WL-TR-96-3099

APPLICATION OF MULTIVARIABLE CONTROL THEORY TO AIRCRAFT CONTROL LAWS

FINAL REPORT: MULTIVARIABLE CONTROL DESIGN GUIDELINES

HONEYWELL TECHNOLOGY CENTER
3660 TECHNOLOGY DRIVE
MINNEAPOLIS MN 55418

LOCKHEED MARTIN SKUNK WORKS
1011 LOCKHEED WAY
PALMDALE CA 93599

LOCKHEED MARTIN TACTICAL AIRCRAFT SYSTEMS
1 LOCKHEED BOULEVARD
FORT WORTH TX 76108

MAY 1996

FINAL REPORT FOR MARCH 1993 TO MARCH 1996

Approved for public release; distribution unlimited

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7562

NOTICE

WHEN GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA ARE USED FOR ANY PURPOSE OTHER THAN IN CONNECTION WITH A DEFINITE GOVERNMENT-RELATED PROCUREMENT, THE UNITED STATES GOVERNMENT INCURS NO RESPONSIBILITY OR ANY OBLIGATION WHATSOEVER. THE FACT THAT THE GOVERNMENT MAY HAVE FORMULATED OR IN ANY WAY SUPPLIED THE SAID DRAWINGS, SPECIFICATIONS, OR OTHER DATA, IS NOT TO BE REGARDED BY IMPLICATION, OR OTHERWISE IN ANY MANNER CONSTRUED, AS LICENSING THE HOLDER, OR ANY OTHER PERSON OR CORPORATION; OR AS CONVEYING ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY IN ANY WAY BE RELATED THERETO.

THIS REPORT IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

WILLIAM C. REIGELSPERGER, Capt, USAF
Stability and Control Engineer
Control Dynamics Branch

SIVA S. BANDA, Chief
Control Dynamics Branch
Flight Control Division

DAVID P. LEMASTER
Chief, Flight Control Division
Flight Dynamics Directorate

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION PLEASE NOTIFY WL/FIGC-3 WRIGHT-PATTERSON AFB OH 45433-7531 TO HELP MAINTAIN A CURRENT MAILING LIST.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	MAY 1996	FINAL REPORT - MAR 93 - MAR 96

4. TITLE AND SUBTITLE
APPLICATION OF MULTIVARIABLE CONTROL THEORY TO AIRCRAFT CONTROL LAWS FINAL REPORT - MULTIVARIABLE CONTROL DESIGN GUIDELINES

5. FUNDING NUMBERS
C: F33615-92-C-3607
PE: 62201F
PR: 2403
TA: 05
WU: 94

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
HONEYWELL TC
3660 TECHNOLOGY DRIVE
MINNEAPOLIS MN 55418

LOCKHEED MARTIN SKUNK WORKS
1011 LOCKHEED WAY
PALMDALE CA 93599

AND

LOCKHEED MARTIN TAS, 1 LOCKHEED BLVD, FORT WORTH TX 76108

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
CAPT WILLIAM C. REIGELSPERGER (513) 255-8678
FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB OH 45433-7531

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-96-3099

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

THE RESEARCH COVERED IN THIS REPORT IS A RESULT OF THE TASKS IN THE CONTROL APPLICATION OF MULTIVARIABLE CONTROL THEORY TO AIRCRAFT CONTROL LAWS. THE REPORT IS WRITTEN TO PROVIDE GUIDELINES TO HELP CONTROL LAW DESIGNERS USE MULTIVARIABLE CONTROL THEORY FOR AIRCRAFT CONTROLS. THE REPORT IS ORGANIZED INTO THREE PARTS. THE FIRST INCLUDES DISCUSSIONS ON VARIOUS CONTROL METHODOLOGIES AND PRESENTATION OF FUNDAMENTALS OF CONTROL. THE SECOND DISCUSSES THE DESIGN AND ANALYSIS OF CONTROLS AS IT RELATES TO FIGHTER AIRCRAFT. FINALLY, THE THIRD SECTION GOES THROUGH THE SPECIFICS OF THREE DESIGN METHODOLOGIES USING FIGHTER AIRCRAFT FOR EXAMPLES.

14. SUBJECT TERMS
MULTIVARIABLE CONTROL, FLIGHT CONTROL, DESIGN REQUIREMENTS, UNCERTAINTY MODELING, AIRCRAFT SYSTEM MODELING, FLYING QUALITIES, DESIGN GUIDELINES

15. NUMBER OF PAGES
267

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Foreword

These guidelines were prepared by the Honeywell Technology Center, Minneapolis, Minnesota; Lockheed Martin Tactical Aircraft Systems, Ft. Worth, Texas; and Lockheed Martin Skunk Works, Palmdale, California under Air Force Contract F33615-92-C-3607. The work was performed between March 1993 and July 1996.

