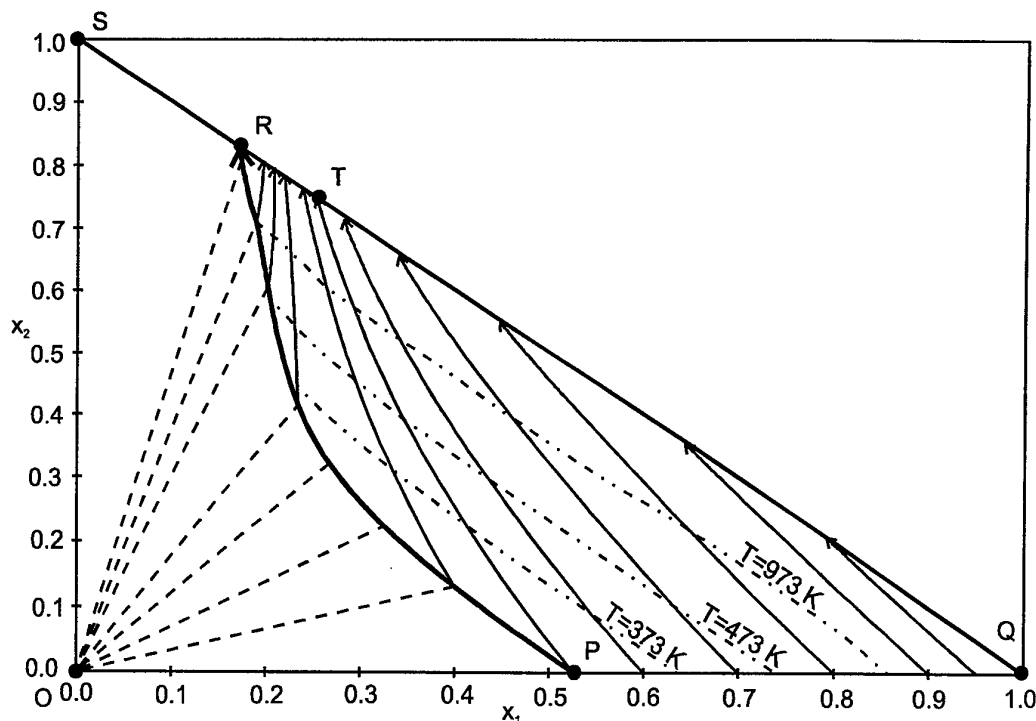


Figure 7: Temperatures in Boundary of AR for Example illustrating DSR with Temperature Profile

The underlying principle of the AR is that it finds the totality of all states that can be achieved by a system that is subject to certain dynamic constraints. By undertaking this exhaustive exploration of the solution space, prior to performing any optimisations, this approach has been shown to offer distinct advantages when dealing with optimisations that involve discontinuities in the state variables or take advantage of states achieved elsewhere in the system to arrive at the desired final state in a more efficient (possibly indirect) manner. The formulations that have been used in this paper have been general in nature, so that it should be easy to apply this approach to non-chemical engineering problems, such as the brachistochrone problem which has been studied by Godorr (1998). Systems where jumps are possible (due to multiple solutions) or where mixing rules apply to the state variables may particularly benefit from this geometric methodology.

The major drawback of the AR approach would appear to lie in the problem of dimensionality. Large optimisations (those involving many state variables) will require the AR to be constructed in a higher dimensional space, which becomes computationally expensive. The conditions, control policies and overall solution structures that are found in the lower dimensional problems, do however offer powerful guidelines and tools that can be used in conjunction with other optimisation techniques. For example, the conditions presented in this paper can be used to determine what sort of interconnections should be allowed in the superstructure defined at the outset of the process synthesis approach. The AR analysis therefore provides a framework overview of what structures can be expected in the final optimal solution and defines control policies that will be required to achieve optimal performance in the various reactors.

Nomenclature

E	activation energy of chemical reaction
$\mathbf{f}$	vector of functions governing the dynamic behaviour of a system
I	measure of system performance, generally a cost function
k	chemical reaction rate constant
k_0	chemical reaction pre-exponential rate constant
L	Lagrangian function
$\mathbf{p}$	vector describing overall process
R	universal gas constant
r_i	components of vector $\mathbf{r}$
$\mathbf{r}$	vector describing a single fundamental process (generally a given reaction)
$\mathbf{u}$	vector of control variables
$\mathbf{v}$	vector describing mixing process

x_i	state variable of a system
$\mathbf{x}$	vector of state variables
T	temperature
$\mathbf{t}$	tangent vector to CSTR locus
t	independent variable (generally time)
τ	independent variable (generally residence time)
α	scalar value $\alpha \geq 0$
φ	defined by equation [8]
φ_i	for $i = 1, \dots, 3$ defined by equations [17], [18] and [23] respectively
$\mathcal{G}$	scalar parameter
ξ_i	for $i = 1, 2$ defined by equations [13] and [14] respectively

Subscripts

m	mixing point
n	total number of elements in a set
i	index referring to elements of a given set
0	initial value or state
<i>initial</i>	initial value or state
<i>final</i>	final value or state
<i>min</i>	minimum permissible value
<i>max</i>	maximum permissible value

References

- Coward, I. & Jackson, R., 1965, *Optimum temperature profiles in tubular reactors : an exploration of some difficulties in the use of Pontryagin's Maximum Principle*, Chem. Eng. Sci., **20**, pp 911-920.
- Horn, F., 1961, *Optimale Temperatur- und Konzentrationsverläufe*, Chem. Eng. Sci., **14**, pp 77-89.
- Horn, F., 1964, *Attainable and non-attainable regions in chemical reaction technique*, Proc. Third European Symposium on Chemical Reaction Engineering, Pergamon, New York, pp 1-10.
- Feinberg, M. & Hildebrandt, D., 1992, presented at 1992 AIChE Annu. Meet., Miami, FL, paper 142C
- Feinberg, M. & Hildebrandt, D., 1997, *Optimal Reactor Design from a Geometric Viewpoint : I. Universal Properties of the Attainable Region*, Chem. Eng. Sci., **52**, (10), pp 1637-1665.
- Glasser, B., Hildebrandt, D. & Glasser, D., 1992, *Optimal Mixing for Exothermic Reversible Reactions*, I.&E.C. Research, **31**, pp 1541-1549
- Glasser, B., Hildebrandt, D. & Glasser, D., 1992, *Optimal Mixing for Exothermic Reversible Reactions*, I.&E.C. Research, **31**, pp 1541-1549
- Glasser, D., Hildebrandt, D. & Crowe, C.M., 1987, *A Geometric Approach to Steady Flow Reactors : The Attainable Region and Optimization in Concentration Space*, I.&E.C. Research, **26**, pp 1803-1810
- Godorr, S.A. (1998) Ph.D Thesis to be submitted to the University of the Witwatersrand, Johannesburg
- Hildebrandt, D., Glasser, D. & Crowe, C.M., 1990, *Geometry of the Attainable Region Generated by Reaction and Mixing : With and without Constraints*, I.&E.C. Research, **29**, pp 49-58
- Hildebrandt, D. & Glasser, D., 1990, *The Attainable Region and Optimal Reactor Structures*, Chem. Eng. Sci., **45**, (8), pp 2161-2168
- Hopley, F., Glasser, D. & Hildebrandt, D., 1996, *Optimal Reactor Structures for Exothermic Reversible Reactions with Complex Kinetics*, Chem. Eng. Sci., **51**, (10), pp 2399-2407.
- Omtveit, T., Tanskanen, J. & Lien, K.M., 1993, *Graphical Targeting Procedures for Reactor Systems*, Proc. Escape-3 Conf., Graz, July 1993

THE DYNAMIC-Q OPTIMIZATION METHOD: AN ALTERNATIVE TO SQP?

J.A. Snyman and A.M. Hay

Multidisciplinary Design Optimization Group
Department of Mechanical Engineering
University of Pretoria
Pretoria 0002
South Africa
Tel: +27 12 4203165 Fax: +27 12 3625087
Email: jan.snyman@eng.up.ac.za

Abstract

In this paper a constrained optimization method, called the Dynamic-Q method, is presented. Simply stated, the method consists of applying an existing dynamic trajectory optimization algorithm to successive spherical quadratic approximations of the actual optimization problem. The Dynamic-Q algorithm has the advantage of having minimal storage requirements, thus making it suitable for problems with large numbers of variables. The Dynamic-Q method is tested and results obtained are compared to results for a sequential quadratic programming (SQP) method. Indications are that the new method is robust and efficient, and particularly well suited to practical engineering optimization problems.

1. Introduction

An efficient constrained optimization method is presented in this paper. The method, called the Dynamic-Q method, consists of applying a DYNAMIC trajectory optimization algorithm to successive Quadratic approximations of the actual optimization problem.

Due to its efficiency with respect to the number of function evaluations required for convergence, the Dynamic-Q method is primarily intended for optimization problems where function evaluations are expensive. Such problems occur frequently in engineering applications where time consuming numerical simulations may be used for function evaluations. Amongst others, these numerical analyses may take the form of a computational fluid dynamics (CFD) simulation, a structural analysis by means of the finite element method (FEM) or a dynamic simulation of a multibody system. Because these simulations are usually expensive to perform, and because the relevant functions may not be known analytically, standard classical optimization methods are normally not suited to these types of problems. Also, as will be shown, the storage requirements of the Dynamic-Q method are minimal. No Hessian information is required. The method is therefore particularly suitable for problems where the number of variables n is large.

In the next section of this paper sequential quadratic programming (SQP) methods are briefly discussed to allow for comparison with the proposed method. Next, the Dynamic-Q methodology is presented as well as the dynamic trajectory “leap-frog” algorithm, which is used for solving the quadratic subproblems. Finally the Dynamic-Q method is tested and its performance compared to that of an SQP method.

2. Sequential quadratic programming methods

Sequential quadratic programming (SQP) methods have been developed over the past thirty years, and are generally considered to be some of the most efficient algorithms available today. Based on Lagrangian methods, it can be shown that the solution $\mathbf{x}^*$ of the nonlinearly equality constrained optimization problem

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}); \quad \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathcal{R}^n \\ \text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned} \quad (1)$$

where $f(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are respectively a scalar and a vector function of $\mathbf{x}$, can be obtained by solving, at successive approximations $\mathbf{x}^i$ to $\mathbf{x}^*$, a sequence of corresponding quadratic programming (QP) subproblems ($QP[i]$, $i=0,1,2,\dots$) containing linearized constraints and of the following form:

$$\begin{aligned} \min_{\mathbf{s}} f(\mathbf{x}^i) + \nabla^T f(\mathbf{x}^i) \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{W}^i \mathbf{s} \\ \text{subject to } \nabla^T \mathbf{h}(\mathbf{x}^i) \mathbf{s} + \mathbf{h}(\mathbf{x}^i) = \mathbf{0} \end{aligned} \quad (2)$$

where $\mathbf{W}^i = \nabla^2 f(\mathbf{x}^i) + \lambda^{iT} \nabla^2 \mathbf{h}(\mathbf{x}^i)$, with λ^i denoting the associated vector of Lagrange multipliers. The solution to subproblem $QP[i]$ is denoted by $\mathbf{s}^i$ and the point at which the next subproblem $QP[i+1]$ is constructed is $\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{s}^i$. If successful, the SQP method yields a sequence $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots$ that converges to $\mathbf{x}^*$. The particular QP subproblem given here is one of a number of possible forms that may be chosen.

Based on the above argument, a simple SQP algorithm is as follows (Papalambros and Wilde [1]):

Simple SQP algorithm

1. Select initial point $\mathbf{x}^0$ and initial Lagrange multipliers λ^0 . Set $i:=0$.
2. Solve the quadratic programming subproblem $QP[i]$ corresponding to (2) to determine $\mathbf{s}^i$ and λ^{i+1} .
3. Set $\mathbf{x}^{i+1} := \mathbf{x}^i + \mathbf{s}^i$.
4. If termination criteria are satisfied, stop; else set $i:=i+1$ and go to Step 2.

Numerous authors have proposed modifications and variations to the above basic algorithm. There are four areas in which the differences are most prominent. The first of these is the way in which *inequality constraints* are also included in the algorithm. For optimization problems containing inequality constraints an active set strategy may be used. This strategy can be implemented in one of two ways, either on the original problem or by including all of the inequality constraints in the QP subproblem, and applying an active set strategy to the subproblem. The second point of

difference lies in the way the QP subproblem is solved. Almost any method for nonlinear programming, such as the augmented Lagrangian method or the dual method, may be specially adapted to the solution of the QP subproblem. A third way in which SQP algorithms differ from each other is in the computation of second derivatives of the problem. In the above simple SQP algorithm it is necessary to evaluate the second derivatives of the objective function and the constraints in the computation of W^i , which will usually be a computationally intensive process. In any event, the storage of Hessian information is required which implies the availability of $O(n^2)$ storage locations, and the determination and manipulation of the elements of the $n \times n$ Hessian matrix. Some authors have avoided the latter difficulties by applying quasi-Newton updating formulae to approximate the second derivatives. Powell [2], for example, has proposed using the BFGS formula to approximate these second derivatives. A fourth point of difference lies in dealing with the feasibility or infeasibility of the constructed subproblems. If the QP subproblem (2) is constructed at a point far from the solution x^* of the constrained optimization problem (1), then the subproblem may have an unbounded or infeasible solution. For this reason many modern SQP algorithms rather use s^i as a search direction. Then the point x^{i+1} at which the next subproblem is constructed is set at $x^{i+1} = x^i + \alpha_i s^i$ with the step size α_i determined by performing a line search on an appropriate merit function in the direction s^i .

3. The Dynamic-Q method

Consider the general nonlinear optimization problem:

$$\begin{aligned} \min_x f(x); \quad x &= (x_1, x_2, \dots, x_n) \in \mathcal{R}^n \\ \text{subject to } g_j(x) &\leq 0; \quad j = 1, \dots, p \\ h_k(x) &= 0; \quad k = 1, \dots, q \end{aligned} \quad (3)$$

where $f(x)$, $g_j(x)$ and $h_k(x)$ are scalar functions of x .

In the Dynamic-Q approach, successive subproblems $P[i]$ $i=0,1,2,\dots$ are generated, at successive approximations x^i to the solution x^* , by constructing *spherically quadratic* approximations $\tilde{f}(x)$, $\tilde{g}_j(x)$ and $\tilde{h}_k(x)$ to $f(x)$, $g_j(x)$ and $h_k(x)$. These approximation functions, evaluated at a point x^i , are given by

$$\begin{aligned} \tilde{f}(x) &= f(x^i) + \nabla^T f(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^T A(x - x^i) \\ \tilde{g}_j(x) &= g_j(x^i) + \nabla^T g_j(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^T B_j(x - x^i); \quad j = 1, \dots, p \\ \tilde{h}_k(x) &= h_k(x^i) + \nabla^T h_k(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^T C_k(x - x^i); \quad k = 1, \dots, q \end{aligned} \quad (4)$$

with the Hessian matrices A , B_j and C_k taking on the simple forms

$$A = \text{diag}(a, a, \dots, a) = aI; \quad B_j = b_j I; \quad C_k = c_k I \quad (5)$$

Clearly the identical entries along the diagonal of the Hessian matrices indicate that the approximate subproblems $P[i]$ are indeed spherically quadratic.

For the first subproblem ($i=0$) a linear approximation is formed by setting the curvatures a , b_j and c_k to zero. Thereafter a , b_j and c_k are chosen so that the approximating functions (4) interpolate their corresponding actual functions at both $\mathbf{x}^i$ and $\mathbf{x}^{i+1}$. These conditions imply that for $i=1,2,3,\dots$

$$\begin{aligned} a &= \frac{2[f(\mathbf{x}^{i+1}) - f(\mathbf{x}^i) - \nabla^T f(\mathbf{x}^i)(\mathbf{x}^{i+1} - \mathbf{x}^i)]}{\|\mathbf{x}^{i+1} - \mathbf{x}^i\|^2} \\ b_j &= \frac{2[g_j(\mathbf{x}^{i+1}) - g_j(\mathbf{x}^i) - \nabla^T g_j(\mathbf{x}^i)(\mathbf{x}^{i+1} - \mathbf{x}^i)]}{\|\mathbf{x}^{i+1} - \mathbf{x}^i\|^2} \\ c_k &= \frac{2[h_k(\mathbf{x}^{i+1}) - h_k(\mathbf{x}^i) - \nabla^T h_k(\mathbf{x}^i)(\mathbf{x}^{i+1} - \mathbf{x}^i)]}{\|\mathbf{x}^{i+1} - \mathbf{x}^i\|^2} \end{aligned} \quad (6)$$

If the gradient vectors ∇f , ∇g_j and ∇h_k are not known analytically, they may be approximated from functional data by means of first-order forward finite differences.

The particular choice of spherically quadratic approximations in the Dynamic-Q algorithm has implications on the computational and storage requirements of the method. Since the second derivatives of the objective function and constraints are approximated using function and gradient data, the $O(n^2)$ calculations and storage locations, which would usually be required for these second derivatives, are not needed. The computational and storage resources for the Dynamic-Q method are thus reduced to $O(n)$. At most, $4+p+q+r+s$ n -vectors need be stored (where p , q , r and s are respectively the number of inequality and equality constraints and the number of lower and upper limits of the variables). These savings become significant when the number of variables becomes large. For this reason it is expected, and has also been shown [3], that the Dynamic-Q method is well suited, for example, to engineering problems such as structural optimization problems where a large number of variables are present.

In many optimization problems, additional simple side constraints of the form $\hat{k}_i \leq x_i \leq \tilde{k}_i$ occur. Constants $\hat{k}_i$ and $\tilde{k}_i$ respectively represent lower and upper bounds for variable x_i . Since these constraints are of a simple form (having zero curvature), they need not be approximated in the Dynamic-Q method and are instead explicitly treated as special linear inequality constraints. Constraints corresponding to lower and upper limits are respectively of the form

$$\hat{g}_l(\mathbf{x}) = \hat{k}_{vl} - x_{vl} \leq 0; \quad l = 1, \dots, r \leq n \quad \text{and} \quad \tilde{g}_m(\mathbf{x}) = x_{wm} - \tilde{k}_{wm} \leq 0; \quad m = 1, \dots, s \leq n. \quad (7)$$

where $vl \in \hat{I} = (v1, v2, \dots, vr)$ the set of r subscripts corresponding to the set of variables for which respective lower bounds $\hat{k}_{vl}$ are prescribed, and $wm \in \tilde{I} = (w1, w2, \dots, ws)$ the set of s subscripts corresponding to the set of variables for which respective upper bounds $\tilde{k}_{wm}$ are prescribed. The subscripts vl and wm are used since there will, in general, not be n lower and upper limits, i.e. usually $r \neq n$ and $s \neq n$.

In order to obtain convergence to the solution in a controlled and stable manner, move limits are placed on the variables. For each approximate subproblem $P[i]$ this move limit takes the form of an additional single inequality constraint

$$g_{\delta}(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^{i-1}\|^2 - \delta^2 \leq 0 \quad (8)$$

where δ is an appropriately chosen step limit and $\mathbf{x}^{i-1}$ is the solution to the previous subproblem.

The approximate subproblem, constructed at $\mathbf{x}^i$, to the optimization problem (3) (plus simple side constraints (7) and move limit (8)), thus becomes $P[i]$:

$$\begin{aligned} \min_{\mathbf{x}} \tilde{f}(\mathbf{x}); \quad & \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathcal{R}^n \\ \text{subject to } & \tilde{g}_j(\mathbf{x}) \leq 0; \quad j = 1, \dots, p \\ & \tilde{h}_k(\mathbf{x}) = 0; \quad k = 1, \dots, q \\ & \tilde{g}_l(\mathbf{x}) \leq 0; \quad l = 1, \dots, r \\ & \tilde{g}_m(\mathbf{x}) \leq 0; \quad m = 1, \dots, s \\ & g_{\delta}(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^{i-1}\|^2 - \delta^2 \leq 0 \end{aligned} \quad (9)$$

with solution $\mathbf{x}^{*i}$.

The Dynamic-Q algorithm can now be stated as follows:

Dynamic-Q algorithm

1. Choose a starting point $\mathbf{x}^0$ and step limit δ . Set $i:=0$.
2. Evaluate $f(\mathbf{x}^i)$, $g_j(\mathbf{x}^i)$ and $h_k(\mathbf{x}^i)$ as well as $\nabla f(\mathbf{x}^i)$, $\nabla g_j(\mathbf{x}^i)$ and $\nabla h_k(\mathbf{x}^i)$. If termination criteria are satisfied then stop.
3. Construct a local approximation $P[i]$ to the optimization problem at $\mathbf{x}^i$ using expressions (4), (5) and (6).
4. Solve the approximated subproblem $P[i]$ (9) using the constrained optimizer LFOPC with $\mathbf{x}^0 := \mathbf{x}^i$ (see section 4) to give $\mathbf{x}^{*i}$.
5. Set $i:=i+1$, $\mathbf{x}^i := \mathbf{x}^{*i}$ and return to Step 2.

4. The dynamic trajectory “leap-frog” optimization method for solving the sub-problems

In the Dynamic-Q method the subproblems generated are solved using the dynamic trajectory, or “leap-frog” method of Snyman [4,5] for unconstrained optimization applied to penalty function formulations (Snyman [6], Snyman et al. [3]) of the constrained problem.

In its unconstrained form, the Leap-Frog Optimizer (LFOP) determines the minimum of a function $f(\mathbf{x})$ by considering the associated dynamic problem of the motion of a particle of unit mass in an n -

dimensional conservative force field where the potential energy of the particle at a point $\mathbf{x}(t)$ at time t is given by $f(\mathbf{x})$. The method thus requires the solution of the equations of motion:

$$\ddot{\mathbf{x}}(t) = -\nabla f(\mathbf{x}(t)) \quad (10)$$

subject to initial conditions

$$\mathbf{x}(0) = \mathbf{x}_0; \quad \dot{\mathbf{x}}(0) = \mathbf{v}_0 \quad (11)$$

To explain how the dynamic trajectory method works, consider the solution of the above problem over the time interval $[0, t]$. It follows that

$$\begin{aligned} \frac{1}{2} \|\dot{\mathbf{x}}(t)\|^2 - \frac{1}{2} \|\mathbf{v}_0\|^2 &= f(\mathbf{x}_0) - f(\mathbf{x}(t)) \\ T(t) - T(0) &= f(0) - f(t); \text{ or } f(t) + T(t) = f(0) + T(0) = K \end{aligned} \quad (12)$$

Here $T(t)$ is used to denote the kinetic energy of the particle at time t and K is a constant determined by the initial values. The last expression in (12) indicates that energy is conserved. It can also be seen that $\Delta f = -\Delta T$, therefore as long as T increases, f decreases. This forms the basis of the dynamic trajectory method.

The LFOP algorithm computes an approximation to the trajectory followed by the particle in the force field. Whenever T is *increasing* along the trajectory, f is decreasing and the algorithm is minimizing the function. However, whenever T is *decreasing* along the trajectory, the objective function (potential energy) is increasing. An interfering strategy is then applied to extract kinetic energy from the particle. The consequence of this strategy, based on an energy conservation argument, is that a systematic reduction in the potential energy f of the particle is obtained. The particle is thus forced to follow a path to a local minimum at $\mathbf{x}^*$. The numerical integration of the initial value problem (10) and (11) is achieved using the "leap-frog" (Euler forward-Euler backward) method. The method contains some heuristic elements relating to time step selection and control.

The LFOP algorithm outlined above can be modified to handle Constrained problems by means of the penalty function approach (LFOPC) [6]. In particular, the penalty function formulation for constrained quadratic optimization problem $P[i]$ (9) is

$$Q(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{p+r+s+1} \alpha_j g_j^2(\mathbf{x}) + \sum_{k=1}^q \beta_k h_k^2(\mathbf{x}) \quad (13)$$

where the vector of inequality constraints functions $\mathbf{g}(\mathbf{x}) = [\tilde{\mathbf{g}}, \hat{\mathbf{g}}, \check{\mathbf{g}}, \mathbf{g}_s]^T$ and $h_k(\mathbf{x}) = \tilde{h}_k(\mathbf{x})$, and

$$\alpha_j = \begin{cases} 0 & \text{if } g_j(\mathbf{x}) \leq 0 \\ \rho_j & \text{if } g_j(\mathbf{x}) > 0 \end{cases}$$

For simplicity the penalty parameters α_j and β_k usually take on the same positive value $\alpha_j = \beta_k = \mu$. It can be shown that as μ tends to infinity, the unconstrained minimum of $Q(\mathbf{x})$ yields the solution to the constrained problem (9).

The dynamic trajectory method is applied to the penalty function formulation of the constrained problem in three phases:

Phase 0: Given some starting point $\mathbf{x}^0$, apply LFOP with some overall penalty parameter $\mu=\mu_0(=10^2)$ to $Q(\mathbf{x}, \mu_0)$ to give $\mathbf{x}^*(\mu_0)$.

Phase 1: With $\mathbf{x}^0:=\mathbf{x}^*(\mu_0)$, apply LFOP with increased overall penalty parameter $\mu=\mu_1(=10^4)>>\mu_0$ to $Q(\mathbf{x}, \mu_1)$ to give $\mathbf{x}^*(\mu_1)$. Identify the set of n_a active constraints corresponding to the set of subscripts $I_a=(u_1, u_2, \dots, u_{n_a})$ for which $g_{uj}(\mathbf{x}^*(\mu_1)) > 0, j=1, 2, \dots, n_a$.

Phase 2: With $\mathbf{x}^0:=\mathbf{x}^*(\mu_1)$, apply LFOP to

$$\underset{\mathbf{x}}{\text{minimize}} Q_a(\mathbf{x}, \mu_1) = \sum_{j=1}^{n_a} \mu_1 g_{uj}^2(\mathbf{x}) + \sum_{k=1}^q \mu_1 h_k^2(\mathbf{x})$$

to give $\mathbf{x}^*$.

The LFOPC algorithm possesses a number of outstanding characteristics, which makes it highly suitable for implementation in the Dynamic-Q methodology. The algorithm requires only gradient information and no explicit line searches or function evaluations are performed. These properties, together with the influence of the fundamental physical principles underlying the method, ensure that the algorithm is extremely robust. This has been proven over many years of testing (Snyman [6]). A further desirable characteristic related to its robustness, and the main reason for its application in Step 4 of the Dynamic-Q algorithm, is that if there is no feasible solution to the problem, the LFOPC algorithm will still find the best possible compromised solution without breaking down. The Dynamic-Q algorithm thus usually converges to a solution from an infeasible remote point without the need to use line searches between subproblems, as is the case with SQP.

5. Numerical results

The Dynamic-Q method requires very few parameter settings by the user. Other than convergence criteria and specification of a maximum number of iterations, the only parameter required is the step limit δ . The algorithm is not very sensitive to the choice of this parameter, however, δ should be chosen of the same order of magnitude as the diameter of the region of interest. For the problems listed in Table 1 a step limit of $\delta=1$ was used except for problems 72 and 106 where step limits $\delta=\sqrt{10}$ and $\delta=100$ were used respectively.

Given specified positive tolerances ϵ_x and ϵ_f , then at step i termination of the algorithm occurs if the

normalized step size $\Delta \mathbf{x}_{\text{norm}} = \frac{\|\mathbf{x}^i - \mathbf{x}^{i-1}\|}{1 + \|\mathbf{x}^i\|} < \epsilon_x$ or if the normalized change in function value

$\Delta f_{\text{norm}} = \frac{|f^i - f_{\text{best}}|}{1 + |f_{\text{best}}|} < \epsilon_f$ where f_{best} is the lowest previous feasible function value. This particular

function termination criterion is used since the Dynamic-Q algorithm may at times exhibit oscillatory behavior near the solution.

Problem #	n	f_{act}	SQP			Dynamic-Q		
			NF	f^*	f_{err}	NF	f^*	f_{err}
2	2	5.04E-02	16~	2.84E+01	2.70E+01	7*	4.94E+00	<1.00E-08
10	2	-1.00E+00	12	-1.00E+00	5.00E-08	13	-1.00E+00	<1.00E-08
12	2	-3.00E+01	12	-3.00E+01	<1.00E-08	9	-3.00E+01	<1.00E-08
13	2	1.00E+00	45	1.00E+00	5.00E-08	50\$	9.59E-01	2.07E-02
14	2	1.39E+00	6	1.39E+00	8.07E-09	5	1.39E+00	7.86E-07
15	2	3.07E+02	5	3.07E+02	<1.00E-08	15*	3.60E+02	5.55E-07
16	2	2.50E-01	6*	2.31E+01	<1.00E-08	5*	2.31E+01	<1.00E-08
17	2	1.00E+00	12	1.00E+00	<1.00E-08	16	1.00E+00	<1.00E-08
20	2	3.82E+01	20	3.82E+01	4.83E-09	4*	4.02E-01	<1.00E-08
22	2	1.00E+00	9	1.00E+00	<1.00E-08	3	1.00E+00	<1.00E-08
23	2	2.00E+00	7	2.00E+00	<1.00E-08	5	2.00E+00	<1.00E-08
24	2	-1.00E+00	5	-1.00E+00	<1.00E-08	4	-1.00E+00	1.00E-08
26	3	0.00E+00	19	4.05E-08	4.05E-08	27	1.79E-07	1.79E-07
27	3	4.00E-02	25	4.00E-02	1.73E-08	28	4.00E-02	9.62E-10
28	3	0.00E+00	5	2.98E-21	2.98E-21	12	7.56E-10	7.56E-10
29	3	-2.26E+01	13	-2.26E+01	8.59E-11	11	-2.26E+01	8.59E-11
30	3	1.00E+00	14	1.00E+00	<1.00E-08	5	1.00E+00	<1.00E-08
31	3	6.00E+00	10	6.00E+00	<1.00E-08	10	6.00E+00	1.43E-08
32	3	1.00E+00	3	1.00E+00	<1.00E-08	4	1.00E+00	<1.00E-08
33	3	-4.59E+00	5*	-4.00E+00	<1.00E-08	3*	-4.00E+00	<1.00E-08
36	3	-3.30E+03	4	-3.30E+03	<1.00E-08	15	-3.30E+03	<1.00E-08
45	5	1.00E+00	8	1.00E+00	<1.00E-08	7	1.00E+00	1.00E-08
52	5	5.33E+00	8	5.33E+00	5.62E-09	12	5.33E+00	1.02E-08
56	7	-3.46E+00	11	-3.46E+00	<1.00E-08	20	-3.46E+00	6.73E-08
60	3	3.26E-02	9	3.26E-02	3.17E-08	11	3.26E-02	1.21E-09
61	3	-1.44E+02	10	-1.44E+02	1.52E-08	10	-1.44E+02	1.52E-08
63	3	9.62E+02	9	9.62E+02	2.18E-09	6	9.62E+02	2.18E-09
65	3	9.54E-01	11~	2.80E+00	9.47E-01	9	9.54E-01	2.90E-08
71	4	1.70E+01	5	1.70E+01	1.67E-08	6	1.70E+01	1.67E-08
72	4	7.28E+02	35	7.28E+02	1.37E-08	30	7.28E+02	1.37E-08
76	4	-4.68E+00	6	-4.68E+00	3.34E-09	8	-4.68E+00	3.34E-09
78	5	-2.92E+00	9	-2.92E+00	2.55E-09	6	-2.92E+00	2.55E-09
80	5	5.39E-02	7	5.39E-02	7.59E-10	6	5.39E-02	7.59E-10
81	5	5.39E-02	8	5.39E-02	1.71E-09	12	5.39E-02	1.90E-10
104	8	3.95E+00	19	3.95E+00	8.00E-09	42	3.95E+00	5.26E-08
106	8	7.05E+03	44	7.05E+03	1.18E-05	79	7.05E+03	1.18E-05
108	9	-8.66E-01	9*	-6.97E-01	1.32E-02	26	-8.66E-01	3.32E-09
118	15	6.65E+02	~	~	~	38	6.65E+02	3.00E-08
Svanberg	21	2.80E+02	150	2.80E+02	9.96E-05	93	2.80E+02	1.59E-06

* Converges to a local minimum - listed f_{err} relative to function value at local minimum

~ Fails

\$ Terminates on maximum number of steps

Table 1: Results for the SQP and Dynamic-Q methods

In Table 1 the performance of the Dynamic-Q method is compared to results obtained for Powell's SQP method as reported by Hock and Schittkowski [7]. The problem numbers given correspond to the problem numbers in the same book by these authors. For each problem, the actual function value f_{act} is given, as well as, for each method, the calculated function value f^* at convergence, the

relative function error $f_{\text{err}} = \frac{|f_{\text{act}} - f^*|}{1 + |f_{\text{act}}|}$ and the number of function evaluations (NF) required for

convergence. In some cases it was not possible to calculate the relative function error due to rounding off of the solutions reported by Hock and Schittkowski. In these cases the calculated solutions were correct to at least eight significant figures. For the Dynamic-Q algorithm, convergence tolerances of $\epsilon_f=10^{-8}$ on the function value and $\epsilon_s=10^{-5}$ on the step size were used. These were chosen to allow for comparison with the reported SQP results. The result for the 12-corner polytope problem of Svanberg [8] is also given. For this problem the results given in the SQP columns are for Svanberg's Method of Moving Asymptotes (MMA). The recorded number of function evaluations for this method is approximate since the results given correspond to 50 outer iterations of the MMA, each requiring about 3 function evaluations.

6. Conclusion

A robust and efficient method for nonlinear optimization, with minimal storage requirements compared to those of the SQP method, has been proposed and tested. The particular methodology proposed is made possible by the special properties of the LFOPC optimization algorithm [6], which is used to solve the quadratic subproblems. Comparison of the results for Dynamic-Q with the results for the SQP method show that equally accurate results are obtained with comparable number of function evaluations. In addition to the testing performed here, the Dynamic-Q has also been applied to, and performed well in a number of practical engineering problems, in the fields of structural optimization [3], the optimization of dynamical systems [9], and in computational fluid dynamics [10,11,12].

References

1. Papalambros, P.Y. and Wilde, D.J., 1993, *Principles of Optimal Design*, Cambridge University Press, Cambridge.
2. Powell, M.J.D., 1978, *Algorithms for nonlinear constraints that use Lagrangian functions*, Mathematical Programming, Vol. 14, pp.224-248.
3. Snyman, J.A., Roux, W.J. and Stander, N., 1994, *A dynamic penalty function method for the solution of structural optimization problems*, Applied Mathematical Modeling, Vol. 18, pp.453-460.
4. Snyman, J.A., 1982, *A new and dynamic method for unconstrained minimization*, Applied Mathematical Modeling, Vol. 6, pp.449-462.
5. Snyman, J.A., 1983, *An improved version of the original leap-frog method for unconstrained minimization*, Applied Mathematical Modeling, Vol. 7, pp.216-218.
6. Snyman, J.A., 1999, *The LFOPC leap-frog algorithm for constrained optimization*, To appear in Computers and Mathematics with Applications (2000).

7. Hock, W. and Schittkowski, K., 1981, *Lecture Notes in Economics and Mathematical Systems. No 187: Test examples for nonlinear programming codes*, Springer-Verlag, Berlin, Heidelberg, New York.
8. Svanberg, K., 1999, *The MMA for modeling and solving optimization problems*, Proceedings of the 3rd World Conference on Structural and Multidisciplinary Optimization, Buffalo, New York, 17-21 May.
9. Smit, W.J., 2000, *The optimal design of a planar Stewart platform for prescribed machining tasks*. Master of Engineering Thesis, Department of Mechanical Engineering, University of Pretoria, Pretoria.
10. Craig, K.J., Venter, P.J., De Kock, D.J. and Snyman, J.A., 1999, *Optimisation of structured grid spacing parameters for separated flow simulation using mathematical optimization*, Journal of Wind Engineering and Industrial Aerodynamics, Vol. 80, pp. 221-231.
11. Craig, K.J., De Kock, D.J. and Snyman, J.A., 1999, *Using CFD and mathematical optimization to investigate air pollution due to stacks*, International Journal for Numerical Methods in Engineering, Vol. 44, pp. 551-565.
12. De Kock, D.J., Craig, K.J. and Snyman, J.A., 2000, *Using mathematical optimization in the CFD analysis of a continuous quenching process*, International Journal for Numerical Methods in Engineering, Vol. 47, pp. 985-999.

AN OPTIMISATION APPROACH TO ENGINE MOUNTING DESIGN

P S Heyns

Department of Mechanical and Aeronautical Engineering
University of Pretoria, Pretoria
E-mail: sheyns@postino.up.ac.za

ABSTRACT

Conventional engine mounting design approaches presuppose considerable experience in the formulation of suitable design objectives. A direct approach in terms of the forces transmitted to the vehicle body, may be more appropriate under unconventional circumstances. In this paper the viability of an optimisation approach to engine mounting design is demonstrated. Mounting positions and stiffness coefficients are selected as design variables to be adjusted to minimise force transmission to the vehicle body.

1. INTRODUCTION

Optimal mounting of power plants has always been one an important concern in automotive NVH design because of its effect on comfort [1]. The conventional mounting configuration for rear wheel drive cars, with two engine mounts and one transmission mount has evolved throughout the history of the automobile and is well developed, as is evident from the fact that most rear wheel drive cars have approximately the same engine mount system. This is however not true of front wheel drive cars where it is more difficult to design the mount system because of factors such as more severe engine compartment spatial constraints, greater equivalent torque tending to rotate the power plant, fewer cylinders and a smaller degree of symmetry which makes intuitive or manual design methods more difficult to apply [2].

Several researchers have proposed the reduction of engine forces by manipulating the system natural frequencies or by the decoupling of specified modes [1,2,3,4].

This approach however presuppose considerable experience in the formulation of suitable design objectives. A direct approach in terms of the forces transmitted to the vehicle body and the acceptable engine motion [5,6], may be more appropriate under unconventional circumstances.

In this paper an optimisation approach to the design of rigid body mounting configurations is presented. It entails the formulation of the equations of motion for a rigid engine supported by an arbitrary number of mounts and excited by an external force function. An objective function value, based on the forces transmitted to the support structure, is then computed. Using optimisation principles, the mount characteristics are systematically varied to minimise the force transmission subject to constraints such as the maximum allowable motion of the body at any of the mount attachment points.

This approach has previously been applied to the optimisation of engine mounts, using time domain simulation to determine the engine response [6]. Assuming periodic excitation and using a frequency domain approach [5,7,8], it is, however, possible to significantly reduce the computational burden which makes the approach suitable for design purposes. Extending this approach, Blanchet and Champoux [9] recently formulated an objective function in terms of sound pressure level.

The basic approach, based on a transmitted force objective function, is demonstrated here through a numerical example in which the mounting positions as well as stiffness coefficients are regarded as design variables. To conform to spatial limitations, restrictions are placed on the allowable range of position vectors. The range of stiffness values is also restricted.

Because of its conceptual simplicity and its ability to handle spatial constraints in a simple way, it is suggested that this approach to the design of engine mountings be further developed, especially for the consideration of lower frequency phenomena such as idle shake.

2. EQUATIONS OF MOTION

A rigid engine of mass m is attached to a rigid support structure by means of an arbitrary number n of elastic mounts at arbitrary positions and orientations with respect to the global co-ordinate system. The origin of the fixed global co ordinate system xyz is located at the centre of mass of the body g when in the initial unloaded state (see figure 1).

Since the mount axes $x'y'z'$ are generally inclined with respect to the global axes, the corresponding mount stiffness matrix must be transformed from the mount to the global co-ordinate system by means of a linear transformation matrix:

$$[T]_i = \begin{bmatrix} \cos(x'x) & \cos(y'x) & \cos(z'x) \\ \cos(x'y) & \cos(y'y) & \cos(z'y) \\ \cos(x'z) & \cos(y'z) & \cos(z'z) \end{bmatrix} \quad (1)$$

where the elements are the direction cosines of the angles between the indicated axes. Using this transformation matrix, the diagonal mount stiffness matrix $[K]_i$ may be transformed to the global system as follows:

$$[K]_i = [T]_i^T [K]_i [T]_i \quad (2)$$

where ^{T} denotes the transpose.

For elastomeric materials such as are often used for engine mounts, a hysteretic damping model with a complex stiffness matrix may be assumed, so that

$$[K]_i = \begin{bmatrix} k'_x(1+i\eta'_x) & 0 & 0 \\ 0 & k'_y(1+i\eta'_y) & 0 \\ 0 & 0 & k'_z(1+i\eta'_z) \end{bmatrix}_i \quad (3)$$

where k'_x, k'_y and k'_z are the spring rates and η'_x, η'_y and η'_z are the loss factors of mount i in terms of the local co-ordinate system, if rotational stiffness of the mount is neglected. $i = \sqrt{-1}$

A second transformation relates the displacements of each mount to displacements and rotations of the engine centre of mass. Assuming small displacements, it follows that

$$\{u\}_i = \begin{Bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{Bmatrix}_i = \begin{bmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{bmatrix}_i \begin{Bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \theta_x \\ \Delta \theta_y \\ \Delta \theta_z \end{Bmatrix}_g = [G]_i \{U\} \quad (4)$$

where $\{u\}_i$ is the translational displacement vector at the mount attachment point i and $\{U\}$ is the engine centre of mass displacements (translational as well as rotational).

With $\{F\}_i$ a vector comprising the three forces and three moments due to mount i acting on the body along x , y and z and about x , y and z respectively, it follows that

$$\{F\}_i = -[[G]_i^T [K]_i [G]_i] \{U\} \quad (5)$$

Disregarding second-order terms in the rigid body equations of motion [10], it may be shown that

$$[M]\{\ddot{U}\} = \{F\} \quad (6)$$

with $\{F\}$ the resultant total force on the engine and the mass matrix

$$[M] = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & I_{xy} & I_{xz} \\ 0 & 0 & 0 & I_{xy} & I_{yy} & I_{yz} \\ 0 & 0 & 0 & I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \quad (7)$$

m is the engine mass and the I_{kl} are moments and products of inertia expressed in terms of the global co-ordinate system.

Adding together the effects of all the mounts on the engine, it follows that

$$\sum_{i=1}^n \{F\}_i = - \left[\sum_{i=1}^n [G]_i^T [K]_i [G]_i \right] \{U\} = -[K] \{U\} \quad (8)$$

From this and equation (5) it now follows that

$$[M]\{\ddot{U}\} + [K]\{U\} = \{F\}_e \quad (9)$$

where $\{F\}_e$ represents all forces on the body other than the mount reaction forces.

In order to take advantage of a frequency domain approach, we now consider periodic excitation forces using a Fourier series approximation. Assuming $\{F\}_e$ to comprise a series of m sinusoidal force phasors $\{\bar{F}\}_j$ with frequencies ω_j and corresponding phase angles α_j , $j=1, 2, \dots, m$, it follows that:

$$\{\bar{U}\} = \sum_{j=1}^m \left[[K] - \omega_j^2 [M] \right]^{-1} \{\bar{F}\}_j \quad (10)$$

where $\{\bar{U}\}$ is the centre of mass displacement phasor vector.

3. OPTIMISATION

As a criterion of the vibration transmitted to the support structure, an objective function [6,7]

$$\phi = \sqrt{\sum_{i=1}^n \{f_{xi}^2(\{X\}) + f_{yi}^2(\{X\}) + f_{zi}^2(\{X\})\}} \quad (11)$$

may be defined. $\{X\}$ is a vector of design variables typically comprising spring rates or mount positions and orientations. f_{xi} , f_{yi} and f_{zi} are the maximum values of forces transmitted to the support structure and are functions of $\{\bar{U}(X)\}$.

It is necessary to minimise ϕ with respect to variables $\{X\}$, subject to the inequality constraints

$$\{g\} = |\{U\}| - \{U\}_c \leq 0 \quad (12)$$

where $|\{U\}|$ is the vector of displacement amplitudes (only translations in this implementation) for a particular design configuration $\{X\}$. $\{U\}_c$ is a vector of specified maximum acceptable translational displacement amplitudes. This implies that

$$\max(\{g\}) = \max\{g_{x1}, g_{y1}, g_{z1}, g_{x2}, \dots, g_{xi}, \dots, g_{zn}\} \leq 0 \quad (13)$$

The theory of the preceding paragraphs was implemented in MATLAB, using its Optimisation Toolbox `constr.m` function that uses a sequential quadratic programming method [11].

4. NUMERICAL EXAMPLE

Consider an engine excited by periodic force F_z and moments M_x and M_y about the engine centre of mass. Each of these forces and moments comprise different harmonic components, with amplitudes and phase angles as follows:

	F_z	M_x	M_y
Amplitude[N,Nm]			
194 rad/s	220	85	19.8
388 rad/s	33	44	2.9
582 rad/s	-	15	-
Phase angle [rad]			
194 rad/s	-3.0	1.8	3.0
388 rad/s	0.35	1.6	0.35
582 rad/s	-	1.5	-

The engine mass is 170 kg and the principal moments of inertia are 4.0, 10.0 and 8.0 kgm^2 at angles 5° , 12.3° and -20.7° with respect to the global x , y and z axes.

Three mounts are positioned at the following global co-ordinates:

	x [m]	y [m]	z [m]
Mount 1	-0.25	0.25	-0.25
Mount 2	0.30	-0.10	0.05
Mount 3	-0.35	-0.35	-0.35

For simplicity the mount loss factor is considered to be the same in all three directions, i.e. $\eta_x = \eta_y = \eta_z = 0.1$.

The maximum acceptable displacement amplitudes at the mount attachment points are assumed to be 0.2 mm in all three orthogonal directions for all three mounts.

To demonstrate the procedure, different starting designs were selected and optimised. For this purpose the 9 stiffness coefficients and the 9 mount position co-ordinates were considered as design variables. The stiffness coefficients were restricted to a range of between 5×10^4 and 1×10^8 N/m to avoid interference with the engine, while each mount position co-ordinate was allowed a ± 50 mm variation with respect to the initial co-ordinate.

The sequential quadratic programming method was then applied to find the corresponding optima. The results of these analyses for the different starting design configurations with different stiffness coefficients are shown in tables 1 and 2. In all cases the same initial position co-ordinates were assumed. From these tables it is clear that the optimisation process converged to different solutions in the selected cases. (During the investigation many other starting designs were used, some of them converging to one of these optimised configurations.)

Designs A to C represent very flexible designs rendering particularly desirable objective functions. However, because of the high degree of flexibility these designs may not be suitable if, for example, significant starting transients are present. Such considerations are not accounted for in the present implementation (although it would be possible if a time domain simulation approach were to be followed [6]), but may be typical of the type of consideration which the designer may take into account in selecting the most suitable design point.

Design F is representative of a stiff design. This design will probably not present any problems during start-up or run-down, but it is clear that significantly higher forces will be transmitted to the support structure during normal operation ($\varphi = 395.63$ compared to $\varphi = 31.40$).

There seems to be little advantage in selecting designs D and E, unless the positions of the mount attachment points would, for some or other reason, be more desirable than for designs A to C or F.

5. CONCLUSIONS

An optimisation approach to vibration isolation of periodically excited rigid bodies, based on the minimisation of forces transmitted to the support structure has been presented. The approach is well suited to handle design constraints and is conceptually (although not necessarily computationally) quite simple.

The approach was demonstrated on an engine, simultaneously excited by a periodic force and two periodic moments and seems to work well. It is, however, clear that multiple local minima exist for these type of problems and that due care should be exercised during optimisation.

REFERENCES

- [1] Johnson, S.R. and Subhedar, J.W. Computer optimization of engine mounting systems. *SAE Paper 790974*, pp 19-26, 1979.
- [2] Geck, P.E. and Patton, R.D. Front wheel drive engine mount optimization. *SAE Paper 840736, Proceedings of the 5th International Conference on Vehicular Structural Mechanics*, pp 123-134, April 1984.
- [3] Bernard, J.E. and Starkey, J.M. Engine mount optimization. *SAE Paper 830257, International Congress & Exposition, Detroit*, pp 1-9, Feb-Mar 1983.
- [4] Feng, Z. and Hou, J. Optimization of engine mounting parameters with forbidden bands of multiple modal frequencies. *Proceedings of the 7th International Modal Analysis Conference, Las Vegas*, pp 884-888, 1989.
- [5] Swanson, D.A., Wu, H.T. & Ashrafiuon, H. Optimization of aircraft engine suspension systems. *Journal of Aircraft*, pp 30, 979-984, 1993.
- [6] Snyman, J.A., Heyns, P.S. & Vermeulen, P.J. Vibration isolation of a mounted engine through optimization. *Mechanism and Machine Theory*, 30, pp 109-118, 1995.
- [7] Heyns, P.S., Nel, C.B. & Snyman, J.A. Optimization of engine mounting configurations. *ISMA19 Tools for Noise and Vibration Analysis*, Leuven, September 1994.
- [8] Heyns, P.S. An optimisation approach to engine mounting design. *Proceedings of the 14th International Modal Analysis Conference, Dearborn*, pp 1124-1129, 1996.
- [9] Blanchet, D. & Chamoux, Y. Comparison of objective functions for engine mounts optimization. *Proceedings of the 18th International Modal Analysis Conference, San Antonio, Texas*, pp 631-636, 2000.
- [10] D'Souza, A.F. & Garg, V.K. *Advanced dynamics Modeling and analysis*, Prentice-Hall, 1994.
- [11] The Mathworks. *Optimization toolbox for use with MATLAB*. The Mathworks, 1998.