The work was performed under the direction of Siva Banda, John Bowlus, and four contract monitors from start to finish throughout the program, Randy Robinson, Rick Adams, Kelly Hammett, and William Reigelsperger, all from Wright Laboratory (WL/FIGC).

Contributors from Honeywell included Mike Barrett, Paul Blue, Dan Bugajski, Dale Enns, Russ Hendrick, Mike Jackson, Blaise Morton, and Gunter Stein.

Contributors from Lockheed Martin Tactical Aircraft Systems were John Bessollo, John Virnig, and Greg Walker.

Contributors from Lockheed Martin Skunk Works were Richard Colgren and Pat Tait.

A consultant, Roy Smith, from the University of California Santa Barbara also made a contribution.

Contents

List of Figures

List of Tables

Guidelines Overview

These guidelines are organized into three parts.

The first part, which sets the stage for the other parts of the guidelines, includes a discussion of various multivariable flight control methodologies with a rationale for choosing three to be highlighted in Part III. This is followed by a presentation of fundamentals of control and some design basics.

The second part begins to focus on fighter aircraft and the issues associated with control design and analysis for these particular vehicles.

Finally, the third part discusses the specifics of three different control design methodologies using fighter aircraft examples for demonstration.

Part I

Fundamentals

Part I Overview

This part of the guidelines presents the fundamentals for understanding and solving multivariable control problems.

The first section gives an overview of multivariable control and the rationale for selecting the three control methods described in Part III.

The second section discusses control fundamentals, and the third section covers the design process itself.

Section 1
Introduction

Flight control laws are necessary for the military fighter aircraft of the present and future. In some cases, the control laws stabilize or augment the stability of configurations that would otherwise be aerodynamically unstable or nearly unstable. In other cases, the control laws efficiently utilize multiple control actuation devices, including aerodynamic surfaces and thrust vectoring, to achieve the highest levels of aircraft performance. In every case, the control laws augment aircraft handling and ride qualities to achieve satisfactory handling characteristics.

Control laws are algorithms that process sensor information (including pilot commands) and produce control actuator commands to achieve the desired responses even in the presence of external disturbances and inaccuracies of the data input to, and needed to design, the algorithm.

A challenge of flight control laws is their multivariable nature:

- Multiple control surfaces and thrust vectoring devices;
- Multiple sensors, including angular rate gyros, accelerometers, air data, and inertial navigation;
- Multiple disturbances, including atmospheric, weapons release, changes in mass properties, and sensor errors;
- Multiple controls objectives for longitudinal, lateral, and directional maneuvering subject to constraints associated with the aircraft and its control devices;
- Multiple uncertainties associated with the math models used for synthesis and analysis of the control laws.

Multivariable flight control laws can be synthesized using a variety of techniques. Needless to say, there are many candidate multivariable design methods. Indeed, a certain senior member of the controls field, who would probably deny this, has claimed that "there are as many methods as there are researchers with egos big enough to choose names for their inventions."

1.1 Categorization of Design Methods

Fortunately, the facts are more modest. Most design methods are variants of a few basic approaches to multivariable design, and each method is the culmination of the work of many researchers, not the creation of a single person. The major methods have been categorized in Table 1.1. This table lists broad categories, specific methods in each category, and some of the principal references that describe them.

The categories include:

- Formal optimization problems,
- Numerical optimization problems,
- Frequency domain methods,
- Eigenstructure assignment methods, and
- Fringe methods.

The first category consists of the big workhorses of the controls field — the linear quadratic Gaussian problem (LQG, H_2) in its various manifestations, the worst-case L_2-induced norm problem (H_∞), the worst case L_∞ induced norm problem (ℓ_1), and the mixed-criteria versions of H_2 and H_∞ developed just recently. It is fair to say that the introduction of these formal methods, together with the use of state-space descriptions, provided the first truly multivariable control design tools, circa 1960.

The second category, numerical optimization, consists of more heuristic approaches to design. They utilize the same optimization philosophy but in parameterized settings that do not necessarily yield analytic solutions or find global optima or even guarantee stabilizing answers. Nevertheless, whenever solutions can be found, these approaches provide very effective tools for the complex tradeoffs present in most realistic engineering designs. (It is recognized that there is a certain overlap between the formal and numerical optimization categories. For instance, ℓ_1-problems must be solved numerically with linear programs. So where do such approaches belong? For these cases, a reasonable assignment has been made based on the spirit behind the method.)