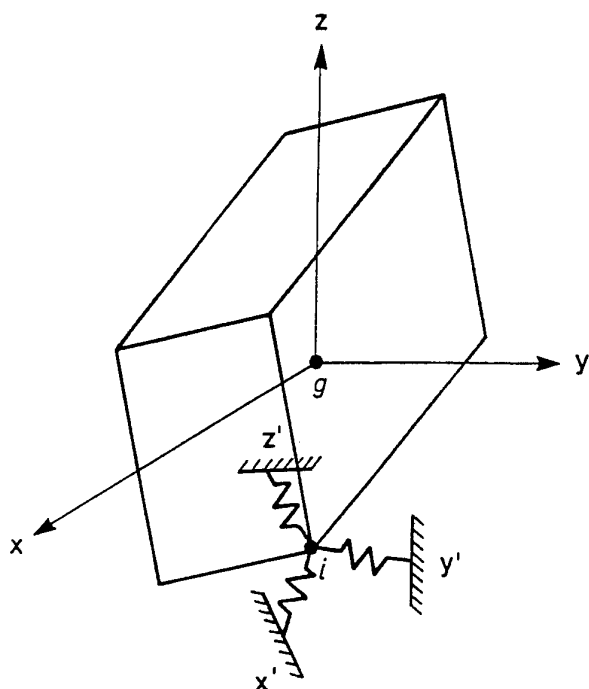


Figure 1 Engine model

Table 1 Optimised design variables (Design cases A to C)

Design variable	Starting design A	Optimised design A	Starting design B	Optimised design B	Starting design C	Optimised design C
x_1	-0.25	-0.2000	-0.25	-0.2000	-0.25	-0.2000
y_1	0.25	0.2000	0.25	0.2000	0.25	0.2000
z_1	-0.25	-0.2000	-0.25	-0.2000	-0.25	-0.2000
x_2	0.30	0.2589	0.30	0.2500	0.30	0.2798
y_2	-0.10	-0.0563	-0.10	-0.0543	-0.10	-0.0629
z_2	0.05	0.0000	0.05	0.0000	0.05	0.0000
x_3	-0.35	-0.3000	-0.35	-0.3000	-0.35	-0.3000
y_3	-0.35	-0.3000	-0.35	-0.3000	-0.35	-0.4000
z_3	-0.35	-0.3000	-0.35	-0.3000	-0.35	-0.3000
k_{x1}	1×10^5	0.8566×10^5	2×10^5	1.5033×10^5	4×10^5	5.2865×10^5
k_{y1}	1×10^5	0.5000×10^5	2×10^5	0.5000×10^5	4×10^5	0.5000×10^5
k_{z1}	1×10^5	0.5000×10^5	2×10^5	0.5000×10^5	4×10^5	0.5000×10^5
k_{x2}	1×10^5	0.9991×10^5	2×10^5	1.9960×10^5	4×10^5	5.1888×10^5
k_{y2}	1×10^5	0.9774×10^5	2×10^5	1.9190×10^5	4×10^5	5.3990×10^5
k_{z2}	1×10^5	0.9920×10^5	2×10^5	1.9790×10^5	4×10^5	3.8163×10^5
k_{x3}	1×10^5	0.9613×10^5	2×10^5	1.8666×10^5	4×10^5	17.672×10^5
k_{y3}	1×10^5	0.5000×10^5	2×10^5	0.5000×10^5	4×10^5	0.5000×10^5
k_{z3}	1×10^5	0.5000×10^5	2×10^5	0.5000×10^5	4×10^5	0.5000×10^5
Objective	78.99	31.40	273.99	34.2793	609.56	77.518

Table 2 Optimised design variables (Design cases D to F)

Design variable	Starting design D	Optimised design D	Starting design E	Optimised design E	Starting design F	Optimised design F
x_1	-0.25	-0.3000	-0.25	-0.3000	-0.25	-0.3000
y_1	0.25	0.3000	0.25	0.2449	0.25	0.2000
z_1	-0.25	-0.3000	-0.25	-0.3000	-0.25	-0.2196
x_2	0.30	0.3500	0.30	0.3500	0.30	0.2552
y_2	-0.10	-0.0884	-0.10	-0.0797	-0.10	-0.0651
z_2	0.05	0.1000	0.05	0.1000	0.05	0.0999
x_3	-0.35	-0.4000	-0.35	-0.4000	-0.35	-0.3000
y_3	-0.35	-0.3405	-0.35	-0.3000	-0.35	-0.3000
z_3	-0.35	-0.4000	-0.35	-0.3000	-0.35	-0.3000
k_{x1}	6×10^5	6.333×10^5	8×10^5	7.278×10^5	10.0×10^5	15.58×10^5
k_{y1}	6×10^5	2.937×10^5	8×10^5	2.733×10^5	10.0×10^5	17.12×10^5
k_{z1}	6×10^5	0.500×10^5	8×10^5	6.385×10^5	10.0×10^5	6.346×10^5
k_{x2}	6×10^5	5.870×10^5	8×10^5	8.085×10^5	10.0×10^5	11.34×10^5
k_{y2}	6×10^5	3.860×10^5	8×10^5	6.276×10^5	10.0×10^5	13.19×10^5
k_{z2}	6×10^5	8.392×10^5	8×10^5	9.593×10^5	10.0×10^5	7.147×10^5
k_{x3}	6×10^5	9.240×10^5	8×10^5	10.24×10^5	10.0×10^5	18.66×10^5
k_{y3}	6×10^5	2.981×10^5	8×10^5	3.871×10^5	10.0×10^5	16.94×10^5
k_{z3}	6×10^5	2.148×10^5	8×10^5	1.328×10^5	10.0×10^5	0.500×10^5
Objective	949.90	502.66	1684.25	551.83	3114.52	395.63

Practical guidelines for training neural networks

Johann E.W. Holm
Department of Electrical, Electronic, and Computer Engineering
University of Pretoria
South Africa
E-mail: jholm@postino.up.ac.za

Abstract

At this stage of development in neural network research, it is necessary to pause and assess the current status of neural network training algorithms. Training a neural network is not a simple task, and a lot of factors must be considered before training proceeds as well as during training. This publication gives an overview of training paradigms, learning rules, training algorithms, data dependencies and some helpful practical advice on training neural networks, in general. We share some of our experience with the reader, which will most certainly save some valuable time in practice.

1. Introduction

Artificial neural networks have found significant application in practice over the past decade, and many new applications are currently in the process of being designed. Many of the theoretical problems associated with neural network training have been successfully addressed, and many excellent text books on this topic exist (see references [1,2]). In spite of this fact, there are still practical problems which have to be addressed. Many solutions to practical problems exist in practice, with many of the solutions kept as trade secrets, simply not published by practitioners, or (in some cases) not available. This document addresses some of the implementation issues surrounding the application of neural networks in practice, with special attention to training algorithms and their associated considerations.

We proceed in Section 2 with a (very) brief overview of different neural networks and their various intended applications - this is done in order to create a framework in which we shall discuss neural-network training. Section 3 is focused on training methods, with special attention to data dependency, effectiveness, efficiency, and complexity. In Sections 4 we briefly discuss data dependency and model selection, in Section 5 we treat statistical approaches to improve neural networks, and in Section 6 we overview environments and parameter selection. We conclude this document in Section 7 with a summary and mention current developments in this field.

2. Neural networks.

We initiate this section by first defining a general neural network. The main building block of a neural network is a perceptron, which is, generally speaking, an entity that computes a weighted sum of inputs (i.e. it computes a vector inner product), and then passes the sum through a mathematical transfer function. Different transfer functions are used in practice, with linear, Gaussian and sigmoidal functions the most popular due to their ease of use. An example of a single perceptron is shown in Figure 1 below (next page). In essence, given the transfer function(s) and network structure, the weights *are* the statistical neural network model, since the weights completely define the network's overall transfer function. There are different ways of physically implementing neurons, with software simulation and silicon hardware (neuron integrated circuits) the most popular approaches.

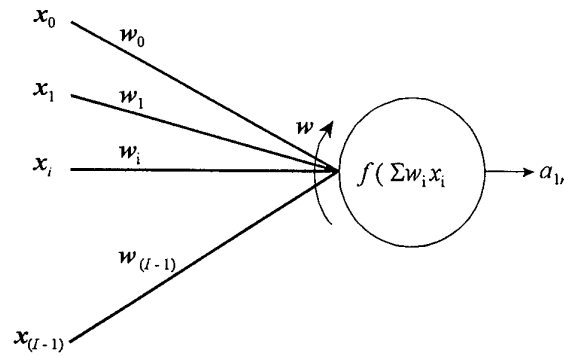


Figure 1 A perceptron with input vector x , weight vector w , transfer function $f()$ and scalar output a .

Generally speaking, all neural networks can be divided into two classes, namely recurrent and feedforward networks. *Feedforward* networks simply combine a number of perceptrons in Figure 1 to form a structure that only feeds information from the input to the output, with no feedback connections. In most cases all perceptrons in a layer are connected to the perceptrons in the following layer through a weight parameter vector. *Feedback* networks have connections between perceptrons so that previously computed outputs (delayed over time) are combined with current inputs to form a new state vector at a given instance in time. This implies that feedback networks must have memory. As a result, feedback networks are more powerful than feedforward networks. An example of a feedback network (more specifically, an Elman network) is shown in Figure 2 on the following page. The feedback connections have to be removed to form a two-layer feedforward network.

2.1 Feedforward multilayered networks.

To proceed, we discuss specific models of feedforward networks. Nonlinear layered feedforward networks are mainly encountered in two forms, namely multilayer perceptrons and radial-basis function networks. Multilayer perceptrons (shown in Figure 2, next page) use the neurons of Figure 1, interconnected through weight vectors in layers. Radial-basis function networks are different in that each hidden-layer “perceptron” has a basis (a vector with a fixed value) from which the distance d is measured to an input vector. The resulting distance d (the Euclidian distance can be used as this functional) is then passed through a nonlinear transfer function to form the output a of the radial-basis neuron. Gaussian transfer functions are commonly used, so that the Gauss function’s standard deviation parameter is a network parameter. The outputs from the hidden layer are then combined through a weight vector in the same fashion as with multilayer perceptrons to form an output layer.

Both multilayer perceptrons and radial-basis function networks are used to approximate functions, where a “function” can be virtually any many-to-one input-output function in real space. In many cases these functions are not actually available, and only observations from some underlying statistical process are recorded and are used to approximate the underlying process function. Therefore, neural networks are commonly referred to as “data-driven” models, since the recorded data determines (through training) the neural network’s weight parameters. In the past, a certain amount of controversy has been associated with neural networks due to the data issue (a further discussion on these issues follows in Section 2).

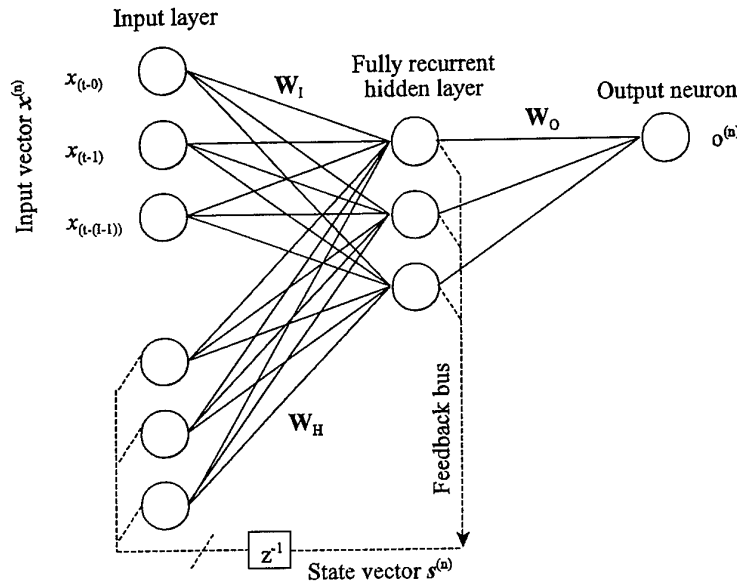


Figure 2 An Elman network as an example of a feedback network. The state vector is delayed with one time delay unit (defined by the process) in the feedback path.

Feedforward networks are used in pattern classification and time-series analyses. Classifiers are used to classify different data patterns (in the form of input vectors) into natural classes. The neural network's task is to distinguish between these vectors in the different classes by indicating (on its outputs, one output for each class) to which class a vector is most likely to belong. This is done by presenting an input vector to the network and then selecting the output with highest value. In the same vein, classifier outputs can be used to indicate the "probability" that a vector belongs to a specific class (not strictly a probability, but rather a probability indicator or estimate). For this reason, a specific type of classifier networks is used to indicate the probability that a person, who applies for a policy at an insurance broker, is going to cancel payment in less than, say 6 months. Classifiers, in turn, are used (for example) to classify objects as being animal or human in security applications. Further applications are endless, and as time progresses, we shall encounter many more creative implementations.

Feedforward networks are also used in time-series analyses, where the idea is to describe time series dynamics in terms of the network's parameters. This is a form of nonlinear regression, where the parameters used to describe the network are analogous to the regression parameters of a classical linear regressor. That is, the dynamics of the time series are captured in the neural network's overall transfer function, which is (incidentally) also the aim of finding discrete nonlinear differential equations. Feedforward networks are also sometimes used to describe the transfer function of a mechanical, chemical, and other temporal systems. Typical examples of time-series neural network applications are found in industry, specifically where it is difficult to build models of complex processes. Output data is recorded by perturbing the system with (carefully controlled!) input data - the recorded pairs of input-output data usually represent the underlying physical process. In application, the network is used to predict the system's output behavior on account of inputs at a specific time, as well as some history of the signal (in the form of previous inputs or network outputs). The predicted outputs are used to control the system, to build simulations, or to detect anomalies that may occur (such as detecting anomalous mechanical vibrations when an expensive roller bearing starts deteriorating).

A specific class of classifier neural networks is represented by a Self-Organizing feature Map (SOFM), first introduced by Kohonen in 1982. Self-organizing maps are used to cluster multi-dimensional data into a two-dimensional output space by following a specific association procedure (see reference [1]). That is, self-organizing maps are effectively used to find correlations between input patterns by investigating the activity of the outputs. This allows a user to identify unknown patterns as belonging to clusters or classes.

Feedforward networks are easy to implement, and numerous commercial and freeware software packages are available with which feedforward networks can be trained and used (see <http://www-ra.informatik.uni-tuebingen.de/SNNS>). A lot of care must be taken to select the correct data, network, and training parameters; more on this topic follows in Section 3.

2.2 Recurrent networks

Recurrent networks are significantly more powerful than feedforward networks in the sense that temporal information is used to enhance the network's ability to approximate or predict. Recurrent networks are usually not used as classifiers, but rather as time-series predictors. The Elman network (previously shown in Figure 2) is a classical example of a recurrent multilayer perceptron predictor. It is not uncommon to reduce the number of network's parameters by orders of magnitude by utilizing a recurrent Elman network instead of a feedforward network. However, the parameters have significantly more weight (sensitivity) in an Elman network than in its feedforward counterpart. This is mainly due to the compactness of the Elman network.

Two similar forms of recurrent networks are Hopfield and Boltzmann networks. Without going into the details of these networks, we briefly describe their definitions and applications. Hopfield and Boltzmann networks are networks with perceptrons that are totally interconnected, with the exception that a neuron is not connected to itself. Boltzmann machines have one hidden layer and an input/output layer, where Hopfield networks have single layers only. Associated with each weight connection is a fixed time delay similar to the recurrent layer of an Elman network. Hopfield and Boltzmann networks are used as Content Addressable Memories (CAM) since they can reconstruct complete images from incomplete or corrupted images.

Two very good text books on this topic are available in the form of references [1] and [2]. Haykin [1] is a good reference text book, while Bishop [2] provides excellent theoretical reading on feedforward networks used in pattern recognition.

3. Training methods

Training is optimization, whether it be with mathematically elegant approaches, or purely heuristical or "forceful" approaches. This is true for any type of neural network. In all cases, the aim of training is to find a set of weight parameters that will sufficiently allow that neural network to fit the data. Different training algorithms apply to the different types of networks discussed in Section 1. For example, training a network to perform function approximation might work with a second-order algorithm, but the same algorithm may not be efficient for training complex classifier networks. This section describes different environments and the appropriate algorithms in each environment.

3.1 General overview of the training problem

When training neural networks, we wish to obtain the network parameters that best describe the input-output relationship of recorded data. This data is a sample of the underlying statistical process which generated the observations. These observations are sometimes called training patterns or events. Memorization of patterns or events in neural networks (biological and artificial) is accomplished through learning, during which a neural network adapts its memory and/or structure by adapting its interconnection weights over time so that it can correctly recall certain external patterns or events. Training patterns are presented to the neural network, on account of which the neural network must adjust its weights so that the network's ability to recall or classify a pattern is enhanced according to some criterium. In this regard there are a number of *learning paradigms* that are important [1]; these are (i) supervised learning, (ii) unsupervised learning, and (iii) reinforcement learning. With supervised learning, data sets with labeled training patterns are available (manually labeled by the trainer, or the like), where the purpose of unsupervised learning is to group data on account of clusters of data that were formed by the underlying process. Reinforcement learning aims at continuously updating the network parameters to be optimal in some sense. Of the learning paradigms, most existing algorithms fall in the supervised learning paradigm.

A further classification of neural network training is done on the basis of a *learning rule* that is used to adapt the neural network's weights. Practical approaches to learning include [1] (i) Error-correction learning, under which most known algorithms fall. The next (ii) learning rule is Hebbian learning, which provides a useful basis for describing the adaptation of weights on account of external stimuli. With Hebbian learning a synapse (or weight) between two neurons is strengthened when activity in one neuron contributes positively to activity in the other neuron and the weight between the two neurons is weakened or reduced when there is negative correlation between activities in the interconnected neurons. Most natural processes adhere to this process. Following this, we have (iii) Boltzmann learning, which is based on the definition of an energy function associated with the activity inside a Boltzmann machine. Weights are adapted in order to reduce this energy (actually, the temperature associated with kinetic energy) so that the neural network reaches an equilibrium where the energy inside the network can not be reduced any further. Boltzmann learning is from the field of statistical physics and details fall beyond the scope of this discussion. The last rule is (iv) competitive learning, where neurons in this network compete against one another for activation. In practice, SOFM is an example of competitive learning and is used for clustering rather than classification, with special application in data-mining applications.

All the learning rules above are used in practice, with error-correction learning the most frequently used of all for discriminatory training. Most research work around error-correction training involves gradient-based optimization methods, so that this topic deserves individual attention. The remaining rules (Hebbian, Boltzmann, and competitive learning) are well described in available literature [1].

Gradient-based optimization algorithms mostly follow an unconstrained approach to learning, with only a few examples of constrained optimization evident in practice [3]. The LS error (for neural networks) is defined for a non-parametric system, with N available training patterns, as

$$E(\mathbf{w}^{(k)}) = \frac{1}{2N} \sum_{n=0}^{N-1} \left(d_n - f(x_n, \mathbf{w}^{(k)}) \right)^2, \quad (2.1)$$

where d_n is a desired output value for input pattern and $f(x_n, w)$ is the actual output of the system for input pattern x_n . The superscript (k) is the iteration index; iterative optimization generates a series of weight vectors $w^{(k)}$ at iterations $k = 0, 1, \dots, K-1$, where K is the total number of iterations up to termination. N is the number of training patterns. Different error functions are used, such as cross-entropy, but the LS error is effective in most applications.

3.2 First-order methods

The most popular training algorithm is online gradient descent with momentum, mostly known as online backpropagation by virtue of the way in which the gradient of equation (2.1) is computed. In its simplest form, a momentum algorithm weight and search-direction update is given by

$$w^{(k+1)} = w^{(k)} + \alpha s^{(k)} + \mu \Delta w^{(k-1)}, \quad s^{(k)} = -g^{(k)}, \quad (2.2)$$

where the update parameter α and the momentum term μ have to be selected by the user. Despite the parameter selection problem, online gradient descent is used due to its ease of implementation. An important aspect is to reduce the step-size parameter (forcibly) during training according to an annealing schedule [4] to ensure less weight variance when the weight-space trajectory passes through a local minimum. However, the Hessian is unknown with neural-network training and the step size and rate of convergence are related to the Hessian. This means that the user is working "blind" until the best possible answer from a number of trial runs becomes available; certainly not an attractive option to the novice! For this reason, more experienced users opt to use more "automatic" algorithms where the step-sizes are determined by the algorithm on the fly (mathematically, or heuristically).

To summarize, select to use online backpropagation only if the environment (Hessian, complexity, and parameter space dimensionality) is familiar. Do not use batch-mode backpropagation for it is a method which gets stuck in local minima without exception. In stead, use some of the superlinear methods in the following subsection.

3.3 Higher-order methods and heuristics

Historically, batch-mode algorithms were only used where large computers were available due to the heavy expense of gradient and error computation. Nowadays, desktop PC's are so powerful that using batch-mode optimization is no longer a tedious affair. The development of a number of powerful algorithms adds to the advantages of using batch-mode training. We will briefly discuss the algorithms which we have tested over the past decade and highlight the important aspects of each algorithm. Details of these algorithms can be found in the references, and a discussion on the algorithms' attributes is all that space permits.

Before we start with the main algorithms, we give a short discussion on the line searches that were tested, with some problems and advantages of each. The first implemented line search was a golden section search, with a good first impression. Line searches were not required to be accurate and the golden section algorithm did not suffer from ill-conditioned error surfaces commonly encountered with neural network error functions. On closer inspection the method was, however, slow despite the use of a proper bracketing algorithm. To rectify this, a double-bracketed line search algorithm was implemented

according to the guidelines in Fletcher [5]. Cubic interpolation and quadratic extrapolation was tested to good effect. This algorithm functioned well and did not suffer from ill-conditioned error surfaces. As usual, with each line search, the initial step size is very important, with extreme computational penalties when an overly conservative or optimistic step is used. To solve this problem, one may use a moving average of the initial step over a number of searches. This solved most of the line search overheads (despite an added parameter that had to be selected).

The first type of main algorithms that were implemented include conjugate gradient [6], LFOP1(b) [7], and ALECO-2 [3]. These algorithms are well-known, apart from the ALECO-2 algorithm which is described in [3,8,9]. Conjugate gradient gave good results on small neural networks (parameter dimensionality of less than 100), provided that the reset conditions are applied according to Powell [6]. We used the Fletcher line search and terminated when the gradient was 10% of the initial value, that is, $0.1 \times \nabla E(\mathbf{w})$. Despite our initial joy, conjugate gradient fell apart when the problem complexity increased in association with an increase in weight parameter dimensionality [10]. The algorithm terminated prematurely and failed to find search directions long before a good local minimum was secured. This is mainly ascribed to the large flat error surfaces, with deep narrow ravines scattered across the surface. To solve the problem of premature termination, LFOP1(b) and ALECO-2 were both implemented according to instructions from the authors. Upon evaluation, it became abundantly clear that the heuristics of LFOP1(b) addressed most of our optimization woes. ALECO-2 performed very well, but failed to find minima when the problem became very ill-conditioned [10]. In terms of parameters of the algorithm, both LFOP1(b) and ALECO-2 were found to be "user friendly" and most settings were left as suggested by the authors [7, 3].

Some of the algorithms that use the Hessian extensively were implemented and evaluated, with results varying from average to poor. These include BFGS and Levenberg-Marquardt. The main reason for the failure of BFGS in neural network environments (despite their popularity) is that the Hessian is not positive definite at all places on the surface. As a result, the approximation of the inverse Hessian becomes ill-conditioned and the algorithm fails to find a direction of decrease in error. Levenberg-Marquardt addresses this problem effectively with good results and shows that (again) heuristics solve most of the actual problems with optimization.

At the end of the day, the user requires an algorithm that works under different circumstances with good results. To the best of our knowledge, when good network models are to be found, the LFOP1(b) algorithm is the best choice, with ALECO-2 a close second. Online gradient descent gives good results after a number of trial runs, but the doubt always exists about the quality of the solution. If a second-order method has to be used (for reasons unknown) for training neural networks, then the best choice would certainly be Levenberg-Marquardt.

4. Data dependency and model selection

Neural networks are data-driven models, and are thus dependent on the amount of data available to the trainer. In this regard, it is usual to assemble training set samples of size $N \in [10W, 32W]$, where W is the dimensionality of the network, with $10W$ the lower limit for reliable results. A generalized result from worst-case analysis gives the relation

$$N_{\min} \approx \frac{W}{\varepsilon}, \quad (2.3)$$

where ε is the required accuracy [11,12]. In addition to selecting a sample size for training, one should assemble an independently recorded test set of similar size with which to test the network after training. During training it is common to use a third set, called a cross-validation set, with which to evaluate network performance during training. Training is stopped when performance on the cross-validation set starts to deteriorate. Although the size of this set is typically 10% of the training set, we advise to use an independent cross-validation set of size at least 30% of the training set, although this number depends on the problem.

An important problem is selection of the optimum number of network parameters, that is, the number of hidden neurons in the network. The best approach, we have found, is to train networks with increasing numbers of hidden neurons, up to the point where the network performance starts to deteriorate on an independent test set. At this point the network starts to specialize on the training set and there is no further generalization possible. After training a number of networks, with a given number of hidden neurons, one should select the network with the best independent test results. We typically train between 3-10 networks before model selection, with 10 networks the best option if time permits. This approach is sometimes called "committees of networks".

There exist some standard techniques for removing excess weights after training. This is done purely from a statistical point of view, since the network is typically not sensitive with respect to these weights. The aim is to find the weights that do not participate in the decision making or prediction process, and then to remove the weights according to some algorithm. This practice is often used in conjunction with regularization, which itself is a method of preventing weights from becoming too large. Regularization adds a penalty function to the standard error (objective) function and controls the size of weights so that the variance on the weights is kept to a minimum, while allowing for a few large weights, if necessary. Both pruning and regularization are good practices to follow, and should be implemented in practice. More detail on these approaches is available in Bishop [2].

A handy mathematical approach that is advisable for unknown data is input conditioning. This is simply a technique that rotates and weighs the input vectors so that their variance-covariance matrix becomes the identity matrix. Whitening, as this process is sometimes known, is a good engineering practice, regardless of its necessity, since normalizing the inputs prevents saturation of the nonlinear transfer functions.

5. Statistical methods

Recently, more statisticians became actively involved in the neural network field department. Statisticians prefer likelihood models for training, with valid reasons. Error functions are required to maximize the likelihood of training pairs (input-output pairs) in the form of a conditional density $p(\mathbf{d} | \mathbf{x})$, where $\mathbf{d}$ is the desired output of a network with a given input $\mathbf{x}$. Instead of maximizing likelihood directly, it is more practical to minimize the negative logarithm of the likelihood function. Further analysis shows that the actual form of the conditional density determines the efficacy of specific error function, and that the LS error strictly maximizes the likelihood function only if the conditional density $p(\mathbf{d} | \mathbf{x})$ is purely

Gaussian. In this regard, an algorithm called the Expectation-Maximization algorithm (EM) algorithm is sometimes employed, specifically with radial-basis function networks. This is a very interesting topic, and much more can be said about statistics and objective functions; the interested reader should consult Bishop [2].

A statistical technique that can reduce computational expense without compromising too much on quality, is statistical sampling. This technique draws samples from the training data and follows an approach between online and batch-mode optimization and is called sampled optimization [8,9]. These samples are used to compute the error and gradient, so that the whole data set is not evaluated, but a sufficiently large sample is used to reduce sampling noise on the weight-space trajectory. Sampling may also be used to train a large number of different networks on different subsets, and then combining these networks to form an averaged output. In doing so, variance on the weight parameter vector is "averaged out" over a number of different networks and a more generalized prediction estimate is formed. We are currently comparing different techniques and results should be available in the foreseeable future. Both these techniques are practical and effective.

Instead of training a number of networks on different data sets to obtain a number of networks to combine, it is possible to sample the weight space around a local minimum, and then use the sampled weights as models for a combined network. Radford Neal introduced Monte-Carlo methods for training these networks, called Bayes networks, and there is freeware available on this topic at [<http://www.cs.toronto.edu/~radford/homepage.html>]. As a matter of interest, Neal uses a leap-frog approach to conserving the Hamiltonian so that the system is conservative. This approach works very well in cases where only small data sets are available (as is often the case), and excellent network models are produced this way.

6. Hostile environments and parameter selection

No discussion about optimization can be complete without mention of optimization parameter selection and sensitivity with respect to parameters and environmental conditions.

While some environments are friendly, there are some environments that are extremely hostile. The question really is: "which?" We have trained mostly classifiers over the past decade, but our reasoning can be applied to time series as well. In short, an environment is hostile when the training problem is ill-conditioned. This happens when the data is not easily separable or when the number of hidden neurons is small in relation to a number of neurons that "easily fit" the data. Now, one may easily state that a problem can be conditioned by increasing the number of hidden neurons. Unfortunately, by increasing the number of hidden neurons, we can (in many cases) also decrease the network's ability to generalize. So, the actual problem is to find the *actual* smallest number of neurons (or parameters) that will fit the data. This is easier said than done. BFGS, conjugate gradient, LDFOP1(b) and ALECO-2 were recently tested to assess their abilities of finding a small number of neurons to fit the data, with the result that both LFOP1(b) and ALECO-2 gave excellent results, and neither BFGS or conjugate gradient gave satisfactory results [10]. LFOP1(b) and ALECO-2 were more expensive than BFGS or conjugate gradient, but this was because they were busy finding good minima in the additional time! In short, environments are defined by the data, that is, by the input space dimensionality, variability and associated complexity.

Parameter selection is not a problem in friendly environments. This is definitely not true for hostile

environments, and special care should be taken when using higher-order methods with line search algorithms (even good ones). Also, when the data set is small (less than 10 W), online gradient descent exhibits excessive noise on the weight-space trajectory, and the choice of step size and annealing parameter is a laborious affair (several trial runs are necessary to find the actual parameters). To our knowledge, LFOP1(b) is the superior algorithm to address these requirements. There is one important parameter here (though), namely a weight-update limiting parameter. By limiting the step size, one can prevent LFOP1(b) from overstepping the very narrow ravines on the largely flat error surface - this was found to be very effective and practical.

7. New developments and summary

In this document we discussed a framework of optimization for training neural networks. There are some areas that were not addressed, but the interested reader can consult some of the many existing references in literature, perhaps starting with references [1] and [2]. In actuality, neural network training has reached a level of maturity for multilayer perceptrons and radial-basis function networks, but a large unexplored area of new developments remains untouched.

Current research into neural networks involve spiking neurons, neurons on chip, new statistical methods and innovative ways with which to implement neurons and weights. Spiking neurons will be used to mimic neuronal activity in the human brain and do what they are good at, namely generating stochastic sequences of spikes with certain electrical and statistical properties. Spikes are combined in neurons to form more complex sequences with which to perform interesting functions such as recognizing words in speech recognition. New statistical methods include more formal models for describing the dynamics of neural networks in time series such as NARMAX modeling and others. Engineers are also attempting to find better ways of implementing neurons on chip, so that computational speeds are improved and size is reduced.

Finally, there exist in practice a few training algorithms that should be classified as zero-order algorithms due to their excessive computational overhead. However, there are examples where no gradient-based approach works, and random searches, genetic algorithms, Hebbian learning, and simulated annealing should be used. References [1] and [] provide sufficient background on these topics, in which references to more advanced literature are contained.

List of references

- [1] Simon Haykin *Neural Networks: A Comprehensive Foundation* MacMillan College Publishing Company Inc., London, 1994.
- [2] Bishop C.M. *Neural Networks for Pattern Recognition* Oxford University Press, New York, 1996.
- [3] Karras D.A. and Perantonis S.J. *An Efficient Constrained Training Algorithm for Feedforward Networks* IEEE Transactions on Neural Networks, Vol 6, No 6, Nov, 1995, pp 1420-1434.
- [4] Darken J. and Moody J. *Notes on learning rate schedules for stochastic optimization* Advances in Neural Information Processing Systems 3, 1991.
- [5] Fletcher R. *Practical Methods of Optimization* Vol 1, J. Wiley & Sons, Chichester, 1980
- [6] Powell M.J.D. *Restart Procedures for the Conjugate Gradient Method* Mathematical Programming, Vol 12, 1977, pp 241-254.
- [7] Snyman J.A. *An Improved Version of the Original Leap-Frog Dynamic Method for Unconstrained*

Minimization: LFOPI(b) Applied Mathematical Modeling, vol 7, 1983, pp 216-218.

[8] Holm J.E.W. and Botha E.C. *Sampled optimization improves on purely stochastic neural network training* Proceedings of the ICSC 2000, Berlin, 2000

[9] Johann E.W. Holm *Sampled optimization for training perceptron neural networks* PhD thesis, University of Pretoria, 1999.

[10] Johann E.W. Holm and Elizabeth C. Botha, *Leap-frog is a robust algorithm for training neural networks* Journal of Physics Publishing, Network: Computation in Neural Systems, Vol 10, No 1, 1999, pp 1-13.

[11] Cohn D. and Tesauro G. *Can neural networks do better than the Vapnik-Chervonenkis bounds?* Advances in Neural Information Processing Systems 3, 1991.

[12] Cohn D. and Tesauro G. *How Tight are the Vapnik-Chervonenkis Bounds?* Neural Computation, Vol 4, 1992, pp 249-269.

[13] Tang K.S. Man K.F. Kwong S. He Q. *Genetic Algorithms and their Applications* IEEE Signal processing magazine, November 1996, pp 22 - 37

[14] Holm J.E.W. and Kotze N.J.H. *Training Recurrent Neural Networks with Leap-frog* International Symposium on Industrial Electronics (ISIE'98), Vol 1, University of Pretoria, July, 1998, pp 99-104

[15] Kotze N.J.H. and Holm J.E.W. *Recurrent Neural Networks in Time Series Prediction* Proceedings of the eighth annual South African workshop on Pattern Recognition, Rhodes University, November, 1997, pp 105-111.

[16] Barnard E. and Holm J.E.W. *A comparative study of optimization techniques for backpropagation* Neurocomputing, Vol 6, 1994, pp 19-30.

ECONOMIC DESIGN OF WELDED I-BEAMS WITH PWT AND CELLULAR PLATES

Károly Jármai

University of Miskolc, H-3515 Miskolc, Egyetemváros, Hungary

e-mail: altjar@gold.uni-miskolc.hu

ABSTRACT

The cost optimization is important in structural design. Using various welding technologies the welding times and costs are different. The investigated structures are predominantly welded from plates. Examples are shown for design of welded I- beams and cellular plates. Fatigue fracture is one of the most dangerous phenomena for welded structures. In order to eliminate or decrease the danger of fatigue fracture several methods have been investigated. Post-welding treatments (PWTs) such as toe grinding, TIG-dressing, hammer peening and ultrasonic impact treatment (UIT) are the most efficient methods. The investigated cellular plates consist of two face sheets and some longitudinal ribs of square hollow section (SHS) welded between them using arc-spot welding technology. The cellular deck panels are subject to axial compression and a transverse load causing bending. In the optimization procedure the dimensions and number of longitudinal SHS ribs as well as the thickness of face sheets are sought which minimize the cost function and fulfil the design constraints. The design constraints relate to the stress due to compression and bending and to the eigenfrequency of the structure.

1. EFFECT OF POST-WELDING TREATMENTS ON THE OPTIMUM FATIGUE DESIGN OF WELDED I-BEAMS

Welding causes residual stresses and sharp stress concentrations around the weld, which are responsible for significant decrease of fatigue strength. Butt welds with partial penetration, toes and roots of fillet welds are points where fatigue cracks initiate and propagate. In order to eliminate or decrease the danger of fatigue fracture several post-welding treatments (PWTs) such as toe grinding, TIG-dressing, hammer peening and ultrasonic impact treatment (UIT) methods have been investigated. These methods have been tested and a lot of experimental results show their effectiveness and reliability. For designers it is important to know the measure of savings in structural weight and cost, which can be achieved by using these treatments. Optimum design is suitable for this task, since the additional cost of PWT can be included in the cost function and the improved fatigue stress range can be considered in the fatigue strength constraint. Thus, our aim is to illustrate this saving by means of a simple numerical example of a welded I-beam. In this case the transverse fillet welds used for vertical stiffeners decrease the fatigue stress range, thus the effect of PWT can be illustrated minimizing the cost function, which contains also the additional cost of PWT and the increased fatigue stress range can be included in the fatigue stress constraint.

1.1 IMPROVEMENT OF FATIGUE STRENGTH USING VARIOUS PWT-S

We have checked a number of articles, which either made measurements on different PWTs, or give data on the improvement in fatigue limit and about the speed of the process: Braid et al (1997), Woodley (1987), Janosch et al (1996), Huther et al (1996). For our purpose those publications are

suitable, in which data are given not only for the measure of improvement (α), but also for the time required for treatment (T_0). These data are summarized in Table 1.

Table 1. Measure of improvement and specific treatment time for various treatments according to the published data

Method	Reference	T_0 (min/m)	Improvement %	α	Remark
Grinding	Woodley (1987)	60	40	1.4	
TIG dressing	Horn (1998)	18	40	1.4	can be 70-100%
Hammer peening	Braid (1997)	4	100	2.0	can be 175-190%
UIT	Janosch (1996)	15	70	1.7	

It should be mentioned that we want to calculate with the minimum value of improvement. A value larger than 100% cannot be realized in our numerical example.

1.2 MINIMUM COST DESIGN OF A WELDED I-BEAM CONSIDERING THE IMPROVED FATIGUE STRESS RANGE AND THE ADDITIONAL PWT COST

In the investigated numerical example transverse vertical stiffeners are welded to a welded I-beam with double fillet welds. PWT is used only in the middle of the span, since near supports the bending stresses are small. The tension part of stiffeners in the middle of span is not welded to the lower flange and to the lower part of the web. Thus, the PWT is needed only for welds connecting the stiffeners to the upper flange (Fig.1). For this reason two types of stiffeners are used as it can be seen in Fig.1.

The beam is loaded by a pair of forces fluctuating in the range of $0 - F_{max}$, so the bending stress range is calculated from F_{max} .

1.2.1 THE COST FUNCTION

In our previous studies we have used a cost function containing the material and fabrication costs as follows:

$$K = K_m + K_f = k_m \rho V + k_f \sum T_i \quad (1)$$

where ρ is the material density, V is the volume of the structure, k_m and k_f are the corresponding cost factors, T_i are the fabrication times.

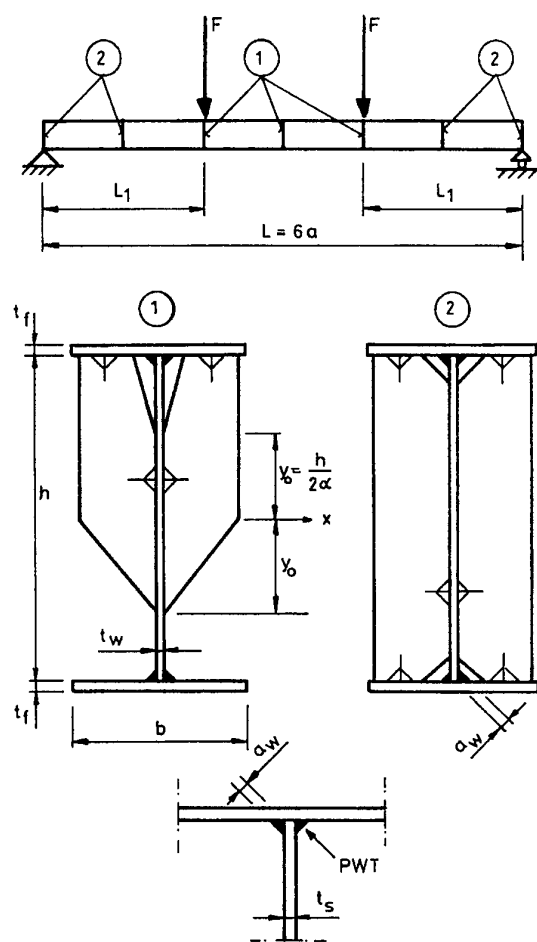


Fig. 1. Welded I-beam with vertical stiffeners. Double fillet welds with (1) and without (2) PWT

Eq.(1) can be written in the form of

$$\frac{K}{k_m} = \rho V + \frac{k_f}{k_m} \sum T_i \quad (2)$$

We use the following cost factors: $k_m = 0.5 - 1$ \$/kg, $k_{f,max} = 60$ \$/h = 1 \$/min, thus the ratio of k_f/k_m can be varied in a wide range of 0 - 2 kg/min. $k_f/k_m = 0$ means that K/k_m is a weight (mass) function, $k_f/k_m = 2$ kg/min can be used for developed countries.

The fabrication times can be calculated as follows:

$$\sum T_i = T_1 + T_2 + T_3 + T_4 \quad (3)$$

Time for preparation, assembly and tacking is

$$T_1 = C_1 \Theta_d \sqrt{\kappa \rho V} \quad (4)$$

where $C_1 = 1$ min/kg^{0.5}, Θ_d is a difficulty factor expressing the complexity of a structure (planar or spatial, consisting of plates or tubes etc.), κ is the number of elements to be assembled.

Time for welding is

$$T_2 = \sum C_{2i} a_{wi}^n L_{wi} \quad (5)$$

where $C_{2i} a_{wi}^n$ is given for different welding technologies and weld shapes according to COSTCOMP (1990) software and Jármai, Farkas (1999), a_w is the weld size, L_w is the weld length.

Time for additional works as deslagging, chipping and electrode changing is

$$T_3 = 0.3 T_2 \quad (6)$$

Time for PWT is

$$T_4 = T_0 L_t \quad (7)$$

T_0 is the specific time (min/mm), L_t is the treated weld length (mm).

The final form of the cost function is

$$\frac{K}{k_m} = \rho V + \frac{k_f}{k_m} \left(\Theta_d \sqrt{\kappa \rho V} + 1.3 \sum C_{2i} a_{wi}^n L_{wi} + T_0 L_t \right) \quad (8)$$

1.2.2 DESIGN CONSTRAINTS

The constraint on fatigue stress range can be formulated as

$$\frac{F_{\max} L_1}{W_x} \leq \frac{\alpha \Delta \sigma_C}{\gamma_{Mf}} \quad (9)$$

$$\text{where } W_x = \frac{I_x}{\frac{h}{2} + \frac{t_f}{2}}; \quad I_x = \frac{h^3 t_w}{12} + 2 b t_f \left(\frac{h}{2} + \frac{t_f}{2} \right)^2 \quad (10)$$

According to Eurocode 3 (EC3) (1992) the fatigue stress range for as welded structure is $\Delta \sigma_C = 80$ MPa, the fatigue safety factor is $\gamma_{Mf} = 1.25$. α expresses the measure of improvement

$$\alpha = \frac{\Delta \sigma_{Cimproved}}{\Delta \sigma_{Cswelded}}$$

The constraint on local buckling of the web according to EC3 is

$$\frac{h}{t_w} \leq 69 \varepsilon; \quad \varepsilon = \sqrt{\frac{235}{\alpha \Delta \sigma_C / \gamma_{Mf}}} \quad (11)$$

Note that we calculate in the denominator of ε with the maximum compressive stress instead of yield stress.

The constraint on local buckling of the compression flange is

$$\frac{b}{t_f} \leq 28\varepsilon \quad (12)$$

1.2.3 NUMERICAL EXAMPLE

Data: $F_{max} = 138$ kN, $L = 12$ m, $L_I = 4$ m, $\Delta\sigma_c / \gamma_{Mf} = 80 / 1.25 = 64$ MPa, $\varepsilon = 1.916 / \sqrt{\alpha}$; $\Theta_d = 3$; number of stiffeners is $2 \times 7 = 14$, thus $\kappa = 3 + 14 = 17$.

The volume of the structure is

$$V = (ht_w + 2bt_f)L + 4bht_s + 1.5bht_s \left(1 + \frac{1}{\alpha}\right); \quad t_s = 6 \text{ mm} \quad (13)$$

The second member expresses the volume of stiffeners without PWT, the third member gives the volume of stiffeners with PWT.

For longitudinal GMAW-C (gas metal arc welding with CO₂) fillet welds of size 4 mm we calculate with

$$C_2 a_w^n L_w = 0.3394 \times 10^{-3} \times 4^2 \times 4L = 260 \text{ min}, \quad (14)$$

for transverse SMAW (shielded metal arc welding) fillet welds the following formula holds

$$C_2 a_w^n L_w = 0.7889 \times 10^{-3} \times 4^2 \left[6 \left(b + \frac{2h}{\alpha} \right) + 16(b + h) \right] \quad (15)$$

For the constrained minimization of the nonlinear cost function the Rosenbrock's Hillelimb mathematical programming method is used complementing it with an additional search for optimum rounded discrete values of unknowns. The results of computation, i.e. the unknown dimensions h , t_w , b and t_f as well as the minimum costs for different values of k_f/k_m and α are given in Table 2.

Table 2. Optimum rounded dimensions in mm and K/k_m (kg) values for different k_f/k_m ratios for various PWT-s. $k_f/k_m = 0$ means the minimum weight design without effect of PWT

PWT	k_f/k_m (kg/min)	h	t_w	b	t_f	K/k_m (kg)
as welded	0	1300	10	320	14	2191
	1	1230	10	310	16	3802
	2	1230	10	310	16	5399
Grinding	1	940	9	340	15	3343
	2	890	8	300	19	4704
TIG dressing	1	1000	9	330	14	3235
	2	1110	10	310	12	4770
Hammer peening	1	820	9	310	13	2762
	2	820	9	310	13	3999
UIT	1	970	10	300	12	3021
	2	810	8	300	17	4202

It can be seen from Table 2. that with the various treatment methods the following cost savings can be achieved: grinding 14-15 %, TIG dressing 13-17 %, hammer peening 35-38 %, UIT 26-28 %.

Thus, the cost savings are significant the most efficient method is the hammer peening. It can be also seen, that PWT methods affect the optimum dimensions.

2 OPTIMUM DESIGN OF WELDED CELLULAR PLATES FOR SHIP DECK PANELS

Cellular plates consist of two face sheets and a grid of ribs welded between them. The main advantage of such plate structure is that the cells have a large torsional stiffness, which allows designers to construct plates of small height. The disadvantage of cellular plates lies in fabrication difficulty, since, when the height is smaller than 800-1000 mm, it is impossible to weld the ribs to the face sheets from inside.

Some applications of cellular plates are as follows: double bottoms of ships, rudders of ships, floating roofs of cylindrical storage tanks, box gates for dry docks, wings of aircraft structures, bridge decks, floating bridges, offshore platforms, elements of machine tool structures (press tables, mounting desks, base plates), mining shields, floors of buildings, lightweight roofs, etc.

Regarding the fabrication of cellular plates there are more possibilities to join the ribs to the face sheets. The simplest but not the cheapest solution is to use face plate elements and weld them to ribs from outside by fillet welds. Special welds such as arc-spot welds, slot or plug welds as well as electron-beam or laser welds can be used without cutting larger face sheet parts. Some applications combine the fillet and arc-spot welds.

The aim of present study is to work out a minimum cost design of such cellular plates considering, besides of stress constraint, the eigenfrequency constraint as well, and the fabrication cost of arc-spot welds. We applied square hollow section (SHS) instead of rectangular hollow section (RHS) to minimise the height of the panel. The main specialities of this application are as follows: 1) only longitudinal ribs of square hollow section (SHS) are used joined to the face sheets by arc-spot welding, thus, in the cost function the fabrication cost of arc-spot welds should be included; 2) to avoid the vibration resonance, the first eigenfrequency of the plate should be larger than a prescribed value.

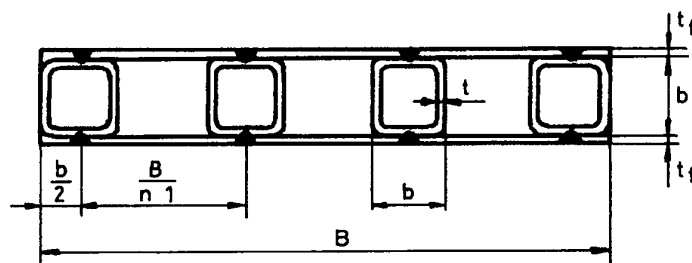


Fig.2. Cross-section of the ship deck panel investigated in longitudinal stiffeners

2.1 THE COST FUNCTION

The cross-section of the deck panel is shown in Fig.2. The cellular plate consists of two face sheets of thickness t_f and longitudinal SHS ribs of number n with dimensions of b and t .

In the longitudinal direction the plate ends are clamped and the panel is supported in two places, thus, it can be calculated as a three-span beam (Fig.3) loaded axially with a compression stress

$\sigma = N / A_{eff}$, A_{eff} being the effective cross-section for compression (Fig.5), and transversely by a uniformly distributed normal load of a factored intensity p .

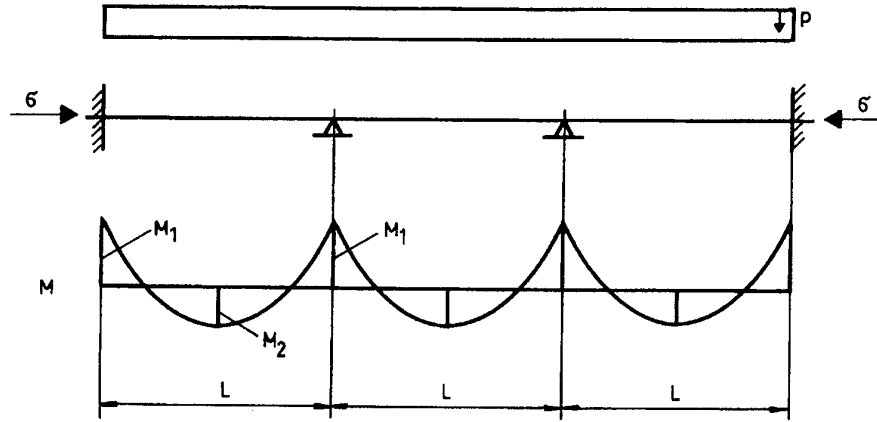


Fig.3. Bending moment diagram of the ship deck panel

The cost of a welded structure consists of material and fabrication cost according to Eq.(2).

The volume of the structure is

$$V = 3L(nA_{SHS} + 2Bt_f) \quad (16)$$

The cross-sectional area of a SHS is, considering the corner roundings according to a formula given by DAST (1986), approximately

$$A_{SHS} = 0.99 * 4(b - t)t \left(1 - 0.43 \frac{t}{b - 3t} \right) \quad (17)$$

At the fabrication time (Eq. 4) κ is the number of assembled structural elements, in our case it is $\kappa = n + 2$.

The time of arc-spot welding is given by

$$T_2 = n_s T_s \quad (18)$$

where n_s is the number of spots, T_s is the time of welding of one spot weld and of the electrode transfer to the next spot.

The additional time for deslagging, chipping and changing the electrode can be calculated as Eq.(6).

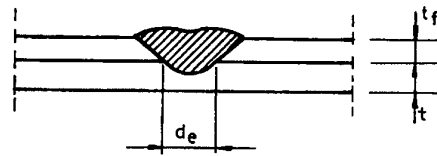


Fig.4. Effective diameter of an arc-spot weld

Since data for T_s cannot be found in literature, we take $T_s = 0.3$ min noting that it depends on the welding equipment and the degree of automation. The number of spots can be calculated by means of the spot pitch a .

The required minimum spot pitch can be determined considering a spot weld as a pin (Blodgett (1978), Füchsel et al. (1990)).

2.2 THE DESIGN CONSTRAINTS

Constraint on eigenfrequency

A serviceability constraint can be defined expressing that the first eigenfrequency of a simply supported bent beam of span length L should be larger than a prescribed value

$$f_1(\text{Hz}) = \frac{\pi}{2L^2} \left(\frac{10^3 EI_x}{m} \right)^{1/2} \geq f_0 \quad (19)$$

where E is the modulus of elasticity, the moment of inertia of the whole cross-section is

$$I_x = nI_{SHS} + Bt_f(b + t_f)^2 / 2 \quad (20)$$

According to DAST (1986) the moment of inertia of a SHS is approximately

$$I_{SHS} = \frac{2}{3}(b - t)^3 t \left(1 - 0.86 \frac{t}{b - 3t} \right) \quad (21)$$

In the mass m an additive mass m_{add} should be considered, thus

$$m = \rho(nA_{SHS} + 2Bt_f) + m_{add} \quad (22)$$

It should be mentioned that f_1 is in reality larger than it is calculated with formula (19) because the beam is clamped and not simply supported, but this approximation can be used since this constraint is not active.

Constraint on stability due to compression and bending

According to EC3, the stress constraint should be defined for a section of class 4 as follows:

$$\frac{N}{\chi A_{eff} f_{y1}} + \frac{k_x \psi M_1}{W_{\xi} f_{y1}} \leq 1 \quad f_{y1} = f_y / \gamma_{M1}; \gamma_{M1} = 1.1 \quad (23)$$

where χ is the overall buckling factor

$$\chi = \frac{1}{\phi + (\phi^2 - \bar{\lambda}^2)^{1/2}} \quad (24)$$

$$\phi = 0.5 \left[1 + 0.34(\bar{\lambda} - 0.2) + \bar{\lambda}^2 \right] \quad (25)$$

$$\bar{\lambda} = \frac{KL}{\lambda_1 r} \beta_A^{1/2} \quad (26)$$

for a beam with clamped ends $K = 0.5$,

$$\lambda_1 = \pi \left(E / f_y \right)^{1/2} \beta_A^{1/2}; \quad r = \left(I_{eff} / A_{eff} \right)^{1/2} \quad (27)$$

$$\beta_A = \frac{A_{eff}}{nA_{SHS} + 2Bt_f} \quad (28)$$

To obtain the effective cross-section, the effective width of face sheets should be calculated according to EC3

$$b_e = \rho_p \frac{B}{n - 1} \quad (29)$$

$$\bar{\lambda}_p = \frac{B / [(n - 1)t_f]}{28.4 \varepsilon k_{\sigma}^{1/2}}, \quad \varepsilon = (235 / f_y)^{1/2} \quad (30)$$

with $k_\sigma = 4$

$$\bar{\lambda}_p = \frac{B}{56.8\epsilon(n-1)t_f} \quad (31)$$

when $\bar{\lambda}_p \leq 0.673$ $\rho_p = 1$ (32a)

when $\bar{\lambda}_p \geq 0.673$ $\rho_p = \frac{1}{\bar{\lambda}_p} - \frac{0.22}{\bar{\lambda}_p^2}$ (32b)

Considering the effective cross-section shown in Fig.5 we get

$$A_{eff} = nA_{SHS} + 2B_e t_f, \quad B_e = b + (n-1)b_e \quad (33)$$

and $I_{eff} = nI_{SHS} + B_e t_f (b + t_f)^2 / 2$ (34)

According to the moment diagram shown in Fig.3

$$M_1 = BpL^2 / 12 \quad (35)$$

this bending moment should be multiplied by a dynamic factor ψ .

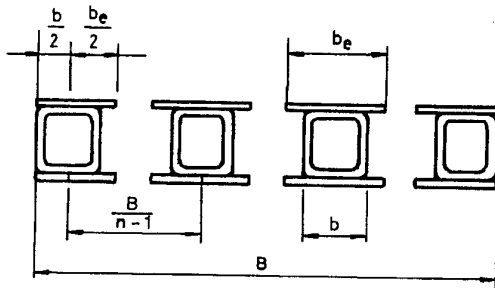


Fig.5. Effective cross-section for compression

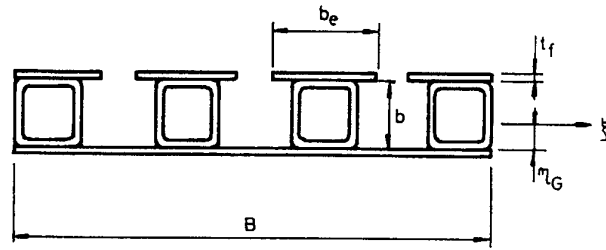


Fig.6. Effective cross-section for bending

$$k_x = 1 - \frac{\mu_x N}{\chi(nA_{SHS} + 2Bt_f)f_y} \quad \text{but } k_x \leq 1.5 \quad (36)$$

$$\mu_x = \bar{\lambda}(2\beta_M - 4) \quad \text{but } \mu_x \leq 0.9 \quad (37)$$

For our case it is $\beta_M = 1.3$ and $\mu_x = -1.4\bar{\lambda}$, thus

$$k_x = 1 + \frac{1.4\bar{\lambda}\beta_A N}{\chi f_y A_{eff}} \quad (38)$$

For bending another asymmetric effective cross-section should be taken into account as shown in Fig.6. The distance of gravity centre G is

$$\eta_G = \frac{nA_{SHS}(b+t_f)/2 + B_e t_f (b+t_f)}{nA_{SHS} + B - 2nt_f + B_e t_f} \quad (39)$$

The deduction of holes caused by arc-spot welds is considered for face sheet subject to tension by taking $B - 2nt_f$ instead of B .

The moment of inertia is given by

$$I_\xi = nI_{SHS} + nA_{SHS} \left(\frac{b+t_f}{2} - \eta_G \right)^2 + (B - 2nt_f)t_f \eta_G + B_e t_f (b+t_f - \eta_G)^2 \quad (40a)$$

the static moment is

$$S_\xi = b_e (b+t_f - \eta_G) \quad (40b)$$

and the section modulus is

$$W_{\xi} = \frac{I_{\xi}}{b + t_f - \eta_G} \quad (41)$$

Stress constraint for the upper face sheet

The upper face sheet is subject to static bending and compression in longitudinal direction as well as to bending in transverse direction. The stress due to longitudinal compression and bending is

$$\sigma_L = \frac{N}{A_{eff}} + \frac{\psi M_1}{W_{\xi}} \quad (42)$$

and the stress due to transverse bending, considering a plate strip with clamped edges of span length $B/(n-1) - b$, is

$$\sigma_T = \frac{\psi p}{2t_f^2} \left(\frac{B}{n-1} - b \right)^2 \quad (43)$$

The stress constraint can be expressed as

$$\left(\sigma_L^2 + \sigma_T^2 + \sigma_L \sigma_T \right)^{0.5} \leq f_{y1} \quad (44)$$

2.3 THE OPTIMIZATION PROCEDURE

In the minimum cost design the optimum values of b , t , t_f and n are sought, which minimize the cost function (2) and fulfil the design constraints (19), (23) and (44). In the first phase the above mentioned variables are treated as continuous ones and the optima are determined using the

Table 3. Optimization results for $f_y = 235$ MPa: number of ribs n , optimum dimensions in mm and K/k_m -values in kg for cost in function of the ratio k_f/k_m

k_f/k_m	n	t_f	b	t	K/k_m
0	4	8	120	3	1987
	5	4	120	3	1212
	6	4	30	2	916
	7	2.5	40	2	639
	8	2	50	2	582
	9	2	50	2	602
1	10	2	50	2	621
	4	8	120	3	2367
	5	4	120	3	1620
	6	4	30	2	1331
	7	2.5	40	2	1161
	8	2	50	2	1232
2	9	2.5	40	2	1306
	4	8	120	3	2747
	5	4	120	3	2027
	6	4	30	2	1745
	7	2.5	40	2	1683
	8	2.5	40	2	1813
	9	2.5	40	2	1943

Rosenbrock's Hillclimb mathematical programming method. In the second phase the discrete values of variables are calculated using a complementary search method. In this search the minimum values are taken as $n_{min} = 30$, $t_{min} = 2$, $t_{fmin} = 2$ mm and $n_{min} = 4$. The discrete values of SHS are sought according to the pre-standard prEN 10219-2 (1992).

Note that the minimum number of ribs $n_{min} = 4$ has been selected, since the transverse stiffness of the panel is in the case of $n = 3$ too small. In the case when the normal loading is not uniformly distributed in transverse direction, it would be necessary to use transverse ribs as well to avoid too large torsional deformations.

The numerical data are as follows: $f_0 = 18$ Hz, $E = 2.1 \cdot 10^5$ MPa, $B = 2000$, $L = 2250$ mm, $\Theta_d = 3$, $\rho = 7850$ kg/m³ = $7.85 \cdot 10^{-6}$ kg/mm³, $m_{add} = 2 \cdot 50 = 100$ kg/m = 0.1

kg/mm, $p = 3.5 \text{ kN/m}^2 = 3.5 \cdot 10^{-3} \text{ N/mm}^2$, $\psi = 1.4$, $\sigma = N / A_{eff} = 150 \text{ MPa}$.

The computational results are summarized in Table 3.

The optimum number of ribs is larger for minimum weight design ($k_f/k_m = 0$) i.e. $n = 8$ for $f_y = 235$ and $n = 6$ for $f_y = 355 \text{ MPa}$, than for minimum cost design ($k_f/k_m = 1$ or 2) i.e. $n = 7$ for $f_y = 235$ and $n = 6$ or 5 for $f_y = 355 \text{ MPa}$. The optimum number of ribs depends on f_y . Cost savings of 14-18% can be achieved using steel of yield stress 355 instead of 235 MPa.

The cost difference between the best and worst solution for $f_y = 235 \text{ MPa}$ and $k_f/k_m = 2$ is $100(2747-1683)/1683 = 63\%$, which emphasizes the importance of structural optimization. Calculations show that the stability and stress constraints are in most cases active and the eigenfrequency constraint is passive.

Acknowledgements

This work has been supported by the Hungarian Fund for Scientific Research grants OTKA 22846, 29326 and the Fund for the Higher Education grant 8/2000.

References

- Braid, J.E.M., Bell, R., Militaru, D.V. (1997): Fatigue life of as-welded, repaired, and hammer-peened joints in high-strength structural steel. *Welding in the World*, **39** (5), 248-261.
- Blodgett, O.W. (1978): Report on proposed standards for sheet steel structural welding. *Weld. J.* **57**, 15-24.
- COSTCOMP (1990): Programm zur Berechnung der Schweisskosten. Düsseldorf, Deutscher Verlag für Schweissttechnik.
- DAST (Deutscher Ausschuss für Stahlbau) (1986): Richtlinie 016. Bemessung und konstruktive Gestaltung von Tragwerken aus dünnwandigen kaltgeformten Bauteilen. Köln.
- Eurocode 3. (1992): Design of steel structures. Part 1.1. Brussels, CEN European Committee for Standardization.
- Farkas, J., Jármai, K. (1997): Analysis and optimum design of metal structures. Balkema, Rotterdam-Brookfield.
- Füchsel, S., Möbius, W. and Steinert, G. (1990): Empfehlung zur Berechnung von MAG-Punktschweißverbindungen. ZIS-Report, Halle, **1**, 31-36.
- Horn, A.A., Huther, I., Lieurade, H.P. (1998): Fatigue behaviour of T-joints improved by TIG dressing. *Welding in the World*, **41** (4), 273-280.
- Huther, I., Lieurade, H.P. et al. (1996): Analysis of results on improved welded joints. *Welding in the World* **37** (5), 242-266.
- Janosch, J.J., Koneczny, H. et al. (1996): Improvement of fatigue strength in welded joints (in HSS and aluminium alloys) by ultrasonic hammer peening. *Welding in the World*, **37** (2), 72-83.
- Jármai, K., Farkas, J. (1999): Cost calculation and optimization of welded steel structures. *Journal of Constructional Steel Research*, **50**, 115-135.
- prEN 10219-2 (1992): Cold formed structural hollow sections of non-alloy and fine grain structural steels. Part 2. Tolerances, dimensions and sectional properties. European Committee for Standardization, Brussels. German version DIN EN 10219 Teil 2. Entwurf. 1993.
- Woodley, C.C. (1987): Practical applications of weld toe grinding. In *Improving the Fatigue Strength of Welded Joints*. The Welding Institute, Abington, Cambridge, UK, 1983. 19-22.

THE USE OF THE DYNAMIC LEAPFROG ALGORITHM IN POWER SYSTEM STATE ESTIMATION

J.A. Jordaan

Potchefstroom University for Christian Higher Education, Potchefstroom

Prof. R. Zivanovic

Technikon Pretoria

1. ABSTRACT

The power system state estimation problem is very large, sparse and non-linear. The following methods to solve the problem will be tested and compared: Gauss-Newton and the dynamic leapfrog method. The IEEE 5-bus and IEEE 14-bus power systems are used to compare the different solvers. The dynamic method proves to be very robust.

2. INTRODUCTION

State estimation is the process of assigning a value to an unknown system state variable. It is based on a redundant set of imperfect measurements taken from that system. Minimising or maximising a selected criterion solves the state estimation problem. A commonly used and familiar criterion is that of minimising the sum of the squares of the differences between the estimated and measured values.

In a power system, the phasor voltage at the system nodes can be expressed in polar co-ordinate form, where the state variables are the voltage magnitudes (V) and relative phase angles (θ). The voltage can also be expressed in rectangular co-ordinate form where the real and imaginary parts of the voltage, e and f , would be the state variables. The relation between V , θ , e and f is as follows:

$$\underline{V} = V(\cos \theta + j \sin \theta) = e + jf. \quad (1)$$

This paper is organised as follows: Section 3 describes the basic principles of state estimation. Section 4 gives a description of the numerical solvers and section 5 gives some numerical results for a 5-bus and a 14-bus system. A conclusion on the most robust method will end the paper.

3. FUNDAMENTALS OF STATE ESTIMATION

3.1. STATE ESTIMATION MODEL

In power system state estimation, the measurement vector $\mathbf{z}$ is related to the state vector $\mathbf{x}$ through a non-linear model given by

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}, \quad (2)$$

where $\mathbf{h}(\mathbf{x})$ is the vector of measurement equations and $\mathbf{e}$ is the error vector that accounts for the uncertainty in the measurements and the model. The errors are usually assumed to be normally distributed random values with zero mean and a known covariance matrix $\mathbf{R} = \text{diag}(\sigma_1^2, \dots, \sigma_m^2)$. To estimate the state variables, we have to define a cost function. The cost function must then be minimised. This cost function is also known as an objective function or an error function. For the

purpose of this paper we shall consider a type of function, called the Quadratic-Tangent (QT) objective function. The QT-objective function is a function of the measurement residuals, where the residuals are defined in the following way:

$$\tilde{r}_{s_i} = \frac{r_{s_i}}{E_s}, \quad i = 1 \dots m, \quad (3)$$

where

$$r_{s_i} = \frac{r_i}{\sigma_i \omega_i}, \quad (4)$$

$$r_i = z_i - h_i(\mathbf{x}), \text{ and} \quad (5)$$

$$E_s = \text{median}_i |r_{s_i}|. \quad (6)$$

The residual r_i is the difference between the i -th measurement z_i taken from the network and the computed value $h_i(\mathbf{x})$ of the corresponding measured quantity. The weight, σ_i , takes into account the standard deviation of measurement errors and ω_i is for measurements that are classified as leverage points. E_s is the scaling factor for the residuals. The objective function is defined as follows [1,2]:

$$\begin{aligned} J(\mathbf{x}) &= \boldsymbol{\omega}^T \boldsymbol{\rho}(\tilde{\mathbf{r}}_s) \boldsymbol{\omega} \\ &= \omega_1^2 \rho(\tilde{r}_{s_1}) + \omega_2^2 \rho(\tilde{r}_{s_2}) + \dots + \omega_m^2 \rho(\tilde{r}_{s_m}), \end{aligned} \quad (7)$$

where

$$\rho(\tilde{r}_{s_i}) = \begin{cases} \tilde{r}_{s_i}^2 / 2 & \text{if } |\tilde{r}_{s_i}| \leq \beta \\ \beta |\tilde{r}_{s_i}| - \beta^2 / 2 & \text{if } |\tilde{r}_{s_i}| > \beta \end{cases}, \quad (8)$$

$$\boldsymbol{\rho}(\tilde{\mathbf{r}}_s) = \text{diag} \{ \rho(\tilde{r}_{s_1}), \rho(\tilde{r}_{s_2}), \dots, \rho(\tilde{r}_{s_m}) \}, \quad (9)$$

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_n]^T, \quad (10)$$

β is the break-even point of the non-quadratic function $\rho(\tilde{r}_{s_i})$, n is the number of state variables and m is the number of measurements taken from the system. The following figure shows a graph of the QT-objective function:

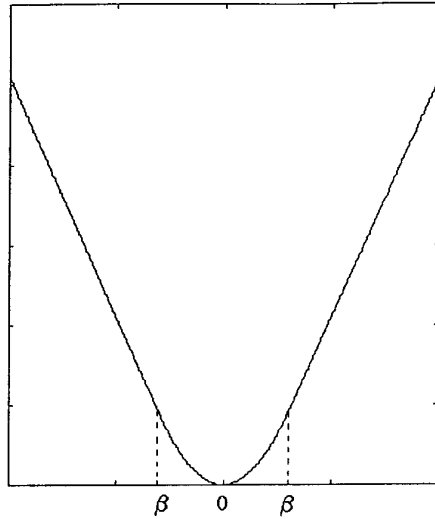


Figure 1: QT-objective function

To minimise the objective function, we can calculate its gradient with respect to the state variables and set it equal to zero [3]:

$$\nabla J = \mathbf{J}_x(\mathbf{x}) = \mathbf{0}. \quad (11)$$

This produces a system of non-linear equations that can be solved by the so-called Newton methods. We will look at the Gauss-Newton method.

The objective function can also be minimised by using the so-called gradient methods. The dynamic leapfrog method will be used.

3.2. MEASUREMENT EQUATIONS

In state estimation we use various types of measurements to solve the state variables. Each measurement is related to the state variables by a measurement equation. These equations can be expressed in polar or rectangular co-ordinate form. Some types of the measurements that we use, will now be discussed. The following figure illustrates the terminology that will be used:

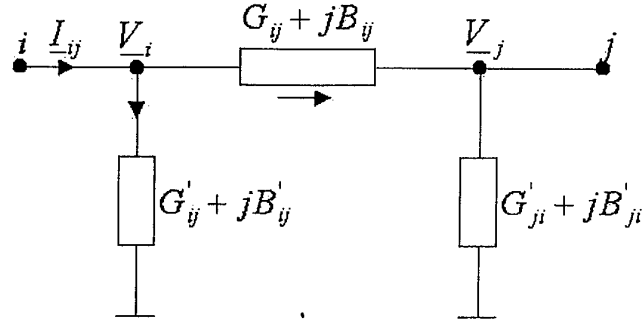


Figure 2: π -model of transmission line

This figure shows the π -model of a transmission line, where $G_{ij} + jB_{ij}$ is the series admittance of the line ij and $G'_{ij} + jB'_{ij} = G'_{ji} + jB'_{ji}$ is the shunt admittance of line ij (take note that j in the front of jB_{ij} means $j = \sqrt{-1}$, while the other j is an index). The active power flow in rectangular form from node i to j is given by

$$P_{ij}^M = (G_{ij} + G'_{ij})(e_i^2 + f_i^2) - f_i(f_j G_{ij} + e_j B_{ij}) - e_i(e_j G_{ij} - f_j B_{ij}) \quad (12)$$

and the reactive power flow by

$$Q_{ij}^M = -(B_{ij} + B'_{ij})(e_i^2 + f_i^2) - f_i(e_j G_{ij} - f_j B_{ij}) + e_i(f_j G_{ij} + e_j B_{ij}). \quad (13)$$

The power injection at a specific node equals the sum of all power flows into that node. The active power injection at any node i is

$$P_i^M = G_{ii}(e_i^2 + f_i^2) - f_i \sum_{j=1, j \neq i} (f_j G_{ij} + e_j B_{ij}) - e_i \sum_{j=1, j \neq i} (e_j G_{ij} - f_j B_{ij}) \quad (14)$$

and the reactive power injection is

$$Q_i^M = -B_{ii}(e_i^2 + f_i^2) - f_i \sum_{j=1, j \neq i} (e_j G_{ij} - f_j B_{ij}) + e_i \sum_{j=1, j \neq i} (f_j G_{ij} + e_j B_{ij}) \quad (15)$$

where $G_{ii} = \sum_{j \neq i} (G_{ij} + G'_{ij})$ and $B_{ii} = \sum_{j \neq i} (B_{ij} + B'_{ij})$.

Besides power measurements, we can also use voltage measurements:

$$V_i^M = \sqrt{e_i^2 + f_i^2} \quad (16)$$

where M indicates a measured value.

For any node i we can rewrite these equations in polar co-ordinate form by using $e_i = V_i \cos \theta_i$ and $f_i = V_i \sin \theta_i$.

4. NUMERICAL SOLVERS

4.1. GAUSS-NEWTON METHOD

To solve the system of non-linear equations in (11), we linearise the system around some working point $\mathbf{x}_k$ by using a Taylor series expansion:

$$\mathbf{J}_{xx}(\mathbf{x}_k) \Delta \mathbf{x}_k + \mathbf{J}_x(\mathbf{x}_k) = \mathbf{0}, \quad (17)$$

where $\mathbf{J}_{xx}(\mathbf{x}_k)$ is the Hessian matrix of the objective function J . The state correction vector $\Delta \mathbf{x}_k$ can then be calculated from the following system of linear equations:

$$\mathbf{J}_{xx}(\mathbf{x}_k) \Delta \mathbf{x}_k = -\mathbf{J}_x(\mathbf{x}_k). \quad (18)$$

An iterative procedure can be used and better iterates of the state vector can be calculated by [3,4]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k. \quad (19)$$

We need to derive expressions for the gradient vector and the Hessian matrix. Let n be the number of state variables. The gradient vector is:

$$\mathbf{J}_x = \left[\frac{\partial J}{\partial x_1}, \frac{\partial J}{\partial x_2}, \dots, \frac{\partial J}{\partial x_n} \right]^T = \mathbf{H}^T \mathbf{D} \mathbf{W}^2 \hat{\mathbf{e}}, \quad (20)$$

where for $i=1, 2 \dots m$ and $j=1, 2 \dots n$ follows

$$[\mathbf{H}]_{ij} = \frac{\partial \tilde{r}_{s_i}}{\partial x_j} = \frac{\partial \left\{ \frac{z_i - h_i(\mathbf{x})}{\sigma_i \omega_i E_s} \right\}}{\partial x_j} = -\frac{1}{\sigma_i \omega_i E_s} \frac{\partial h_i(\mathbf{x})}{\partial x_j}, \quad (21)$$

and the elements of the diagonal matrices are

$$[\mathbf{D}]_{ii} = \frac{\partial \rho}{\partial \tilde{r}_{s_i}} = \begin{cases} \tilde{r}_{s_i} & \text{if } |\tilde{r}_{s_i}| \leq \beta \\ \text{sign}(\tilde{r}_{s_i}) \cdot \beta & \text{if } |\tilde{r}_{s_i}| > \beta \end{cases}, \quad (22)$$

$$[\mathbf{W}]_{ii} = \omega_i. \quad (23)$$

$\hat{\mathbf{e}}$ is an m dimensional column vector with all its entries equal to one. The Hessian matrix is given by:

$$\mathbf{J}_{xx} = \alpha_k \sum_{i=1}^m \left(\omega_i^2 \frac{\partial \rho}{\partial \tilde{r}_{s_i}} \mathbf{G}_i \right) + \mathbf{H}^T \tilde{\mathbf{D}} \mathbf{W}^2 \mathbf{H}, \quad (24)$$

where for $i=1,2\dots m$ and $j,k=1,2\dots n$ follows

$$[\mathbf{G}_i(\mathbf{x})]_{jk} = \frac{\partial^2 \tilde{r}_{s_i}}{\partial x_j \partial x_k} = \frac{\partial^2 \left\{ \frac{z_i - h_i(\mathbf{x})}{\sigma_i \omega_i E_s} \right\}}{\partial x_j \partial x_k} = -\frac{1}{\sigma_i \omega_i E_s} \frac{\partial^2 h_i(\mathbf{x})}{\partial x_j \partial x_k}, \quad (25)$$

and the elements of the diagonal matrix are

$$[\tilde{\mathbf{D}}]_{ii} = \frac{\partial^2 \rho}{\partial \tilde{r}_{s_i}^2} = \begin{cases} 1 & \text{if } |\tilde{r}_{s_i}| \leq \beta \\ 0 & \text{if } |\tilde{r}_{s_i}| > \beta \end{cases}. \quad (26)$$

The parameter α_k in (24) determines the amount of second order information we use in constructing the Hessian. If we use equation (24) for the Hessian matrix, we are using the full Newton method. But to calculate the second order derivative matrix $\mathbf{G}_i(\mathbf{x})$ takes a lot of computational time if the system to solve gets big.

The Gauss-Newton method approximates the Hessian matrix by setting the second order derivative term $\mathbf{G}_i(\mathbf{x})$ equal to zero. The Hessian then becomes

$$\mathbf{J}_{xx} = \mathbf{H}^T \tilde{\mathbf{D}} \mathbf{W}^2 \mathbf{H}. \quad (27)$$

This method has fast local convergence on mildly non-linear problems, but it may also be non-convergent for very non-linear problems or problems that have large residuals [4].

4.2. DYNAMIC LEAPFROG METHOD

This method differs conceptually from the other gradient methods, like the conjugate gradient method. It considers the analogous physical problem of the motion of a particle of unit mass in an n -dimensional conservative force field. The potential energy is represented by the function $J(\mathbf{x})$ to be minimised. In such a force field the total energy of a particle is conserved. The total energy consists of the kinetic and potential energy. This method simulates the motion of the particle in the force field and by monitoring the kinetic energy an interfering strategy is adopted such that the potential energy is systematically reduced. The particle is thus forced to follow a trajectory to the local minimum in the potential energy.

The method can be stated as follows [5]: given initial values $\mathbf{x}_0, \mathbf{v}_0$ and time step Δt , compute in each iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k \Delta t, \quad (28)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{a}_{k+1} \Delta t, \quad (29)$$

$$\mathbf{a}_k = \nabla J_k. \quad (30)$$

5. NUMERICAL RESULTS

MATLAB was used to implement the prototypes of the different numerical solvers for the state estimation problem. The computer used in the simulations was an Intel Celeron 300 MHz. The state estimators were tested on IEEE 5-bus and 14-bus cases, having 22 and 70 measurements respectively [6]. The rectangular co-ordinates of the voltage were used as state variables.

For the IEEE 5-bus the following errors were generated:

1. the active power injection measurement at node 3 was changed from -0.4500 to 10.0000
2. the reactive power injection measurement at node 1 was changed from -0.0742 to 10.0000
3. the reactive power injection measurement at node 4 was changed from -0.0500 to 10.0000
4. the active power injection measurement (P_i) at node 1 was changed from 1.2959 to : -8.0, -5.0, 1.2959, 5.0, 10.0

Errors 1, 2 and 3 were kept constant, while error 4 was changed according to the five values indicated. All the errors were applied at the same time. The number of steps and CPU time (in seconds) until convergence are shown in Table 1 for the above cases.

Table 1: Results of IEEE 5-bus test case

P_i	Gauss-Newton		Dynamic	
	Steps	Time	Steps	Time
-8	90	12.7	398	47.0
-5	97	13.6	484	57.4
1.2959	174	24.1	544	64.4 *
5	No convergence		490	58.0
10	No convergence		835	98.4

* Did not converge to the correct solution

For the IEEE 14-bus the following errors were generated:

1. the active power injection measurement at node 1 was changed from 1.8989 to 10.0000
2. the active power injection measurement at node 2 was changed from -0.8101 to 10.0000
3. the active power injection measurement at node 6 was changed from -0.1142 to 10.0000
4. the reactive power injection measurement at node 3 was changed from 0.0335 to 10.0000
5. the reactive power injection measurement at node 5 was changed from -0.0147 to 10.0000
6. the reactive power injection measurement (Q_i) at node 1 was changed from -0.0555 to : -10.0, -5.0, -0.0555, 5.0, 10.0

Errors 1 to 5 were kept constant, while error 6 was changed according to the five values indicated. All the errors were applied at the same time. The number of steps and CPU time (in seconds) until convergence are shown in Table 2.

Table 2: Results of IEEE 14-bus test case

Q_i	Gauss-Newton		Dynamic	
	Steps	Time	Steps	Time
-10	150	143.6	1598	1447.0
-5	148	140.9	1598	1454.5
-0.0555	140	134.0	2438	2399.4
5	113	108.6	225	206.4
10	127	122.0	3313	3000.0

In one of the 5-bus simulations the dynamic method converged to the incorrect solution. The method probably found a local minimum of the objective function instead of the global minimum. Although for the 14-bus simulations both methods converged for all cases, the dynamic method sometimes found a better solution. But both methods' solutions were good enough to detect the bad measurements by looking at the residuals defined by equation (5).

6. CONCLUSIONS

In this paper we formulated the non-linear power system state estimation problem with quadratic-tangent objective function. The Gauss-Newton and dynamic leapfrog methods were tested. In some cases where the Gauss-Newton method failed to converge, the dynamic method did converge. We can use the Gauss-Newton method to solve the state estimation problem, and when it fails to converge, switch to the dynamic method. Although the dynamic method is very slow to obtain convergence, it can be used in situations where convergence must be obtained.

7. REFERENCES

- [1] R.C. Pires, A.S. Costa and L. Mili, "Iteratively Re-weighted Least-Squares State estimation through Givens Rotations", *IEEE-PES Summer Meeting*, San Diego, 12-16 July, 1998.
- [2] R. Baldick, K.A. Clemens, Z. Pinjo-Dzagal and P.W. Davis, "Implementing Nonquadratic Objective Functions for State Estimation and Bad Data Rejection", *IEEE Transactions on Power Systems*, Vol. 12, No. 1, pp. 376-382, February 1997.
- [3] R.A.M. van Amerongen, "On Convergence Analysis and Convergence Enhancement of Power System Least Squares State Estimators", *IEEE-PES Winter Meeting*, New York, 29 January - 2 February, 1995.
- [4] Björck, *Numerical Methods for Least Squares Problems*, SIAM, 1997.
- [5] J.A. Snymann, "A new and dynamic method for unconstrained minimisation", *App. Math Modeling*, Vol. 6, pp. 449-462, 1982.
- [6] Y. Wallach, *Calculations & Programs for Power System Network*, Wayne State University, Prentice-Hall, 1986.

PARAMETER ESTIMATION OF A POLYCRYSTAL MODEL THROUGH IDENTIFICATION STUDIES

S. KOK

A.J. BEAUDOIN

D.A. TORTORELLI

*Department of Mechanical and Industrial Engineering,
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

ABSTRACT

A temperature and rate-dependent viscoplastic polycrystal model is presented. The individual crystal response is used to obtain the macroscopic response through the extended Taylor hypothesis. A Newton-Raphson algorithm is used to solve the set of fully implicit nonlinear equations for each crystal. Material parameter estimates are obtained through an identification study, where the error between experimental and computed stress response is minimized. The BFGS method, which is used to solve the identification problem, requires first-order gradients. These gradients are computed efficiently via the direct method of design sensitivity analysis. Texture augmentation is performed in a second identification study by changing crystal weights (volume fractions).

1 INTRODUCTION

One of the first attempts to model the plastic behavior of polycrystalline metals, based on dislocation glide on slip systems of single crystals, was made by Taylor [1]. This pioneering work has been extended by numerous authors [2, 3, 4] to model the rate-dependent response of polycrystal aggregates.

The primary components of all polycrystal models are the constitutive behavior of single crystals and a mean field hypothesis which delivers the macroscopic response, given the individual crystal response. We make use of a multiplicative kinematic decomposition of the deformation gradient into elastic and plastic parts [4]. In rigid-viscoplastic models, the elastic part is reduced to rotation of the lattice. Considering the mean field hypotheses, we employ the Taylor hypothesis [1, 4], in which the deformation gradient in all grains in a material point region are assumed to be equal to the macroscopic deformation gradient.

The main contribution of this paper is a material parameter identification procedure which uses experimental stress-strain data at various temperatures, strain rates and loading conditions (e.g. compression, tension, torsion) simultaneously. The experimental data is compared to the calculated stress response of a polycrystal plasticity model [5] which includes temperature and rate effects. The material parameters are adjusted to minimize the error between the experimental and calculated response by solving an inverse problem. A similar procedure has previously been used by Chastel *et al* [6] and Signorelli *et al* [7] to calculate critical resolved shear stress (CRSS) ratios for a Taylor model and selfconsistent model, respectively. However, only a single load condition was considered in these efforts.

In the numerical implementation of our model, we evaluate the deviatoric stress and the crystal hardness and rotation state variables at each time step. The set of fully implicit nonlinear equations are coupled and solved simultaneously through a Newton-Raphson scheme. The crystal lattice rotation is integrated implicitly, whereas the hardness state variable is integrated analytically. The algorithm appears to be stable, and it exhibits results that are independent of

temporal discretization for constant strain rate simulations. Response gradients are calculated efficiently through the direct method of design sensitivity analysis and are used in an identification study to find material parameter estimates for HY100 steel and Tantalum. We also present an example where the volume fractions of individual crystals, used to represent initial texture of a rolled tantalum plate, are modified slightly through an identification study.

2 RATE DEPENDENT TAYLOR MODEL

We first consider the response of a single crystal. Given a prescribed velocity gradient $\mathbf{L}$ and the current state variables (crystal orientation and 'flow stress'), we need to compute the crystal Cauchy stress $\mathbf{T}$ and the evolution of the state variables for a given increment of time.

Neglecting elastic effects, the deformation gradient $\mathbf{F}$ is expressed as (see Figure 1)

$$\mathbf{F} = \mathbf{R}\mathbf{F}_p \quad \text{with} \quad \det \mathbf{R} = 1, \quad \det \mathbf{F}_p = 1. \quad (1)$$

Here $\mathbf{F}_p$ represents the plastic deformation gradient, due solely to the cumulative effect of dislocation motion on active slip systems and $\mathbf{R}$ represents the crystal lattice rotation. A consequence of neglecting elasticity is that only the deviatoric part of the Cauchy stress tensor, $\mathbf{T}' = \mathbf{T} - \frac{1}{3}\text{tr}(\mathbf{T})\mathbf{I}$, can be determined constitutively since the deformation is isochoric.

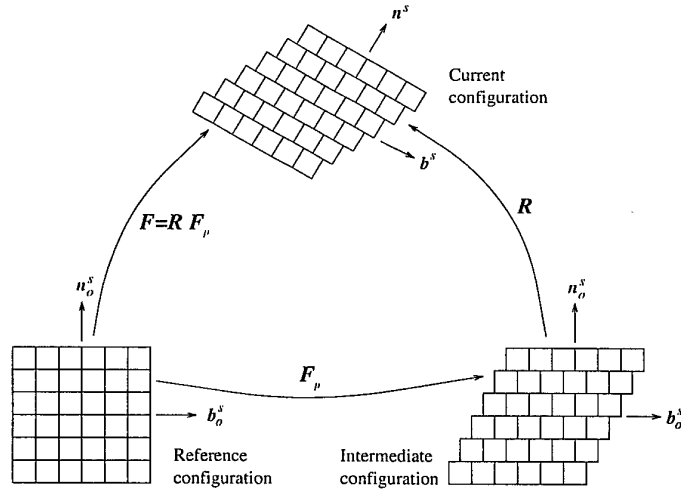


Figure 1: Schematic diagram of the multiplicative decomposition $\mathbf{F} = \mathbf{R}\mathbf{F}_p$

Using Eq.(1), the velocity gradient in the current configuration, $\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}$, is expressed as

$$\mathbf{L} = \dot{\mathbf{R}}\mathbf{R}^T + \mathbf{R}\dot{\mathbf{F}}_p\mathbf{F}_p^{-1}\mathbf{R}^T \quad (2)$$

where $\mathbf{L}_p = \dot{\mathbf{F}}_p\mathbf{F}_p^{-1}$ is the plastic velocity gradient with respect to the intermediate configuration. We assume that all plastic straining is due to slip on slip planes, hence $\mathbf{L}_p$ is expressed as

$$\mathbf{L}_p = \sum_{s=1}^M \dot{\gamma}^s \mathbf{b}_o^s \otimes \mathbf{n}_o^s. \quad (3)$$

Here $\dot{\gamma}^s$ is the shear rate of slip system s and $\mathbf{b}_o^s$ and $\mathbf{n}_o^s$ are unit vectors in the reference configuration, directed along the slip direction and along the normal to the slip plane, respectively. The summation is performed over the M potentially active slip systems.

Expressing $\mathbf{L} = \mathbf{W} + \mathbf{D}$ as the sum of skew $\mathbf{W} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^T)$ and symmetric $\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T)$ components, Eq.(2) is rewritten as

$$\mathbf{D} = \mathbf{R} \left(\sum_{s=1}^M \dot{\gamma}^s \mathbf{m}_o^s \right) \mathbf{R}^T \quad \text{and} \quad \mathbf{W} = \dot{\mathbf{R}} \mathbf{R}^T + \mathbf{R} \left(\sum_{s=1}^M \dot{\gamma}^s \mathbf{q}_o^s \right) \mathbf{R}^T, \quad (4)$$

where $\mathbf{m}_o^s = \frac{1}{2}(\mathbf{S}_o^s + \mathbf{S}_o^{sT})$ and $\mathbf{q}_o^s = \frac{1}{2}(\mathbf{S}_o^s - \mathbf{S}_o^{sT})$ are the symmetric and skew parts of the Schmid tensor $\mathbf{S}_o^s = \mathbf{b}_o^s \otimes \mathbf{n}_o^s$. The lattice rotation evolution equation is obtained by solving Eq.(4.2) for $\dot{\mathbf{R}}$,

$$\dot{\mathbf{R}} = \mathbf{W} \mathbf{R} + \mathbf{R} \mathbf{A} \quad \text{with} \quad \mathbf{A} = - \sum_{s=1}^M \dot{\gamma}^s \mathbf{q}_o^s. \quad (5)$$

To proceed further, we require a constitutive expression for the slip system shear rate $\dot{\gamma}^s$ as a function of the traction component τ^s acting on the slip plane in the slip direction. Using the transformations $\mathbf{b}^s = \mathbf{R} \mathbf{b}_o^s$ and $\mathbf{n}^s = \mathbf{R} \mathbf{n}_o^s$, with $\mathbf{b}^s$ and $\mathbf{n}^s$ unit vectors in the current configuration, τ^s is expressed as

$$\tau^s = \tilde{\tau}^s(\mathbf{T}', \mathbf{R}) = \mathbf{T}' : (\mathbf{b}^s \otimes \mathbf{n}^s) = \mathbf{T}' : (\mathbf{R} \mathbf{m}_o^s \mathbf{R}^T). \quad (6)$$

We use a modified power law [3, 4] to describe $\dot{\gamma}^s$ as a function of τ^s , i.e.

$$\dot{\gamma}^s = \dot{\gamma}^s(\tau^s) = \dot{\epsilon} \left| \frac{\tau^s}{\tau_o} \right|^n \text{sign}(\tau^s) \quad (7)$$

where τ_o is the slip system flow stress and $\dot{\epsilon} = \sqrt{\frac{2}{3} \mathbf{D} : \mathbf{D}}$ is the equivalent scalar strain rate. The slip system flow stress τ_o is expressed as a function of strain rate and temperature through the mechanical threshold stress (MTS) model [8] as

$$\frac{\tau_o}{\mu} = \frac{\tau_a}{\mu} + S_i(\dot{\epsilon}, T) \frac{\hat{\tau}_i}{\mu_o} + S_\epsilon(\dot{\epsilon}, T) \frac{\hat{\tau}_\epsilon}{\mu_o}. \quad (8)$$

In the above $\hat{\tau}_i$ describes the thermal portion of the yield stress (which does not evolve) and $\hat{\tau}_\epsilon$ the dislocation structure (which does evolve). μ_o is a reference value of the shear modulus μ , which is modeled by [9]

$$\mu = \tilde{\mu}(T) = \mu_o - \frac{D_o}{\exp\left(\frac{T_o}{T}\right) - 1}, \quad (9)$$

in which T_o and D_o are empirical constants. S_i and S_ϵ are scaling factors given by

$$S_i = \left[1 - \left(\frac{kT}{g_{oi}\mu b^3} \ln \frac{\dot{\epsilon}_{oi}}{\dot{\epsilon}} \right)^{1/q_i} \right]^{1/p_i} \quad S_\epsilon = \left[1 - \left(\frac{kT}{g_{oe}\mu b^3} \ln \frac{\dot{\epsilon}_{oe}}{\dot{\epsilon}} \right)^{1/q_\epsilon} \right]^{1/p_\epsilon}. \quad (10)$$

Here k is the Boltzmann constant, b is the magnitude of the Burger's vector, g_o is the normalized activation energy for dislocations to overcome the obstacles, $\dot{\epsilon}_o$ is a constant and p and q are statistical constants [10]. The evolution of $\hat{\tau}_\epsilon$ is given in rate form as

$$\frac{d\hat{\tau}_\epsilon}{dt} = \theta = \theta_o \frac{\mu}{\mu_o} \left(1 - \frac{\hat{\tau}_\epsilon}{\hat{\tau}_{\epsilon s}} \right)^\kappa \sum_{s=1}^M |\dot{\gamma}^s| \quad (11)$$

where θ_o is the initial hardening rate, $\hat{\tau}_{\epsilon s}$ is the saturation threshold stress and κ is a fitted constant. The saturation threshold stress $\hat{\tau}_{\epsilon s}$ is a function of both strain rate and temperature, through the relation [11]

$$\ln \frac{\dot{\epsilon}}{\dot{\epsilon}_{\epsilon so}} = \frac{g_{o\epsilon s} \mu b^3}{kT} \ln \frac{\hat{\tau}_{\epsilon s}}{\hat{\tau}_{\epsilon so}} \quad (12)$$

where $\dot{\epsilon}_{\epsilon so}$, $g_{o\epsilon s}$ and $\hat{\tau}_{\epsilon so}$ are empirically obtained constants.

Substituting Eq.(7) into Eq.(4.1), $\mathbf{T}'$ can be solved from the nonlinear equation

$$\mathbf{D} = \mathcal{P} \mathbf{T}' \quad (13)$$

where $\mathcal{P} = \tilde{\mathcal{P}}(\mathbf{T}', \mathbf{R})$ is a fourth order tensor given by

$$\mathcal{P} = \tilde{\mathcal{P}}(\mathbf{T}', \mathbf{R}) = \sum_{s=1}^M \frac{\dot{\epsilon}}{\tau_o} \left| \frac{\tilde{\tau}^s(\mathbf{T}', \mathbf{R})}{\tau_o} \right|^{n-1} (\mathbf{R} m_o^s \mathbf{R}^T) \otimes (\mathbf{R} m_o^s \mathbf{R}^T). \quad (14)$$

The response of a polycrystal is now discussed. To obtain the macroscopic response, we use the extended Taylor hypothesis [1], i.e. the velocity gradient of all crystals are equal to the macroscopic velocity gradient (or equivalently the deformation gradient in each grain is homogeneous and equal to the macroscopic deformation gradient in each material region). In such a deformation driven problem, the macroscopic deviatoric stress $\bar{\mathbf{T}}'$ is then obtained from the individual crystal stress responses according to [12]

$$\bar{\mathbf{T}}' = \sum_{c=1}^N w^c \mathbf{T}'^c / \sum_{c=1}^N w^c \quad (15)$$

where $\mathbf{T}'^c$ and w^c are the deviatoric stress and volume fraction of crystal c respectively and N is the total number of crystals that are considered to affect the macroscopic stress at a given material point.

3 NUMERICAL IMPLEMENTATION

Using subscripts $n+1$ and n to indicate variables at time $t + \Delta t$ and t respectively, the problem is stated as: Given the velocity gradient $\mathbf{L}_{n+1}$ and the state variables $\mathbf{R}_n$ and $\hat{\tau}_n^\epsilon$, compute the deviatoric crystal stress $\mathbf{T}'_{n+1}$ and the updated state variables $\mathbf{R}_{n+1}$ and $\hat{\tau}_{n+1}^\epsilon$ which satisfy, in a discrete setting, Eqs.(5), (11) and (13). To perform the identification study, we also need to compute the derivatives $\frac{D\mathbf{T}'_{n+1}}{D\mathbf{d}}$, given $\frac{D\mathbf{T}'_n}{D\mathbf{d}}$, where $\mathbf{d}$ is a vector containing the constitutive model parameters.

We choose a fully implicit method to integrate the lattice rotation evolution equation (cf. Eq.(5)), while we integrate the crystal hardness evolution equation (cf. Eq.(11)) analytically. The implicit integration scheme is chosen in part due to its unconditionally stability, and hence large timesteps are possible, but even more importantly it allows for efficient evaluation of design sensitivities. The Newton-Raphson method is selected to solve the nonlinear equations, since the decomposed Jacobian is used to compute the response sensitivities through a simple backsubstitution.

To solve this system of equations at each iteration i we first solve Eq.(5) for $\mathbf{R}_{n+1}^i$. We also evaluate the derivatives $\left(\frac{\partial \mathbf{R}_{n+1}}{\partial \hat{\tau}_{n+1}^\varepsilon}\right)^i$ and $\left(\frac{\partial \mathbf{R}_{n+1}}{\partial \mathbf{T}_{n+1}'}\right)^i$ because they are needed to solve Eqs.(13) and (11). The updated lattice rotation $\mathbf{R}_{n+1}^i$ is calculated from [13]

$$\mathbf{R}_{n+1}^i = \exp\left(\mathbf{W}_{n+1}\Delta t\right)\mathbf{R}_n\exp\left(-\sum_{s=1}^M\dot{\gamma}_{n+1}^s\mathbf{q}_o^s\Delta t\right). \quad (16)$$

Numerous closed-form expressions for the exponential map which appear in Eq.(16) exist, here we use quaternion parameters [14]. Since $\mathbf{R}_{n+1}^i$ is an explicit function of $\dot{\gamma}_{n+1}^s$, which is an explicit function of the stress $\mathbf{T}_{n+1}'$ and hardness $\hat{\tau}_{n+1}^\varepsilon$ (cf. Eqs.(6), (7) and (8)), it is a straightforward calculation to evaluate $\left(\frac{\partial \mathbf{R}_{n+1}}{\partial \hat{\tau}_{n+1}^\varepsilon}\right)^i$ and $\left(\frac{\partial \mathbf{R}_{n+1}}{\partial \mathbf{T}_{n+1}'}\right)^i$.

For the given $(\mathbf{R}_{n+1})^i$ we now solve the nonlinear system

$$[\mathcal{R}_{n+1}] = \left[\begin{array}{c} \mathbf{D}_{n+1} - \mathcal{P}(\mathbf{T}_{n+1}', \mathbf{R}_{n+1})\mathbf{T}_{n+1}' \\ \hat{\tau}_{n+1}^\varepsilon - \hat{\tau}_{\varepsilon s} - \left[(\hat{\tau}_{\varepsilon s} - \hat{\tau}_n^\varepsilon)^{1-\kappa} + \frac{(\kappa-1)\mu\theta_0\Delta t\sum_{s=1}^M|\dot{\gamma}_{n+1}^s|}{\mu_0\hat{\tau}_{\varepsilon s}^\kappa} \right]^{1/(1-\kappa)} \end{array} \right] = \left[\begin{array}{c} \mathbf{0} \\ 0 \end{array} \right] \quad (17)$$

iteratively for $\mathbf{x}_{n+1} = [\mathbf{T}_{n+1}'^T \quad \hat{\tau}_{n+1}^\varepsilon]^T$ from the Newton-Raphson method, i.e.

$$\mathbf{J}_{n+1}^i \Delta \mathbf{x}_{n+1}^i = -\mathcal{R}_{n+1}^i \quad (18)$$

where $\mathbf{J}_{n+1}^i = \left(\frac{\partial \mathcal{R}_{n+1}}{\partial \mathbf{x}_{n+1}}\right)^i$ is the Jacobian and $\Delta \mathbf{x}_{n+1}^i = \mathbf{x}_{n+1}^{i+1} - \mathbf{x}_{n+1}^i$ is the solution update at iteration i . Eq.(17.2) is obtained by integrating Eq.(11) analytically. Eqs.(16) and (18) are repeatedly solved until the solution converges.

Upon solving Eq.(17) for $\mathbf{T}_{n+1}'$ for each of the N crystals, the macroscopic Cauchy stress $\bar{\mathbf{T}}_{n+1}'$ is calculated from Eq.(15).

4 IDENTIFICATION STUDIES

Here we use an identification study to address a primary disadvantage of our model i.e. the large number of material parameters. We assemble the model parameters into the vector $\mathbf{d} = (\hat{\tau}_i/\mu_0, \tau_a, \theta_0, \kappa, g_{0i}, g_{0\varepsilon}, g_{0\varepsilon s}, \hat{\tau}_{\varepsilon s0}, \dot{\varepsilon}_{0i}, \dot{\varepsilon}_{0\varepsilon}, \dot{\varepsilon}_{0\varepsilon s}, p_i, q_i, p_\varepsilon, q_\varepsilon)$ and use reference values for these parameters to define the vector $\bar{\mathbf{d}}$. We then compute the material parameters $\mathbf{d}$ by solving the following problem:

$$\underset{\mathbf{d}}{\text{Minimize}} \quad \hat{e}(\mathbf{d}) = \sum_{p=1}^P \left(1 - \frac{\bar{T}_{ij}^p(\mathbf{d})}{\tilde{T}_{ij}^p}\right)^2 + \alpha \sum_{i=1}^N \left(1 - \frac{d_i}{\bar{d}_i}\right)^2. \quad (19)$$

The first summation in Eq.(19) is an error measure between the calculated stress response $\bar{\mathbf{T}}^p$ and the experimentally measured $\tilde{\mathbf{T}}^p$. Regularization is required to make the problem well-posed [15], so that small changes in experimental data lead to small changes in the parameter estimates. The second summation in Eq.(19), which penalizes deviation from reference values

*The superscript i denotes functions evaluated at $[\mathbf{R}_{n+1}^i, \mathbf{T}_{n+1}'^i, \hat{\tau}_{n+1}^\varepsilon]$, where i is the iteration number.

$\bar{\mathbf{d}}$, achieves this goal. The identification problem is solved with the BFGS method [16], a first-order optimization algorithm, which requires the derivative $\frac{\partial \hat{\epsilon}}{\partial \mathbf{d}}$. The efficient calculation of this gradient is discussed next.

4.1 DESIGN SENSITIVITY ANALYSIS

Gradients of the error function in Eq.(19) involve the derivative $\frac{D\bar{\mathbf{T}}'_{n+1}}{Dd_i}$. From Eq.(15) it follows that

$$\frac{D\bar{\mathbf{T}}'_{n+1}}{Dd_i} = \sum_{c=1}^N w^c \left(\frac{D\mathbf{T}'^c_{n+1}}{Dd_i} \right) / \sum_{c=1}^N w^c. \quad (20)$$

The direct method [17] is used to calculate $\frac{D\mathbf{T}'^c_{n+1}}{Dd_i}$ by differentiating Eq.(17) with respect to d_i ,

$$\frac{\partial \mathcal{R}_{n+1}}{\partial \mathbf{x}_{n+1}} \frac{D\mathbf{x}_{n+1}}{Dd_i} + \frac{\partial \mathcal{R}_{n+1}}{\partial \mathbf{x}_n} \frac{D\mathbf{x}_n}{Dd_i} + \frac{\partial \mathcal{R}_{n+1}}{\partial d_i} = 0 \quad (21)$$

and then solving $\frac{D\mathbf{x}_{n+1}}{Dd_i} = \left[\frac{D\mathbf{T}'^c_{n+1}}{Dd_i}^T \quad \frac{D\hat{\epsilon}_{n+1}}{Dd_i} \right]^T$ from

$$\mathbf{J}_{n+1} \frac{D\mathbf{x}_{n+1}}{Dd_i} = - \left(\frac{\partial \mathcal{R}_{n+1}}{\partial \mathbf{x}_n} \frac{D\mathbf{x}_n}{Dd_i} + \frac{\partial \mathcal{R}_{n+1}}{\partial d_i} \right). \quad (22)$$

Note that the evaluation of $\frac{D\mathbf{x}_{n+1}}{Dd_i}$ requires only a backsubstitution into the decomposed Jacobian from the analysis. Regarding the right hand side, we see that $\frac{D\mathbf{x}_n}{Dd_i}$ is available from the sensitivity analysis at the previous timestep. $\frac{\partial \mathcal{R}_{n+1}}{\partial \mathbf{x}_n}$ is evaluated analytically while $\frac{\partial \mathcal{R}_{n+1}}{\partial d_i}$ is calculated via the finite difference method, making this a so called semi-analytical sensitivity analysis [18].

4.2 EXAMPLE 1: HY100 STEEL

An identification study is performed using our polycrystal MTS model to determine the material parameters of HY100 steel. A total of 48 potentially active slip systems (12 $\{110\}/\langle 111 \rangle$, 12 $\{112\}/\langle 111 \rangle$ and 24 $\{123\}/\langle 111 \rangle$) are considered for this bcc metal. The experimental data comes from compression and torsion tests at various strain rates and temperatures, that are performed at the Los Alamos National Laboratory (LANL) and the Naval Surface Warfare Center, Indian Head Division.

The simulation, starting with a random initial texture, reaches an equivalent plastic strain of 20% after 10 equal time step increments. The initial material parameter estimates for the identification study, i.e. $\mathbf{d}$, are obtained from an available isotropic MTS model fit [19]. This particular realization used a tanh hardening law, which we replace with the power law of Eq.(11).

The identification problem was solved for penalty coefficients of $\alpha = 0$, $\alpha = 0.001$ and $\alpha = 0.01$. The material parameters were only allowed a $\pm 10\%$ variation. Exceptions to these side constraints were $\dot{\epsilon}_{oi}$ which was only allowed a 10% decrease (the initial estimate is on the upper bound), q_ϵ which was only allowed a 10% increase (the initial estimate is on the lower bound) and θ_o and κ which were both unrestricted (the initial estimates are not well known

due to the different hardening law form). The error function iteration history, including and excluding the regularization term, is plotted in Figures 2 and 3, respectively. Figure 4 illustrates the iteration history of θ_o and κ . The computed stress response using the identified parameters for $\alpha = 0.01$ is plotted in Figure 5, where it is compared to the experimental data.

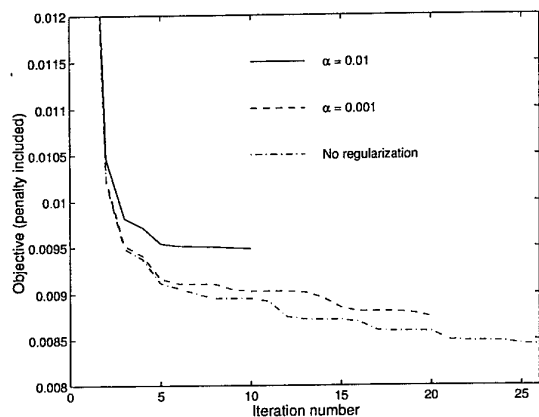


Figure 2: Iteration history of error function, including penalty

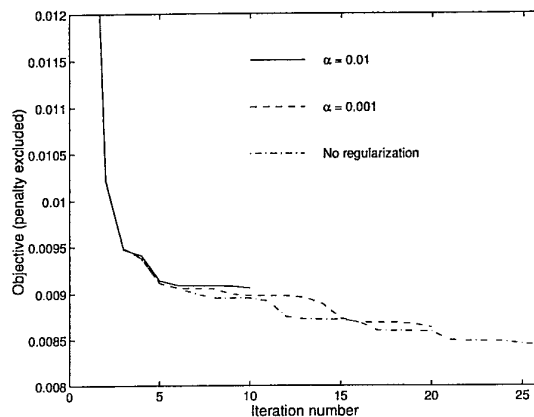


Figure 3: Iteration history of error function, excluding penalty

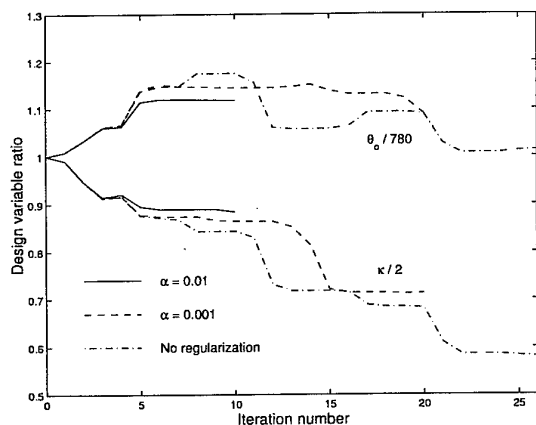


Figure 4: Iteration history of hardening law parameters

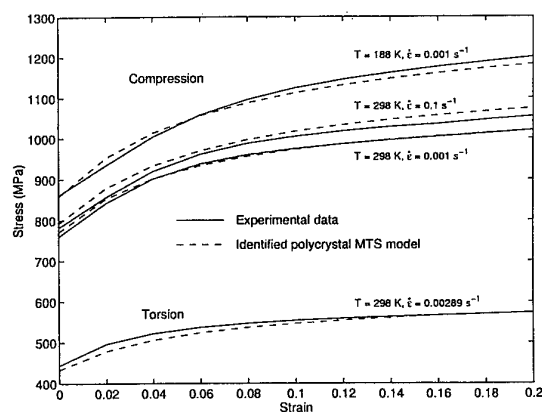


Figure 5: Identified MTS model stress response

These results clearly illustrate the benefit of regularization. The number of BFGS iterations required for convergence decreases from 26 to 10, as α increases from 0 to 0.01. Furthermore, the changes in the material parameter estimates from their initial values (notably θ_o and κ) are limited, with only a slight increase in the objective function.

4.3 EXAMPLE 2: TEXTURED TANTALUM PLATE

In this example, data [†] for through thickness compression and in-plane compression at 0° and 90° are supplied for a Tantalum plate with initial texture. The tests are performed at room-temperature and at quasi-static strain rates.

[†]The data is supplied by P.J. Maudlin of LANL Theoretical Division

The 12 $\{110\}/\langle 111 \rangle$ and 12 $\{112\}/\langle 111 \rangle$ slip systems are included for this bcc metal. The initial texture of the plate is described by 218 crystals with varying weights and orientations. Simulations are performed at room temperature ($T=298$ K) and a compression rate of 0.01 s^{-1} . Initial estimates for the material parameters are obtained from Chen and Gray [11].

Since all tests are performed at the same temperature and strain rate, only τ_a , θ_o , κ and $\hat{\tau}_{\text{eso}}$ are identified in this study. Our experience shows that these parameters are not sensitive to changes in the experimental data, hence no regularization was required for this problem ($\alpha=0$). Each test was simulated with 11 time steps which each generate a 1% strain increment. The initial data point of each test, i.e. the stress at 0% strain, was not included in the error function due to the complex initial yielding behaviour of tantalum.

Instead of the usual approach to use only x-ray diffraction data to obtain a discrete texture representation i.e. volume fractions, we use both x-ray diffraction and stress-strain data. The discrete set of crystal weights, that is initially obtained from x-ray diffraction data, is adjusted slightly to minimize the discrepancy between numerical and experimental stress-strain data. By obtaining the crystal weights in this manner we maximize the accuracy of the polycrystal model. Keeping the material parameters fixed for the Tantalum plate model, the 218 crystal weights (volume fractions) for the initial texture are tweaked, i.e. augmented, through a second identification study. The volume fraction for each crystal is calculated as $w^c = w_o^c d^c$, where w_o^c is the original volume fraction and d^c is the crystal weight scale factor to be determined in the identification study. The identification problem is now stated as

$$\text{Minimize } \hat{e}(\mathbf{d}) = \sum_{p=1}^P \left(1 - \frac{\bar{T}_{ij}^p(\mathbf{d})}{\tilde{T}_{ij}^p} \right)^2 + \alpha \left[\sum_{c=1}^N w_o^c d^c (d^c - 1)^2 \right] / \sum_{c=1}^N w_o^c d^c. \quad (23)$$

The penalty term ensures that the augmented texture will not deviate significantly from the original texture. The specific form of the penalty term penalizes large weights more than small weights, which should retain the peaks in the orientation distribution.

The texture identification problem is computationally must less costly than the material parameter identification, since the individual crystal equations are solved only once to obtain $\mathbf{T}'^c$, $c = 1, 2, \dots, N$. The appropriate weighting of these crystal stress solutions are performed afterwards to calculate the material response via Eq.(15).

The experimental data and computed reponse, both before and after TA, are plotted in Figure 6. The initial texture, augmented texture for $\alpha = 0.02$ and texture difference of the Tantalum plate are plotted in Figure 7, as $\{110\}$ pole figures. It is clear from Figure 7 that the penalty term is successful in not allowing severe departure from the original texture. Table 1 emphasizes this point, where initial volume fractions and augmented volume fractions for $\alpha = 0$ and $\alpha = 0.02$, for selected grains, are summarized. The penalty term greatly reduces volume fraction changes, with only a slight increase in the minimum obtainable error function.

5 CONCLUSIONS AND RECOMMENDATIONS

1. The use of a fully implicit algorithm, solved with the Newton-Raphson method, allows the efficient calculation of response sensitivities. These sensitivities are used to solve a

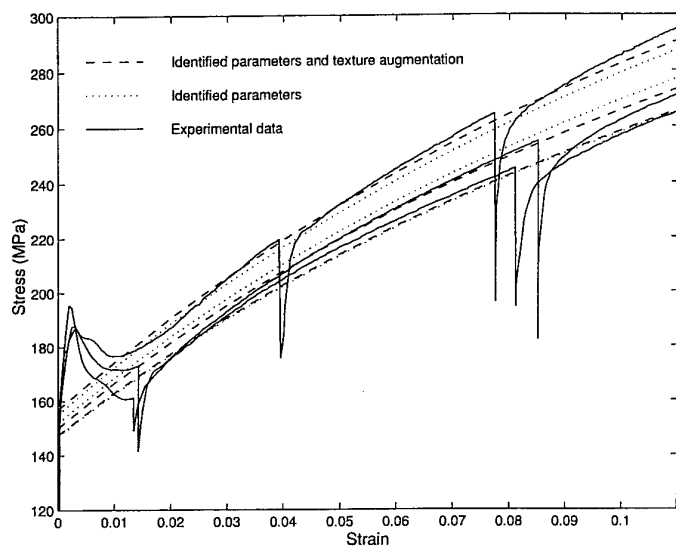


Figure 6: Stress response of identified polycrystal MTS model for Ta plate

Volume fractions		
Initial	$\alpha = 0$	$\alpha = 0.02$
2.816	4.585	3.118
2.103	1.574	1.921
2.090	2.471	2.293
1.969	1.216	1.787
1.780	2.444	1.806
1.505	0.719	1.366
1.282	0.606	0.920
Objective (error only) $\times 10^{-3}$		
9.046	4.112	4.380

Table 1: Volume fractions before and after texture augmentation

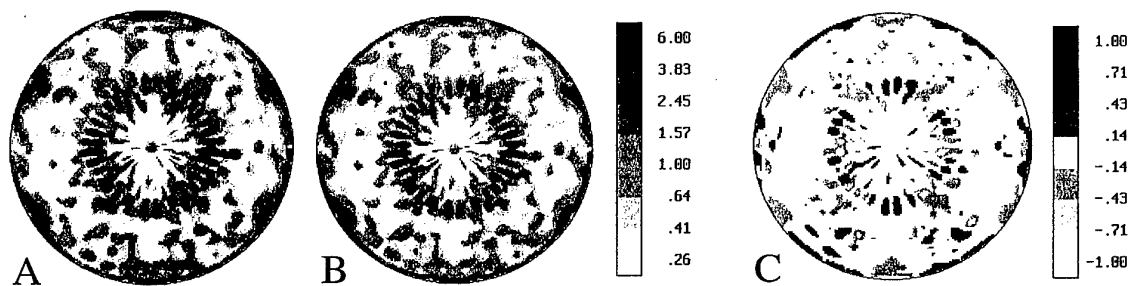


Figure 7: $\{110\}$ Pole figures of Ta plate, using 218 weighted crystals. A) Original texture B) Augmented texture C) Texture difference

material parameter identification problem, using gradient based optimization algorithms. Regularization is required to make the problem well-posed.

2. Optimization provides a powerful method of parameter estimation since numerous material parameters must be determined.
3. Analytical design sensitivities are necessary to provide a computationally efficient parameter identification algorithm.
4. Texture augmentation can be used to fine tune discrete texture representation for use in numerical polycrystal model simulations.

REFERENCES

- [1] G.I. Taylor, Plastic Strain in Metals, *J. Inst. Metals*, Vol. 62, 307-324, 1938.
- [2] J.F.W. Bishop, R. Hill, A Theoretical Derivation of the Plastic Properties of a Polycrystalline Face-Centered Metal, *Phil. Mag.*, Vol. 42, 1298-1307, 1951
- [3] J. Pan, J.R. Rice, Rate Sensitivity of Plastic Flow and Implications for Yield-surface Vertices, *Int. J. Solids Structures*, Vol. 19, No. 11, pp. 973-987, 1983.
- [4] R.J. Asaro, A. Needleman, Texture Development and Strain Hardening in Rate Dependent Polycrystals, *Acta Metall.*, Vol. 33, No. 6, pp. 923-953, 1985.
- [5] S. Kok, A.J. Beaudoin, D.A. Tortorelli, A Polycrystal Plasticity Model Based on the Mechanical Threshold, *Submitted*, 2000.
- [6] Y. Chastel, J. Signorelli, J.-L. Chenot, Modelling Metal Forming Processes Using Polycrystalline Plasticity: An Inverse Method for Parameter Identification, In *Int. Conf. on Thermomech. Processing of Steels & Other Materials*, Ed. T. Chandra, T. Sakai, The Minerals, Metals & Materials Society, pp. 2101-2107, 1997
- [7] J.W. Signorelli, R.E. Logé, Y.B. Chastel, R.A. Lebensohn, Parameter Identification Method for a Polycrystalline Viscoplastic Selfconsistent Model Based on Analytical Derivatives of the Direct Model Equations, *Modelling Simul. Mater. Sci. Eng.*, Vol. 8, pp. 1-17, 2000.
- [8] P.S. Follansbee and U.F. Kocks, A Constitutive Description of Copper Based on the Use of the Mechanical Threshold Stress as an Internal State Variable, *Acta Metall.*, Vol. 36, No. 1, pp. 81-93, 1988.
- [9] Y.P. Varshni, Temperature Dependence of the Elastic Constants, *Phys. Rev. B*, Vol. 2, pp. 3952-3958, 1970.
- [10] U.F. Kocks, A.S. Argon, M.F. Ashby, *Thermodynamics and Kinetics of Slip*, Progress in Materials Science, Vol. 19, Pergamon Press, New York, NY, 1975.
- [11] S.R. Chen, G.T. Gray, Constitutive Behaviour of Tantalum and Tantalum-Tungsten Alloys, *Metall. and Mat. Trans. A*, Vol. 27A, pp. 2994-3006, October 1996.
- [12] R. Hill, The Essential Structure of Constitutive Laws for Metal Composites and Polycrystals, *J. Mech. Phys. Solids*, Vol. 15, pp. 79-95, 1967.
- [13] C.-T. Chen, *Linear System Theory and Design*, Holt, Rinehart and Winston, New York, 1984.
- [14] J.C. Simo, T.J.R. Hughes, *Computational Inelasticity*, Springer-Verlag, New York, 1998.
- [15] A.N. Tikhonov, V.Y. Arsenin, *Solution of Ill-Posed Problems*, V.H. Winston and Sons, Washington D.C., 1977.
- [16] DOT Users Manual, Version 3.00, VMA Engineering, 1992.
- [17] P. Michaleris, D.A. Tortorelli, C.A. Vidal, Tangent Operators and Design Sensitivity Formulations for Transient Non-linear Coupled Problems with Applications to Elastoplasticity, *Int. J. Numer. Meth. Engng.*, Vol. 37, pp. 2471-2499, 1994.
- [18] N. Olhoff, J. Rasmussen, E. Lund, A Method of Exact Numerical Differentiation for Error Elimination in Finite Element Based Semi-Analytical Shape Sensitivity Analysis, *Mechanics of Structures and Machines*, Vol. 21(1), pp. 1-66, 1993.
- [19] S.R. Chen, Personal communication, March 1997.

VEHICLE SUSPENSION OPTIMISATION USING VEHSIM2D AND LFOPC

A.F. Naudé¹ and J.A. Snyman²

¹ AF Naude Engineering Technology cc, Pretoria, South Africa

² Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria, South Africa

ABSTRACT

In order to fulfil the mobility requirements for a specific vehicle, decisions must be made during the concept design phase regarding the characteristics of the suspension components. These include the characteristics of the springs, dampers, bump stops, etc. As an aid during the concept design phase a two dimensional vehicle simulation program Vehsim2d was developed. The Leap Frog Optimisation algorithm for Constrained problems (LFOPC) was included in Vehsim2d to enable optimisation of certain vehicle and suspension characteristics. This paper describes the simulation program and gives an example of the optimisation of the damper characteristics of a 22 ton three axle vehicle. By using this approach the optimised damper characteristics with respect to ride comfort for a vehicle were computed. With these damper characteristics a 28.5 % improvement in the ride comfort of the vehicle was experienced.

INTRODUCTION

Vehicle suspension characteristics have an important influence on the ride comfort and mobility of vehicles [1,2]. The suspension system of a vehicle consists mainly of the springs, dampers, bump stops and tyres. The characteristics of each of these components can be non-linear. During the vehicle concept design phase it is important to prescribe the required characteristics for these suspension components to enable the vehicle to attain the required ride comfort and mobility over specified terrain and speed.

An approach to determine the required suspension characteristics can be to simulate the vehicle motion for different suspension characteristics using a vehicle dynamic simulation program. From an analysis of the results for the different combinations, the configuration with the best characteristics can be selected. Such an approach was used by Lee [3]. Using the simulation program DADS [4] he simulated the ride dynamics of a medium truck for different spring stiffnesses of the front/rear suspension and different damping characteristics of the shock absorbers. The drawback of this approach is the large number of configurations that must be simulated and analysed.

Alternatively a more formal mathematical programming approach may be used to search through the design space to give the optimum configuration accurately. Etman [5], however, states that the coupling of a multibody code to a mathematical programming algorithm may be difficult to implement and can lead to high computational cost. Nevertheless this approach was adopted in the work presented in this paper. A two-dimensional vehicle computer simulation program Vehsim2d was developed and linked to the LFOPC optimisation algorithm [6] to optimise an objective function with respect to the chosen characteristics (design variables). The objective function may be the ride comfort of the vehicle and the design variables the suspension characteristics. The user can prescribe both the relevant objective function and the specific choice of design variables.

SIMULATION PROGRAM

As an aid in the concept design phase a two-dimensional (the influence of roll is neglected) vehicle dynamics simulation program Vehsim2d was developed. The program is able to simulate vehicles with up to four axles. Figure 1 shows the main input screen for the program. The main input consists of the geometrical dimensions of the vehicle, the mass and inertia characteristics, the suspension characteristics, the route profile for the simulation, general simulation data and names of files used for input and recording of results during the simulation.

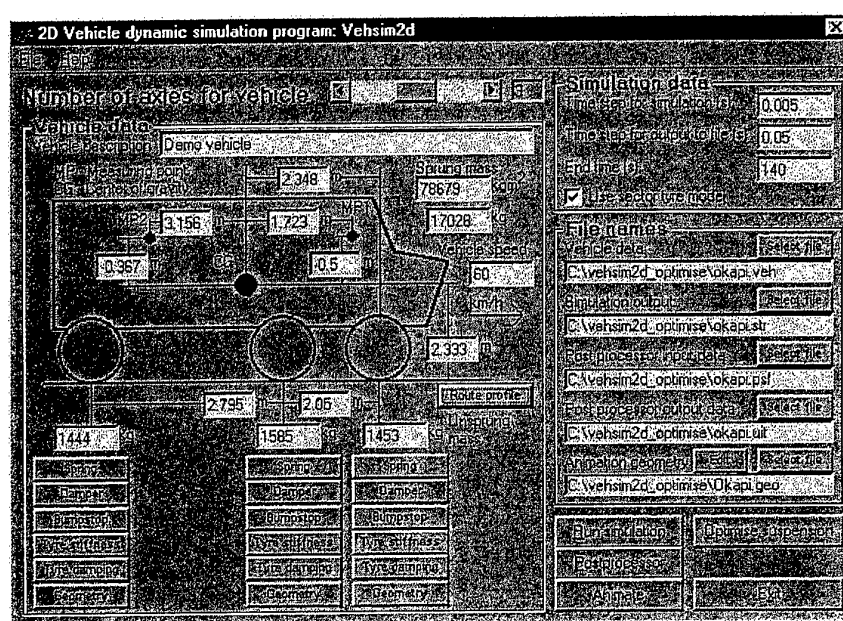


Figure 1: Vehsim2d – Main input screen

The characteristic data for the different suspension components are prescribed using six piece-wise continuous linear approximation. For example for a particular damper six damper stiffness values and six corresponding deflection speed values define the characteristic. Figure 2 shows a schematic for the damper characterisation. Figure 3 shows a typical screen input for damper characterisation.

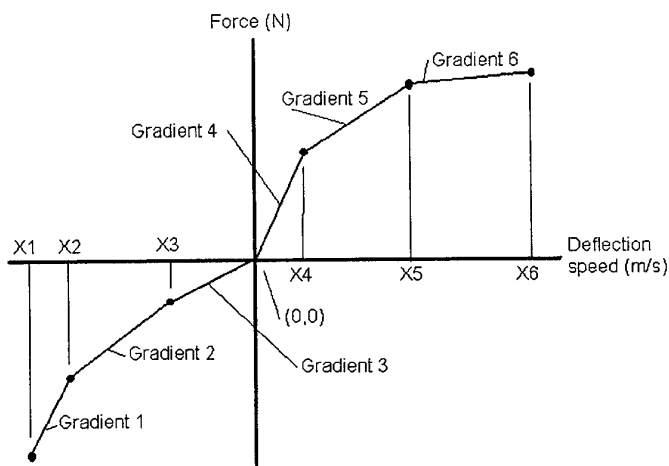


Figure 2: The six piece-wise continuous linear approximation for suspension force characterisation

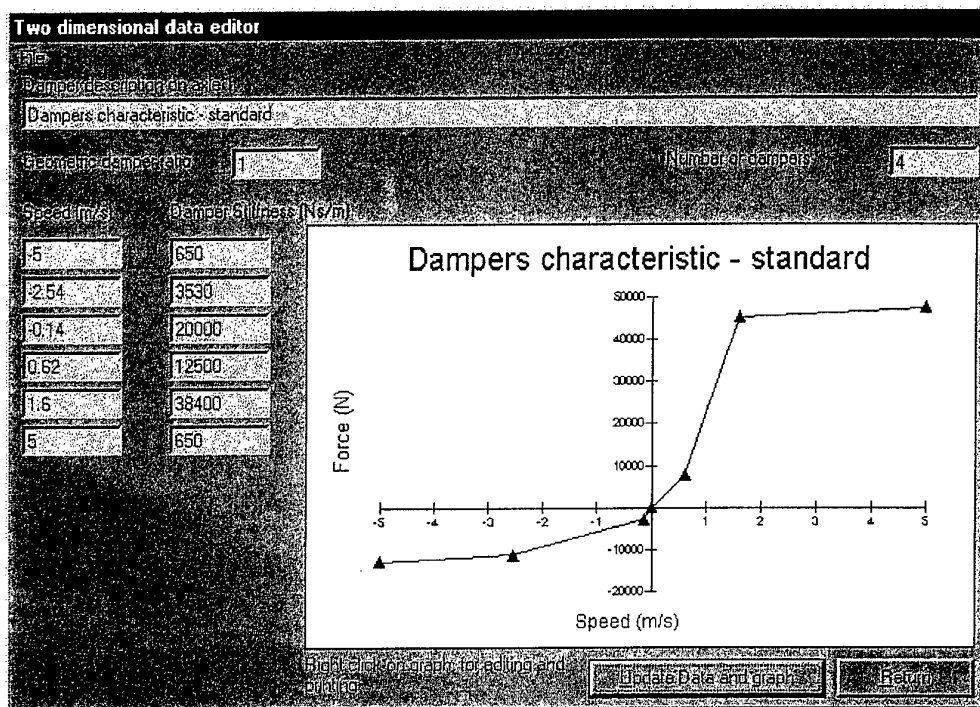


Figure 3: Input of suspension characterisation using a six piece-wise continuous linear approximation

Output of the simulation program consists of the accelerations, velocities and displacements at the centre of gravity (CG), two other specified measuring points (MP1 and MP2) and the wheels. In addition forces, deflections and deflection rates of the wheels and of the suspension components are also saved in the simulation result files. Figure 4 shows the output (postprocessor) options available.

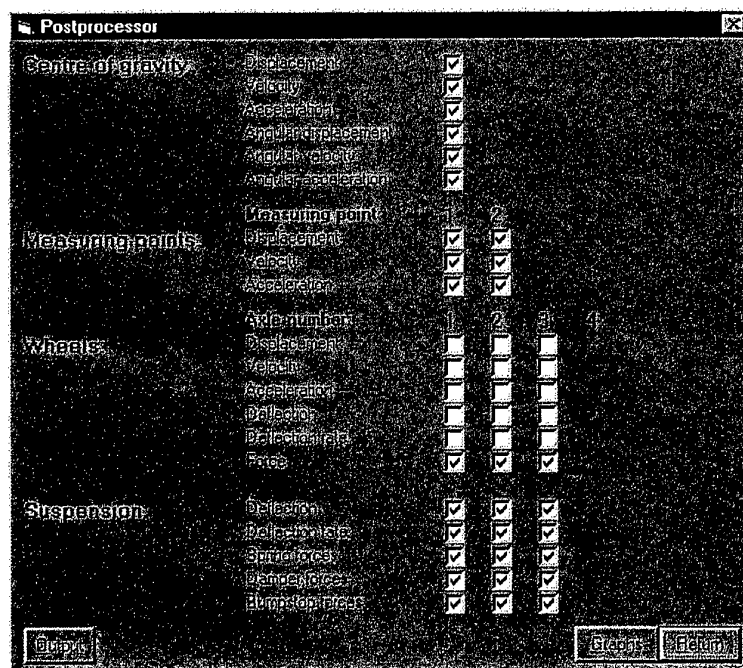


Figure 4: Vehsim2d output options

To objectively measure the ride comfort of the vehicle the simulation output of the program can be analysed by calculating the vibration dose value (VDV) according to BS6841 [7]. In a study performed by Els [8] it was shown that the VDV according to BS6841 gives good correlation with subjective ride comfort comments for the type of vehicles, terrain and speed considered in this study.

The simulation program also allows for a subjective evaluation of the resultant motion of the vehicle through an animation of the simulated behaviour. Figure 5 shows an example of the animation output at a particular instant in time.

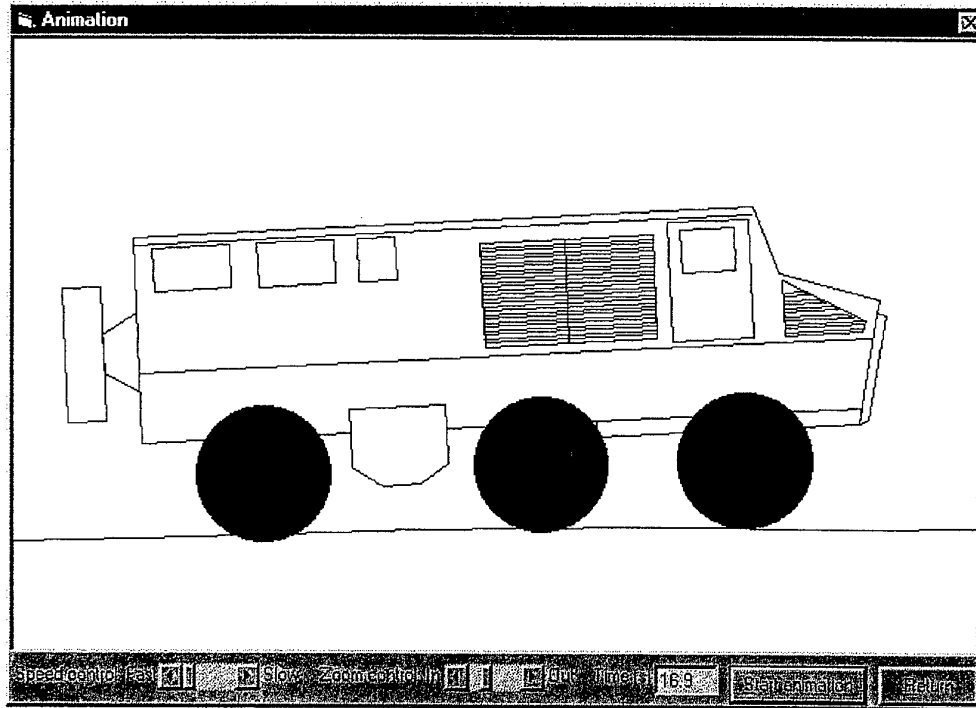


Figure 5: Vehsim2d – Animation output

OPTIMISATION METHODOLOGY

Engineering optimisation problems where simulation programs are used in computing objective functions, as is the case in this study, present unique challenges because of

- i. *the presence of noise* in the objective or constraint functions (due to numerical inaccuracies in the analyses or simulations); and
- ii. *the presence of discontinuities* in the objective/constraint functions arising from formulations of the optimisation problem in a convenient form for engineers (e.g. absolute value functions, and penalty function formulations of constrained problems).

These aspects, presently of great world wide interest to design engineers, have been addressed by the second author. Central and essential to tackling the above difficulties has been the development of novel optimisation algorithms suitable for engineering problems. This required both the

construction of new algorithms, and the testing of these methods on appropriate standard, and new, engineering design problems. A particular successful development has been the leapfrog trajectory methods [9,10].

The 'leap-frog' unconstrained optimisation algorithms were originally proposed in the early eighties [9,10]. These algorithms have the unique characteristics, for gradient based methods, that only the gradient of the objective function is used and that no explicit line searches are performed. These algorithms were later refined and extended to constrained problems. The methods were found to be extremely reliable and robust. In particular, the methods are relatively insensitive to problems where discontinuities and noise are present. The most current version of the code for constrained optimisation, embodying the leap-frog method, is called LFOPC. For a brief review of the most recent developments regarding the leap-frog method the reader is referred to the paper by Snyman [6].

The LFOPC code was linked to Vehsim2d. The user specifies design variables linked to the optimisation variables and the objective function that is also linked to a specified objective computable from the simulation output, e.g. the VDV for ride comfort [7]. Side constraints on the design variables may also be prescribed. Thus the simulation of the vehicle motion over the prescribed terrain and speed, for a given set of design variables, allows for the computing of the associated objective function and function gradients to be used by LFOPC.

DAMPER OPTIMISATION

Figure 6 shows a 22 ton three axle military vehicle that was developed locally.

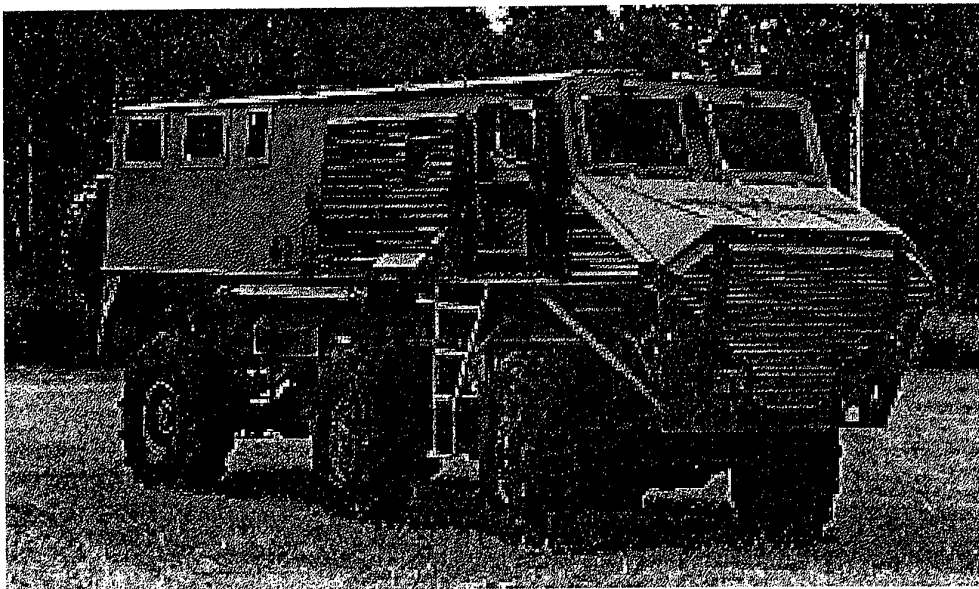


Figure 6: Vehicle used for suspension optimisation

The vehicle had gone through a series of design changes to improve mobility, fatigue life on the axles and other general features of the vehicle. In spite of these improvements certain suspension failures remained and it was decided to perform a suspension optimisation on the vehicle using the

program Vehsim2d and the LFOPC algorithm. The suspension optimisation was limited to the dampers of the vehicle. Similar dampers are used on all three axles. Four dampers are used on the front and rear axle and two dampers are used on the middle axle.

Since the vehicle was experiencing extreme damper forces during off-road movement, it was decided to simulate the vehicle under typical off-road conditions. A 2215 m dirt road track at the Gerotek vehicle testing facility was measured using the CSIR profilometer. Due to the fact that the program Vehsim2d is two-dimensional, the average profile of the left and right track was used as input for the route profile in the simulations. Figure 7 shows the resultant dirt road profile used.

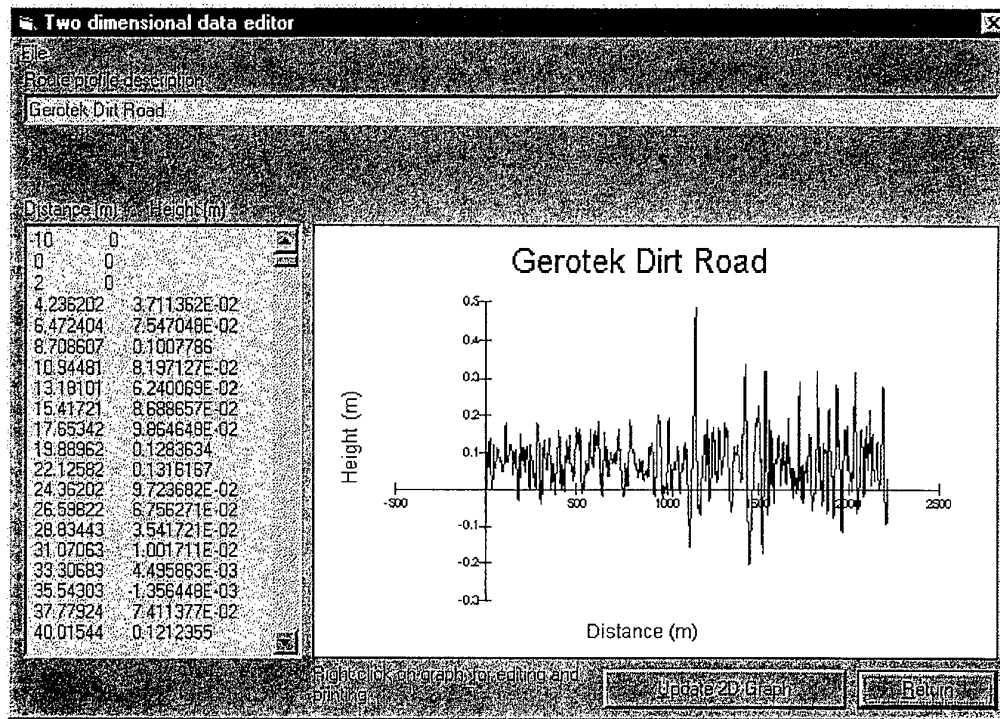


Figure 7: Route profile for the optimisation of the damper characteristics

During analyses of experimentally measured values of damper deflections over similar terrain it was found that the typical damper deflection rates experienced were ± 2 m/s. It was therefore decided to optimise the damper characteristics in the -2.5 to 2.5 m/s range, i.e. only the x-value 3 and 4 and gradient 2, 3, 4, 5 (see figure 2) were used as design variables to be optimised.

The objective function that was used in the study is an average of the four VDV (4-hour) values for the vertical acceleration at the centre of gravity, measuring point 1 (driver) and measuring point 2 (rear), as well as that for pitch acceleration.

The LFOPC optimisation results are shown in figure 8 to 11. Figure 8 indicates the reduction in the objective function value, as well as in the respective VDV (4-hour) values, as the optimisation progresses through the successive iterations. According to Els [8] a VDV (4-hour) value of $26 \text{ m/s}^{1.75}$ for vertical acceleration indicates an uncomfortable ride. The computed optimum damper characteristics decreased the VDV (4-hour) value of MP1 from $29.1 \text{ m/s}^{1.75}$ to $18.9 \text{ m/s}^{1.75}$ and for

MP2 from $28.9 \text{ m/s}^{1.75}$ to $21.8 \text{ m/s}^{1.75}$, indicating a more comfortable ride for both the driver and rear passengers.

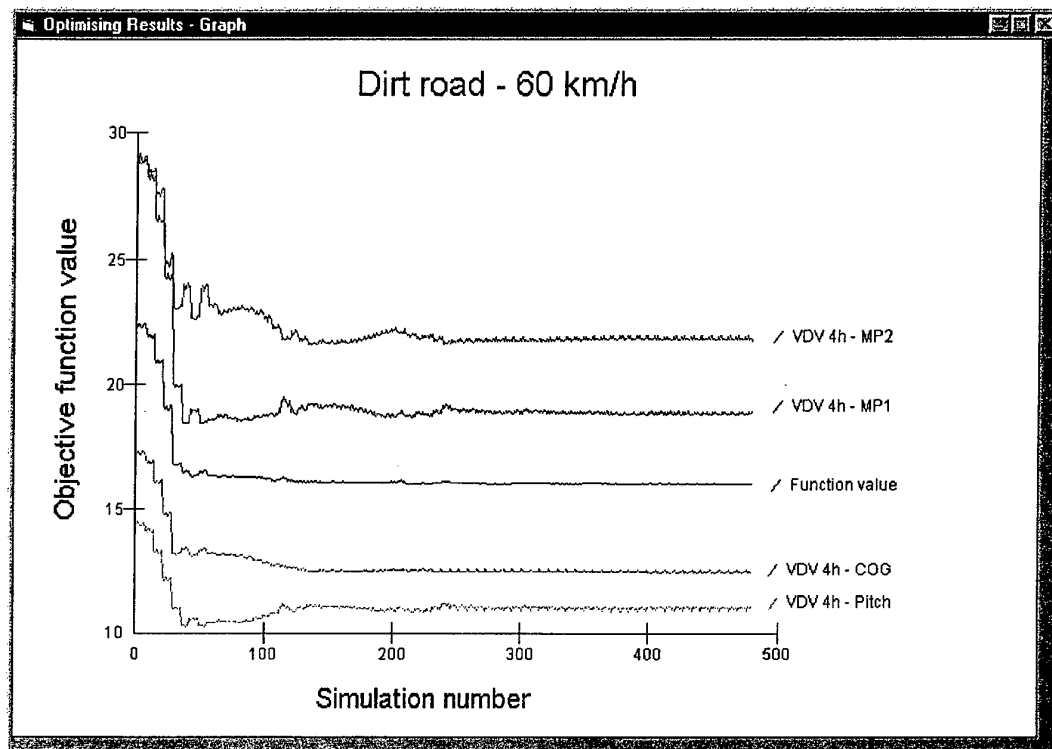


Figure 8: Objective function reduction during the optimisation of the damper characteristics

Figure 9 shows the change in the design variables during the optimisation. A value of 1 indicates the baseline value and the new value is obtained by the product of the design variable value and the baseline value. The schematic in figure 2 shows the interpretation of the legends as used in figure 9. The apparent fluctuations shown in figures 8 and 9 are due to the inclusion of perturbed values used in computing the gradients by finite difference.

Figure 10 compares the final optimum damper characteristic with that of the baseline damper characteristic. With this optimum damper characteristic an improvement of 28.5% in the objective function was obtained. Interesting is the increase in the bump stiffness (negative speed) in comparison with the baseline and the similarity in the bump and rebound damping force. This is contrary to the typical ratio of three to one between rebound and bump damping usually prescribed [2]. The increase in the bump damping can possibly be explained by the relative small suspension travel available before contact is made with the bump stops.

Further analyses of simulations for various configurations relative to that of the optimum configuration were done. All the damper configurations that give an objective function value within 5% of that for the optimum damper are shown in figure 11 relative to that of the baseline damper configuration. From figure 11 it is clear that the rebound damping characteristic is not as critical as the bump damping which forms a much narrower band than the rebound damping characteristics.

Table 1 compares the simulation results for the respective baseline and optimum damper characteristics as shown in figure 10. Notice the large reductions in the maximum acceleration at MP1 and MP2, the reduction in the minimum bumpstop force and the lower damper deflection rates on axle 1, 2 and 3 given by the optimum configuration.

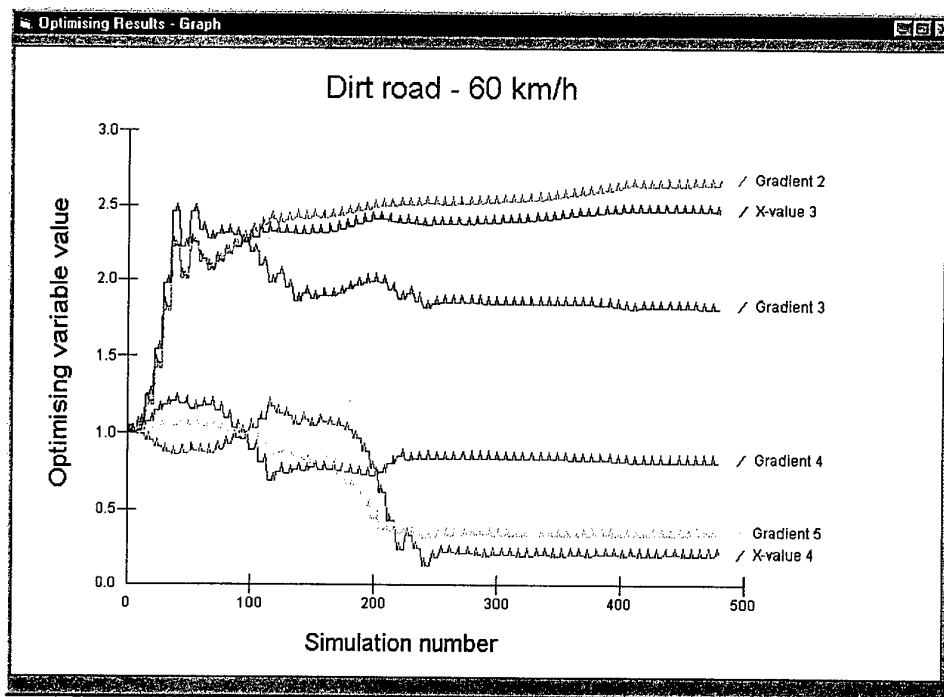


Figure 9: Change in design variables during the damper optimisation

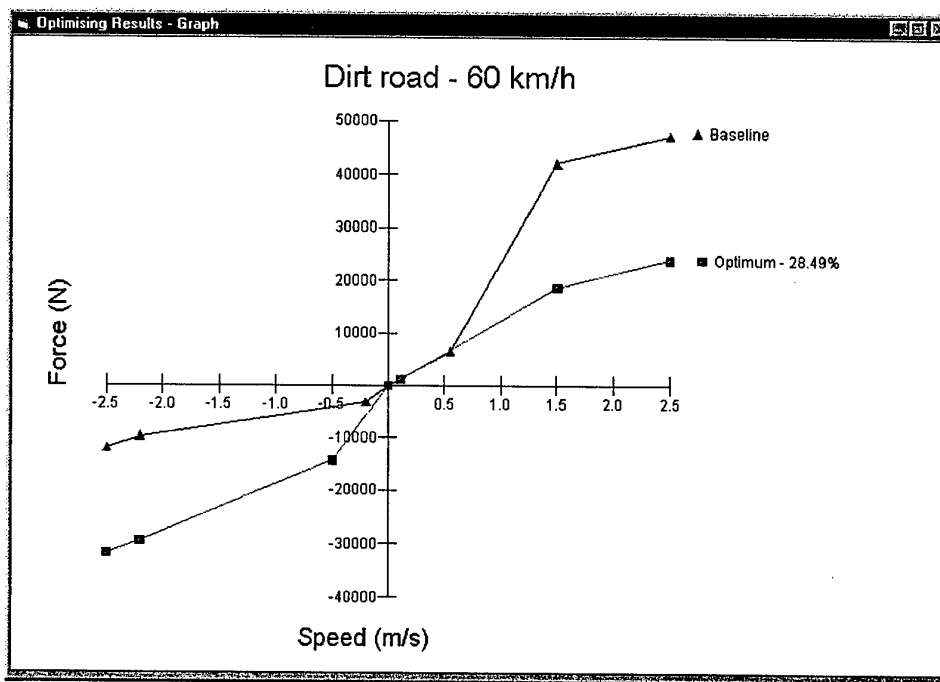


Figure 10: Comparison between the baseline and the optimum damper characteristics

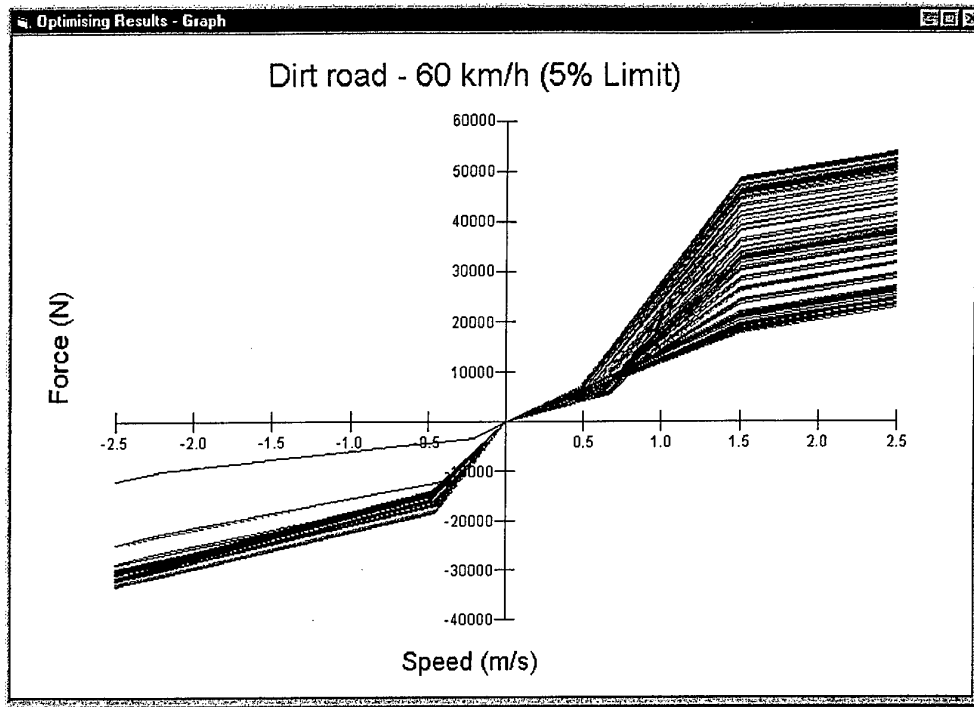


Figure 11: Damper characteristics giving a 5% margin on the objective function.
(Top line is that for the baseline damper)

Table 1: Comparison of simulation results between the baseline and optimum damper configurations (RMS = Root Mean Square value)

Characteristic	Baseline			Optimum		
	RMS	Min	Max	RMS	Min	Max
Displacement - COG (m)	0.107	-0.193	0.476	0.111	-0.188	0.482
Displacement - MP1 (m)	0.118	-0.287	0.525	0.119	-0.236	0.524
Displacement - MP2 (m)	0.121	-0.318	0.541	0.120	-0.238	0.549
Angular displacement - COG (rad)	0.023	-0.087	0.108	0.019	-0.069	0.096
Displacement - wheel 1 (m)	0.109	-0.222	0.496	0.108	-0.209	0.496
Displacement - wheel 2 (m)	0.105	-0.201	0.484	0.105	-0.200	0.485
Displacement - wheel 3 (m)	0.106	-0.208	0.487	0.106	-0.206	0.491
Acceleration - COG (m/s ²)	1.782	-9.649	14.659	1.436	-8.156	8.424
Acceleration - MP1 (m/s ²)	3.311	-10.949	26.042	2.507	-10.868	16.282
Acceleration - MP2 (m/s ²)	2.906	-9.295	28.749	2.600	-8.255	18.749
Angular acceleration - COG (rad/s ²)	1.126	-6.607	6.547	0.898	-4.560	4.776
Acceleration - wheel 1 (m/s ²)	4.232	-70.110	46.635	3.311	-23.071	29.913
Acceleration - wheel 2 (m/s ²)	3.702	-31.299	29.568	3.309	-24.876	27.109
Acceleration - wheel 3 (m/s ²)	3.762	-82.128	29.218	3.144	-24.572	28.044
Spring force - 1 (N)	55858.9	-116925.5	7146.1	50243.4	-108538.9	7399.3
Spring force - 2 (N)	54365.3	-87516.1	-877.5	51381.0	-74561.0	-11095.1
Spring force - 3 (N)	59642.7	-119187.8	-3767.4	55705.5	-107359.0	-7126.9
Damper force - 1 (N)	15353.4	-46924.6	88921.6	21426.9	-98812.0	62768.2
Damper force - 2 (N)	4061.7	-13208.5	24588.4	5303.8	-34789.9	17830.4
Damper force - 3 (N)	11030.2	-40053.0	53449.3	16427.8	-98701.4	42496.2

Table 1: Continue	Baseline			Optimum		
Characteristic	RMS	Min	Max	RMS	Min	Max
Bumpstop force - 1 (N)	13385.7	-231519.7	40878.6	4633.2	-89742.0	46152.2
Bumpstop force - 2 (N)	875.9	-21291.8	0.0	129.2	-6297.5	0.0
Bumpstop force - 3 (N)	7323.9	-164943.0	0.0	2078.3	-41548.5	0.0
Suspension deflection - 1 (m)	0.038	-0.111	0.104	0.031	-0.096	0.105
Suspension deflection - 2 (m)	0.014	-0.056	0.094	0.013	-0.033	0.077
Suspension deflection - 3 (m)	0.028	-0.105	0.095	0.022	-0.084	0.090
Suspension deflection rate - 1 (m/s)	0.342	-2.461	0.971	0.260	-1.625	1.266
Suspension deflection rate - 2 (m/s)	0.171	-1.218	0.705	0.128	-0.788	0.730
Suspension deflection rate - 3 (m/s)	0.248	-2.222	0.734	0.196	-1.622	0.865

CONCLUSION

Using the mathematical programming approach by linking the LFOPC optimisation algorithm to the two-dimensional simulation program Vehsim2d a vehicle suspension optimisation methodology was successfully developed. Coupling this method to the six piece-wise continuous linear approximation of the suspension components, it was shown that the algorithm can be used to optimise the non-linear damper characteristics for a specific vehicle over a typical off-road terrain. Using this approach an improvement of 28.5 % in the ride comfort for the vehicle over the specified terrain was obtained by optimising the damper characteristics. By using this method during the concept design phase, the required suspension characteristics can be optimised to fulfil the mobility requirements of the vehicle.

REFERENCES

1. Sternberg, E.R., Heavy-duty truck suspensions, SAE760369, SAE, 1976
2. Gillespie, T.D., Fundamentals of vehicle dynamics, Society of Automotive Engineers, 1992
3. Lee, H., Lee, G., Kim, T., A study of ride analysis of medium trucks with varying the characteristics of suspension design parameters, SAE973230, SAE, 1997
4. Haug, E.J., Computer-aided kinematics and dynamics of mechanical systems, Allyn and Bacon, 1989
5. Etman, L.F.P., Van Campen, D.H., Schoofs, A.J.G., Optimization of multibody systems using approximation concepts, IUTAM Symposium on optimization of mechanical systems, Kluwer Academic Publishers, 1996
6. Snyman, J.A., The LFOPC leap-frog method for constrained optimisation, *Computers & Mathematics with Applications*, in press, 2000
7. British Standard guide to measurement and evaluation of human exposure to whole body mechanical vibration and repeated shock, BS6841, British Standard Institution, 1987
8. Els, P.S., The application of ride comfort standards to off-road vehicles, Presented at: Human response to vibration 34 th meeting of the UK group, Dunton, Essex, 1999
9. Snyman, J.A., A new and dynamic method for unconstrained minimisation, *Appl Math Modelling*, Vol 6, pp. 449-462, 1982
10. Snyman, J.A., An improved version of the original leap-frog dynamic method for unconstrained minimisation LFOP1(b), *Appl Math Modelling*, Vol 7, pp. 216-218, 1983

SOUND AND VIBRATION OPTIMIZATION OF CARILLON BELLS AND MRI SCANNERS

A.J.G. Schoofs, P.H.L. Kessels, A.H.W.M. Kuijpers, M.H. van Houten
Eindhoven University of Technology, The Netherlands
Faculty of Mechanical Engineering
P.O. BOX 513, 5600 MB Eindhoven, The Netherlands
E-mail: A.J.G.Schoofs@tue.nl

ABSTRACT

Although bells and MRI scanners are very different products, design optimization of such systems show interesting similar aspects, enabling a fruitful crossover between involved research projects. The aim of bell design is to achieve eigenfrequencies that constitute a musically nice sound spectrum, while modal acoustic damping parameters must lie within acceptable bounds. The mechanism of image generation used in MRI scanners may lead to unacceptable noise levels. The dilemma is that if images are to be improved noise levels will increase too. Therefore, the optimization goal for MRI scanners is to minimize the sound radiation at a given image quality.

For the mechanical and acoustical behavior of bells and MRI scanners, accurate and highly efficient analysis and optimization tools have been developed such as Fourier-FEM and Fourier-BEM codes, and appropriate optimization tools. In the paper the main topics of these tools will be presented together with design optimization goals and results from both application fields.

INTRODUCTION

Bell design

During the summer of 1982 the research staff of the Division of Mechanical Engineering Fundamentals of the Eindhoven University of Technology visited the Royal Eijsbouts Bell Foundry at Asten, near Eindhoven. André Lehr invited them, at the time campanologist and general manager of the bell foundry. This visit was the beginning of a fruitful cooperation between foundry and university. In this paper some obtained results will be presented on the basis of the bell profiles A through D, shown in Figure 1.

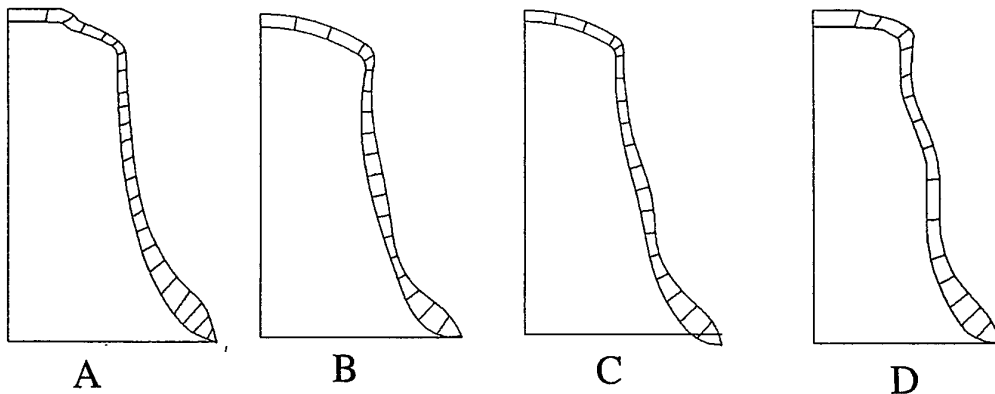


Figure 1: (A): Minor third bell, and three major third bells: (B): Second-generation bell, (C): Damping optimized bell, (D): First-generation bell.

Figure 1A displays the typical profile of a traditional minor third bell, developed in the 17th century by the Dutch bell founders François and Pieter. As the name indicates, this bell is characterized by a strong minor chord in its overtone structure. Such bells are very well suited to play music in minor. However, if the carillonneur likes to play in major he is in trouble, because of the conflicting minor chords in individual bells. Therefore, aforementioned André Lehr tried to find a bell in which the third overtone was raised halve a tone, in order to achieve a major chord within the bell's sound spectrum. This must be achieved remaining other important overtones unchanged. But, despite his great experience in bell design, his trial and error approach (for more than 30 years!) was not successful. Indeed, it proved to be a difficult design problem. The reason is that the vibration mode of the third overtone resembles the vibration mode of three other important overtones. Therefore, shape modifications of the bell affect these overtones in almost the same way as is the case for the third overtone, and as a consequence it is difficult to change the third overtone without affecting the other ones. Then in 1982, knowing that by that time modern FEM analysis methods were available, André Lehr requested us for help at the bell design problem.

MRI scanners

A Magnetic Resonance Imaging (MRI) scanner (see Figure 2) is a diagnostic device for medical imaging of soft tissues, flow and other physiological phenomena in the human body. The imaging process is based on nuclear magnetic resonance: when a static magnetic field is applied, the spins of the nuclei of particular atoms in the human body are aligned and start to precess, i.e., spin around the magnetic field lines. The precession frequency is linearly dependent on the magnetic field strength. The alignment causes a net longitudinal magnetization. For the imaging process, this alignment is disturbed by sending a radio frequent (RF) pulse, which excites the nuclei and brings the spins in phase. The frequency of the pulse must be the same as the precession frequency. When the RF signal is switched off, the spin axes of the nuclei will realign and dephase. This results in a transverse magnetization, which gives a measurement signal, which depends on the tissue properties. The measured signals are processed and form an image. Spatial encoding of the imaging information is achieved by superimposing a gradient magnetic field on top of the static field, which directly influences the precession frequency.

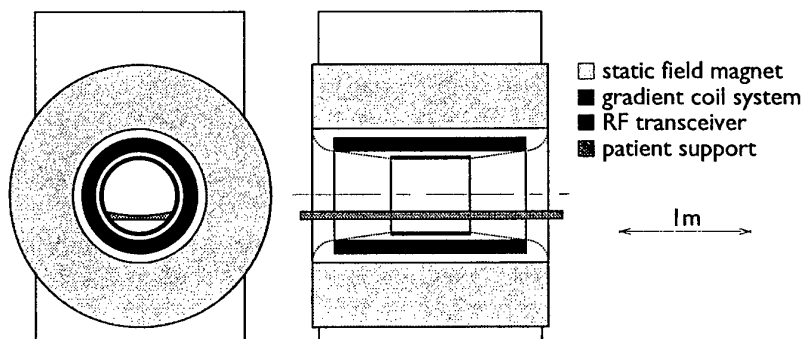


Figure 2: Schematic frontal and cross-sectional view of an MRI scanner.

A well-known problem of MRI scanners is their large noise production during the imaging process. Both patient and operator are exposed to high noise levels (up to 90 – 120 dB(A)), mainly caused by the vibration of the so-called gradient coil system. To understand this noise production mechanism it is important to look at an MRI scanner's construction. A strong magnetic field (0.1 – 1.5 T) is

produced by the (superconducting) static field magnet. On this field, a gradient magnetic field is dynamically superimposed which is created by the gradient coil magnets. The gradient coil consists of a composite tube in which copper conductors are embedded. The coils are driven by a sequence of pulse-like currents of about 300 A in magnitude and with a frequency in the range of 100 – 2000 Hz. Because of this rapidly switching current in the strong static magnetic field, dynamic Lorentz forces are generated in the gradient coils, which results in vibration of the coil tube. This vibration causes direct sound radiation at the surface of the gradient coil system. Indirectly, acoustical energy is transported through the coil tube to the remainder of the scanner and magnetically to the superconducting magnet housing.

ANALYSIS METHODS

Fourier-FEM method

For structural analysis bells are almost perfect axisymmetric structures. They are excited non-axisymmetrically by the stroke of the clapper. The gradient coil of MRI scanners shows cyclic deviations from perfect structural axisymmetry. Same as the bell, it is also excited non-axisymmetrically, by Lorentz forces. Despite mentioned deviations, Kessels (2000a) has shown that for preliminary engineering design purposes, the gradient coil can be structurally modeled axisymmetrically. Of course, such structures can be discretized with general 3D finite element like 8-node bricks. The displacement field in each brick is described in three independent directions, leading to three degrees of freedom per node. Alternatively, for axisymmetric structures, the circumferential variations of the displacement amplitudes can be expanded into a Fourier series. The displacements are split into a part $\bar{u}^a$ and a part $\bar{u}^b$ that contain sine and cosine terms. A truncated Fourier series approximates the displacement field, in combination with a common finite element discretization for the axial and radial coordinate directions:

$$\bar{u}(r, \theta, z) \approx \bar{\mathbf{e}}^T \sum_{m=0}^{n_f} \mathbf{N}(r, z) \begin{bmatrix} u_{m_r}^a \cos(m\theta) + u_{m_r}^b \sin(m\theta) \\ u_{m_\theta}^a \sin(m\theta) + u_{m_\theta}^b \cos(m\theta) \\ u_{m_z}^a \cos(m\theta) + u_{m_z}^b \sin(m\theta) \end{bmatrix} \quad (1)$$

where u_m^a and u_m^b contain Fourier coefficients of the nodal displacement amplitudes. The parameter m indicates the Fourier mode. Using Fourier expansions in combination with FEM, a ring can then be discretized using a single element. Applying such ring elements, the number of nodes needed for the finite element discretization is reduced considerably. The number of degrees of freedom per node, however, has increased from three to $6n_f$. Fortunately, using the Fourier expansion in the discretization of the differential equations of motion leads to a set of $n_f + 1$ smaller uncoupled systems:

$$(-\omega^2 \mathbf{M} + \mathbf{K}_m) \mathbf{u}_m = \mathbf{f}_m \quad m = 0, \dots, n_f \quad (2)$$

Due to the visco-elastic material properties of the gradient coil the stiffness matrix becomes frequency dependent: $\mathbf{K}_m = \mathbf{K}_m(\omega)$. The package SATURN (Kessels et al., 1998) can cope with such material behavior in a Fourier-FEM formulation. For the analysis and optimization of bells the Fourier-FEM package DYNOPT (Van Asperen, 1984) has been developed. Modal analysis of a bell is carried using a zero load vector: $\mathbf{f}_m = 0$. Furthermore, in DYNOPT transient bell responses on a clapper stroke can be analyzed (Roozen-Kroon, 1992).

Methods for acoustic analysis

The sound radiation of a vibrating 3D body can conveniently be analyzed using the boundary element method (BEM). In this method only the surface of the vibrating body needs to be discretized. Roozen-Kroon (1992) used the 3D BEM package SYSNOISE (1996), and 3D surface meshes to compute the sound radiation of bells. However, such analyses are very time consuming, since the discretization leads to a very large system of equations, involving a full matrix. Kuijpers developed a very efficient tool, called bArd (1998) to analyze sound radiation of axisymmetric structures, and based on Fourier-BEM.

The differential equations that describe the relations between velocities and acoustic pressure in an acoustic fluid comprise the Helmholtz equation, the momentum equation and an interface condition between the vibrating body and the fluid. Discretization of these equations using Fourier-BEM leads to a set of linear equations in matrix form (Kuijpers, 1999):

$$\mathbf{A}_m(\omega)\mathbf{p}_m = \mathbf{B}_m(\omega)\mathbf{v}_m \quad (3)$$

where the matrices $\mathbf{A}_m$ and $\mathbf{B}_m$ have to be assembled for each Fourier harmonic and for each frequency for which an acoustic analysis has to be performed. The column matrix $\mathbf{v}_m$ contains the Fourier coefficients of the surface normal velocity amplitudes for the Fourier harmonic m . These velocity amplitudes are determined from surface normal displacement amplitudes at the radiating surface, as computed in the structural analysis. Solving this system would lead to the Fourier coefficients of the nodal sound pressure amplitudes $\mathbf{p}_m$ at the surface. From these surface pressures, the pressure in any point of the acoustic domain can be determined. Alternatively, the radiated sound power can be obtained (Kuijpers, 1999):

$$\bar{P}(\omega) = \sum_{m=1}^{n_f} \bar{P}_m(\omega), \quad \bar{P}_m(\omega) = \frac{1}{2} \mathbf{v}_m^H \mathbf{C}_m \mathbf{v}_m \quad (4)$$

where an overline denotes temporal averaging, and $\mathbf{C}_m$ is the power coupling matrix:

$$\mathbf{C}_m(\omega) = \text{Re}((\mathbf{A}_m^{-1}(\omega)\mathbf{B}_m(\omega))^T \mathbf{N}) \quad (5)$$

in which the matrix $\mathbf{N}$ is defined as:

$$\mathbf{N} = \int_{\Gamma_a} \boldsymbol{\varphi} \boldsymbol{\varphi}^T d\Gamma_a \quad (6)$$

where $\boldsymbol{\varphi}$ is a column matrix containing the basis functions that are used to discretize the pressure and the surface normal velocity. Γ_a is the common boundary of the vibrating body and the acoustic fluid. The radiated sound power can be determined for any velocity distribution, but that would require the storage of the power coupling matrices $\mathbf{C}_m$ for each Fourier harmonic m and for each frequency of interest. A considerable reduction in memory requirement can be achieved by applying a very powerful reduction method, namely the so-called radiation modes formulation (Kuijpers, 1999).

Radiation modes form an orthogonal modal basis that decomposes the velocity distribution into independently radiating components with respect to the total acoustic power. This implies that the

radiated sound power can be determined almost instantly once the radiation modes have been computed. Since the radiation modes depend only on the acoustic geometry and on the frequency, they can be reused as long as the acoustic geometry remains unchanged. As a consequence for the MRI design, evaluating the effects that modifications to structural design have on the radiated power requires only a single full numerical acoustic analysis. Similarly, in bell design, different clapper configurations can be analyzed using only one full acoustic analysis as long as the shape of the bell remains unchanged.

There are cases where only part of the outer geometry is radiating. Also, an analyst may be interested in the contribution of a certain part of the geometry to the total sound radiation. Kuijpers (1999) showed that computational efforts in determining the radiation modes can then be reduced. Instead of extracting the modes from the full matrix system, a smaller system can be derived to which a modal analysis method can be applied. This way, an additional CPU time reduction is achieved for the analysis in the MRI project, as the larger part of the casing of the MRI scanner will be assumed rigid.

3. OPTIMIZATION OF THE SOUND SPECTRUM OF CARILLON BELLS

We formulated André Lehr's bell design problem as a shape optimization problem. Radii of the midplane and the wall thicknesses in a number of discrete points of the bell profile were applied as design variables. To solve the problem we developed the Fourier-FEM analysis and optimization program DYNOPT (Van Asperen, 1984). For optimization, Sequential Linear Programming (SLP) including move limits was applied. Furthermore, in this project we extensively used response surfaces based on numerical analyses to investigate the design space (Schoofs, 1987). The result is depicted in Figure 1D, showing a bell with a typical bulge. It is regarded as the first-generation major third bell. Several carillons were built applying the new bell shapes, and it was a great success (Schoofs et al., 1987a). However, there was also a fierce discussion within the carillonneurs' community. Compared with the old bells, the damping of the major third bells seemed to have increased. The carillonneurs were right indeed: due to the new bell shape especially the acoustic damping of the lowest vibration mode proved to be increased considerably. In the bell's sound spectrum this was experienced as a flaw.

A new, multidisciplinary, research project was defined to incorporate damping parameters in the bell design problem. Roozen-Kroon (1992) combined response surfaces for frequencies (derived using DYNOPT) with response surfaces for modal acoustic damping parameters based on 3D BEM analyses, carried out with the SYSNOISE (1996) package. Acoustic damping of a bell is defined as:

$$\eta_a = \frac{\bar{P}_m(\omega)}{\omega E_k} \quad (7)$$

where ω is the frequency (in radians), and $\bar{P}_m(\omega)$ is the radiated sound power of a vibration mode, given by Eq.(4). E_k is the kinetic energy of the same vibration mode given by:

$$E_k = \frac{1}{2} \rho \pi \int_A u^2 r dr dz \quad (8)$$

where u is the normalized velocity of the mode shape, ρ is the density of the bell material, and A is the area of the bell profile in the rz -plane. The kinetic energy is related to the maximum amplitude of the vibration mode and hence is not a time-average value.

Optimization was performed using an SQP algorithm from the NAG numerical library. The aim was to design a major bell with the same acoustic damping parameters as the minor bell. The resulting bell is shown in Figure 1C. Computed and measured damping parameters of the bell are displayed in Figure 3C. Although, compared with the first-generation bell (see Figure 1D), damping was improved considerably, the design goal was not achieved entirely. One reason was that the acoustic response surfaces could not be derived accurately enough due to lack of sufficient data points, caused by the computationally expensive acoustic analyses. Nevertheless, the project rendered valuable new insights in damping of bells.

Bell B (see Figure 1B) was designed (Schoofs, 1994) in an attempt to avoid the bulged shape of the first-generation major bells (see Figure 1D). The bulge was another, esthetical, complaint on the first major bells. Although acoustic damping was not an explicit design goal, bell B proved to show pretty good acoustic damping parameters, comparable with those of damping optimized bell C, see Figure 3B and Figure 3C respectively. Because of its improved shape and acoustic damping, bell B is regarded as the second-generation major third bell.

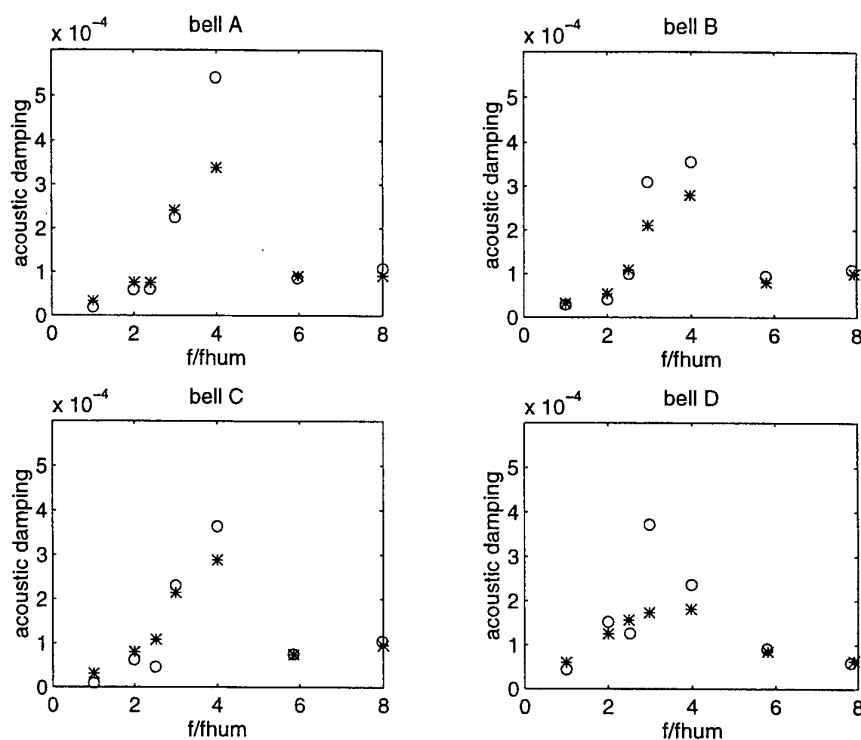


Figure 3: Measured (o) and calculated (*) acoustic damping values of different bells.

Van Houten (1998) wrote an interface between the structural Fourier-FEM program DYNOPT and the program bArd (1998). First, the bell is analyzed with DYNOPT giving eigenfrequencies, vibration modes and kinetic energy of the bell. Then, bArd is used to compute the radiated sound power of the modes, and finally the modal acoustic damping is computed using Eq. (7). This way, modal acoustic damping values were computed for the four bells shown in Figure 3. The results were compared with measured values. Computed and measured values agree rather well, except for the

fourth overtone (bells B and D), and for the fifth overtone (all bells). Where these deviations originate from is not yet clear, and is subject of further research. Anyway, the program bArd is regarded as a very valuable analysis tool in bell design.

Van Houten (1998) adopted the bell design problem as a test example in his research on approximation concepts for multidisciplinary optimization. The design goal was similar as the one in the work of Roozen-Kroon(1992): find a major third bell with damping parameters of a minor bell. He applied a mid-range multi-point approximation method in the optimization, both for frequencies and for acoustic damping. Since the efficient acoustic analysis program bArd (1998) was available now, it was used in the optimization. In this optimization, five radii of the bell's mid-plane were used as design variables. The wall thickness of the bell was kept constant. Starting points for the optimization were taken within a broad vicinity of bell B, see Figure 1. The following objective function was applied:

$$F = \sum_{i=1}^7 w_{f_i} (f_i - f_{opt_i})^2 + \sum_{i=1}^7 w_{a_i} (\eta_{a_i} - \eta_{a_{opt_i}})^2 + w_{fD} (fD / fD_{opt} - 1)^2 \quad (9)$$

where f_{opt_i} are frequency ratio target values, $\eta_{a_{opt_i}}$ are acoustic damping target values, and fD_{opt} is the target value for the size of the bell related to the lowest eigenfrequency. Weighing factors w_{f_i} , $w_{\eta_{a_i}}$ and w_{fD} were applied to the different terms. Additional constraints were placed on the frequency ratio values (target $\pm 1\%$), acoustic damping values (target $\pm 20\%$), and fD-value (target $\pm 5\%$). The first optimization runs showed violated acoustic damping constraints. After the damping constraints were relaxed considerably, convergence occurred within 10 design cycles. In Table 1 targets, initial and optimal design results are listed.

Mode	Frequency ratio			Acoustic damping (*E-04)		
	initial	target	final	initial	target	final
1	1.07	1.000	1.00	0.24	0.21	0.30
2	2.17	2.000	2.02	0.45	0.74	0.56
3	2.46	2.520	2.51	0.66	0.72	1.19
4	3.45	2.997	3.02	2.25	2.81	1.99
5	4.00	4.000	4.00	1.65	2.92	2.81
6	6.00	5.993	5.82	2.64	0.71	0.79
7	8.40	8.000	7.96	0.93	0.65	1.00

Table 1: Initial, target and final values of frequency ratio and acoustic damping.

The optimization tool worked well, but, substantially improved damping values were not obtained. The optimization runs repeatedly converged to bells that were very close to bell B, which is not so surprising. Based on the results, future detailed studies with more and other design variables included in the optimization process are advisable. With 47 structural analyses and 259 acoustic analyses the computation time for this optimization run was 70 minutes. The approach presented by Roozen-Kroon (1992), using a 3D BEM model, would have required 64 hours of computation time for the same number of analyses on the same computer. Hence, a great reduction in total computation time is achieved by using the Fourier-BEM approach.

4. SOUND POWER REDUCTION OF MRI SCANNERS

To demonstrate the potential of the developed acoustic tools, Kuijpers (1999) performed four parameter studies for different MRI scanner models, the first of which will be presented here. To compute the vibration excitation, a layered finite element model of the gradient coil was used, see Figure 4. In this study, the inside surface of the gradient coil radiates directly into the bore of the scanner. The remainder of the scanner was assumed to be rigid. The finite element model was excited with a Lorenz force distribution defined by a single circumferential harmonic. The response of the Fourier-FEM model was computed with the SATURN package (Kessels et al. (1998)). In SATURN it is possible to perform automatic parameter studies. This feature was used here to access the influence of the thickness of different material layers in the gradient coil system, see Figure 4.

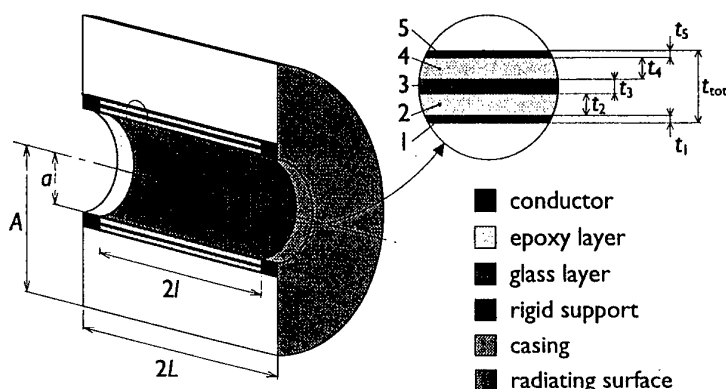


Figure 4: MRI model with layered gradient coil system.

The total thickness was kept constant ($t_{tot} = 100$ mm) in the parameter study. The thicknesses of the conductor layers were also kept constant ($t_1 = t_5 = 10$ mm). Then the thickness of the first damping layer, t_2 , and the stiff layer, t_3 , were both varied between 5 and 35 mm in steps of 5 mm, making a grid of 49 different gradient coil designs. Layer thickness t_4 was treated as a dependent variable.

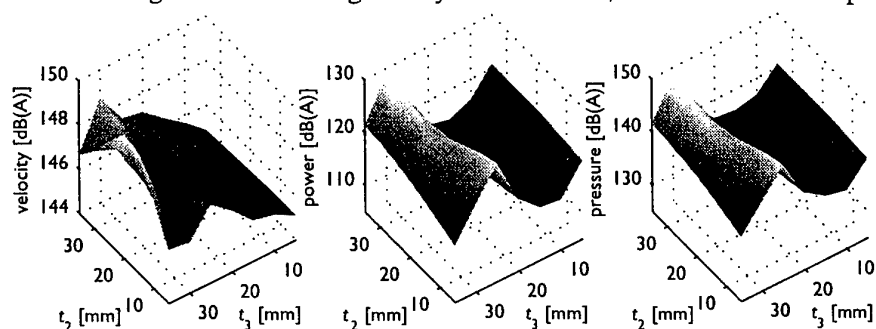


Figure 5: Response surfaces for MRI scanner parameter study.

The acoustic response of the MRI models was computed with the program bArd (1998). The response surfaces that resulted from this design study are depicted in Figure 5. A comparison between the response surfaces for velocity level and sound power level shows that they do not display similar parameter sensitivities. This means that gradient coil designs with low vibration levels do not necessarily coincide with designs that have low sound power levels. The correlation between sound power and sound pressure level, however, is very good. This means that a low sound power

design is also a low sound pressure design here. This close agreement makes the pressure calculations superfluous, and it is possible to decrease the computational effort for the acoustic analyses by using the radiation modes techniques as described in section 'Methods for acoustic analysis'.

Whereas Kuijpers(1999) used rather arbitrary layer configurations in his parameter studies, Kessels (2000a) performed parameter studies applying models that better resemble the actual structural coil behavior. Here, the goal was to identify relevant design variables, and to determine magnitudes of sound power reductions that can be obtained. These studies revealed the modulus of a bonding layer and the thickness of the fiber reinforced carrier layer as interesting design variables. Using these design variables, Kessels minimized the total sound power at constrained weight of the gradient coil. For that purpose, a multi-point path mid-range method has been implemented in MATLAB, which has resulted in the Approximate Optimization Toolbox (Kessels,2000). The user can select from different approximation models. This toolbox is combined with the MATLAB Optimization Toolbox (Coleman et al., 1999). The optimization resulted in a lowering of the sound power level from 103 dB to 90 dB, and a coil weight increase from 660 kg to 685 kg. The sound power reduction was obtained by increasing the thickness of the carrier layer, resulting in an increase of the total mass of the coil.

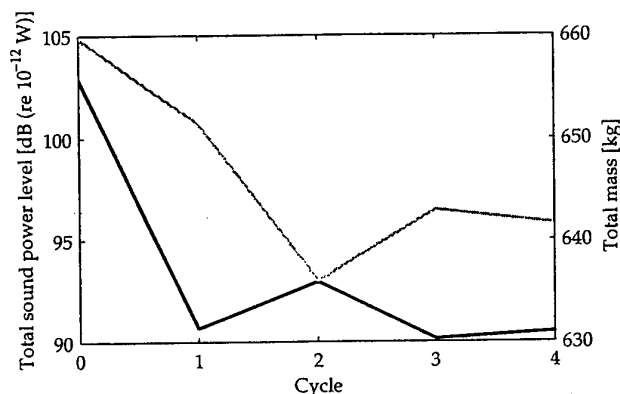


Figure 6: History of the multi-objective optimization example.
 — : total sound power level; - - - : total mass.

As an example of multi-objective optimization, both the sound radiation and the mass of the coil were treated as objectives. The goals were set as 90 dB for the sound radiation, and 640 kg for the mass. A particular objective can be given a larger relative importance by assigning appropriate weighing factors. The same starting point as in the previous optimization was used. As can be seen from the optimization history in Figure 6, the goals are almost met after four cycles, taking 13 Fourier-FEM analyses. Using the mass as a second objective has forced the optimization process to lower sound radiation by reducing the thickness of the carrier layer. The total weight has been reduced by almost 20 kg, and this is an improvement of 45 kg compared to the optimum of the weight constrained optimization example.

CONCLUSIONS

There has been a fruitful crossover between the bell design project and the MRI scanner project. Analysis and optimization methods for bell design have helped to create such tools dedicated to

design more quiet MRI scanners. For bell design, the program bArd, developed within the MRI scanner project, helped much in understanding modal acoustic damping behavior.

The acoustic analysis program bArd has proven to be very efficient. This enables direct implementation of acoustic analyses in the design process of bells and MRI scanners. For MRI scanners the structural-acoustic design tools already gave nice multidisciplinary design optimization results. However, for structural-acoustic optimization of bells the potential of the developed acoustic analysis tools have not been yet fully exploited. This will be subject of future research.

REFERENCES

bArd (1998) *Version 3.0 manual*. Computational and Experimental Mechanics, Department of Mechanical Engineering, Eindhoven University of Technology.

Coleman, T., Branch, M., Grace, A. (1999) *Optimization Toolbox User's Guide, Version 2*. The Mathworks, Inc., Natick, USA.

Kessels, P.H.L., Kuijpers, A.H.W.M., Verbeek, G., Verheij, J.W. (1998) *Structural-Acoustics Toolbox for Understanding and Reducing Noise (SATURN)*. INTERNOISE 98, Christchurch New Zealand.

Kessels, P.H.L. (2000) *Approximate Optimization Toolbox*. Paragon Numerical Engineering, <http://www.dolfyn.nl>.

Kessels, P.H.L. (2000a) *Engineering toolbox for structural-acoustic design – applied to MRI-scanners*. Ph.D. thesis, Eindhoven University of Technology, The Netherlands.

Kuijpers, A. (1999) *Acoustic modeling and design of MRI scanners*. Ph.D. thesis, Eindhoven University of Technology, The Netherlands.

Roozen-Kroon, P.J.M. (1992) *Structural optimization of bells*. Ph.D. thesis, Eindhoven University of Technology, The Netherlands.

Schoofs, A.J.G. (1987) *Experimental design and structural optimization*. Ph.D. thesis, Eindhoven University of Technology, The Netherlands.

Schoofs, A., Van Asperen, F., Maas, P., Lehr, A. (1987a) A carillon of major-third bells I : Computation of bell profiles using structural optimization. *Music Perception*, vol. 4, no. 3, pp 245-254.

Schoofs, A.J.G. (1994) A new model major third bell (in Dutch). *Messages from The National Carillon Museum*, No. 9 April 1994, pp. 12-14, Asten, The Netherlands.

SYSNOISE (1996). *Revision 5.3 On-Line Documentation*. LMS Numerical Technologies, Leuven Belgium.

Van Asperen, F.J.G. (1984) *Optimization of the eigen frequencies of axisymmetric structures, applied to swinging bells and carillon bells* (in Dutch). Master thesis, report WFW84.012.

Van Houten, M.H. (1998) *Function approximation concepts for multidisciplinary design optimization*. Ph.D. thesis, Eindhoven University of Technology, The Netherlands.

AEROELASTIC TAILORING OF AERODYNAMIC SURFACES AND LOW COST WIND TUNNEL MODEL DESIGN

Otto Sensburg, J. Schweiger
Daimler Chrysler Aerospace, Germany
and

V.A. Tischler and V.B. Venkayya
Air Force Research Laboratory
Air Vehicles Directorate
Wright Patterson Air Force Base, OH 45433-7531

Abstract

The theme of this paper is to demonstrate the advantages of the multidisciplinary optimization approach for the aeroelastic tailoring of lifting surfaces to significantly improve their stability and control performance. Although the procedure is valid for lifting surfaces, in general, the present emphasis is on the design of empennage surfaces, such as, vertical tails and by implication horizontal tails and canards as well. The tailoring concepts are demonstrated using a vertical tail of a typical fighter aircraft with variations on the boundary conditions.

1. Introduction

The stability and control aspect of an airplane is a well studied subject from the very early development of aeronautics. The wing control surfaces, canards, vertical and horizontal tails are some of the control surface configurations proposed for enhancing dynamic stability. Earlier studies have treated the parent structures (structures on which the control surfaces are mounted) as rigid elements. However, it was soon recognized that structural flexibility cannot be ignored as the systems are designed for higher speeds. In fact the structural flexibility can be used very creatively for enhancing aerodynamic performance.

The control surface effectiveness, reversal and divergence are some of the static aeroelastic measures considered in the design of lifting surfaces. Flutter, although it represents a dynamic aeroelastic phenomena, can be included in the same category.

The aeroelastic behavior of simple wings is often explained in terms of the relative location of three important axes⁽¹⁾ a) aerodynamic axis, b) elastic axis and c) mass cg axis. The aerodynamic axis is the locii of aerodynamic centers along the wing span. Similarly, the elastic and mass cg axes are the locii of shear centers and mass cgs of the wing sections. For simple, high aspect ratio, unswept wings with constant cross-

section, the elastic axis can be computed easily. These computations can be extended, in an approximate sense, to moderately swept and variable section wings made of isotropic materials. However, the elastic axis definition becomes quite murky when applied to low aspect ratio, highly swept wings made of non-isotropic materials. The aerodynamic axis definition is generally expressed in terms of percentage of chord length from the leading edge. This definition may not bear with reality in the case of severe wing twist. The mass cg axis, on the other hand, is easy to compute. Of course, the mass cg axis has no role in explaining the static aeroelastic behavior. The relative location of these three axes, with respect to each other, affects the aeroelastic response of the lifting surfaces. Here the implication being that if these relative locations can be altered by formal optimization procedures, then the aeroelastic response can be tailored for improved performance of the lifting surfaces.

For aeroelastic analysis in the context of the use of software systems such as NASTRAN, ASTROS^(2,3), etc, an explicit computation of the location of these axes is not required. Although it would be very useful, if the axes could be located with reasonable accuracy, in providing insight into the understanding of the aeroelastic behavior. In NASTRAN and ASTROS the structure is represented by finite element models, and the aerodynamics is modeled by discrete panels. This analysis implicitly accounts for the effect of these axes.

Vertical tails designed for high speed aircraft suffer from reduced stability and control effectiveness at high dynamic pressures due to large aeroelastic deformations. Therefore, to assure adequate tail performance results in an increased size. Since the tails are most effective furthest from the cg of the aircraft, the increased size is tantamount to higher aspect ratio tails. These considerations invariably lead to stiff and heavy tail structures. These large tails are also subject to burst vortex or shock induced buffet which causes fatigue problems. Their size and structural constraints

cause weight, drag, and radar-cross section penalties. These penalties can be significantly reduced by the application of divergent flexible technologies to the vertical tail design problem, which results in a lighter structure and potentially smaller size to reduce buffet, drag and observables. In some cases the smaller size requirement could remove the necessity of two vertical tails.

The vertical tail is a stiffness design, because flight loads are much lower than on the wings of fighter airplanes. Therefore there is a wider scope for variation of the carbon fiber composite layers, and it can have a more significant impact on performance than on wings, where much of the stiffness is defined by strength constraints.

The objective of this paper is to demonstrate increased tail effectiveness at high speeds. This could lead to decreased tail size and structural weight that meets or exceeds all tail performance and observables goals.

The technology applied is called 'Active Flexible Technology' which is a multidisciplinary, synergistic technology that integrates aerodynamics, controls, and structures together to maximize air vehicle performance by allowing thinner, higher aspect ratio surfaces that are aeroelastically deformed into shapes for optimum performance. This was first described extensively in References (4-7).

For high speed the vertical tail is sized to give a certain minimum value of the directional static stability derivative. For low speed the rudder power unit must be adequate to hold a sideslip of $\beta=11.5^\circ$ at the approach speed for a cross wind landing. It also must cover the one engine out case, which is important for transport aircraft. This low speed requirement may reduce the possibility to cut the fin span and area commensurate with positive high speed aeroelastics.

2. Test Case for the Design Study

A typical vertical tail of a fighter was selected for this design study. A number of other investigators have used various versions of this configuration as a benchmark to test new concepts in the emerging multidisciplinary approach to design under the title, 'Diverging Flexible Vertical Tail (DFVT) Technology' ^(8,9). Some results of the conventional design are also available from DASA sources in Germany for comparison as well as for establishing the advantages of the new technology.

DASA engineers, in particular, have made extensive studies on this configuration using their LAGRANGE

optimization program. This configuration consists of a fin and a rudder as shown in Figure 1.

Because of the low aspect ratio of the chosen vertical tail design, $AR=1.2$, this is an ideal candidate for applying aeroelastic tailoring for a carbon fiber composite structure (Fig. 2). As can be seen in the figure the higher the aspect ratio is, the higher the weight penalties to meet the performance goals.

2.1 Structural Description of the Fin and Rudder

The overall geometry of the fin is given in Figure 1. The surface area is 5.46 m^2 and the leading edge sweep angle is 45° . The fin box has one shear pick-up in the front and one bending attachment at the rear. The rudder actuator is connected with two rods for control actuation. Fin box and rudder skins are built as carbon fiber symmetric laminates. A quasi isotropic glass fiber laminate is used for the tip structure with contains avionics equipment. The fin box and rudder are coupled by three hinges.

Four materials were used: Carbon Fiber Composite, CFC; Graphite Fiber Composite, GFC; Aluminum and Titanium. The materials were distributed as follows:

1. Fin Box Skin - Eight layer CFC laminate
2. Rudder Skin - Six layer CFC laminate
3. Tip Skin - Quasi Isotropic GFC
4. Fin Box Rear Spar - Eight layer CFC laminate
5. Rudder Main Spar - Eight layer CFC laminate
6. Remaining Spars - Aluminum
7. Fin Box End Rib - Titanium
8. Rudder End Ribs - Titanium
9. Remaining Ribs - Isotropic CFC

3. Theoretical Basis

A statement of the optimization problem for the design study is as follows:

Minimize or maximize a desired objective function

$$F(\mathbf{X}) = F(X_1, X_2, \dots, X_n) \quad (1)$$

subject to a set of system response constraints

$$G_i(X_1, X_2, \dots, X_n) \leq G_{i0} \quad i = 1, 2, \dots, p \quad (2)$$

and constraints on the elements of the $\mathbf{X}$ vector

$$X^L \leq X \leq X^U \quad (3)$$

The weight of the fin was selected as the objective function, $F(\mathbf{X})$, for this study. The response constraints, $G(\mathbf{X})$, were derived from the simultaneous analysis of the structure in three areas: a) static strength, b) static aeroelastic response and c) dynamic aeroelasticity - flutter. The variable vector, $\mathbf{X}$, represents the thicknesses and directional properties of the fiber reinforced composite skins.

The optimization problem was to find the optimal variable vector, $\mathbf{X}$, that corresponds to the minimum weight of the structure without violating any of the response constraints defined by Equation (2). Also, this variable vector had to be within the range defined by Equation (3).

The solution of the optimization problem stated by Equations (1-3) involves three major steps:

1. Selection of an initial variable vector, $\mathbf{X}^0$
2. Evaluation of the objective and constraint functions, Equations (1-2)
3. A search strategy to move to a new variable vector, $\mathbf{X}^1$, in an n -dimensional design space, and eventually to the optimal vector, $\mathbf{X}_{opt}$.

This is an iterative procedure, and the issues of solution convergence need to be addressed. The first step, the selection of the initial vector, is generally arbitrary, although it will have a significant impact, if there is a potential for multiple minimums. The second step, objective and constraint function evaluation, is referred to as an analysis of the system, and it requires a significant computational effort for complex systems. Additional details of this step are discussed in the next subsection. The third step, the search strategy in n -dimensional space to locate the optimum point, depends on the type of algorithm selected. A first order optimization algorithm, also known as a gradient method, was used in the ASTROS software system in this study. Except for the zero-order methods, all other algorithms require an additional step called a sensitivity analysis. The ASTROS system uses analytical gradients based on the analysis used in the second step.

As stated previously, the function evaluations in the second step require an analysis, and in general, it is multidisciplinary, because of the type of response functions involved. In this study two types of aeroelastic response functions are addressed, in addition to the static strength of the structure:

- static aeroelasticity
- dynamic aeroelasticity

A brief description of the first function is given here. A more detailed discussion can be found in References 3 and 10.

The response of a structure subjected to external forces can be described by a simple equation in the context of a finite element discretization of the structure, i.e.

$$\mathbf{KU} = \mathbf{P}(\mathbf{U}) \quad (4)$$

where $\mathbf{K}$ is the $n \times n$ stiffness matrix of the structure, and $\mathbf{U}$ is an $n \times 1$ vector of displacements, defined in reference to the structure's degrees of freedom. The left hand side of the equation represents the elastic forces in equilibrium with the aerodynamic forces on the right hand side. However, the forces due to the airflow are not independent of the deformation of the structure, and this is indicated by expressing $\mathbf{P}$ as a function of $\mathbf{U}$ as well. This is usually referred to as aeroelastic interaction. If the right hand side of the equation does not include inertia forces (accelerations are assumed to be zero), then it is referred to as static aeroelasticity.

An approximate form of $\mathbf{P}(\mathbf{U})$ can be written as

$$\mathbf{P}(\mathbf{U}) = q\mathbf{AU} + q\mathbf{A}^\alpha \alpha + q\mathbf{A}^\delta \delta \quad (5)$$

where q is the dynamic pressure, $\frac{\rho V^2}{2}$, ρ is the air density and V is the free stream velocity. $\mathbf{A}$ is the aerodynamic influence coefficient matrix with respect to the displacement degrees of freedom. $\mathbf{A}^\alpha$ and $\mathbf{A}^\delta$ are due to the lifting surface and the control surface angles of attack. α is the initial angle of attack and δ is the control surface displacement.

The case of static divergence of the lifting surface is represented by

$$[\mathbf{K} - q\mathbf{A}]\mathbf{U} = 0 \quad (6)$$

The solution of this complex eigenvalue problem (corresponding to the lowest real positive value) yields both the divergence dynamic pressure and the divergence velocity.

The lift effectiveness is another important static aeroelastic parameter, and it is defined as the ratio of the flexible lift to the rigid lift. It can be written as

$$LE = \frac{C_{LF}}{C_{LR}} = \frac{q\mathbf{h}^T \mathbf{A}^\alpha \alpha + q\mathbf{h}^T \mathbf{AU}}{\mathbf{h}^T \mathbf{A}^\alpha \alpha} \quad (7)$$

where C_{LF} is the flexible lift curve slope and C_{LR} is its rigid counterpart, and where $\mathbf{h}$ is a vector consisting of the aerodynamic panel lengths. Solving Equation 5 for U with $\delta = 0$ and substituting into Equation 7 gives the lift effectiveness equation

$$LE = \frac{\mathbf{h}^T [\mathbf{I} + q\mathbf{A}[\mathbf{K} - q\mathbf{A}]^{-1}] \mathbf{A}^a \alpha}{\mathbf{h}^T \mathbf{A} \alpha} \quad (8)$$

Similar expressions can be derived for control surface effectiveness (aileron effectiveness) as well as flutter velocity constraints. They are implemented as constraints in the ASTROS system.

4. ASTROS Capabilities

ASTROS is a multidisciplinary analysis and optimization system for the design of airframe structures. It is a multidisciplinary system, because it integrates the essential and relevant elements, the aerodynamics, structures and controls, that are the design drivers of the airframe structures representing the wings, fuselage and empennage surfaces. The multidisciplinary aspect of the program has been implemented in an integrated way, so that all the critical design conditions are considered simultaneously. A concerted effort has been made to provide the user with a modeling and design tool that has general capabilities and flexibility in its application. In addition to capturing the interaction of several disciplines, ASTROS can treat multiple boundary conditions and, within each boundary condition, multiple subcases.

Although ASTROS was primarily intended for the preliminary phase of aerospace structures design, with some finesse and understanding of modeling nuances, it can be adeptly used to address many of the conceptual and detailed design requirements in an optimization environment. It is a sophisticated, but a convenient, mathematical modeling tool intended for the perusal of a skilled modeler to solve complex engineering problems and obtain valuable information on the behavior of the flight vehicle.

Compatibility with the current aerospace analysis and design environment was another major consideration in the development of the ASTROS system. Its data input stream and pre- and post-processor interfaces resemble those of the most popularly used aerospace structural analysis program, NASTRAN. Its solution commands, however, were tailored for multiple analysis and design conditions required for simultaneous optimization.

The specific analysis and design capabilities of ASTROS include:

1. Static structural analysis
2. Dynamic structural Analysis, including modal/eigenvalue analysis, frequency response and transient response analysis
3. Steady and unsteady flight loads analysis including trim conditions
4. Static and dynamic aeroelastic response
5. Optimization

ASTROS computes over two dozen response quantities from the various disciplines cited. Some of these are stresses, strains, frequencies, mode shapes, stability derivatives, static aeroelastic effectiveness parameters, flutter, etc. Any of these response quantities or combination of them can be formulated as objective or constraint functions in optimization.

The static analysis methodology is based on a finite element representation of the structure, as are all the structural analysis disciplines in ASTROS. The static analysis module computes responses to statically applied mechanical (e.g. discrete forces and moments), thermal and gravity loadings. Static deformations and their resultant stresses are among the computed responses. An extensive design capability is provided for the static analysis discipline. It provides the capability to analyze and design linear structures subjected to time invariant loading.

The modal analysis feature in ASTROS provides the capability to analyze and design linear structures for their modal characteristics, i.e. eigenvalues and eigenvectors. The design aspect of ASTROS places limits on the frequencies of the structure. The modal analysis is not only useful in its own right, but it also provides the basis for a number of further dynamic analyses. Flutter and blast response analyses in ASTROS are always performed in modal coordinates. Transient and frequency response analyses can be performed in either modal or physical coordinates, at the selection of the user.

Steady aerodynamics are used for the computation of external loads on aircraft structures. The static aeroelastic analysis features in ASTROS provide the capability to analyze and design linear structures in the presence of steady aerodynamic loading. This provides the ASTROS user with a self-contained capability to compute loads experienced by a maneuvering aircraft and to redesign the structure based on these loads. The capabilities available for steady aerodynamics design include specifying limits on: (1) the allowable stress or strain response due to a specified trimmed maneuver, (2) the flexible to rigid ratio of the aircraft's lift curve slope, (3) the flexible roll control effectiveness of any

antisymmetric control surface and (4) the values of the flexible stability derivatives and trim parameters.

Flutter analysis in ASTROS provides the capability to assess the aeroelastic stability characteristics of the designed structure and to correct any deficiencies in a systematic fashion. Both subsonic and supersonic analyses are available and, reflecting the multidisciplinary character of the procedure, the design task can be performed with any number of boundary conditions and flight conditions. In this way, all critical flutter conditions can be analyzed and designed for simultaneously.

Dynamic response analysis is performed for loadings which are a function of time or frequency.

The final discipline listed above is that of optimization. If only stress or strain constraints are included in the design task, the fully stressed design option may be used. For more general design tasks, a mathematical programming approach has been implemented.

5. Design Conditions

- Objective function is weight

The constraints are:

- Maximum Strength Constraint

Allowable strain in the fiber/transverse direction

Tension : $\epsilon = .0037$

Compression : $\epsilon = -.0028$

- Composite Constraints

Minimum gauge for a single layer was .125mm for a symmetric laminate distribution over the thickness, i.e. two layers of the same fiber orientation are necessary.

Minimum gauge for the composite laminate was 2.0mm

Not more than 2/3 of the total thickness could belong to one fiber orientation.

- Static aeroelastic efficiency constraints were imposed for both the vertical tail and the rudder while being trimmed for lift.

- Minimum flutter constraint was 530 m/sec at Ma 1.2.

6. Comparison of NASTRAN and ASTROS Results with Existing DASA Data

In order to become familiar with the DASA finite element model of the fin and rudder, several NASTRAN and ASTROS analyses were performed, and the results were compared with existing DASA data. Correlation was found to be excellent. After that exercise the DASA model was changed. To allow different attachment conditions the general stiffness element, GENEL, giving the effect of the fuselage stiffness, was removed and replaced with single attachment springs. These springs were tuned so that the model would give the original DASA result. ASTROS and NASTRAN results are identical, because the ASTROS-code uses the finite element description of NASTRAN. The results of this comparison can be found in Table 1.

7. Results of the Optimization with ASTROS

Several computer runs were made subject to the constraints defined in Section 5 trying to match the DASA results for a fin efficiency of 0.814 at Ma 1.8, 102kPa. The rudder efficiency was fallout at 0.3799. The ASTROS code reduced the weight for this configuration to 81.1 kg. The weight of the initial design was 99.4 kg. When a rudder efficiency of .5 was imposed, the weight was 94.3 kg for a fin efficiency of 0.814.

Higher fin efficiencies were requested, and the weights for these designs are plotted in Fig. 3. While 0.9 can be reached with additional weight, higher efficiencies need excessive weight penalties. When the rudder efficiency was treated as fallout, then the weight reduces considerably. The fallout is quite reasonable and sufficient for a feasible design. From Fig. 3 it can be seen that a fin efficiency of 1.0 can only be achieved with infinite weight.

The picture changes completely when Ma 0.9 subsonic air forces are used (Fig. 4). Now efficiencies higher than 1.0 are reached. As can be seen, with little additional weight, 1.3 can be reached for a high dynamic pressure, q , of 102 kPa, which is not possible for air. The highest q is 57 kPa for Ma 0.9, i.e. sea level in air. This trend is also verified in Fig. 5 which clearly shows that the wash-in angle increases for higher efficiencies, which simulates basically a forward swept fin behavior (diverging!), and in Fig. 6 which shows a positive wash-in angle despite that it is a swept back surface.

8. Physical Explanation of the Basic Mechanism of the DFVT

In order to understand the elastic behavior of the fin, an equivalent beam is assumed which contains the

stiffness of the fin. This beam would be located at the elastic axis, which is a spanwise line thru the shear centers of each cross section. The shear center of each cross section is computed by establishing the point in the plane of the section at which a normal shear load can be applied without twisting the section or where a torsion moment can be applied to the section without producing a deflection at the shear center. An effective elastic axis was defined by using the deflection of two points fore and aft on the chord, where a moment was applied at the tip, assuming small angles and that the deflection varies linearly along the chord. Fig. 7 shows the elastic axis location. From this figure one can assess why it is impossible to get a wash-in effect (diverging) for the supersonic Ma 1.8 case. The center of pressure – at 30% span and 50% chord – just reduces any initial angle attack of the fin, and therefore the best fin efficiency which can be reached with aeroelastic tailoring is 1.0, which is the rigid behavior and needs a lot of structural weight. At the subsonic case, Ma 0.9, there exists some possibilities for wash-in, because the aeroelastic tailoring also shifts the so called elastic axis. This behavior is shown in Fig. 4 and also in Fig. 8 for an optimized case of Ma 0.9, 102kPa and fin efficiency of 1.3.

9. Results for Shifting the Fin Attachments Back

This behavior changes drastically when the fin attachments are shifted back. The x-position for the forward attachment was shifted back from $x = 450\text{mm}$ to $x = 950\text{mm}$. The x-position for the rear attachment was shifted from $x = 1750\text{mm}$ to $x = 2300\text{mm}$. The new positions can be seen in Fig. 9. Now the centers of pressure are forward of the elastic axis, and wash-in behavior can be expected for both the subsonic and the supersonic cases (Fig. 9). For Ma 0.9, 57 kPa a fin efficiency of 1.3 can be reached with practically no weight increase. Also the rudder efficiency increases from 0.5 to about 0.7. This can be seen in Figure 10. For the supersonic case Ma 1.8, 102kPa the behavior is similar (Fig. 11), and 1.3 can also be reached with an optimized laminate. The rudder efficiency is now reduced to 0.5. The flutter speed is 530m/sec. As an item of interest an analysis was performed (no optimization) to find the fin and rudder efficiency at Ma 0.9, 57kPa for the laminate of Ma 1.8, 102kPa. This shows a fin efficiency of 1.3 and a rudder efficiency of 0.8. Figure 12 shows the thicknesses of Layer 1 and 8 for the fin box and rudder skin for Ma 1.8, 102kPa and an effectiveness of 1.3.

10. All Moveable Vertical Tail

An all moveable vertical tail is not a new invention. It was utilized on the very successful Lockheed SR71

Blackbird and on the Lockheed F117 stealth fighter. Also a British prototype aircraft of the 1960's, the TSR2, had a vertical tail. It was also considered seriously on the European Fighter Aircraft, EFA. Lately it was discussed in reference 6. In the context of the DFVT it has several advantages:

- The rear yaw attachment can be moved far backward on the fin, because there is no rudder.
- It can also be utilized for the low speed regime (engine out or side wind requirement) where there is no aeroelastic effect, because the whole surface is rotated.

10.1 Structural Representation of the All Moveable Vertical Tail

The rudder was attached with stiff rods to the fin. The forward attachment was reduced to a very low stiffness. Reducing this stiffness results in a low yaw stiffness, which in turn reduces the flutter speed considerably as shown in Table 2.

10.2 Optimization Results

When the optimization code is used, a fin efficiency of 1.61 with a slightly reduced flutter speed of 500 m/sec can be achieved, which gives a 23% flutter margin at Ma 1.2 at sea level which is sufficient. Higher than 1.61 efficiency cannot be achieved as can be seen in Figure 13.

11 A Method to Design Low Cost Aeroelastic Wind Tunnel Models.

Considerable progress has been made in calculating the aeroelastic behavior of aircraft surfaces such as:

- Static deformations under airloads
- Static aeroelastic efficiencies
- Control surface reversal
- Divergence
- Flutter

by applying finite element methods for structural analysis and CFD methods for aerodynamics. To prove these methods with experiments, wind tunnel tests are the only way before airplane flight test commences. Finding errors by flight test is dangerous, time consuming and costly.

The cost of wind tunnel tests can be split up into two blocks:

- Manufacturing a wind tunnel model

- Running the wind tunnel
 - Cheap in low Speed
 - High in transonic Speed

including all the necessary measuring, recording and data reduction equipment.

In this report a method is described and applied whereby the use of a modern structural optimization code allows one to define the skin thicknesses of an aeroelastic model made from composite irrespective of whether the real aircraft is manufactured from metal or composite material. A similar attempt was made in reference 11.

11.1 Description of the Wind Tunnel Model Structure

The fabrication of the model was successfully applied in reference 12. To meet the skin scale requirements for the model the number of plies had to be reduced so that the percentages for the different fiber orientations within the local skin laminate can be realized. An unusually thin gage graphite prepreg unidirectional tape was specially fabricated for the model skins with .002 inch thickness instead of the commonly available .005 inch gage. This tape allowed a faithful scaled replica of the full scale skin ply thickness distributions at the various ply orientations. The skins of the model were fabricated in a negative mould. The core of the model consists of epoxy-foam. To avoid expensive machining the core was thermoplastically formed by pressing the heated foam together with the prefabricated skins in the mould. Ribs and spars were also made from carbon fiber material and alter the assembly with the skins the model was glued together in the mould. The fin and rudder fittings are metal parts and also replicas of the full scale design. Fig 14 shows the model which was built by the DFVLR Stuttgart (Institut für Bauweisen)

11.2 Design of a Transonic Wind Tunnel Model

The design of the model was performed by multiplying all the dimensions with the appropriate scale factor. The model length factor chosen was 0.2, because then the model span would be 0.5m, which would fit in most transonic wind tunnels. Table 3 defines the model scale factors

Length	0.2
Stiffness	0.2
Frequency	5.0
Mass	0.2 ³

Table 3

The original fin sizes were all scaled down with 0.2. The attachment springs were also multiplied with 0.2. The mass was scaled with 0.2³.

11.2.1 Design of Wind Tunnel Model 1

The vibration modes for the model were calculated and agree very well with the A/C fin. Also the efficiencies and the maximum deflection for the fin aeroelastic load case correlate very well (see Table 4). The flutter speed and the frequency are well met. This is for a model representing the rear attachment locations and the original DASA sizes.

The next step taken was to not allow any lower skin size than 0.125 mm. for one CFC layer. There is good correlation again. Efficiencies and maximum deflection can also be found in Table 4. This model seems to be 10% too stiff, so the deflection is 10% less and the efficiency for the fin case is also 10% reduced. This deficiency can be corrected with using a lower ply size of 0.050 mm which is available on the market.

11.2.2 Design of Wind Tunnel Model 2

Obviously, it is more rewarding to test a model which represents the diverging fin optimized for Ma 0.9, 57kPa with an efficiency of the fin at 1.4, the rudder at 0.77 and a flutter speed of 530m/sec. The same procedure as an wind tunnel model 1 was performed, but now with a minimum layer gauge of 0.050 mm. Results for this model are shown in Table 5 Now the fin and rudder efficiencies are reproduced very well Because the frequencies and modes are very well matched, there will be the same flutter speed as an the airplane.

11.3 Design of a wind tunnel model using glass fibre reinforced composites (GFC)

Using a material with a much lower elasticity modulus such as fiber glass has two advantages:

- The thicknesses of the various layer stacks are greater
- There is no need for an autoclave.

The constraints for wind tunnel model 2 were used:

- Fin efficiency 1,4
- Rudder efficiency 0,5
- Flutter speed 530 m/sec
- 57 kPa dynamic pressure which corresponds to Ma 0,9, sea level.

After optimisation the following results were achieved:

- Weight 136,0 kg, (almost twice the carbon fiber weight)
- Fin efficiency 1,4
- Rudder efficiency 0.6175
- Flutter speed 530 m/sec

Such a model can be used in a transonic wind tunnel to represent the initial optimum design at very low cost.

12. Conclusions and Recommendations

A list of possible benefits is presented below:

- The reduced tail size reduces the CD_0 drag.
- The reduced span and area reduces the exposure to upstream induced burst vortex and separated flow unsteady pressure fields which increases tail buffet fatigue life. The increase in life reduces repair and replacement life cycle costs.
- The reduced planform size reduces observable signatures to increase stealth mission capability and reduce detectability.
- Because of the possible size reduction one vertical tail should be sufficient even for carrier based airplanes.

- With proper multidisciplinary optimization a carbon fiber vertical tail can be made 30% more efficient than a rigid surface at the same weight.

- If the low speed requirement is not relevant, the area of the vertical tail can be reduced by 30%.

- An all moveable vertical tail could be the optimum solution for a fighter aircraft, because the yaw axis would be brought very far to the rear. It would also be a solution for a subsonic aircraft, because moving the whole tail would fulfill the low speed requirement.

- These conclusions are based on analytical studies, and they need to be validated experimentally in wind tunnel tests. An analytical method to lay out and fabricate a low cost wind tunnel model is presented.

References

1. Bisplinghoff, R.L., Ashley, H., and Halfman, R. L., Aeroelasticity, Addison-Wesley Publishing Co., Mass., 1955
2. Rodden, W.P. and Johnson, E.H., MSC/NASTRAN Aeroelastic Analysis Users's Guide – Version 68, The MacNeal-Schwendler Corp., Los Angeles, Calif., 1994
3. Neill, D.J., Herendeen, D.L., and Venkayya, V.B., ASTROS Theoretical Manual, USAF WL-TR-95-3006
4. Shirk, M. H., Hertz, T. J., and Weisshaar, T. A., "A Survey of Aeroelastic Tailoring Theory, Practice, Promise", AIAA Paper AIAA-84-0982-CP, 25th Structures, Structural Dynamics and Materials Conference, Palm Springs, California, 1984
5. Love, M. H., "Integrated Airframe Design at Lockheed Martin Tactical Aircraft Systems", AGARD Report 814, Integrated Airframe Design Technology, NATO, Oct 1996
6. Pendleton, E., Bessette, D., Field, P., Miller, G., and Griffin, K., "The Active Aeroelastic Wing Flight Research Program", 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 1998
7. Flick, P. and Love, M., "The Impact of Active Aeroelastic Wing Technology on Conceptual Aircraft

Design", AVT Panel Meeting Proceedings, Ottawa, Canada, Fall 1999

8. Schneider, G., Krammer, J., and Hornlein, H.R.E.M., "First Approach to an Integrated Fin Design", AGARD Report 784, Integrated Design Analysis and Optimization of Aircraft Structures

9. Schweiger, D. and Krammer, J., "Active Aeroelastic Aircraft and its Impact on Structure and Flight Control System Design", AVT Panel Meeting Proceedings, Ottawa, Canada, Fall 1999

10. Bowman, K.B., Grandhi, R.V., and Eastep, F.E., "Structural Optimization of Lifting Surfaces with Divergence and Control Reversal Constraints", Structural Optimization 1, pgs 153-161, Springer-Verlag 1989

11. French, M., Eastep, F.E. "Aeroelastic Models Design Using Parametric Identification", AIAA-94-1422-CP.

12. Schneider, G., Hoenlinger, H., Guldner, W., Manser, R. "Aeroelastic Tailoring Validation by Windtunnel Model Testing ", European Forum on Aeroelasticity and Structural Dynamics, Aachen, Germany 1989.

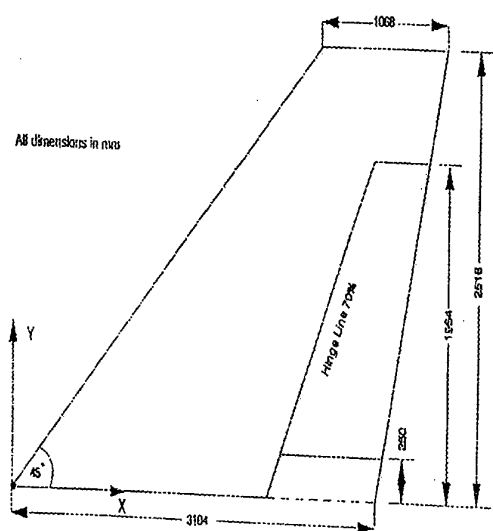


Figure 1 Fin Geometry

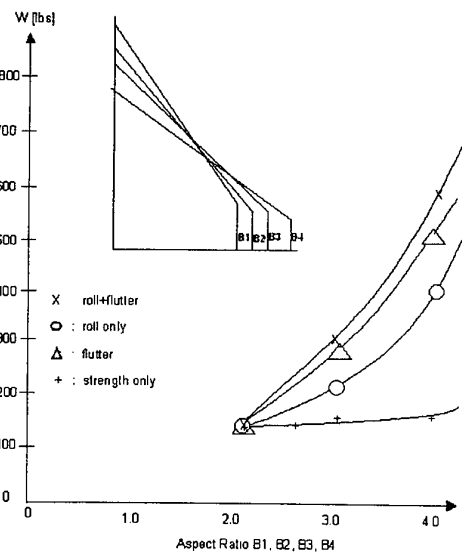


Figure 2 Structural Weight for Various Constraints vs Aspect Ratio

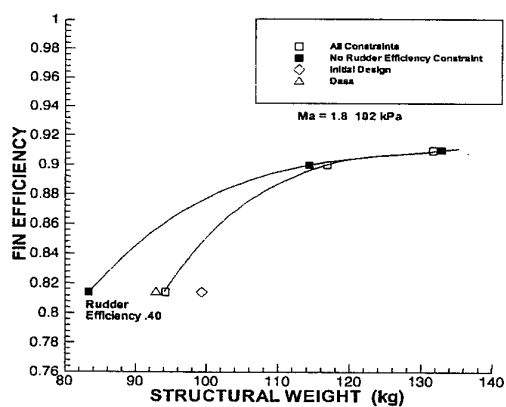


Figure 3 Fin Efficiency vs Structural Weight for Ma 1.8 102kPa

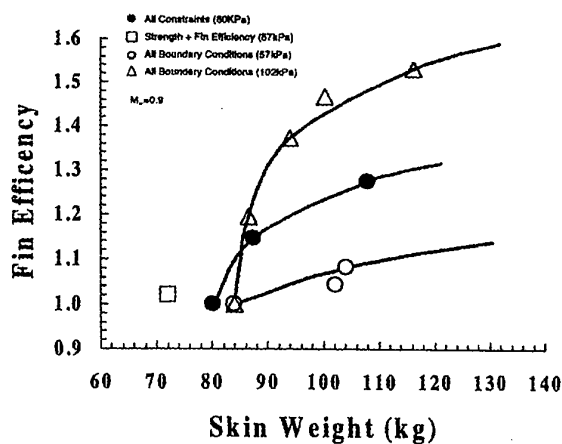


Figure 4 Fin Efficiency vs Structural Weight for Ma = 0.9

	Initial Design ASTROS	Initial Design DASA	With Single Springs	Optimum Design	
				ASTROS	DASA
Weight [kg]					
Structure	99.4	99.4	99.4	94.3	92.9
Non Structure	53.6	53.6	53.6	53.6	53.6
Total	153.0	153.0	153.0	147.9	146.5
Deflections [mm]					
Load Case 1	304	291			
Load Case 2	384	367			
Load Case 3	148	154			
Load Case 4	220	231			
Load Case 5	146	159			
Frequencies [Hz]					
f_1	9.1	8.9	9.0	8.89	9.2
f_2	30.5	29.8	30.0	28.84 (f+a)	30.2 (f+a)
f_3	32.5 (fore+aft)	31.2 (f+a)	43.9 (f+a)	41.03	30.6
f_4	41.4	40.0	41.6	42.39	41.08
f_5	55.7	54.9	57.6	59.36	58.31
Ma 1.2 S.L.					
Flutter Frequency – f_f [Hz]	20.2	21.2	20.0		
Speed – v_f [m/s]	493.4	495.0	534.0	530.0	530.0
Ma 1.8 102kPa – Aeroelastics					
Fin		0.753	0.740	0.814	0.814
Rudder		0.441	0.423	0.500	0.500
Aeroelastic Deflections [mm]					
Fin 1^0	65.34	53.7			
Rudder 1^0	8.88	8.29			

Table 1 Comparison of DASA and ASTROS Results

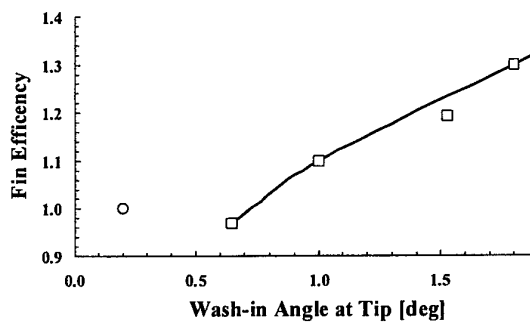


Figure 5 Fin efficiency vs Wash-in Angle at Tip

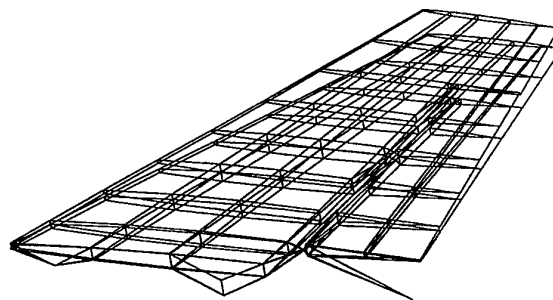


Figure 6 Displacement in Aeroelastic Case Side Slip -
Fin Efficiency 1.2, Ma = 0.9 102 kPa

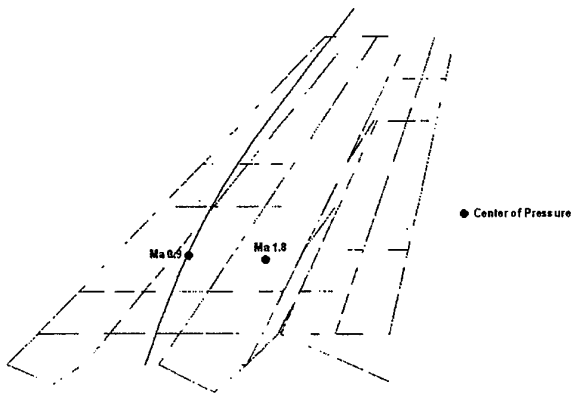


Figure 7 Elastic Axis Location (Original Attachments and DASA Skin Thicknesses)

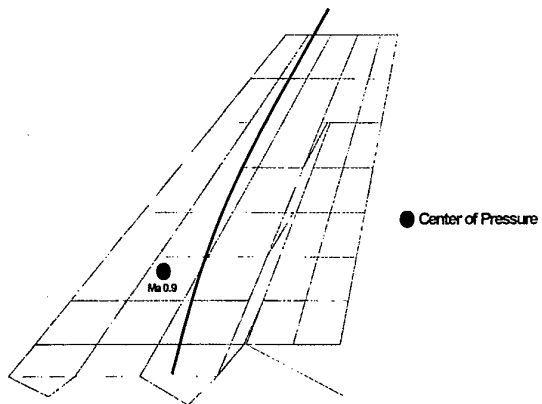


Figure 8 Elastic Axis Location for Ma 0.9 102kPa
Fin Efficiency 1.3

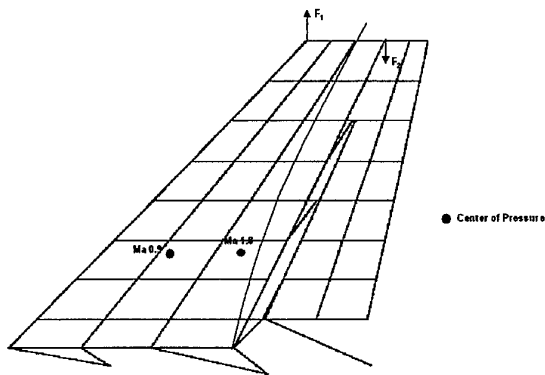


Figure 9 Elastic Axis Location (Rear Location)

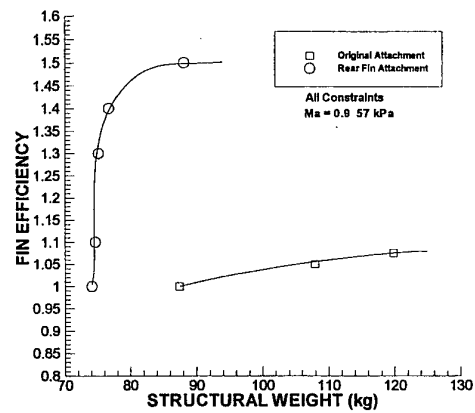


Figure 10 Fin Efficiency vs Structural Weight for
Ma = 0.9 57 kPa

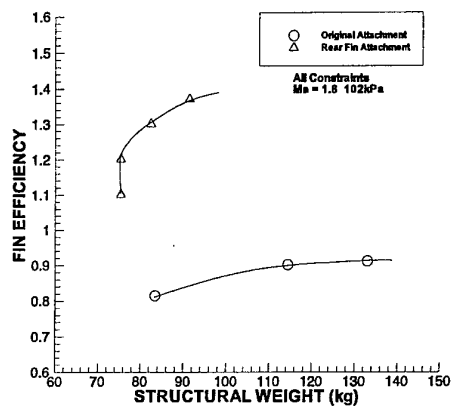


Figure 11 Fin Efficiency vs Structural Weight for $Ma = 1.8$ 102kPa

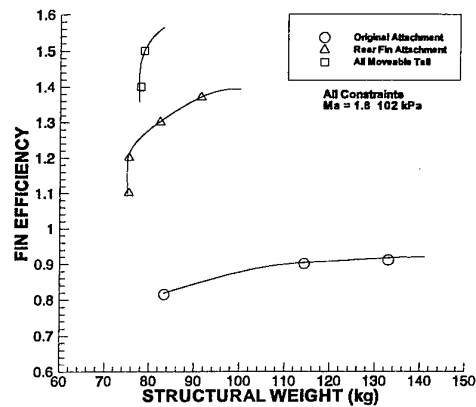


Figure 13 Fin Efficiency vs Structural Weight for $Ma = 1.8$ 102kPa

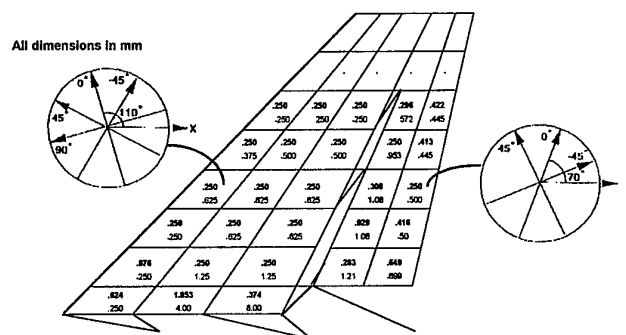


Figure 12 Thickness of Layer 1 After Optimization ($Ma = 1.8$ 102kPa, 1.3 Efficiency)

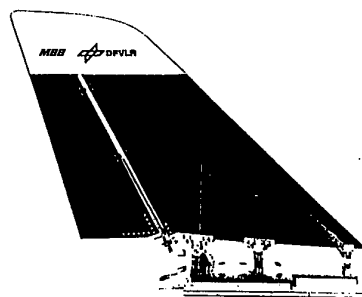


Fig. 14 Aeroelastic Fin Model

	Original DASA Sizes With Forward Attachment	Original DASA Sizes With Low Stiffness Forward Attachment
Mode 1	8.31 Hz	8.20 Hz
Mode 2	26.72 Hz	19.69 Hz
Mode 3	43.01 Hz (fore and aft)	43.01 Hz (fore and aft)
Mode 4	46.11 Hz	36.88 Hz
Mode 5	53.71 Hz	52.40 Hz
	Flutter Speed m/sec 526.00 Flutter Frequency Hz 18.27	375.04 13.95

Table 2 Influence of the Forward Attachment on the Dynamic Properties

	ORIGINAL SIZES [Hz]	WIND TUNNEL MODEL [Hz]	0.125 mm MINIMUM PLY SIZE [Hz]	MODE DESCRIPTION
MODE 1	8.36	8.02	8.06	First bending
MODE 2	26.38	26.22	27.46	First torsion
MODE 3	40.39	44.62	40.14	Rudder mode
MODE 4	44.33	42.20	42.84	Fore and Aft Mode
MODE 5	51.97	52.62	55.46	Tip torsion
Efficiency Fin	0.9872	0.9966	0.9075	
Efficiency Rudder	0.6667	0.7884	0.7906	
Max deflection 1 degree fin case	74.79 mm	74.55 mm	67.90 mm	
Flutter speed M/sec, Hz	514 (17.84	524 (91.66) (18.33)		

Table 4: Modal Frequencies, Efficiencies, Deflections for Model 1

	ORIGINAL SIZES [Hz]	WIND TUNNEL MODEL [Hz]	0.050 mm MINIMUM PLY SIZE [Hz]	MODE DESCRIPTION
MODE 1	7.55	7.24	7.28	First bending
MODE 2	24.46	24.42	24.62	First torsion
MODE 3	34.79	39.00	38.38	Rudder mode
MODE 4	42.70	40.60	40.56	Fore and Aft Mode
MODE 5	47.15	47.50	46.38	Tip torsion
Efficiency Fin	1.4	1.38	1.36	
Efficiency Rudder	0.77	0.84	0.82	
Max deflection 1 degree fin case	156.53 mm	154.95 mm	150.26 mm	

Table 5: Modal Frequencies, Efficiencies, Deflections for Model 2

THE OPTIMAL DESIGN OF A PLANAR PARALLEL PLATFORM FOR PRESCRIBED MACHINING TASKS

W. J. Smit

Multidisciplinary Design Optimization Group (MDOG)
Department of Mechanical and Aeronautical Engineering
University of Pretoria, South Africa

1. ABSTRACT

In this paper a general methodology is presented for determining the optimum design of a planar parallel platform to be used in machining [1]. The method, based on a mathematical optimization approach, is used to find a platform design and placement such that firstly, the execution of a prescribed task path is feasible, and secondly, such that the actuator forces required to execute the prescribed task are minimized. The application of the method is illustrated for two prescribed tasks, five design variables and a number of geometrical inequality constraints such as actuator length limits. The method succeeds in finding locally optimum and feasible platform designs for which the required task lies inside the workspace. Two optimization algorithms are implemented and their respective results are compared. The first algorithm is a robust and reliable trajectory algorithm, *LFOPC*, which is however expensive in terms of the number of required function evaluations. As the simulations performed here in evaluating the objective and constraint functions may be computationally intensive, an approximation method, *Dynamic-Q*, is also used to find the optimum design with greater efficiency. The effectiveness of this approximation approach is evaluated.

2. INTRODUCTION

Recently parallel platforms, also known as Stewart platforms, have been the subject of much active research and development because of their distinct advantages in many practical applications over serially linked manipulators [2]. In particular machine tool manufacturers are already designing and fabricating platform devices to replace conventional milling machines [3].

Currently a research group at the University of Pretoria are investigating the possibility of extending the capability of existing 3-axis machining centers through the retro-fitting of parallel platforms. Introductory to such a development the workspaces of such platforms have already been investigated [4,5,6]. As a first step in the practical implementation of Stewart platforms in machining, the fitting of a planar type parallel platform is being investigated. Such a planar manipulator will be easier and cheaper to manufacture than a spatial platform, and its 4-axis CNC capability may, to some extent, fulfill in the machining requirements of the South African injection moulding industry.

A major and essential requirement for the practical use of a planar platform in machining is that, for a prescribed tool path in the workpiece, the design (or set-up) of the platform should be such that the completion of the tool path within the workspace is guaranteed. This problem is addressed in this paper. An optimization methodology is proposed and demonstrated which not only yields a design which accommodates the prescribed tool path, but also places the tool path in a manner which minimizes the actuator forces required for the execution of the task.

3. FORMULATION OF DESIGN PROBLEM

3.1. PLANAR PARALLEL PLATFORM MACHINING CENTER

The problem is to design a planar parallel platform for a machining application. Figure 1 gives an example of how a planar parallel platform may be utilized as a machine tool. For the set-up shown, the machining process is referred to as form-milling.

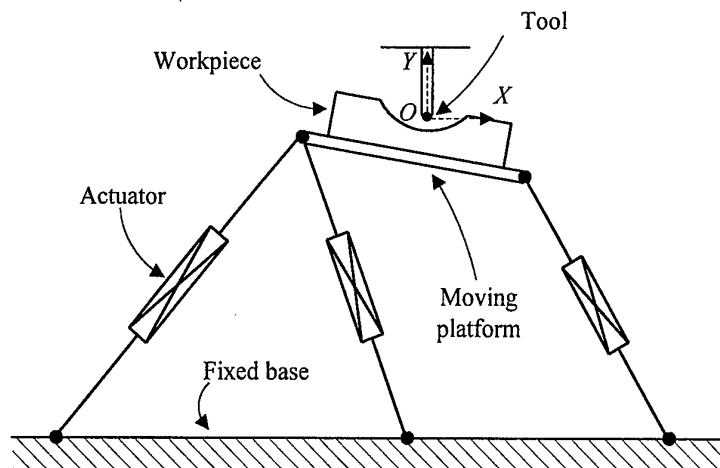


Figure 1: A planar parallel platform utilized as a machine tool.

A planar parallel platform consists of a fixed base, a moving platform and three actuators. The moving platform is connected to the fixed base via three actuators in parallel. The workpiece is fixed on top of the moving platform. The tool is stationary, and usually has a fast rotating tool piece which cuts the material from the workpiece. The position and orientation of the moving platform, and therefore the workpiece, can be controlled by controlling the lengths of each of the three actuators. In this manner a given profile can be cut in the workpiece.

The global coordinate system OXY is fixed to the tool tip. Forces induced by the milling process are modeled as external forces to the platform. For the application considered here, the dimensions (length $\times$ height $\times$ width) of the moving platform are chosen as $0.75m \times 0.1m \times 0.5m$. The workpiece has dimensions of $0.5m \times 0.201m \times 0.3m$. The platform mass is $135kg$ and is made out of steel. This platform weight takes fitting-holes in the platform into account. The workpiece is made out of aluminium and has a mass of $80kg$.

The actuators are modeled as driving constraints, with negligible mass and moment of inertia compared to the platform and workpiece. The driving constraints are distance drivers, i.e. the lengths of the actuators are specified as functions of time.

The tool force is modeled as a force of constant magnitude and direction relative to the global reference system. The effect of the tool force on the platform, i.e. the resultant of the tool force that works through the platform origin, and the moment of the tool force about the platform origin, are calculated and applied to the platform origin as external forces. The tool force components are dependent on the particular prescribed type and speed of the machining operation. In the applications considered here, the tool force has a component perpendicular to the tool path (and

axially along OY) of $440N$, and a tangential component of $1600N$ perpendicular to OY and opposite to the direction of relative motion between the tool and the workpiece surface.

3.2. DESIGN VARIABLES

The platform designer can decide on the geometrical design of the platform as well as the placement of the fixed base relative to the tool. These design parameters can be incorporated into the design methodology by means of design variables. Figure 2 shows five possible design variables. The origin O of the fixed global coordinate system OXY coincides with the fixed tool tip as shown. The global coordinates of the base of the left leg D is denoted by (x_4, x_3) . The local coordinate system $O_P\xi\eta$ is attached to the platform as shown with the origin O_P at the midpoint of AC .

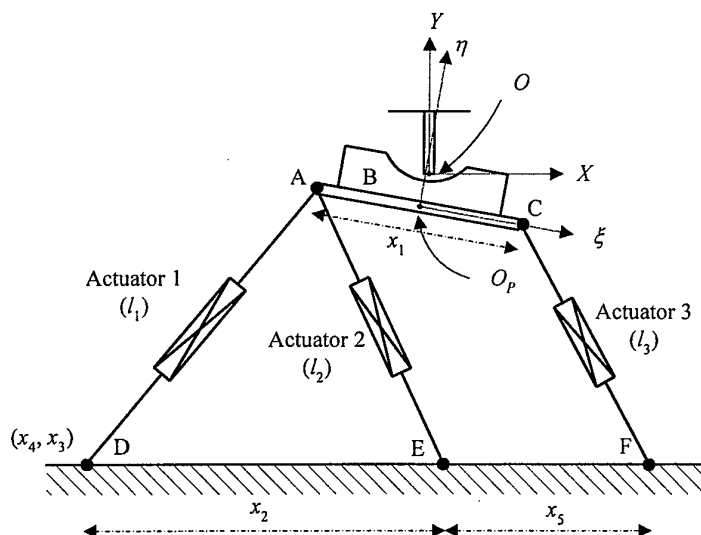


Figure 2: Possible design variables $x_i, i=1, \dots, 5$.

Design variables x_1, x_2 and x_5 define the geometrical design of the platform, while x_3 and x_4 give the placement of the base point of the left leg, relative to the tool, on the fixed horizontal base. Many other design variables may be chosen, but only these five are used here to illustrate the design methodology.

3.3. CONSTRAINTS

Constraints on the manipulator are specified by the designer, and may differ from one design problem to another. Here only inequality constraints are considered. The constraints are mathematically formulated for implementation in an optimization algorithm, and take on the general form:

$$c_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \quad (1)$$

where m is the number of inequality constraints and $\mathbf{x}$ denotes the vector of design variables, i.e. $\mathbf{x} = (x_1, x_2, \dots, x_5)^T$.

Limits are imposed on the maximum and minimum actuator lengths. It is important to realize that for a prescribed path the maximum and minimum values that are attained by the actuators are dependent on the design vector $\mathbf{x}$. For a given design $\mathbf{x}$, the respective maximum and minimum values to be attained by the different actuators for a prescribed path over the time interval $[0, T]$ are given by $l_i^{\max} = \max_t[l_i(t, \mathbf{x})]$ and $l_i^{\min} = \min_t[l_i(t, \mathbf{x})]$ for $i=1,2,3$ and $t \in [0, T]$. The allowable maximum and minimum actuator lengths are given by $\bar{l}_i$ and $\underline{l}_i$ respectively. The first six constraints, written in standard form (see expression (1)), are therefore:

$$\begin{aligned} c_i(\mathbf{x}) &= \max_t[l_i(t, \mathbf{x})] - \bar{l}_i \leq 0, \quad i=1,2,3 \quad \text{with } t \in [0, T] \\ c_{i+3}(\mathbf{x}) &= \underline{l}_i - \min_t[l_i(t, \mathbf{x})] \leq 0, \quad i=1,2,3 \quad \text{with } t \in [0, T] \end{aligned} \quad (2)$$

Physical bounds on the design variables define the following additional constraints:

$$\begin{aligned} c_{6+i}(\mathbf{x}) &= x_i - \bar{x}_i \leq 0 \quad i=1,2,\dots,n \\ c_{6+n+i}(\mathbf{x}) &= \underline{x}_i - x_i \leq 0 \quad i=1,2,\dots,n \end{aligned} \quad (3)$$

with $\bar{x}_i$ and $\underline{x}_i$ respectively denoting the prescribed upper and lower limits on variable x_i .

3.4. COST FUNCTION

For a given prescribed task, many feasible designs may be possible for which the constraints on the actuator lengths and on the physical dimensions are not violated. Out of the set of feasible designs, it would be wise to choose the best one in some sense. The choice of the optimum design depends on the specific design criterion considered. In the design of the milling machine considered here, a good choice for a design criterion may be based on the actuator forces. In particular one may choose a feasible design for which the actuator forces required to execute the prescribed task are minimized.

The design criterion is given in mathematical form by a cost function to be minimized. Here the manipulator is to be designed such that the forces in the respective actuators, $f_k(t)$, $k=1, 2, 3$, are minimized over the time interval $[0, T]$. This is done by minimization of the maximum absolute value of the actuator forces over all the actuators and over the total time interval $[0, T]$, i.e. the cost function $F(\mathbf{x})$ is defined as:

$$F(\mathbf{x}) = \max_{k=1,2,3} |f_k(t)|, \quad t \in [0, T] \quad (4)$$

The overall constrained design optimization problem may now be formally stated as:

$$\text{minimize}_{\mathbf{x}} F(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, x_3, x_4, x_5)^T \quad (5)$$

such that $c_i(\mathbf{x}) \leq 0$, $i=1,2,\dots,m=6+2n$.

In practice the value of $F(\mathbf{x})$ is determined by inspecting the values of the actuator forces, $f_k(t_i)$, at discretised time instants $t_i=i\Delta t$, $i=0, 1, \dots, N=T/\Delta t$, where Δt is the time sub-interval at which the actuator forces are sampled, and $N+1$ is the number of points for which it is sampled. The optimization process is usually initiated from a given starting design $\mathbf{x}^0$. The solution to the problem gives the optimum design which is denoted by $\mathbf{x}^*$.

3.5. PRESCRIBED TASKS

The prescribed task should uniquely describe the manner in which the moving platform is to be controlled. Therefore, a prescribed task consists of prescribing the profile that the tool should trace out on the workpiece (the tool path), the relative angle between the tool axis, YO , and the tool path at each moment, and finally the velocity profile with which uncut material is fed to the tool.

This tool path in the workpiece should be specified in the local coordinate system, $O_p\xi\eta$, because coordinates in this system are independent of the design, x . The two tool paths, A and B, that are considered here are shown in Figure 3(a) and (b) respectively. Note from Figure 3 that here the tool paths are chosen to start and end at the same local level, $\eta = 0.2m$. Also note that for these cases the tool paths are centered about $\xi=0$ (the η -axis). For both tasks, it is required of the tool axis OY to remain perpendicular to the tool path throughout the execution of the milling task.

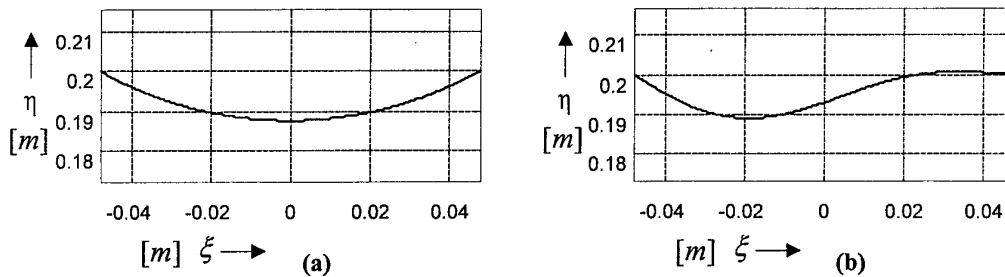


Figure 3: Respective tool paths (a) A and (b) B.

In machining applications the tool velocity v is tangential to the tool path and its magnitude is a function of time, and has a velocity profile that is shown in Figure 4. This profile is calculated for maximum allowable tool velocity, $v_{\max}=0.01 \text{ m.s}^{-1}$, and the maximum magnitude of constant acceleration of the workpiece tangential to the tool path, $a_{\max}=0.005 \text{ m.s}^{-2}$.

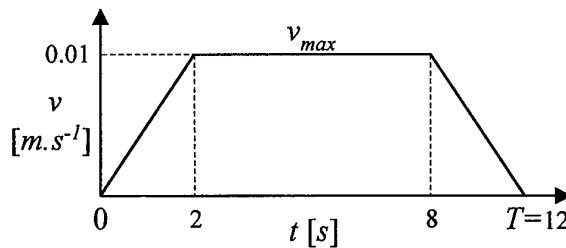


Figure 4: Velocity profile of tool relative to workpiece.

The time taken to complete each of the two prescribed task considered here is $T=12s$.

4. PRACTICAL IMPLEMENTATION OF OPTIMIZATION METHODOLOGY

4.1. OPTIMIZATION PROCEDURE

The procedure of finding the optimum platform design x^* , given an initial design x^0 , is discussed in this section. Figure 5 gives a diagrammatic representation of the optimization process. The constraint and cost functions for the initial design are evaluated. The design variables are then changed, and the effect of these changes on the constraint and cost functions is monitored. The

process of changing design variables and monitoring the constraint and cost functions is continued until a feasible design is obtained and no further improvement in the cost function is possible. This design is then taken as the optimum design, x^* .

The optimization was done using the well-established *LFOPC* constrained optimization code of Snyman [7,8,9]. This code uses a very robust gradient descent optimization algorithm which handles discontinuities that may occur in the gradients, as well as noise in the constraint and objective functions, with ease. Considering the nature of the object function and the fact that the gradients are computed by forward finite differences, the occurrence of both discontinuities and noise are distinct possibilities here. The manipulator dynamics is simulated using the *Dynamic Analysis Design System* (DADS v.9.0) [10]. The time interval at which the forces are sampled is $\Delta t=0.2s$.

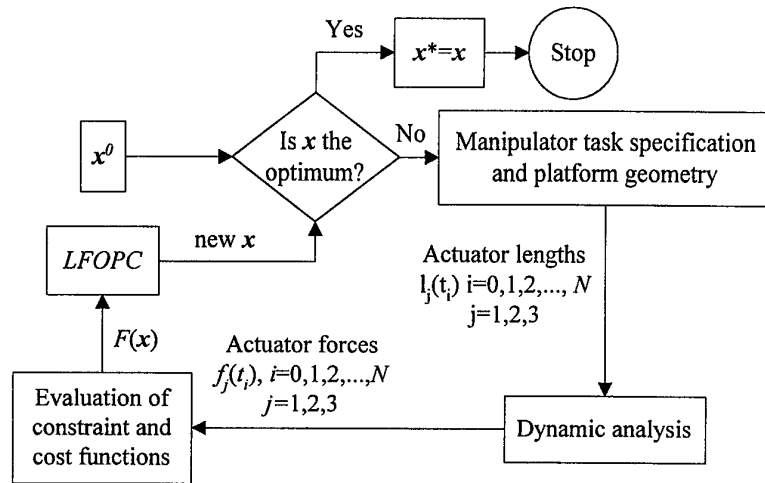


Figure 5: Diagrammatic representation of optimization procedure.

4.2. OPTIMUM DESIGNS

4.2.1. TOOL PATH A

The results obtained from the optimization procedure with tool path A (Figure 3(a)) as the prescribed path are given in this section. For this example, the upper and lower limits on the actuator lengths (2) were set to $\bar{l}_i = 1.2m$ and $\underline{l}_i = 0.6m$, $i=1, 2, 3$. The constraints on the variables (3) in meters were set to: $0.1 \leq x_1 \leq 0.75$, $0.1 \leq x_2$, $x_3 \leq 0$, $0.1 \leq x_5$, $x_2 + x_5 \leq 1.5$. The initial design is chosen as $x^0 = (0.4, 0.6, -0.6, -0.8, 0.7)^T$, with again all dimensions in meters. For the given initial design the value of the cost function (3) is $F(x^0) = 10.107 \text{ kN}$. The optimum design was found to be $x^* = (0.750, 0.833, -0.961, -1.148, 0.101)^T$, with a cost function value of $F(x^*) = 1.525 \text{ kN}$. The actuator forces for the initial and optimum designs are shown in Figure 6 and Figure 7 respectively.

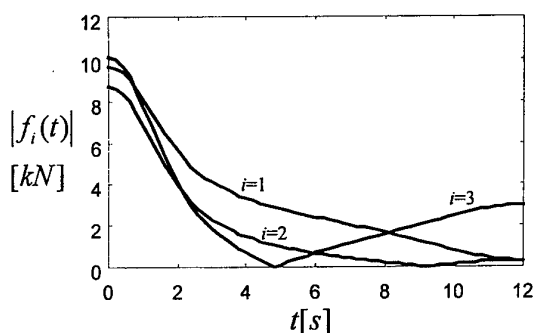


Figure 6: The actuator forces vs. time plot for the initial configuration.

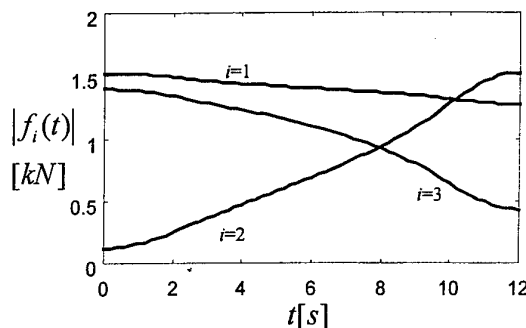


Figure 7: The actuator forces vs. time plot for the optimal configuration.

The actuator length variation for the initial configuration clearly does not satisfy the constraints on the actuator lengths, as shown in Figure 8. After optimization the obtained optimum configuration does not only satisfy the actuator length constraints as shown in Figure 9, but also corresponds to a design with an 85% reduction in the cost function!

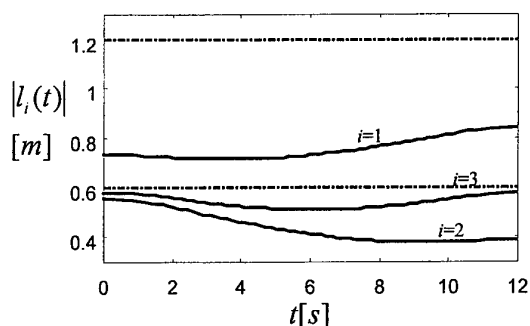


Figure 8: The actuator lengths $l_i(t)$ for the initial configuration.

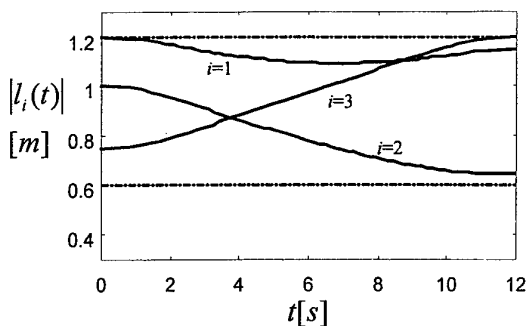


Figure 9: The actuator lengths $l_i(t)$ for the optimum configuration.

The convergence histories of three of the design variables x_i , $i=1, 2, 3$ and the cost function $F(x)$ are depicted in Figure 10 and Figure 11 respectively, for the first 120 iterations at which point the procedure has effectively converged. The listed optimum, obtained with an extremely accurate specified convergence tolerance of $\|\Delta x\| \leq 10^{-5}$, was reached after 727 iterations, with a total computer time of around one hour on a Pentium II-266.

By using different initial designs, $x_I^0 = (0.7, 0.4, -1.0, -1.1, 0.2)^T$ and $x_{II}^0 = (0.5, 0.8, -0.6, -0.6, 0.5)^T$, the same optimum was reached as before, which strongly suggest that the obtained local optimum is indeed the global optimum.

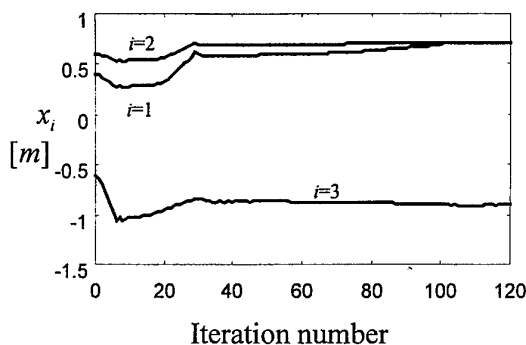


Figure 10: The convergence histories of three design variables x_i $i=1,2,3$ are shown.

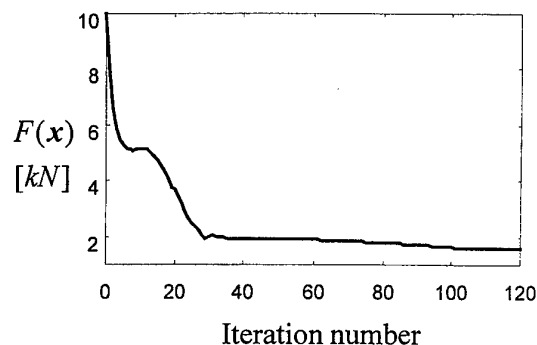


Figure 11: The convergence history of the cost function $F_{A.F.D}(x)$ is shown.

4.2.2. TOOL PATH B

The results obtained from the optimization procedure with tool path B (Figure 3(b)) as the prescribed path are given here. The constraints (2) and (3) are the same as for path A. With an initial design $\mathbf{x}^0 = (0.4, 0.6, -0.6, -0.8, 0.7)^T$, the cost function value is $F(\mathbf{x}^0) = 4.332 \text{ kN}$. The optimum feasible design was found to be $\mathbf{x}^* = (0.750, 0.737, -1.021, -1.135, 0.337)^T$, with a cost function value of $F(\mathbf{x}^*) = 1.812 \text{ kN}$ representing a 58% reduction. Different initial designs yielded the same optimum, again suggesting that the global optimum has been obtained.

4.3. APPROXIMATION METHOD

The time it takes for *LFOPC* trajectory method to solve the optimization problem with five or more design variables may be excessive. This situation is exacerbated due to the fact that the optimization algorithm is a gradient based algorithm, requiring the use of gradients computed by forward finite differences. This implies $n+1$ cost function evaluations to approximate the gradient vector at each design point, where n is the number of design variables. The evaluation of one cost function value, for a specific design vector, therefore requires a full dynamical analysis of the platform as it executes the prescribed task. This analysis is computationally expensive resulting in a correspondingly time consuming evaluation of the cost function.

In view of the above an investigation into the possibility of reducing the computational time by using a different optimization procedure, which requires less cost function evaluations is justified. One solution to the problem is to use an approximation optimization algorithm, such as *Dynamic-Q* proposed by Snyman et.al. [11]. This algorithm has been successfully applied to various engineering problems [12,13,14], which have expensive cost or constraint functions.

Dynamic-Q constructs and solves successive sub-problems with constraint and cost functions that are approximations to the actual constraint and cost functions. For each approximate sub-problem spherical quadratic functions are constructed using function and gradient information of the actual constraint and cost functions, evaluated at the solution point of the previous sub-problem. In this manner solutions to a sequence sub-problems are obtained which, in practice and under relatively general conditions, converges to the solution of the original problem.

The convergence histories of the *LFOPC* and *Dynamic-Q* algorithms are shown in Figure 12 for the design problem with prescribed tool path A and with the same constraints as specified earlier in this paper. Figure 12 clearly shows that *Dynamic-Q* is successful in effectively obtaining the optimum design after only 31 iterations, which required about 5 minutes computational time on a Pentium II-266. The periodic spikes that occur in the convergence curve for *Dynamic-Q*, is typical of that obtained when the optimum of the objective function lies along an extremely narrow and steep valley.

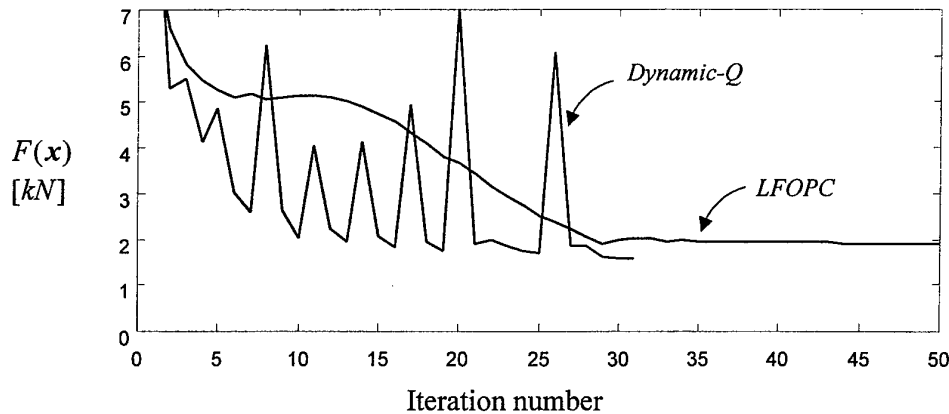


Figure 12: Convergence histories for *LFOPC* and *Dynamic-Q*, with $x^0 = (0.4, 0.6, -0.6, -0.8, 0.7)^T$.
 $F_{LFOPC}(x^*) = 1.525$ after 727 iterations and $F_{Dyn-Q}(x^*) = 1.564$ after 31 iterations.

5. CONCLUSION

This study shows that the optimization methodology presented here, can successfully be applied to the optimal design of a planar parallel platform required to perform a prescribed milling task. The mathematical optimization approach yields a good platform design based on the proposed design criterion, whilst at the same time satisfying stipulated design constraints. In particular this method solves the following problems that are inherent to the design of parallel platforms:

- i. it finds designs for which the given tool path lies inside the workspace,
- ii. it solves the optimum placement problem of the platform relative to the tool, and
- iii. it gives a platform design, which has the property that the actuator forces required to execute the prescribed task are minimal.

The design methodology described in this study has the important potential application that it can form an integral part of a design software package for planar parallel platforms to be used in machining operations. This methodology may also be extended to be applicable to the optimal design of spatial parallel platforms.

6. REFERENCES

- [1] Smit W.J., "The optimal design of a planar Stewart platform for prescribed machining tasks", *Master's thesis in Mechanical Engineering*, University of Pretoria, 2000.
- [2] Merlet J-P, "Parallel manipulators: state of the art and perspectives", http://www-sop.inria.fr/saga/personnel/merlet/Etat/etat_de_lart.html, 1996.
- [3] Abassi W.A., Ridgeway S.C., Adsit P.D., Crane C.D. and Duffy J., "Development of a spatial 6-6 parallel platform for contour milling", *Proceedings of the ASME Manufacturing*

Engineering Division 1997 International M.E. Congress and Exposition (IMECE), Dallas, 1997, p.373-380.

- [4] Snyman J.A., Du Plessis L.J. and Duffy J., "An optimization approach to the determination of the boundaries of manipulator workspaces". To appear in the *ASME Journal of Mechanical Design*, 2000.
- [5] Du Plessis L.J. and Snyman J.A., "A numerical method for the determination of dextrous workspaces of Stewart platforms", Submitted for publication in the *International Journal for Numerical Methods in Engineering*, 1999.
- [6] Hay A.M. and Snyman J.A., "The determination of non-convex workspaces of generally constrained planar Stewart platforms". To appear in *Computers and Mathematics with Applications*, 2000.
- [7] Snyman J.A., "A new dynamic method for unconstrained minimization", *Applied Mathematical Modelling*, Vol. 6, p.449-462, 1982.
- [8] Snyman J.A., "An improved version of the original leap-frog dynamic method for unconstrained minimization LFOP1(b)", *Applied Mathematical Modelling*, Vol. 7, p.216-218, 1983.
- [9] Snyman J.A., "The LFOPC leap-frog algorithm for constrained optimization", To appear in the *International Journal for Computers and Mathematics with Applications*, 2000.
- [10] Haug E.J., "Computer aided kinematics and dynamics of mechanical systems", *Allen and Bacon*, Boston, 1989.
- [11] Snyman J.A., Stander N., "A new successive approximation method for optimum structural optimization problems", *AIAA Journal*, 1994, vol.32, p.1310-1315.
- [12] Craig K.J., de Kock D.J., Snyman J.A., "Using CFD and mathematical optimization to minimize stack pollution", *International Journal for Numerical Methods in Engineering*, 1999, vol.44, p.551-566.
- [13] Craig K.J., Venter P., de Kock D.J., Snyman J.A., "Optimization of structured grid spacing parameters for separated flow simulation using mathematical optimization", *Journal of Wind Engineering and Industrial Aerodynamics*, 1999, vol.80, p.221-231.
- [14] de Kock D.J., Craig K.J., Snyman J.A., "Using mathematical optimization in CFD analysis of a continuous quenching process". *International Journal for Numerical Methods in Engineering*, Vol.47, pp.985-999, 2000.

THE TREATMENT OF LOCK-UP IN THE OPTIMAL DESIGN OF SERIALY LINKED MANIPULATORS PERFORMING PRESCRIBED TASKS*

J. A. Snyman

Multidisciplinary Design Optimization Group
Department of Mechanical Engineering
University of Pretoria, Pretoria 0002, South Africa
e-mail: jan.snyman@eng.up.ac.za

ABSTRACT

In the application of optimization algorithms to the optimal design of mechanical manipulators performing prescribed tasks, unique difficulties are encountered due to the fact that the optimization algorithms may drive the design to a configuration that experiences lock-up, and thus resulting non-assembly along the task path. If this happens no objective function can be evaluated for the current design, and the optimization procedure is forced to terminate prematurely. In previous work on planar serially linked manipulators, a heuristic procedure was proposed to deal with this problem. Here the author carefully examines the behavior of such systems near lock-up, in order to place the proposed heuristic on a more solid foundation.

INTRODUCTION AND PROBLEM FORMULATION

In spite of the mathematical sophistication of existing *gradient-based* algorithms, certain inhibiting difficulties remain when these algorithms are applied to *real-world problems*. This is particularly true in the field of engineering, where *unique difficulties* occur, that have prevented the general application of mathematical optimization techniques to design problems. Typical optimization difficulties that arise are:

- (i) that the functions are *very expensive to evaluate* requiring, for example, time-consuming finite element analyses, CFD or multi-body dynamical simulations,
- (ii) the *existence of noise*, numerical or experimental, in the functions,
- (iii) the occurrence of *discontinuities* in the functions,
- (iv) the existence of *multiple local minima*, requiring global optimization techniques,
- (v) the occurrence of an *extremely large number* of design *variables* disqualifying, for example, the SQP method if Hessian information is required, and
- (vi) the *existence of regions* in the design space where the *functions are not defined*, and therefore cannot be evaluated.

Problems (i) to (v) have often been addressed, and for the different complications that may arise, various methods and techniques have been developed to overcome the respective problems. Difficulty (vi),

* This paper is similar to that read by the author at the 3rd WCSMO Congress in Buffalo, New York, May 17-21, 1999.

however, has by comparison been neglected. The probable reason for this is that there does not appear to be a general method for solving this particular problem. Often the nature of the system being optimized dictates the method to be adopted in overcoming this particular difficulty. Here we present a technique that was successfully applied to this problem as it presented itself in the optimal dimensional synthesis of serially linked manipulators.

In the optimal design of serially linked manipulators, it may be required to minimize an objective function associated with the execution of a prescribed task path to be followed by the end-effector. In particular it may, for example, be required that for a typical planar serially linked manipulator, depicted in Figure 1, the average torque for the execution of the task path over the time interval $[0, T]$, be minimized with respect to the link lengths and base placement.

A major problem that may occur, corresponding to difficulty (vi) above, is that during the optimization procedure the optimization algorithm in adjusting the link lengths and base coordinates $[\ell_1, \ell_2, \ell_3, x_b, y_b]$, may drive the design to one where the workspace no longer fully encloses the prescribed task path. Thus, assuming that the initial point of the task path is in the workspace, the end-effector in following the prescribed task path will reach a point on the boundary of the workspace, beyond which assembly of the manipulator is no longer possible. Consequently the objective function associated with the completion of the path cannot be evaluated, i.e., difficulty (vi) is encountered, and the optimization procedure terminates prematurely and unsuccessfully.

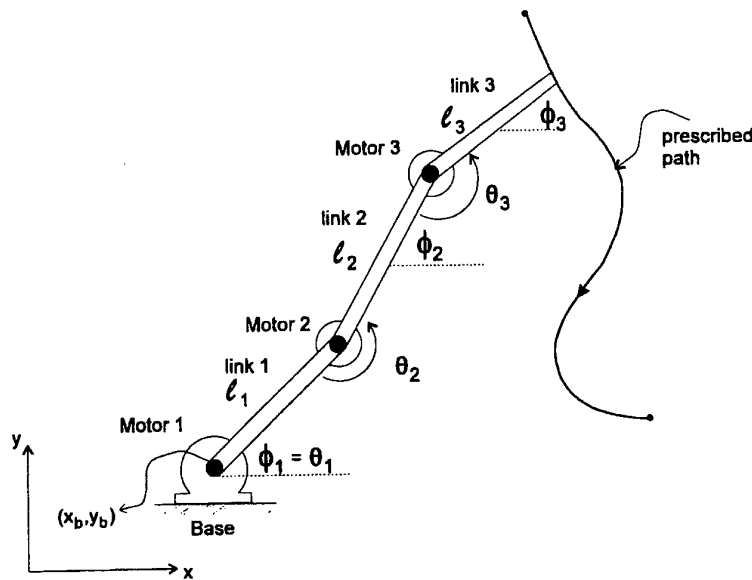


Figure 1. Schematic representation of a planar manipulator following a prescribed path.

Recently Snyman and Berner [1,2,3] proposed and successfully applied an ad hoc heuristic procedure for avoiding non-assembly during optimization. Here this heuristic procedure, involving artificial

objective function values, is carefully re-examined in order to place it on a more solid theoretical foundation.

KINEMATICALLY DRIVEN SYSTEM AND LOCK-UP

In general a kinematically driven system is described by a combination of n kinematic and driving constraint equations of the form [4]:

$$\Phi(\mathbf{q}, t) = 0 \quad (1)$$

where $\mathbf{q}$ is a n -vector of generalized coordinates and t denotes the instant in time. It is assumed that (1) explicitly or implicitly specifies the position vector $\mathbf{r}(t)$ of the end-effector at any instant t in the interval $[0, T]$. Suppose now that for a specific design the path is initially ($t=0$) inside the reachable workspace of the manipulator, but crosses the boundary at $t=t^* > 0$. This implies that assembly is lost beyond the boundary and that, for $t > t^*$, no solution to (1) exists. According to the definition of Haug [4] the configuration $\mathbf{q}^*$ at time t^* corresponds to *lock-up*, and further, since the solution cannot be continued beyond t^* , it follows by the implicit function theorem that the Jacobian of (1) at t^* , $\Phi_{\mathbf{q}}(\mathbf{q}^*, t^*)$, is singular. This has an important influence on the behavior of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ as t approaches t^* . The velocity equation derived from (1) is

$$\Phi_{\mathbf{q}}(\mathbf{q}, t) \dot{\mathbf{q}} = -\Phi_t \quad (2)$$

Since $\Phi_{\mathbf{q}}(\mathbf{q}, t)$ is singular at time t^* , it follows that in the situation described above, where the manipulator approaches a lock-up configuration, that the solution of (2) yields values for $\dot{\mathbf{q}}$ that tend to infinity as t approaches t^* . Clearly the same applies to the solution $\ddot{\mathbf{q}}$ of the corresponding acceleration equation obtained from (2). Thus the behavior of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ serves as a warning that a lockup configuration and non-assembly is being approached. In the following section a simple example is presented that illustrates such a situation and shows how the behavior is monitored in practice.

SIMPLE EXAMPLE OF LOCK-UP AND NON-ASSEMBLY

Consider a two-link revolute manipulator [5] with the configuration specified by relative coordinates as shown in Figure 2(a), and with the task path of end-effector specified by $\mathbf{r}(t) = [t, \sqrt{2}] \forall t \in [0, 2]$. Setting $q_1 = \theta_1$, $q_2 = \theta_2$, equation (1) assumes the form

$$\Phi(\theta, t) = \begin{bmatrix} \cos\theta_1 + \cos(\theta_1 + \theta_2) - t \\ \sin\theta_1 + \sin(\theta_1 + \theta_2) - \sqrt{2} \end{bmatrix} = 0 \quad (3)$$

For any instant t equation (3) may be solved for θ_1 and θ_2 [5]. By further differentiation of (3) with respect to time, the velocity and acceleration equations may be derived, from which one may solve for

$\dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1$ and $\ddot{\theta}_2$. These values, computed at intervals of $\Delta t=0.01s$, are plotted in Figure 2(b). Clearly as t approaches $t^*=\sqrt{2}$ the velocities and accelerations increase dramatically, as is expected from theoretical considerations, and beyond t^* no solutions exist.

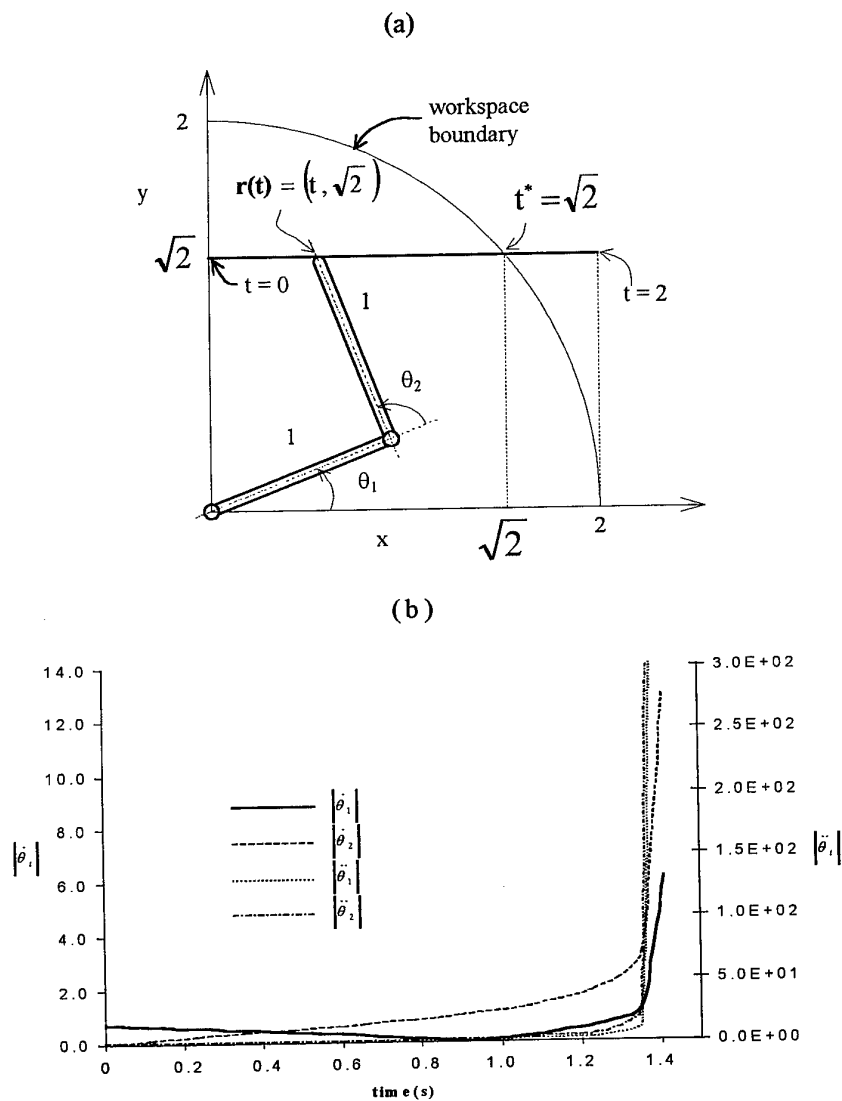


Figure2. (a) Task path of two-link manipulator, and (b) behavior of $\dot{\theta}_i$ and $\ddot{\theta}_i$ along task path.

Important here is the conclusion that, if the further inverse dynamics is done, then, since the accelerations increase to infinity, the corresponding *required couples*, τ_1 and τ_2 , on the respective links

and required to maintain the prescribed motion, will also increase to infinity as t approaches t^* . The theory in the previous section and evidence presented here, indicate that this behavior is in general true for a large number of practical cases where "lock-up trajectories" may be encountered.

EVALUATION OF OBJECTIVE FUNCTION FOR LOCK-UP TRAJECTORIES

Consider the manipulator depicted in Figure 1, and where the objective function for a given design, $\mathbf{x} = [\ell_1, \ell_2, \ell_3, x_b, y_b]$, is the average torque requirement for execution of the task path, given by

$$T_{av}(\mathbf{x}) = \frac{1}{T} \int_0^T (|\tau_1(t)| + |\tau_2(t)| + |\tau_3(t)|) dt \quad (4)$$

where, $\tau_i(t)$ denotes the torque applied to link i . Assume for the moment that it can be ensured that at all design points during the optimization procedure, the initial point and endpoint of the prescribed path lie within the workspace [1]. If during the optimization procedure a design is obtained that results in lock-up and resultant non-assembly further along the path (between the initial and end point), then obviously the evaluation of the objective function (4) is not possible.

The behavior of the accelerations near lock-up, as discussed in the previous sections, now points to a practical strategy for obtaining a meaningful value for the objective function even if non-assembly is encountered. In practice the analysis is done at discrete time steps Δt , starting at the initial point where assembly is possible. Lock-up and non-assembly will therefore be detected at some discrete point along the path if a solution to (1) cannot be found. Non-assembly will be evident from the failure of the formula for the closed form solution, if it exists, or failure of the Newton-Raphson method to converge, if a numerical solution is sought. If the system is now forced through the lock-up position and the analysis continued for the prescribed path, the analysis will fail at intermediate steps where assembly is no longer possible. However as assembly at the end position is guaranteed by our assumption above, a point will be reached where assembly is again possible, and the analysis can successfully be continued.

A heuristic procedure to compute a meaningful objective function value for such a lock-up trajectory, and previously proposed and successfully applied to the optimal design of planar manipulators [1,2,3], is now justified in terms of the theory and evidence on the behavior of the accelerations presented above. At the integration points t where assembly fails, set the respective torques $\tau_i(t) := \tau_i(t_s)$, where t_s (close to t^* , $t_s < t^*$), corresponds to the last step at which assembly was successfully carried out. Use these artificial values in the numerical integration of (4) to give an associated artificial value for the objective function. This situation is depicted in Figure 3. Because t_s is close to t^* , one expects the values of $|\tau_i(t_s)|$ to be relatively large, so that the computed value for T_{av} will be very high. Also, the longer the non-assembly time interval, the larger the expected value for the artificial objective function. Any minimization procedure employing the above artificial integration procedure for evaluating the objective function, should drive the design away from lock-up trajectories with artificially high objective function values. This is indeed borne out by the work of Snyman and Berner [1,2,3].

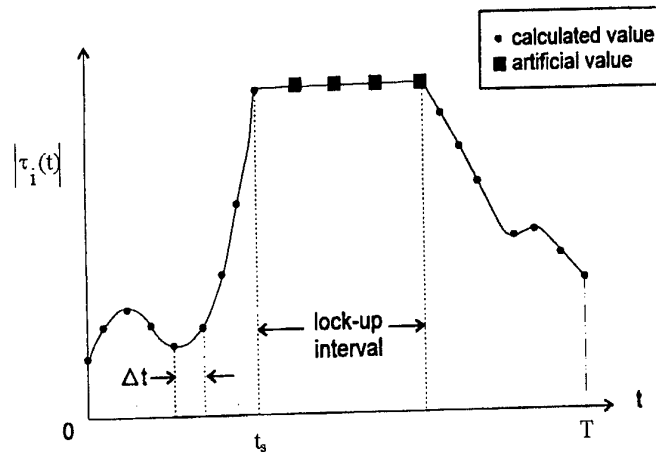


Figure 3. Numerically computed absolute torque value, $|\tau_i(t)|$, along lock-up trajectory.

CONCLUSION

The behavior of manipulators following prescribed paths and in the neighborhood of lock-up has carefully been investigated. This leads to a well-founded justification of a previously proposed heuristic for avoiding non-assembly during the optimal dimensional synthesis of manipulators.

REFERENCES

- [1] Snyman J. A. & Berner D. F., A mathematical optimization methodology for the optimal design of a planar robotic manipulator. *International Journal for Numerical Methods in Engineering*, Vol 44, 535-550 (1999).
- [2] Berner D. F. & Snyman J. A., The influence of joint angle constraints on the optimum design of a manipulator following a complicated path. *Computers Math Applic*, Vol 37, 111-124 (1999).
- [3] Snyman J. A. & Berner D. F., The design of a planar robotic manipulator for optimum performance of prescribed tasks. *Structural Optimization*, Vol 18, 95-106 (1999).
- [4] Haug E. J., Computer-aided kinematics and dynamics of mechanical systems, Vol 1. Allyn & Bacon, Boston, 1989.
- [5] Stadler W., Analytical robotics and mechatronics. McGraw-Hill, Inc. 1995.

SHAPE OPTIMIZATION FOR CRASHWORTHINESS DESIGN USING RESPONSE SURFACES

Akkerman, A.

Ford Motor Company, Dearborn, MI, U.S.A.

Thyagarajan, R.

Visteon Corporation, Dearborn, MI, U.S.A.

Stander, N. Burger, M.

Livermore Software Technology Corporation, Livermore, CA, U.S.A.

Kuhn, R. Rajic, H.

Kuck & Associates, Inc., Champaign, IL, U.S.A.

ABSTRACT

A successive response surface method is applied to the design optimization of problems with non-linear response. The response surfaces are built using linear approximations within a dynamic sub-region. The method employs two dynamic parameters to adjust the size of the sub-region. These are determined by the proximity of successive optimal points and the degree of oscillation. The algorithm has been implemented in the LS-OPT optimization code.

A complex example in occupant safety design is used to demonstrate the methodology. In the example, the crashworthiness of an instrument panel was enhanced using LS-OPT in conjunction with the explicit dynamics code, LS-DYNA. The successive response surface method employed in LS-OPT results in a fully automated method. The paper studies the need for shape optimization in the design of instrument panels for crashworthiness and concludes that, although a large number of shape variables can benefit the design, the largest gain in efficiency is made using the thickness variables.

INTRODUCTION

The automotive instrument panel (IP) has evolved over time to become one of the most complex subsystems in today's automobile (Jira, 1996), both from an appearance and functionality viewpoint. Not only does it lend a distinctive character to the interior of an automobile from an aesthetic point of view, it must also house beneath the styled surface several components required for functional reasons. Some examples of these are cross-vehicle structure, steering column supports, climate control system, electronic modules and wiring, airbags, and a knee bolster system. Of particular significance to this paper is the knee bolster portion of an IP, which is designed to perform several functions (Kulkarni, 1998). Most notably it provides the first contact surface for the knees in a frontal impact situation. Also, it participates in cushioning and directing the knees and in energy management of the lower torso of the occupant.

Design variables for problems of this type can create a very large design space that the engineer must explore. Typical parameters are:

- Gauge and modulus of the material in the energy absorption (EA) brackets,
- Gauge and modulus of the knee bolster material,
- Steering column isolator (a.k.a. yoke) cross-section radius, and

- Lightening holes and flange depth in the EA brackets.

Recently, optimization methods have been increasingly adopted as an aid to explore the design space effectively and with minimal user intervention. In this endeavor, the Response Surface Method (Myers, 1995) has become a popular method for addressing the “step-size dilemma” (Haftka, 1992) in sensitivity analysis and optimization, especially as it may occur in nonlinear dynamical response. The purpose of the method is primarily to avoid the necessity for analytical or numerical gradient quantities as these are either too complex to formulate, discontinuous or sensitive to roundoff error. Automated methods have also been formulated to address problems in rigid body dynamics (Etman, 1997), sheet metal forming (Kok, 1999) and material parameter identification (Stander, 2000). The method presented here incorporates sophisticated features into these approaches. These are:

- the use of *D*-optimal experimental design within the feasible design space,
- the use of move characteristics to determine the contraction rate of the region of interest and
- the identification of oscillation vs. ‘panning’ (translation of the region of interest in the design space) to determine the maximum shrinkage rate of the region of interest.

The first paper to study the present design problem using gauge and material parameters as variables was presented in (Akkerman, 1999). Since the design problem was small, second order response surfaces could be used to construct a trade-off diagram of maximum knee force versus intrusion. A follow-up paper (Akkerman, 2000a) extended the example to include eight additional shape variables assigned to various geometric parameters of the brackets and the yoke radius. Because of the additional design complexity, two major features were introduced:

- A parametric preprocessor, TrueGrid[®], to allow geometric modeling.
- A *successive linear approximation* procedure to reduce the number of simulations.

Although the latter method is linear versus the quadratic method used (Akkerman, 1999), it was shown that only three or four iterations were required for convergence. This could be achieved by reducing the size of the region of interest in the design space (Stander, 2000). Since these earlier studies pointed to the importance of the yoke in occupant safety design, a further paper (Akkerman, 2000b) investigated the effects of enlarging the design space with respect to the yoke radius and setting a deformation constraint for the yoke.

As the previous paper confirmed the role of the yoke in reducing the knee forces, the present paper focuses on optimizing the left and right bracket gauges, knee bolster gauge and yoke radius (four variables) as opposed to including all eleven parameters as variables. A comparison is therefore made with the results of (Akkerman, 2000b).

RESPONSE SURFACE METHODOLOGY

The method presented here is based on the design of experiments. The experimental designs are constructed within the bounds of a region of interest in the design space as determined by the upper and lower bounds of the design variables. For experimental design the *D*-optimality criterion is used (Myers, 1995).

Approximations. In response surface methodology surfaces are fitted to the responses of the design points determined by the experimental design. A common approximation method is the

[®] TrueGrid is a registered trademark of XYZ Scientific Applications, Inc.

fitting of polynomials although other types of surfaces can also be used. Quadratic polynomials are usually accurate for a mid-range region of the design space but because the expense is a function of n^2 (where n is the number of design variables) they are normally avoided for large design problems. This applies particularly to nonlinear problems involving large finite element models. A possible solution is to use linear approximations. These are generally less accurate but can be used in a successive response surface procedure (Etman, 1997; Kok, 1998; Kok, 1999, Stander 2000). The difficulty with using successive linear approximations is that *cycling* or oscillation may occur. This phenomenon can be countered by manipulation of the size of the region of interest, a measure analogous to applying move limits in successive linear programming. Heuristic measures are typically introduced (Etman, 1997)

Successive response surface method. The successive response surface method uses the region of interest, a subspace of the design space, as a trust region to determine an approximate optimum. A range is chosen for each variable to determine its initial size. A new region of interest centers on each successive optimum. In the following procedure, two parameters have been used to drive a successive linear response surface method:

1. A *maximum contraction* parameter is determined based on whether the current and previous optima are on the opposite or same side of the region of interest. The former case signals the onset of oscillation while the latter suggests that the optimum lies beyond the current region of interest. The parameter determines the maximum shrinkage rate and should therefore be large for the oscillatory case and small for the 'panning' case. One parameter is chosen for each design variable.
2. An *effective contraction* parameter interpolates between the maximum contraction parameter and a constant minimum contraction parameter using the distance of the current optimum to the center of the region of interest as input.

Software: LS-OPT. The afore-mentioned methods have been incorporated in the program LS-OPT (LSTC, 1999) a command language-based, standalone general optimization program with a comprehensive LS-DYNA interface. The core solver LFOPC (Snyman, 1999) is used to solve the approximate subproblem constructed within the region of interest using the linear response surfaces.

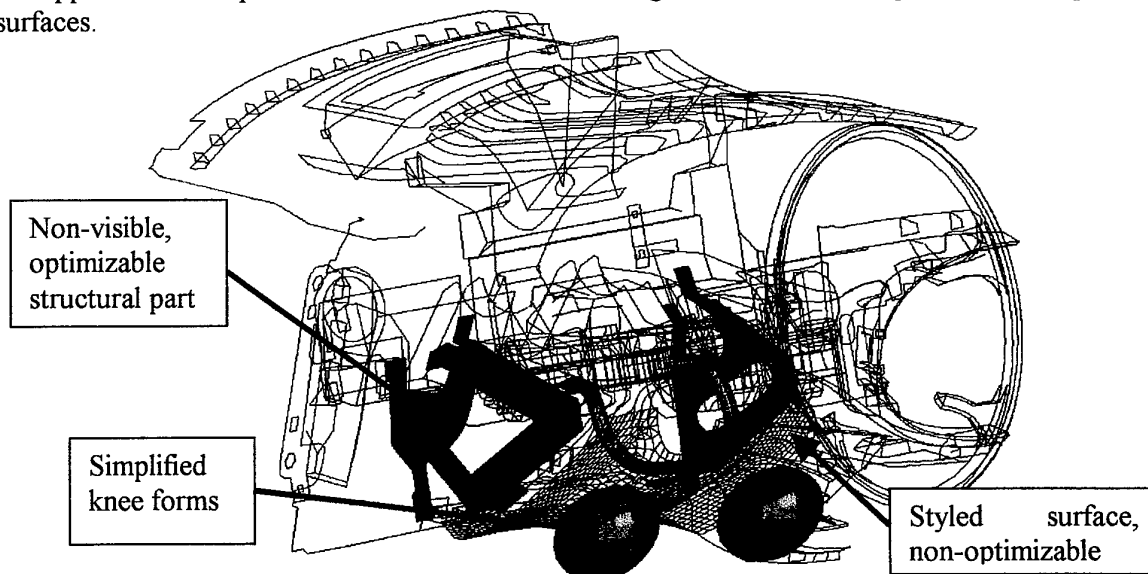


Figure 1: Typical instrument panel prepared for a "Bendix" component test

GENERAL PROBLEM STATEMENT

Figure 1 shows the finite element model of a typical automotive IP. For model simplification and reduced per-iteration computational times, only the driver's side of the IP is used in the analysis as shown, and consists of around 25,000 shell elements. Symmetry boundary conditions are assumed at the centerline, and to simulate a bench component "Bendix" test, body attachments are assumed fixed in all 6 directions. Also shown in Figure 1 are simplified knee forms which move in a direction as determined from prior physical tests. As shown in the figure, this system is composed of a knee bolster (steel, plastic or both) that also serves as a steering column cover with a styled surface, and two EA brackets (usually steel) attached to the cross vehicle IP structure. The brackets absorb a significant portion of the lower torso energy of the occupant by deforming appropriately. Sometimes, a steering column isolator (also known as a yoke) may be used as part of the knee bolster system to delay the wrap-around of the knees around the steering column. The last three components are non-visible and hence their shape can be optimized. The design variables are shown in Figure 2.

The simulation is carried out for a 40 ms duration by which time the knees have been brought to rest. It may be mentioned here that the Bendix component test is used mainly for knee bolster system development; for certification purposes, a different physical test representative of the full vehicle is performed. Since the simulation used herein is at a subsystem level, the results reported here may be used mainly for illustration purposes.

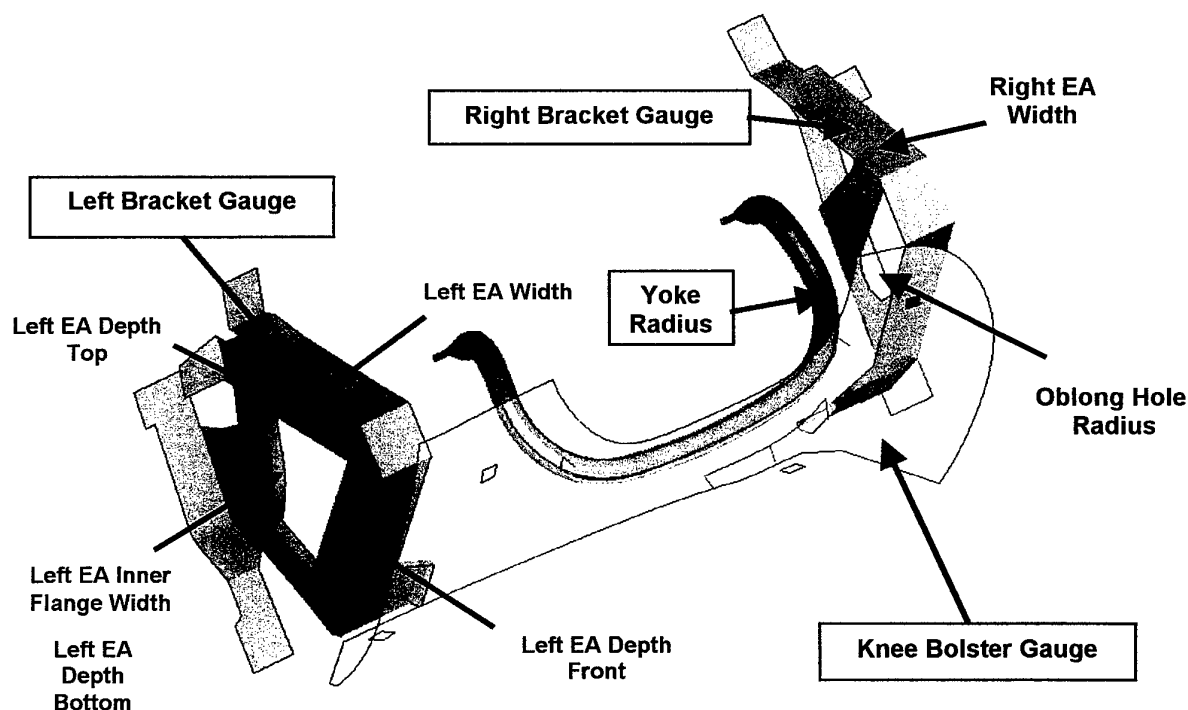


Figure 2: Design variables of the knee bolster system

DEFINING THE DESIGN SPACE

The gauge of the EA brackets and the knee bolster and the yoke diameter are firstly defined. The ranges of the variables are continuous and limited by manufacturing to specific values. Table 1 shows which part thicknesses were varied and the range over which they were varied.

Description	Lower Limit (mm)	Variable	Upper Limit (mm)	Baseline (mm)	Baseline Material
Left Bracket Gauge	0.7	G_{RB}	3	1.1	Steel
Right Bracket Gauge	0.7	G_{LB}	3	1.1	Steel
Knee Bolster Gauge	1	G_{KB}	6	3.5	SMC
Yoke Radius	2	R_Y	8	4	Steel

Table 1: Thickness variables

The EA brackets and yoke were modeled with steel in all the analyses. The knee bolster was modeled with SMC (Sheet Molding Compound).

To specify a shape optimization problem, a baseline design is parameterized. The complexity of doing this depends on the part being optimized. Figure 2 shows the parameterization of the three parts to be optimized in this design. For the yoke, one design variable, the cross-section radius of the yoke, R_Y , is varied.

The shape optimization of the right EA bracket will restrict itself to two areas:

- **Right EA Hole Radius**—The hole in the right EA bracket has been modeled as an oblong hole. The radius of the hole is one design variable, R_{RB} . The out-of-round dimension of the hole was arbitrarily fixed and is not considered a design variable.
- **Right EA Flange Width**—Around the outside of the bracket the metal has been folded at 90 degrees to make a mounting surface and to provide flange stiffness. The width of the flange is W_{RB} .

For the left EA bracket, the shape optimization will restrict itself to three areas:

- **Size and shape of the hole**—There are three design variables, D_{LBT} , D_{LBF} , D_{LBB} . These correspond to the flange depths in the bracket plane for 3 of the 4 primary edges of the left EA bracket, the top, the front, and the bottom, as seen from the driver's position. The depth of the back of bracket is currently fixed.
- **Inner Flange Width**—The hole in the left EA bracket has been folded inward to form a metal rim which acts to stiffen the bracket. The width of the flange is a design variable, W_{FLB} .
- **Width of the bracket**—As with the right AE bracket, the outside of the bracket has been folded at 90 degrees. The width of the flange is W_{LB} .

Fillets to round the four sides of the hole have not been used because they are not likely to have global effects. There is no distinct treatment of the shape and sizing variables, both having been incorporated in the TrueGrid input file. The shape variables are summarized in the Table 2 with Figure 3 depicting the extreme shapes on the boundaries of the design space.

Description	Lower Limit (mm)	Variable	Upper Limit (mm)	Baseline (mm)
Right EA Hole Radius	10	R_{RB}	25	15
Right EA Width	10	W_{RB}	50	32
Left EA Depth Top	10	D_{LBT}	50	28.3
Left EA Depth Front	10	D_{LBF}	50	27.5
Left EA Depth Bottom	10	D_{LBB}	50	22.3
Left EA Inner Flange Width	5	W_{LBF}	15	7
Left EA Width	10	W_{LB}	50	32

Table 2: Shape variables

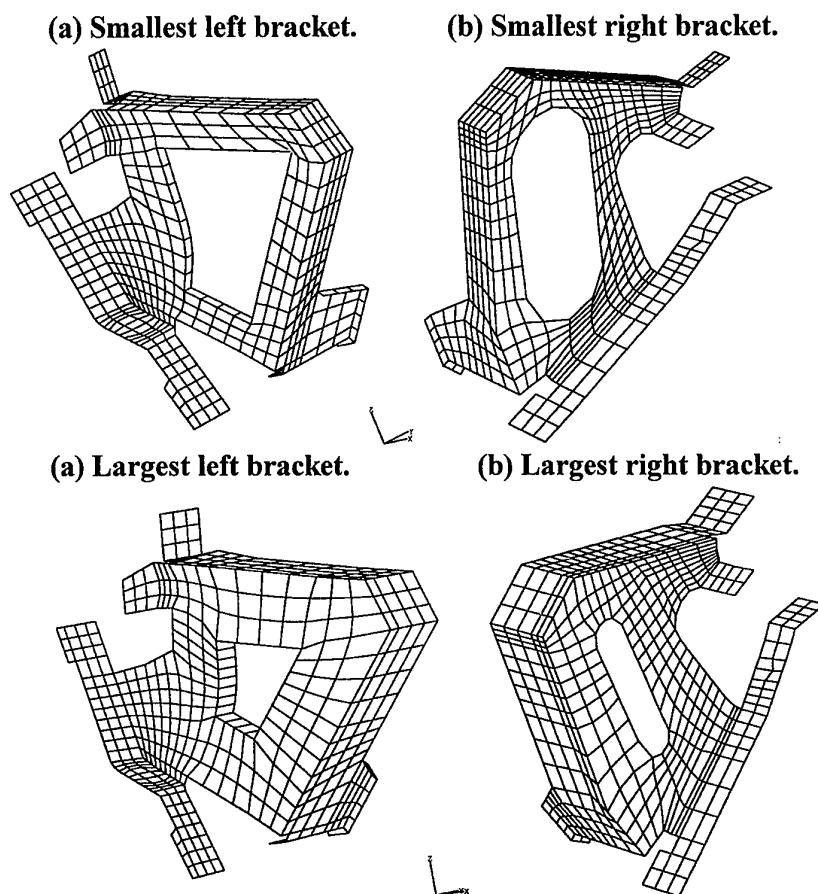


Figure 3: Views of the smallest and largest shapes in the design space.

FORMULATION OF THE OPTIMIZATION PROBLEM

Constraints. For optimal occupant kinematics, it is essential that knee intrusion into the IP be limited to desired values. Upper bounds of the left and right knee displacements, D_{LK} and D_{RK} , are used to limit knee intrusions to **115 mm**. The yoke displacement is limited to **85mm**.

Objective. In general, the primary object of knee bolster crashworthiness engineering is to minimize the forces on the occupant to specific program targets within the limits of the design envelope. Therefore, the primary optimization criterion is to minimize the force on the occupant's left and right knees, F_{LK} and F_{RK} . The selection of a low force constraint value forced the optimization formulation to minimize the maximum knee force subject to the constraints above, i.e.

$$\min (\max (F_{LK} , F_{RK})).$$

The maximization considers both the knee forces over time. The knee forces have been filtered, SAE 60 Hz, to improve the approximation accuracy.

RESULTS

Based on the over-sampling thumb rule of 1.5 (Roux, 1998), LS-OPT generated 8 LS-DYNA design points for each iteration ($n = 4$) compared to 19 for $n = 11$. Table 3 lists the design variable values and computed responses for the baseline design, the shape optimized design (7 iterations) and the thickness-only optimization (7 iterations). The fixed parameters are shown in bold type. The points and lines in Figures 4 and 5 represent the simulated and predicted results respectively. The knee forces have been scaled by 6500N (Figure 4) and the yoke displacements by 85mm (Figure 5). The predicted knee force pairs are (5757;5757) for the full shape design vs. (6113;6009) for the sizing design (a difference of approximately 6%). It should be noted that, except for the lower bound of the right bracket gauge, only the yoke displacement constraint is active.

Parameter	Baseline	Shape Optimized (11 variables)	Gauge & Yoke dia. (4 variables)
Left Bracket Gauge, G_{RB}	1.1	0.94	1.22
Right Bracket Gauge, G_{LB}	1.1	0.7*	0.7
Knee Bolster Gauge, G_{KB}	3.5	5.58	5.7
Yoke Cross-Section Radius, R_Y	4	2.90	2.69
Right EA Hole Radius, R_{RB}	15	14.4	15
Right EA Width, W_{RB}	32	15.4	32
Left EA Depth Top, D_{LBT}	28.3	25.2	28.3
Left EA Depth Front, D_{LBF}	27.5	26.4	27.5
Left EA Depth Bottom, D_{LBB}	22.3	14.9	22.3
Left EA Inner Flange Width, W_{LBF}	7	6.9	7
Left EA Width, W_{LB}	32	46.8	32
Maximum Left Knee Force	6626 N	6045 N	6136
Maximum Right Knee Force	8602 N	5763 N	6110
Maximum Left Knee Disp.	96.4 mm	100.9 mm	97.2
Maximum Right Knee Disp.	98.7 mm	99.9 mm	91.9
Yoke displacement	85.9 mm	70.4	93.8

Table 3: Baseline and Optimal design characteristics

The force history (Figure 4) confirms that there is a significant difference between the baseline and optimal designs but that the sizing variables cause most of the design improvement. It takes about two iterations for the maximum force to reduce to its minimum value. Figure 5 also shows that the predicted yoke displacement is active. However, it appears as if the yoke displacement response is severely noisy, probably as a result of unstable structural behavior which activates different failure mechanisms for different designs. Figure 6 illustrates the convergence of the yoke radius towards a lower value. The dotted lines represent the bounds of the dynamic sub-region. Figure 7 shows that the filtered knee forces tend to equalize and distribute evenly over time.

It must be mentioned that because this analysis is based on a single size occupant, a defined vehicle environment, and a single test mode, more simulations would be necessary to determine the overall effectiveness of this design.

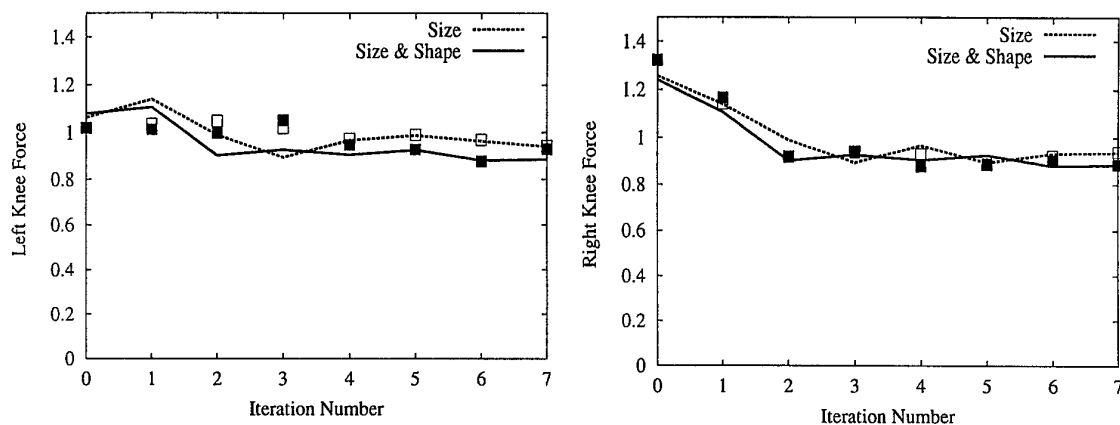


Figure 4: Optimization History: Left and Right Knee Forces

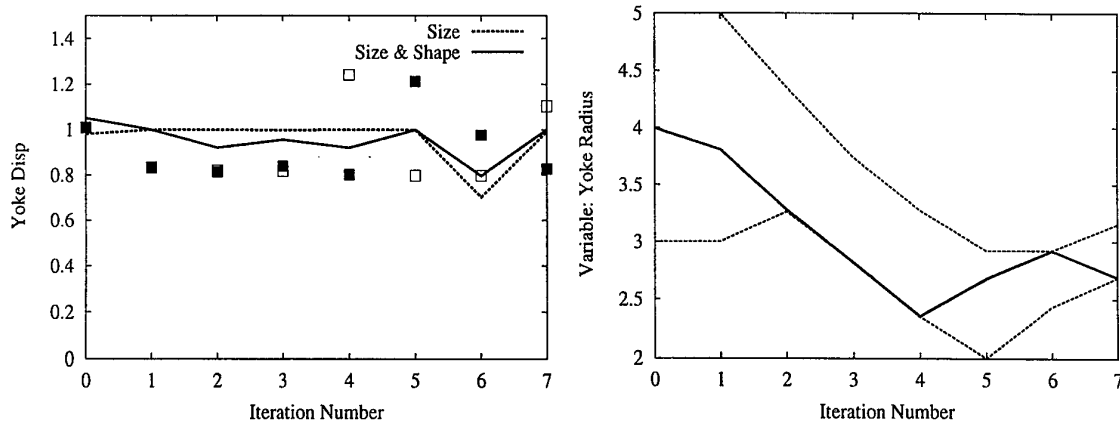


Figure 5: Yoke Displacement History

Figure 6: Yoke Radius History

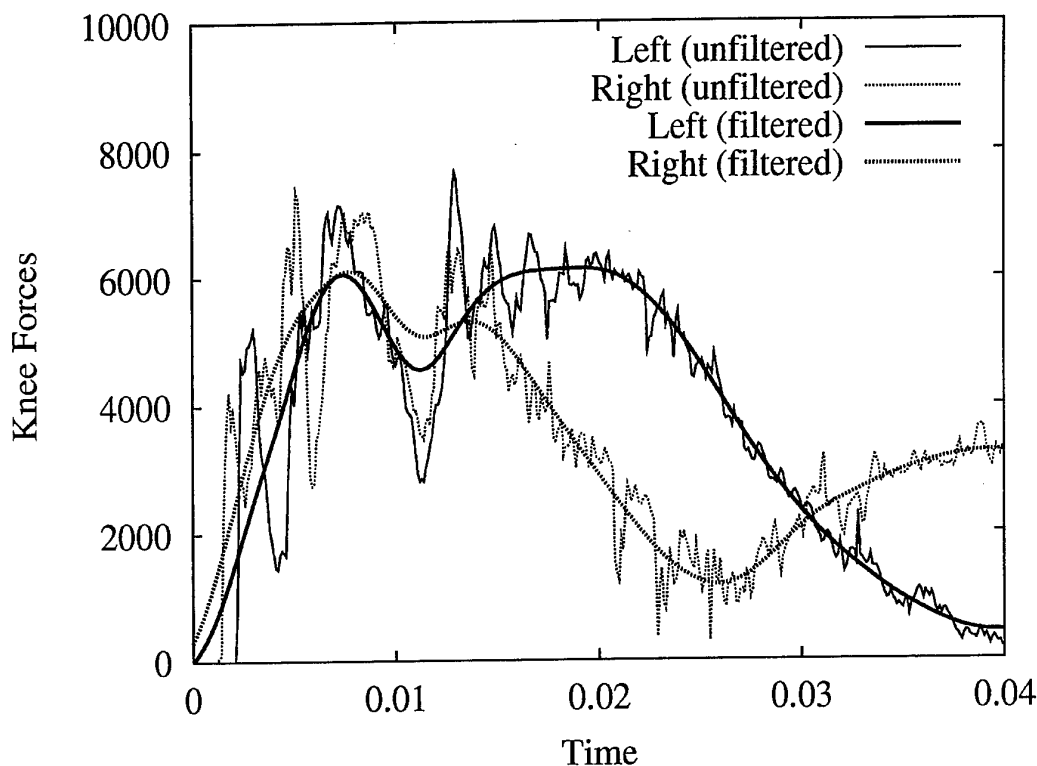


Figure 7: Time history of knee forces (four variable problem)

CONCLUSIONS

The paper demonstrates the use of a successive response surface method to optimize part thicknesses and shapes of an instrument panel in order to enhance its crashworthiness.

The predicted LS-OPT results were all well within acceptable levels of correlation with the actual LS-DYNA runs for the same design parameters. However, as would be expected, some responses are more susceptible to random response variations than others as was demonstrated by the yoke displacement (noisy) and the filtered knee force (reasonably smooth) responses. It was shown that there is a small advantage (~6%) in using a large number of shape variables in the optimization. An almost equally good design could be achieved by using a subset of the variables, in this case the thickness variables of all the components involved.

Future optimization studies will focus the investigation on the effect of multiple starts on the global optimum and the use of a gradient-based optimization approach.

Acknowledgement. We would like to express our gratitude and appreciation to Kumar Kulkarni of Visteon Corporation who built the finite element model used in this study.

REFERENCES

- Akkerman, A., Kuhn, R., Rajic, H., Stander, N., Thyagarajan, R., "Optimization of Instrument Panel Crashworthiness", 2nd European LS-DYNA Users Group Meeting, May, 1999.
- Akkerman, A., Burger, M., Kuhn, R., Rajic, H., Stander, N., Thyagarajan, R., "Shape optimization of instrument panel components for crashworthiness using distributed computing. Proceedings of the 6th LS-DYNA User's Conference, Dearborn, MI, April 9-11, 2000.
- Akkerman, A., Burger, M., Kuhn, R., Rajic, H., Stander, N., Thyagarajan, R., "Shape optimization of instrument panel components for crashworthiness using distributed computing. Proceedings of the International Body Engineering Conference (IBEC), October 3-5, 2000.
- Etman, P. Optimization of Multibody Systems using Approximation Concepts. Ph.D. thesis, Technical University Eindhoven, The Netherlands, 1997.
- Jira, J., Kulkarni, K., and Thyagarajan, R., "Deformation Control in Automotive Instrument Panels Exposed to High Sunload Temperatures With the Use of Unique Cowltop Attachments", International Body Engineering Conference, 1996, Interior and Safety Systems, Detroit, MI, Oct 1996.
- Kok, S. and Stander, N. Optimization of a sheet metal forming process using successive multi-point approximations. *Structural Optimization*. Vol. 18, pp. 277-295, 1999.
- Kok, S., Stander, N. and Roux, W.J. Thermal optimization in transient thermo-elasticity using response surface approximations. *International Journal for Numerical methods in Engineering*, Vol. 43, 1-21, 1998.
- Kulkarni, K., and Thyagarajan, R., "A Brief Look at Instrument Panel Knee Bolster Designs and Materials", 1998 Regional Technical Conference of Society of Plastics Engineers, Detroit, MI, May 1998.
- LSTC, LS-DYNA Keyword User's Manual. Version 950. May 1999.
- LSTC, LS-OPT User's Manual. Version 1. September, 1999.
- Roux, W.J., Stander, N., and Haftka, R., "Response Surface Approximations for Structural Optimization", *International Journal for Numerical Methods in Engineering*, Vol. 42, pp. 517-534, 1998.
- Snyman, J.A. The LFOPC leap-frog algorithm for constrained optimization. *In press, Computers and Mathematics for Applications*.
- Standar, N., "Crashworthiness Design Optimization Using LS-OPT and Full-Model LS-DYNA Analyses", LS-DYNA Users Conference, Japan, 1998.
- Standar, N., Reichert, R. and Frank, T., "Optimization in Nonlinear Dynamics Using Successive Linear Approximations", 6th International LS-DYNA Users Conference, Detroit, MI, April 2000.
- Standar, N., Roux, W., Pattabiraman, S., and Dhulipudi, R., "The Application of LS-OPT to Design Optimization Problems in Nonlinear Dynamics Using LS-DYNA", 5th International LS-DYNA Users Conference, Detroit, MI, September 1998.

THE SPHERICAL APPROXIMATION GRAPH MATCHING ALGORITHM

B.J. van Wyk^a M.A. van Wyk^b

^a University of the Witwatersrand, Johannesburg, South Africa, vanwykbj@xsinet.co.za

^bTechnikon South Africa, Roodepoort, South Africa, mvanwyk@tsa.ac.za

ABSTRACT

Many engineering applications require the matching of images based on their structural descriptions. The structural data of an image can be represented by a weighted graph and the similarity between object descriptions can therefore be established using graph matching algorithms. The Spherical Approximation Graph Matching (SAGM) algorithm capable of performing full- and sub-graph matching is presented and its performance compared to the linear programming, eigen-decomposition, polynomial transform, RKHS Interpolator-based and graduated assignment graph matching algorithms. Simulation results indicate that the SAGM algorithm is more robust against noise than any of the other algorithms it was compared against.

1 INTRODUCTION

Structural descriptions are general methods to describe visual objects. The structural data of an image can be represented by a weighted graph. In image processing applications it is often required to match different images of the same object or similar objects based on the structural descriptions constructed from these images. If the structural descriptions of objects are represented by attributed relational graphs, different images can be matched by performing Attributed Graph Matching (AGM). The AGM problem is a special case of the more general graph isomorphism problem which is proved to be NP-complete [10]. Because of the combinatorial nature of this problem it can be efficiently solved by an exhaustive search only when dealing with extremely small graphs.

Several algorithms have been proposed to solve graph matching problems during recent years. You and Wong [1] and Tsai and Fu [2] [6] proposed tree search techniques but the combinatorial nature of these approaches renders them impractical for moderate to large graphs. Other proposed algorithms include a symmetric *Polynomial Transform Graph Matching* (PTGM) algorithm by Almohamad [3], a graph distance measure algorithm by Eshera and Fu [7], a *Linear Programming Graph Matching* (LPGM) algorithm by Almohamad and Duffuaa [4], an *Eigen-decomposition Graph Matching* (EIGGM) method by Umeyama [5], a Lagrangian relaxation method by Rangarajan and Mjolsness [8], an *RKHS Interpolator-based Graph*

Matching (RIGM) algorithm by [12], genetic algorithms [9] and a multitude of neural network and relaxation based methods. Recently Gold and Rangarajan [11] introduced the *Graduated Assignment Graph Matching* (GAGM) algorithm which combines graduated nonconvexity, two-way assignment constraints and sparsity.

In this paper the *Spherical Approximation Graph Matching* (SAGM) algorithm is presented. This algorithm is unique in the way a constrained quadratic cost function is obtained and minimised. The minimisation part of the algorithm resembles the Dynamic-Q methodology [14] [15] in the sense that it also uses iterative spherical approximations. However, instead of using the LFOPC [16] or similar techniques to enforce constraints at each iteration, iterative spherical approximations is combined with an optimal assignment method presented in [18] to ensure that the optimal answer to the constrained minimisation problem is a permutation matrix (when performing full-graph matching) or sub-matrix (when performing sub-graph matching).

2 COST FUNCTION FORMULATION

The focus of this paper is on matching undirected graphs where a reference graph, say

$$G' = (V', E', \{\mathbf{A}'_i\}_{i=1}^r, \{\mathbf{B}'_j\}_{j=1}^s), \quad (1)$$

is matched to a duplicate graph, say

$$G = (V, E, \{\mathbf{A}_i\}_{i=1}^r, \{\mathbf{B}_j\}_{j=1}^s), \quad (2)$$

where $\mathbf{A}'_i$, $\mathbf{A}_i$, $\mathbf{B}'_j$ and $\mathbf{B}_j$ respectively represent the edge attribute adjacency matrices and vertex attribute vectors. The reference and duplicate graphs each have r edge attributes and s vertex attributes. The number of vertices of G' (respectively, G) is $n' := |V'|$ (respectively, $n := |V|$). Here we consider the general case of sub-graph matching ($n' \geq n$). The Attribute Graph Matching Problem (AGM) can be expressed as the combinatorial optimisation problem

$$\min_{\mathbf{P}} \left(\sum_{i=1}^r \|\mathbf{A}_i - \mathbf{P}\mathbf{A}'_i\mathbf{P}^T\|_F^2 + \sum_{j=1}^s \|\mathbf{B}_j - \mathbf{P}\mathbf{B}'_j\|_2^2 \right), \quad (3)$$

where $\mathbf{P} \in \text{Per}(n, n')$, the set of all $n \times n'$ permutation matrices or sub-matrices. Here $\|\cdot\|_F$ denotes the Frobenius matrix norm and $\|\cdot\|_2$ the Euclidean norm.

The approximate solution to (3) takes the form (refer to [13])

$$\max_{\bar{\mathbf{p}}} \bar{\mathbf{p}}^T \mathbf{p}, \quad \text{devec}(\bar{\mathbf{P}}) \in \text{Per}(n, n'), \quad (4)$$

where $\bar{\mathbf{p}} \in \mathbf{R}^{nn'}$ minimises the quadratic form

$$J(\mathbf{p}) := \mathbf{p}^T \mathbf{X} \mathbf{p} - 2\mathbf{y}^T \mathbf{p} + z \quad (5)$$

with the matrices $\mathbf{X} \in \mathbf{R}^{nn' \times nn'}$ and $\mathbf{y} \in \mathbf{R}^{nn' \times 1}$ and the scalar $z \in \mathbf{R}$ given as:

$$\begin{aligned}
\mathbf{X} &:= \sum_{i=1}^r \mathbf{M}_{AA'i}^T \mathbf{M}_{AA'i} + \sum_{j=1}^s \mathbf{N}_{A'j}^T \mathbf{N}_{A'j}, \\
\mathbf{y} &:= \sum_{j=1}^s (\mathbf{B}_j^T \mathbf{N}_{A'j}), \\
\mathbf{z} &:= \sum_{j=1}^s (\mathbf{B}_j^T \mathbf{B}_j), \\
[\mathbf{N}_{A'j}]_{k, k+n(l-1)} &:= \begin{cases} [\mathbf{B}'_j]_l & \text{for } k = 1, \dots, n \text{ and } l = 1, \dots, n' \\ 0 & \text{otherwise} \end{cases} \\
\mathbf{M}_{AA'i} &:= \mathbf{M}_{A'i} + \mathbf{M}_{Ai}.
\end{aligned}$$

The matrices $\mathbf{M}_A$ and $\mathbf{M}_{A'}$ are defined by

$$\begin{aligned}
[\mathbf{M}_{Ai}]_{kl} &:= \begin{cases} [\mathbf{A}_i]_{l(k-1)(\text{mod } n)+1, (k-1)(\text{mod } n)+l-k+1}, & \text{for} \\ k = 1, \dots, nn' \text{ and} \\ l = k - (k-1)\text{mod } n, \dots, k+n-1 - (k-1)\text{mod } n, & \\ 0, & \text{otherwise} \end{cases} \\
[\mathbf{M}_{A'i}]_{k, (k-1)\text{mod } n + n(l-1)+1} &:= \begin{cases} -[\mathbf{A}'_i]_{l, [(k-1)-(k-1)\text{mod } n]/n+1}, & \text{for} \\ k = 1, \dots, nn' \text{ and } l = 1, \dots, n' & \\ 0, & \text{otherwise} \end{cases}
\end{aligned}$$

3 ITERATIVE COST FUNCTION MINIMISATION

3.1 TRADITIONAL GRADIENT BASED METHODS

To find the minimum of the quadratic function given by equation (5) either the steepest descent method, Newton's method or the conjugate gradient method could be used. Newton's method is attractive because of the *quadratic termination* property, but the inverse of the Hessian is required. The steepest descent algorithm uses orthogonal search directions at each consecutive iteration. For quadratic functions with highly elliptical contours this produces a zig-zag trajectory of short steps and results in slow convergence. The problem of slow convergence can be overcome by using the conjugate gradient method which executes a sequence of exact linear searches along a set of conjugate directions. Although the conjugate gradient algorithm is guaranteed to converge to the optimum within a certain number of iterations it is sensitive to machine roundoff error, especially when dealing with problems of large dimension.

The optimal answer obtained, say $\hat{\mathbf{P}}$, by the above mentioned gradient methods will in general not be a permutation matrix or sub-matrix. We consequently use $\hat{\mathbf{P}}$ as the weight matrix of an *Optimal Assignment Problem*, the solution of which is a permutation matrix

or sub-matrix $\bar{\mathbf{P}}$ representing the optimal assignment. The matrix $\bar{\mathbf{P}}$ is an approximation to $\mathbf{P}$, the permutation matrix which appears in equation (3). It can be shown that $\bar{\mathbf{P}}$ is the permutation matrix closest to $\hat{\mathbf{P}}$ in the sense that $\|\bar{\mathbf{P}} - \hat{\mathbf{P}}\|_F$ is a minimum. In contrast, the optimal answer obtained by the SAGM algorithm will inherently be a permutation matrix (or sub-matrix) and hence the additional requirement to solve an optimal assignment problem is avoided.

3.2 GRADUALLY CONSTRAINED SPHERICAL APPROXIMATIONS

The essence of the SAGM algorithm is constituted by the following equations for $k > 2$ where k represents the iteration number:

$$\begin{aligned}\bar{\mathbf{p}}_k &= \text{vec}(\bar{\mathbf{P}}_k), \\ \bar{\mathbf{P}}_k &= \xi(\mathbf{P}_k), & (\text{see text below}) \\ \mathbf{P}_k &= \text{devec}(\mathbf{p}_k), \\ \mathbf{p}_k &= \exp\left(\beta_k \left(\bar{\mathbf{p}}_{k-1} - \frac{\nabla J(\bar{\mathbf{p}})|_{k-1}}{c_k}\right)\right), \quad (\exp(\mathbf{p}) := (\exp(p_i))) \\ \nabla J(\bar{\mathbf{p}})|_{k-1} &= 2(\mathbf{X}\bar{\mathbf{p}}_{k-1} - \mathbf{y}), \\ c_k &= 2 \left(\frac{(J(\bar{\mathbf{p}}_{k-2}) - J(\bar{\mathbf{p}}_{k-1}) - (\nabla J(\bar{\mathbf{p}})^T|_{k-1}) \mathbf{d}_{k-1})}{g_{k-1}} \right), \\ \beta_k &= \beta_{k-1} + \alpha, \\ \mathbf{d}_{k-1} &= \bar{\mathbf{p}}_{k-1} - \bar{\mathbf{p}}_{k-2}, \\ g_{k-1} &= (\mathbf{d}_{k-1})^T (\mathbf{d}_{k-1}).\end{aligned}$$

The above process is repeated until $c_k < \epsilon_1$. For most of our experiments we have chosen $\epsilon_1 = 0.00001$. For all our experiments the initial value for β was chosen as 5 and the incremental value, α , was set to 0.5. The operation $\xi(\mathbf{P})$ can be described as follows: First normalise across all rows of $\mathbf{P} = (P_{ij})$

$$P_{ij} = \frac{P_{ij}}{\sum_{i=1}^I P_{ij}}, \quad (6)$$

and then we normalise across all columns of $\mathbf{P}$,

$$P_{ij} = \frac{P_{ij}}{\sum_{j=1}^J P_{ij}}, \quad (7)$$

where I represents the number of rows of $\mathbf{P}$ and J represents the number of columns of $\mathbf{P}$. The row and column normalisation procedures are repeated until

$$\sum_{i=1}^I \sum_{j=1}^J |P_{ij} - \bar{P}_{ij}| < \epsilon_2. \quad (8)$$

In our experiments we set $\epsilon_2 = 0.05$. If a value smaller than ϵ_2 was not achieved within 30 iterations the normalisation process was terminated. As mentioned earlier a permutation matrix is not obtained when performing subgraph matching, but rather a permutation sub-matrix $\in \mathbb{R}^{n \times n'}$. As there were now more columns than rows a slack variable per column was introduced during normalisation to ensure that $\bar{\mathbf{P}}$ will have the desired properties of a permutation sub-matrix. The exponentiation of $\mathbf{p}_k$ together with the row and column normalisations gradually forces $\bar{\mathbf{P}}_k$ to only assume discrete values. As β_k gets large $\bar{\mathbf{P}}_k$ will tend to the the desired permutation matrix or sub-matrix. (See also [11] and [18].)

The initial value $\bar{\mathbf{p}}_0$ can be chosen as a random vector and c_1 very small or zero. After $\bar{\mathbf{p}}_0$ and c_1 were initialised $\bar{\mathbf{p}}_1$ was calculated as

$$\bar{\mathbf{p}}_1 = \mathbf{p}_0 - \frac{2(\mathbf{X}\bar{\mathbf{p}}_0 - \mathbf{y})}{c_1}.$$

4 SIMULATION RESULTS

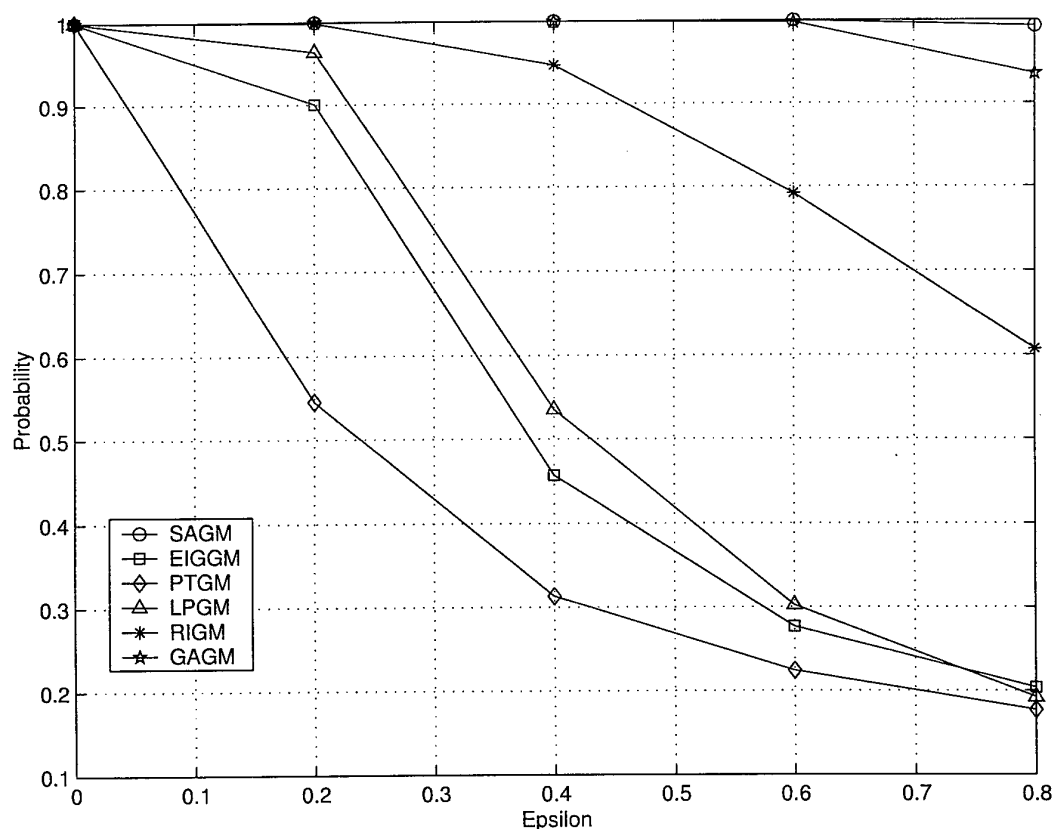


Figure 1: Probability of correct vertex-vertex matching versus ϵ for (10,3,3) attributed graphs.

In order to evaluate the performance of the SAGM algorithm the following procedure was used: Firstly, the parameters n', n, r and s were fixed. For every iteration a reference graph

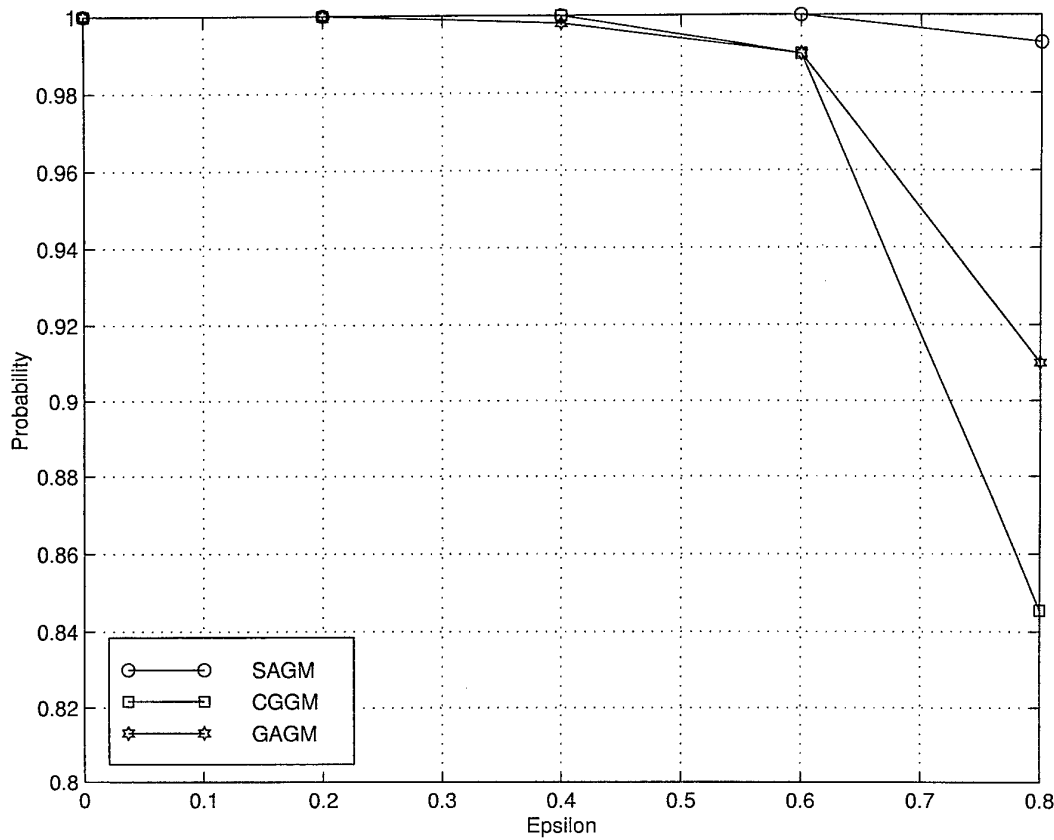


Figure 2: Probability of correct vertex-vertex matching versus ε for $(20,3,3)$ attributed graphs.

G' was generated randomly with all attributes distributed between 0 and 1. An $n \times n'$ permutation matrix (or sub-matrix), $\mathbf{P}$, was also generated randomly and then used to permute the rows and columns of the edge attribute adjacency matrices and the elements of the vertex attribute vectors of G' . Next, an independently generated noise matrix (vector, respectively) was added to each edge attribute adjacency matrix (vertex attribute vector, respectively) to obtain the duplicate graph G . The element of each noise matrix/vector was obtained by multiplying a random variable —uniformly distributed on the interval $[-1/2, 1/2]$ — by the noise magnitude parameter ε . Different graph matching algorithms were then used to determine a permutation matrix (or sub-matrix) which approximates the original permutation matrix (or sub-matrix) $\mathbf{P}$.

In figure 1 the performance of the SAGM algorithm with ϵ_1 set to 0.00001 is compared to the performance of the GAGM, EIGGM, RIGM, PTGN and LPGM algorithms for $n' = 10$, $n = 10$, $r = 3$ and $s = 3$. The probability of a correct vertex-vertex assignment was calculated for a given value of ε after every 500 trials. From a probabilistic point of view this reflects how well the proposed algorithm performs for a given noise magnitude.

In figure 2 the performance of the SAGM algorithm with ϵ_1 set to 0.00001 is compared

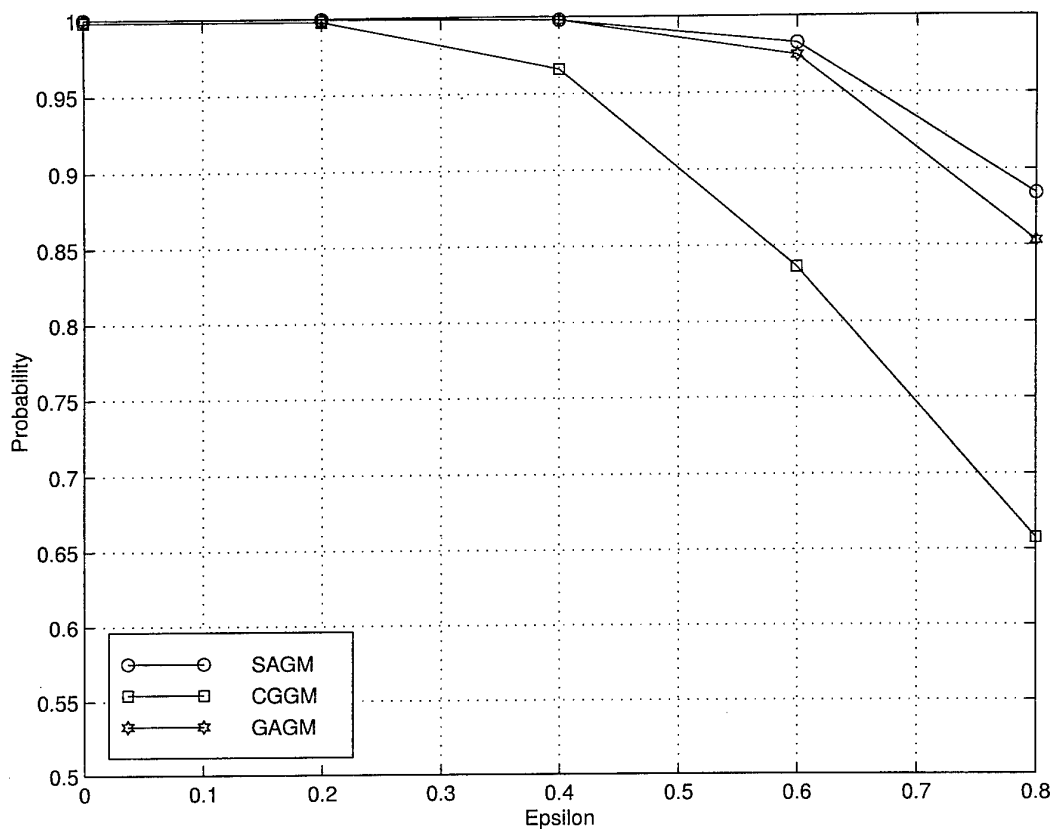


Figure 3: Probability of correct vertex-vertex matching versus ϵ for $(15/10,3,3)$ attributed graphs.

to the performance of the GAGM and *Conjugate Gradient Graph Matching* (CGGM) algorithms for $n' = 20$, $n = 20$, $r = 3$ and $s = 3$. The CGGM algorithm minimises equation (5) using the conjugate gradient method followed by an optimal assignment routine. The conjugate gradient part of the CGGM algorithm was terminated when the value $\|\mathbf{p}_k - \mathbf{p}_{k-1}\|_2^2$ became smaller than 0.00001. The probability of a correct vertex-vertex assignment was again calculated for a given value of ϵ after every 500 trials. Comparable results were obtained for $n' = 30$, $n = 30$, $r = 3$ and $s = 3$.

In figure 3 the sub-graph matching performance of the SAGM algorithm with ϵ_1 set to 0.00001 is compared to the performance of the GAGM and CGGM algorithms for $n' = 15$, $n = 10$, $r = 3$ and $s = 3$. The conjugate gradient part of the CGGM algorithm was again terminated when the value $\|\mathbf{p}_k - \mathbf{p}_{k-1}\|_2^2$ became smaller than 0.00001. The EIGGM, LPGM and PTGM algorithms are not suitable for performing sub-graph matching.

From the results it is evident that the SAGM algorithm outperformed all the other algorithms it was compared against. It must however be noted that the SAGM algorithm is sensitive to the value of ϵ_1 . Its performance degrades notably for values of ϵ_1 larger than 0.00001. For $n' = 10$ and $n = 10$, the algorithm took on average 43 iterations to converge.

For $n' = 20$ and $n = 20$, the algorithm took on average 55 iterations to converge.

5 CONCLUSION

A robust algorithm for performing attributed full- and sub-graph matching was presented and its performance compared to the performance of the EIGGM, GAGM, RIGM, PTGM and LPGM algorithms. It was demonstrated that the SAGM algorithm performed significantly better than the algorithms it was compared against and that the algorithm is extremely robust against noise.

Further work will include the characterisation of the algorithm using real image data, convergence analysis, detailed complexity analysis and DSP implementation.

Acknowledgement

The authors would like to thank Prof Jan Snyman for several valuable discussions.

References

- [1] M. You, K.C. Wong, *An Algorithm for Graph Optimal Isomorphism*, Proc. ICPR. pp. 316-319, 1984,
- [2] W.H. Tsai, K.S. Fu, *Error-Correcting Isomorphisms of Attributed Relation Graphs for Pattern Recognition*, IEEE Trans. Syst. Man Cybern., vol. SMC-9, pp. 757-768, 1979.
- [3] H.A.L. Almohamad, *Polynomial Transform for Matching Pairs of Weighted Graphs*, Appl. Math. Modelling, vol. 15, no. 4, pp. 216-222, 1991.
- [4] H.A.L. Almohamad, S.O. Duffuaa, *A Linear Programming Approach for the Weighted Graph Matching Problem*, IEEE Trans. Patt. Anal. Machine Intell., vol. 15, no. 5, pp. 522-525, 1993.
- [5] S. Umeyama, *An Eigendecomposition Approach to Weighted Graph Matching Problems*, IEEE Trans. Patt. Anal. Machine Intell., vol. 10, no. 5, pp. 695-703, 1988.
- [6] W.-H. Tsai and K.-S. Fu, *Subgraph Error-Correcting Isomorphisms for Syntactic Pattern Recognition*, IEEE Trans. Systems, Man, Cybernetics., vol. 13, pp 48-62, 1983.
- [7] M.A. Eshera and K.S. Fu, *A Graph Distance measure for Image Analysis*, IEEE Trans. Systems, Man, Cybernetics., vol. 13, pp 398-407, 1984.
- [8] A Rangarajan and E. Mjolsness, *A Lagrangian Relaxation Network for Graph Matching*, IEEE Int'l Conf. Neural Networks (ICNN), vol. 7, pp 4629-4634, IEEE Press, 1994.
- [9] M. Krcmar and A Dhawan, *Application of Genetic Algorithms in Graph Matching*, IEEE Int'l Conf. Neural Networks (ICNN), vol. 6, pp 3872-3876, IEEE Press, 1994.
- [10] M.R. Garey and D.S. Johnson, *Computers and Intracability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, 1979.

- [11] S. Gold and A Rangarajan, *A Graduated Assignment Algorithm for Graph Matching*, IEEE Trans. Patt. Anal. Machine Intell, vol. 18, pp 377–388, 1996.
- [12] M.A. van Wyk, B.J. van Wyk and T.S. Durrani, *A RKHS Interpolator-based Graph Matching Algorithm*, Submitted to IEEE Trans. Patt. Anal. Machine Intell, 1999.
- [13] M.A. van Wyk, *Graph Matching Algorithms: Analytical and Preliminary Simulation Results*. Internal Report, Technikon SA, 1998.
- [14] K.J. Craig, D.J. de Kock and P Gauché, *Minimisation of Heat Sink Mass using CFD and Mathematical Optimisation*, ASME Journal of Electronic Packaging, vol. 121, No. 3, pp 143–147, 1999.
- [15] K.J. Craig, L.T. Visser and D.J. Penning, *Optimisation of the Lift-to-Drag Ratio of a Touring Car*, Submitted to ASME Journal of Fluids Engineering, 1999.
- [16] J.A. Snyman, *The LFOPC Leap-Frog Algorithm for Constrained Optimisation*, In press, Computers Math. Appl., 1999.
- [17] J.A. Snyman and N Stander, *A New Successive Approximation Method for Optimum Structural Design*. AIAA Journal, vol. 32, pp 1310–1315, 1994.
- [18] J.J. Kowsowsky and A.L. Yuille, *The Invisible Hand Algorithm: Solving the Assignment Problem with Statistical Physics*, Neural Networks, vol. 7, pp 477–490, 1994.

OPTIMISATION OF HEAT SINKS USING MATHEMATICAL OPTIMISATION

J.A. Visser, D.J. de Kock

*Department of Mechanical and Aeronautical Engineering
University of Pretoria, Pretoria, South Africa*

Abstract

Heat sink designers have to balance a number of conflicting parameters to maximise the performance of a heat sink. This must be achieved within the given constraints of size or volume of the heat sink as well as the mass or material cost of the heat sink. This multi-parameter problem lends itself naturally to optimisation techniques. Traditionally, an experimental approach was used where different heat sink designs were constructed and their performance measured. This approach is both time-consuming and costly. More recently, Computational Fluid Dynamics (CFD) techniques have been used, but mostly on a trial-and-error basis. This leads to long design cycles and is basically the numerical equivalent of the experimental approach. A better approach is to combine a semi-empirical simulation program with mathematical optimisation techniques. This paper describes the use of mathematical optimisation techniques to optimise heat sinks. The simulation uses the Qfin 2.1 code, while the optimisation is carried out by means of the DYNAMIC-Q method. This method is specifically designed to handle constrained problems where the objective and/or constraint functions are expensive to evaluate. The paper illustrates how the parameters considered influence the heat sink mass and how mathematical optimisation techniques can be used by the heat sink designer to design compact heat sinks for different types of electronic enclosures.

Nomenclature

ar	Heat sink channel aspect ratio	r	Number of equality constraints
c_j	Curvature of sub-problem	Ra^*	Modified Raleigh number
δ_i	Specified move limit for i -th design variable	S	Source term in governing equation [W/m ³]
d_h	Hydraulic diameter	sh	Heat sink fin gap [m]
$f(x)$	Objective function	T	Temperature [K]
$g_j(x)$	j -th inequality constraint	T_{HS}	Maximum heat sink temperature [K]
$\bar{h}$	Mean heat transfer coefficient [W/m ² K]	T_{spec}	Maximum specified heat sink temperature [K]
$h_k(x)$	k -th equality constraint	V_h	Velocity between the fins [m/s]
k	Thermal conductivity [W/mK]	VL	Constant
L	Length of heat sink [m]	x	Design vector
m	Number of inequality constraints	$x_i^{\min}$	Minimum value of i -th design variable
m_{HS}	Heat sink mass [kg]	$x_i^{\max}$	Maximum value of i -th design variable
m_{spec}	Specified maximum heat sink mass [kg]	Δx_i	Step size for first-order differencing scheme
n	Number of design variables		
$\overline{Nu}$	Mean Nusselt number		
$p(x)$	Penalty function		
ρ, β, μ	Penalty parameters		

1 Introduction

The continuing increase of power densities in electronics packages and the simultaneous drive to reduce the size and weight of electronic products have led to an increased importance on thermal management issues in this industry. The temperature at the junction of an electronics package has become the limiting factor determining the lifetime of the package and much attention has been dedicated to understanding the heat transfer mechanisms involved in this problem. The most common method for cooling packages is the use of aluminium heat sinks. These heat sinks provide a large surface area for the dissipation of heat and effectively reduce the thermal resistance of a package. Unfortunately, heat sinks often take up much space and contribute to the weight and cost of the product.

The performance of a heat sink depends on a number of parameters including the thermal conduction resistance, dimensions of the cooling channels, location and concentration of heat sources as well as the airflow bypass due to flow resistance through the heat sink. These parameters make the optimal design of a heat sink very difficult. Traditionally, the performance of heat sinks is measured experimentally and the results are made available in the form of design graphs in heat sink catalogues. This characterisation method has been the topic of much debate as vendors have applied different standards or interpretations to determine the characterisation of heat sinks [1,2]. Analytical and empirical formulations for the fin efficiency, pressure drop and the heat transfer coefficient have also been used in the design process to determine the optimal heat sink design. Knight et al [3,4] developed and verified a generalised model to determine the optimal geometrical design of closed-fin heat sinks. Lee [5] analytically determined the optimal design of heat sinks by performing a parametric analysis that takes flow bypass into account. Computational Fluid Dynamics (CFD) techniques have been used more frequently in the last few years [6,7], but mostly on a trial-and-error basis due to the computational cost of performing parametric studies. A better approach is to combine a simplified semi-empirical simulation program with a mathematical optimisation technique, thereby incorporating the influence of the design variables automatically.

The simulation presented in this paper uses the Qfin 2.1 code that solves the heat conduction in the solid while using an empirical model for the heat transfer between the heat sink and the air. The optimisation is carried out by means of the DYNAMIC-Q method [8], which is specifically designed to handle constrained problems where the objective or constraint functions are expensive to evaluate. The optimisation method is a gradient method for constrained optimisation applied to successive approximate quadratic sub-problems.

2 Problem definition and formulation

The problem considered in this paper is the minimisation of heat sink mass given a specified heat source, duct geometry and fan characteristics in the case of forced convection. The problem is depicted schematically in Figure 1. The five design variables, fin height (x_1), fin thickness (x_2), extrusion length (x_3), base thickness (x_4), and number of fins (x_5), are indicated in the figure.

For practical considerations, the design variables are confined to certain limits. These are shown in the results section. These considerations are derived from manufacturing limitations and geometry considerations.

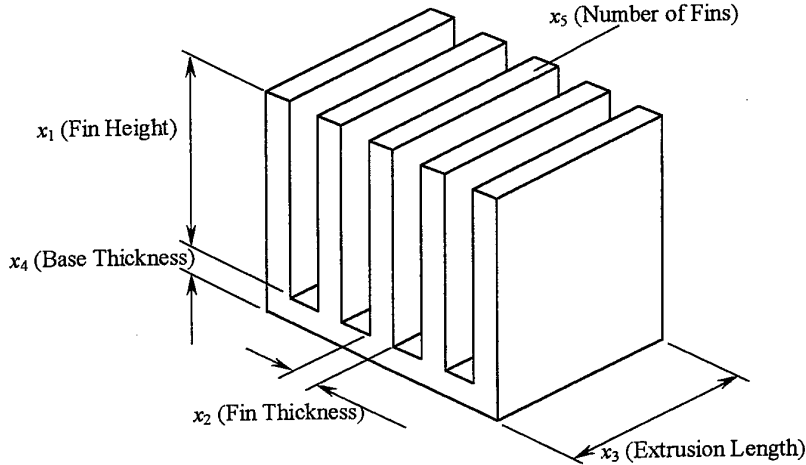


Figure 1: Graphical presentation of design variables

The complete mathematical formulation of the optimisation problem, in which the constraints are written in the standard form $g(\mathbf{x}) \leq 0$, where $\mathbf{x}$ denotes the vector of the design variables $(x_1, x_2, x_3, x_4, x_5)^T$, is as follows:

$$\begin{aligned} &\text{Minimise } f(\mathbf{x}) = m_{HS}(\mathbf{x}), \text{ subject to} \\ &g_j = -x_j + x_j^{\min} \leq 0; j = 1, 3, 5 \dots (2n-1) \\ &g_j = x_j - x_j^{\max} \leq 0; j = 2, 4, 6 \dots, 2n \\ &g_{2n+1} = T_{HS}(\mathbf{x}) - T_{spec} \leq 0 \end{aligned} \quad (1)$$

It can be seen from the formulation that the objective function f is straightforward but that the last constraint requires a simulation for evaluation, and therefore need to be approximated.

An alternate approach would be to minimise thermal resistance of the heat sink while limiting the mass of the heat sink. The formulation of the optimisation problem is then as follows:

$$\begin{aligned} &\text{Minimise } f(\mathbf{x}) = R_{HS}(\mathbf{x}), \text{ subject to} \\ &g_j = -x_j + x_j^{\min} \leq 0; j = 1, 3, 5 \dots (2n-1) \\ &g_j = x_j - x_j^{\max} \leq 0; j = 2, 4, 6 \dots, 2n \\ &g_{2n+1} = m_{HS}(\mathbf{x}) - m_{spec} \leq 0 \end{aligned} \quad (2)$$

3 Theoretical modelling

3.1 Thermal Modelling

The heat sink itself is modelled numerically with a curvilinear form of the discretised diffusion equation. In Cartesian co-ordinates the diffusion equation is:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) + S \quad (3)$$

The reason for using a numerical simulation is that the temperature variation through the heat sink can be accurately determined with the aim of calculating the maximum temperature (T_{HS}) for the definition of the thermal resistance (R_{HS}). This method prevents any ambiguity regarding the size and location of a device and hence any confusion regarding the formal definition of what the thermal performance of the heat sink actually is. On this note, it is worth pointing out that technically, the thermal resistance cannot be specified for a heat sink alone. The size, location and number of devices have an effect on the thermal resistance because of the finite thermal conductivity of the heat sink material. This method therefore gives an *application dependent* heat sink thermal resistance.

The boundary conditions are specified using compact analytical and empirical models. Van de Pol and Tierney [9] developed an empirical correlation for vertical U-channel geometries exposed to natural convection and is given as:

$$Nu_r = \frac{Ra^*}{\psi} \left[1 - \exp \left[-\psi \left(\frac{0.5}{Ra^*} \right)^{\frac{3}{4}} \right] \right] \quad (4)$$

where

$$\psi = \frac{24(1 - 0.483e^{-0.17/ar})}{\left[(1 + ar/2) \left[1 + (1 - e^{-0.83ar}) (9.14ar^{1/2} e^{V_{Lsh}} - 0.61) \right] \right]^3} \quad (5)$$

The forced convection heat transfer coefficient between the heat sink and the surrounding air is calculated by using the analytical formulation:

$$\overline{h_{ca}} = CF \times 3.455 \sqrt{\frac{\rho V_h}{L}} \quad (6)$$

where CF is a modification factor used to account for turbulence caused by the fins.

An analytical model proposed by Butterbaugh and Kang [10] was adapted and used to determine the flow velocities around a heat sink. For a given approach velocity and known duct dimensions, the airflow rate, Q is used to determine the flow velocity in the finned region by applying pressure balance and mass conservation conditions. Each flow path has to conform to a pressure drop (ΔP) balance and mass conservation laws so that

$$\Delta P_h = \Delta P_t = \Delta P_l \text{ and } Q_h + Q_t + Q_l = Q \quad (7)$$

This is achieved by estimating the initial flow rates and then calculating the sum of each individual pressure drop mechanism in each flow path. The pressure drop in each flow path is then used to redefine the flow rates in each path and the process is repeated iteratively until both equations are satisfied. The flow leakage through the fin tips causes a longitudinal velocity gradient. The fin flow velocity decreases along the length of the heat sink and the effect is more pronounced with heat sinks more densely packed with fins.

3.2 Mathematical optimisation

The optimisation method used in this study is the DYNAMIC-Q method [8]. This approach involves the application of a dynamic trajectory method for unconstrained optimisation [11], adapted to handle constrained problems through appropriate penalty function formulations [8]. This DYNAMIC method is applied to successive approximate Quadratic sub-problems [8] of the original problem. The successive sub-problems are constructed from sampling, at relative high computational expense, the behaviour of the constraint functions or objective functions at successive

approximate solution points in the design space. The sub-problems, which are analytically simple, are solved quickly and economically using the adapted dynamic trajectory method. A brief outline of the DYNAMIC-Q methodology now follows [7].

Consider the typical inequality constrained optimisation problem of the following form:

$$\text{Minimise } f(\mathbf{x}), \mathbf{x} \in R^n \text{ subject to the inequality constraints} \quad (8)$$

$$g_j(\mathbf{x}) \leq 0 \quad j=1,2,\dots,m \text{ and equality constraints} \quad (9)$$

$$h_k(\mathbf{x}) = 0 \quad k=1,2,\dots,r \quad (10)$$

An initial trial design $\mathbf{x}^{(1)}$ is available, and the solution is denoted by $\mathbf{x}^*$.

The penalty function used is defined by

$$p(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^m \alpha_j g_j^2(\mathbf{x}) + \sum_{k=1}^r \beta_k h_k^2(\mathbf{x}) \quad (11)$$

$$\text{where } \alpha_j = \begin{cases} 0 & \text{if } g_j(\mathbf{x}) \leq 0 \\ \rho_j & \text{if } g_j(\mathbf{x}) > 0 \end{cases} \text{ and } \beta_k = \text{large positive number}$$

For simplicity the penalty parameters, $\rho_j, j=1,2,\dots,m$ and $\beta_k, k=1,2,\dots,r$, take on some positive value, $\rho_j = \beta_k = \mu$. The unconstrained minimum of $p(\mathbf{x})$ tends to the constrained minimum of problem (equation (8)-(11)) as μ tends to infinity. In the application of the dynamic trajectory method used here, and with the objective and gradient functions appropriately scaled, the penalty parameter μ is introduced at a certain specified value, here $\mu = 10^2$, and then increased to $\mu = 10^4$ when the intersection of active constraints is found. The dynamic trajectory method is applied to approximate sub-problems as follows.

Successive approximate quadratic sub-problems, $P[l] : l=1,2,\dots$, are formed at successive design points $\mathbf{x}^{(l)}$. The inequality constraint approximation is discussed in the following, but an approximation for the objective function constraint is obtained in a similar fashion. The approximation $\tilde{g}_j(\mathbf{x})$ to $g_j(\mathbf{x})$ is given by

$$\begin{aligned} \tilde{g}_j(\mathbf{x}) = & g_j(\mathbf{x}^{(l)}) + \nabla^T g_j(\mathbf{x}^{(l)})(\mathbf{x} - \mathbf{x}^{(l)}) \\ & + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(l)})^T C_j^{(l)}(\mathbf{x} - \mathbf{x}^{(l)}) \quad \text{for } j=1,2,\dots,m \end{aligned} \quad (12)$$

where ∇g_j denotes the gradient vector. The approximate Hessian matrix is given by

$$C_j^{(l)} = \text{diag}(c_j^l, c_j^l, \dots, c_j^l) = c_j^l I \quad (13)$$

The initial value $c_j^{(l)}$ depends on the specific problem being considered. Here a value of 0.001 was arbitrarily used for the first sub-problem. Thereafter the $c_j^{(l)}$ are calculated using the expression:

$$c_j^{(l)} = \frac{2\{g_j(\mathbf{x}^{(l-1)}) - g_j(\mathbf{x}^{(l)}) - \nabla^T g_j(\mathbf{x}^{(l)})(\mathbf{x}^{(l-1)} - \mathbf{x}^{(l)})\}}{\|\mathbf{x}^{(l-1)} - \mathbf{x}^{(l)}\|^2} \quad (14)$$

As a further aid in controlling convergence, intermediate move limits are imposed on the design variables during the minimisation of the subproblem. These constraints are described by

$$x_i - x_i^{(l)} - \delta_i \leq 0 \text{ and } -x_i + x_i^{(l)} - \delta_i \leq 0 \quad ; \quad i = 1, 2, \dots, n \quad (15)$$

where δ_i is a user-specified move limit for each variable.

The gradient vector of the last constraint in (1) at a specific design point $\mathbf{x}$, with respect to each of the design variables x_i , is approximated by the first-order forward differencing scheme

$$\frac{\partial g(\mathbf{x})}{\partial x_i} \approx \frac{g(\mathbf{x} + \Delta \mathbf{x}_i) - g(\mathbf{x})}{\Delta x_i} \quad ; \quad i = 1, 2, \dots, n \quad (16)$$

where $\Delta \mathbf{x}_i = [0, 0, \dots, \Delta x_i, \dots, 0]^T$, Δx_i is a suitable step size determined from a sensitivity study. It is clear that a maximum of $n+1$ numerical analyses are required at each design point $\mathbf{x}$ to determine the constraint gradient vectors. The successive simple quadratic sub-problems are solved economically using the trajectory method [8] referred to above.

4 Results and Discussion

The method outlined above are now applied to a typical extruded heat sink exposed to different boundary conditions. The heat sink consists of a 52×50×3mm base with 9 fins on top of the base. The fins are 1.59mm thick, 23mm high and 50mm in length. The heat sink is made out of aluminium with a conductivity of 225.94W/m °C, a density of 2698kg/m³ and a specific heat of 920J/kg °C. The heat source is a 25×25mm square component. This heat sink will now be exposed to different boundary conditions in the three different case studies.

4.1 Case 1: Mass Minimisation with natural convection (3 design variables)

In the first case the heat sink mentioned above is exposed to natural convection with the fins upwards and the environment at 25°C. In this case the heat source is dissipating a constant flux of 10W. When solved without any optimisation this set-up weighs 66g, has a thermal resistance of 5.2°C/W with a maximum temperature of 77°C on the heat sink.

In the first optimisation case, the heat sink is optimised to obtain the lowest possible mass while not exceeding 78°C (equation (1)). During this study the optimising algorithm is allowed to change three design variables simultaneously, i.e. the height of the fins, the thickness of the fins and the number of fins. As outlined previously the optimisation algorithm requires minimum and maximum constraints for each variable. These constraints are given in Table 1.

Table 1: Constraints used for Case 1

	Minimum	Maximum
Fin thickness	1 mm	10 mm
Fin height	1 mm	100 mm
Number of fins	2	12

The change in the heat sink mass and maximum heat sink temperature are shown in Figure 2. The heat sink mass starts with a value of 66g and after 8 iterations the mass declines to 40g. In the second iteration the optimiser predicts a very small heat sink resulting in a very high temperature. The optimiser are however able to recover and converges in 8 iterations. The thermal resistance increased by a small amount from 5.2°C/W to 5.3°C/W.

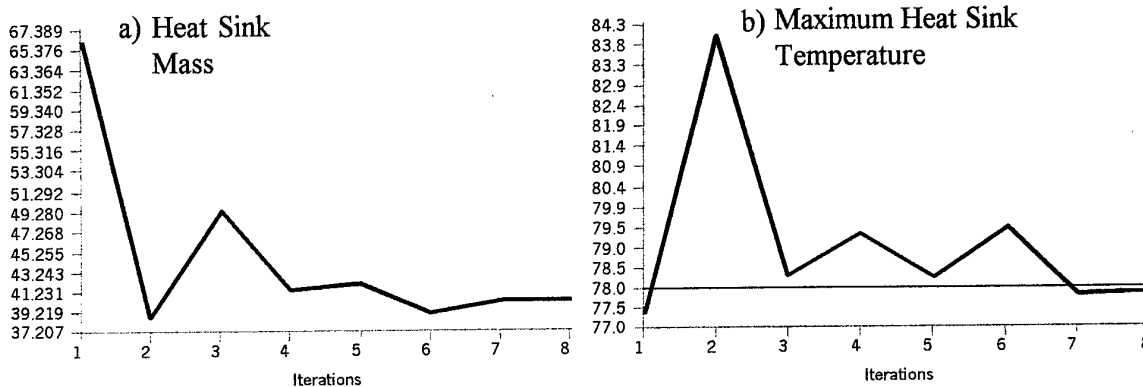


Figure 2: a) Optimisation history of heat sink mass and
b) Maximum temperature on heat sink (Case 1)

Table 2: Initial and optimum heat sink configuration (Case 1)

	Initial	Optimum
Fin thickness	1.59 mm	1.00 mm
Fin height	22.81 mm	26.82 mm
Number of fins	9	5
Base thickness	3.11 mm	3.11 mm
Extrusion length	50 mm	50 mm
Thermal resistance	5.24 °C/W	5.28 °C/W
Heat sink mass	66.0 g	40.1 g
Maximum temperature	77.4 °C	77.8 °C

The initial and optimum heat sink configuration are summarised in Table 2. As expected the fin thickness was reduced to its minimum allowable value while the fin height and number of fins increased. The impressive part of the optimisation is that the heat sink mass is reduced by 39% without a major increase in the temperature of the heat sink.

4.2 Case 2: Mass Minimisation with natural convection (5 design variables)

During the optimisation carried out in Case 1, only three design variables were allowed to change. In this case two additional variables are allowed to change, i.e. base thickness and extrusion length. All the other variables were kept the same and the new minimum and maximum constraints are shown in Table 3.

Table 3: Additional constraints used for Case 2

	Minimum	Maximum
Base thickness	1 mm	10 mm
Extrusion length	30 mm	100 mm

The optimiser again converged after 8 iterations. A summary of the initial and optimum heat sink is given in Table 4. It can be seen again, that the fin thickness is reduced to its minimum allowed value.

The base thickness is also reduced to its minimum allowable value while the extrusion length is reduced slightly.

Table 4: Initial and optimum heat sink configuration (Case 2)

	Initial	Optimum
Fin thickness	1.59 mm	1.00 mm
Fin height	22.81 mm	28.71 mm
Number of fins	9	6
Base thickness	3.11 mm	1.00 mm
Extrusion length	50 mm	46.06 mm
Thermal resistance	5.24 °C/W	5.28 °C/W
Heat sink mass	66.0 g	27.9 g
Maximum temperature	77.4 °C	77.8 °C

The heat sink mass in this case reduced by an impressive 58%. It must be noted that the optimum heat sink mass is much lower than in Case 1. This is due to the fact that more design variables are allowed to vary, resulting in a better optimum.

4.3 Case 3: Mass minimisation with forced convection (5 design variables)

In the third case, the same heat sink is now exposed to forced convection. The heat sink is placed in an 80×60mm tunnel with an approach velocity of 1m/s. The heat dissipated by the source is increased to 25W. The constraints were kept the same as for Case 2.

The optimiser converged in 10 iterations. The initial and optimum heat sink configuration for this case are shown in Table 5. The minimum constraints on the fin and base thickness were again reached. The optimum heat sink, compared to that of the optimum heat sink in Case 2, has longer fins, a longer extrusion length and one more fin. The reduction in heat sink mass is however still a very high 44%.

Table 5: Initial and optimum heat sink configuration (Case 3)

	Initial	Optimum
Fin thickness	1.59 mm	1.00 mm
Fin height	22.81 mm	29.17 mm
Number of fins	9	7
Base thickness	3.11 mm	1.00 mm
Extrusion length	50 mm	53.1 mm
Thermal resistance	2.06 °C/W	2.09 °C/W
Heat sink mass	66.0 g	36.8 g
Maximum temperature	76.5 °C	77.2 °C

4.4 Case4: Thermal resistance minimisation with forced convection (5 design variables)

In the last case, the thermal resistance is minimised with a constraint on the maximum mass of the heat sink (equation (2)). The same forced convection case as in Case 3 was used with a heat source of 30W and a specified maximum mass of 80g.

The optimised converged in 9 iterations. The initial and final design for this case are shown in Table 6. From the table it can be seen that the thermal resistance is reduced from 1.986 °C/W to 1.080 °C/W which is an impressive 45.6% reduction. This is achieved by increasing the fin length, number of fins and the extrusion length. It can also be seen that the constraint on the maximum heat sink mass is met.

Table 6: Initial and optimum heat sink configuration (Case 4)

	Initial	Optimum
Fin thickness	1.59 mm	1.00 mm
Fin height	22.81 mm	38.10 mm
Number of fins	9	11
Base thickness	3.11 mm	2.325 mm
Extrusion length	50 mm	54.82 mm
Thermal resistance	1.986 °C/W	1.080 °C/W
Heat sink mass	66.0 g	88.0 g
Maximum temperature	84.45 °C	57.41 °C

5 Conclusions

This paper illustrated the application of a semi-empirical simulation and mathematical optimisation to the field of heat sink design. A simple heat sink geometry is considered, but the techniques can readily be extended to more complex electronics cooling geometry. The effectiveness of the DYNAMIC-Q optimisation method has been illustrated as a robust optimisation tool in conjunction with the Qfin 2.1 analysis code.

6 References

- [1] Kim, S.K., & Lee S. On Heat Sink Measurement and Characterisation. Proceedings of the Pacific Rim/ASME International Intersociety Electronic & Photonic Packaging Conference (INTERpack '97), Hawaii, June 15-19, 1997.
- [2] Biber, C.R. & Belady, C.L. Pressure Drop Prediction for Heat Sinks: What is the Best Method? *Proceedings of the Pacific Rim/ASME International Intersociety Electronic & Photonic Packaging Conference (INTERPACK '97)*, Hawaii, June 15-19, 1997.
- [3] Knight R.W., Goodling, J.S. & Hall, D.J. Optimal Thermal Design of Forced Convection Heat Sinks - Analytical. *Journal of Electronic Packaging*, **113**, pp. 313-32, 1 September 1991.
- [4] Knight, R.W., Goodling, J.S., & Gross, B.E. Optimal Thermal Design of Air Cooled Forced Convection Finned Heat Sinks - Experimental Verification. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, **15(5)**, pp. 754-760, October 1992.

- [5] Lee, S. Optimum Design and Selection of Heat Sinks. *Proceedings of the 11th Annual IEEE Semi-Therm Symposium*, pp. 48-52, 1995.
- [6] Obinelo, I.F. Characterisation of Thermal and Hydraulic Performance of Longitudinal Fin Heat Sinks for System Level Modeling Using CFD Methods. *Proceedings of the Pacific Rim/ASME International Intersociety Electronic & Photonic Packaging Conference (INTERPACK '97)*, Hawaii, June 15-19, 1997.
- [7] Craig, K.J., De Kock, D.J., & Gauche, P. Optimization of Heat Sink Mass using CFD and Mathematical Optimization. *ASME Journal of Electronic Packaging*, **121**(3), pp 143-147, 1999.
- [8] Snyman, J.A., Stander, N., & Roux, W.J. A dynamic penalty function method for the solution of structural optimization problems. *Appl. Math. Modelling*, **18**, pp. 453-460, 1994.
- [9] Van de Pol, D.W., & Tierney, J.K. Free convection Heat Transfer from Vertical Fin Arrays. *IEEE Transactions on Parts, Hybrids, and Packaging*, **(10)**4, pp. 267-271, 1974.
- [10] Butterbaugh, M.A., and Kang, S.S. Effect of Airflow Bypass on the Performance of Heat Sinks in Electronic Cooling. *Advances in Electronic Packaging*, **(10)**2, ASME 1995
- [11] Snyman, J.A. An improved version of the original leap-frog dynamic method for unconstrained minimization LFOP1(b), *Appl. Math. Modelling*, **7**, pp. 216-218, 1983.

AUTHOR INDEX

- Akkerman A., 270
Ali M.M., 26
Baker C.A., 50
Beaudoin A.J., 210
Bolton H.P.J., 134
Brusa E., 124
Bühmann T., 37
Burger M., 270
Cox S.E., 50
Craig K.J., 60, 70
De Kock D.J., 60, 70, 289
Du Plessis L.J., 79
Farkas J., 89
Fourie P.C., 97
Frangos C., 107
Genta G., 124
Glasser D., 144
Godorr S., 144
Groenwold A.A. 97, 134
Grossman B., 50
Haftka R.T., 50
Haug E.J., 16
Hausberger B., 144
Hay A.M., 79, 163
Heyns P.S., 173
Hildebrandt D., 144
Holm J.E.W., 181
Jármai K., 192
Jordaan J.A., 202
Kessels P.H.L., 230
Kok S., 210
Kuhn R., 270
Kuijpers A.H.W.M., 230
Mason W.H., 50
McGregor C., 144
Michelena N.F., 1
Naudé A.F., 220
Papalambros P.Y., 1
Rajic H., 270
Schoofs A.J.G., 230
Schutte J.F., 134
Schweiger J., 240
Sensburg O., 240
Serban R., 16
Smit W.J., 254
Snyman J.A., 163, 220, 264
Stander N., 270
Thyagarajan R., 270
Tischler V., 240
Tortorelli D.A., 210
Van Houten M.H., 230
Van Wyk B.J., 280
Van Wyk M.A., 280
Venkayya V.B., 240
Visser J.A., 289
Watson L.T., 50
Yavin Y., 107
Zivanovic R., 202

AD NUMBER	DATE	DTIC ACCESSION NOTICE
1. REPORT IDENTIFYING INFORMATION A. ORIGINATING AGENCY University of Pretoria, South Africa B. REPORT TITLE AND/OR NUMBER Internet: wksp. on Multichip processors Design optimization C. MONITOR REPORT NUMBER KR D8957-AN-03 D. PREPARED UNDER CONTRACT NUMBER N68171-00-M-5704		20010508 105 REQUE 1. Put you on review 2. Complete 3. Attach 1 mailed 4. Use uninform 5. Do not for 6 h DIIC: 1. Assign 2. Return
2. DISTRIBUTION STATEMENT APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED PROCEEDINGS		

DTIC ACCESSION NOTICE ARE OBSOLETE

DTIC OCT 95