The third category, frequency domain methods, consists of various adaptations of classical Bode and Nyquist SISO techniques to multivariable design. It includes early attempts by the "British school," using diagonal dominance, inverse Nyquist arrays, and characteristic loci. This category also includes "quantitative feedback theory," which closes sequential loops around multivariable plants, and singular-value-based loop shaping accomplished using either direct inversion (LQG/LTR or H_∞) or dynamic inversion. Although good multivariable controllers can and have been designed using all of these methods, only singular-value-based loop shaping probably qualifies as truly multivariable.

Table 1.1: Categorization of Design Methods

Formal Optimization Problems	
Basic Linear-Quadratic-Gaussian (LQG)	[Athans 1971]
LQG with Explicit/Implicit Model Following	[Asseo 1970, Tyler 1966]
Frequency Weighted LQG	[Gupta 1980]
Worst-case Induced L_2 Norm (H_∞)	[Zames 1983, Doyle 1989]
Worst-case Induced L_∞ Norm (ℓ_1)	[Dahleh 1987]
Mixed Criteria (H_2/H_∞)	[Zhou 1989, Rotea 1991, Yeh 1992]
Numerical Optimization Problems	
Fixed Structure LQ-Control	[Axsater 1966, Levine 1970, Stein 1971]
Fixed-Plus-Variable Gain LQ-Control	[VanDierendonck 1972]
Fixed Structure H_∞ Control	[Bernstein 1990]
Fixed Structure H_2/H_∞ Control	[Bernstein 1989, Ridgely 1992]
Q-Parameter Design (QDES)	[Boyd 1991]
μ-Synthesis via $D-K$-Iteration	[Doyle 1983, Stein 1991]
Frequency Domain Methods	
Diagonal Dominance/Inverse Nyquist Array	[Rosenbrock 1974]
Characteristic Loci	[Postlethwaite 1979]
Upper Triangular Structures	[Mayne 1973]
Singular-Value-Based Loop Shaping via Direct Inversion	[Hung 1982]
via LQG/LTR	[Stein 1987]
via H_∞	[McFarlane 1992]
via Dynamic Inversion	[Bugajski 1992b]
Quantitative Feedback Theory (QFT)	[Horowitz 1979]
Eigenstructure Assignment Methods	
via Full-State Feedback	[Andry 1983]
via Output Feedback	[Calvo-Ramon 1986, Sobel 1990]
via Quadratic Weights in LQ	[Harvey 1978]
via Numerical Optimization	[Garg 1989, Wilson 1990]
Fringe Methods	
Model Predictive Control (MPC, DMC, MAC)	[Morari 1989]
Covariance Control	[Skelton 1989]
Stochastic Parameters (maximum entropy)	[Hyland 1982]
Variable-Structure Control	[Utkin 1977]
Geometric Methods	[Wonham 1979]
Polynomial-Matrix Methods	[Peczkowski 1978, Wolovitch 1974]
Lyapunov-Based Methods	[Barmish 1985, Boyd 1989]

The fourth category, eigenstructure assignment, focuses on the internal structure of closed-loop systems. Its methods are concerned with locations of closed-loop eigenvalues and directions of closed-loop eigenvectors, as constrained by the limitations of linear feedback. Included are methods based on full-state feedback, output feedback, asymptotic LQ regulator solutions, as well as numerical optimization methods that trade off desired eigenstructures against other desired properties. (Again, there is a certain overlap here with the numerical optimization category.)

The final category carries the name "fringe methods." This label is not intended in a derogatory sense. Rather, it is chosen to communicate the fact that these methods exist at the fringes of the flight control world, some used extensively in other arenas but rarely tried in flight control. Examples include the model-predictive-control techniques from process control, covariance assignment and maximum-entropy-optimal-projection methods from large space structure control, and variable-structure methods from robotics.

1.2 Evaluation Criteria

Design techniques will be selected based on an evaluation of how well each method handles the specific requirements of fighter aircraft flight control design. They lead us to the following list of evaluation criteria:

- Nominal stability guarantees;
- Robust stability guarantees (for unstructured uncertainties, for real parameter variations);
- Suitability for achieving nominal performance (decoupling/handling qualities, gust and noise responses);
- Suitability for achieving robust performance (for unstructured uncertainties, for real parameter variations);
- Controller complexity (state dimension, scheduling difficulty, iteration difficulty);
- Suitability for satisfying practical considerations (actuator limits, elastic interactions, antiwindup/boundary control logic, necessary nonlinear terms, classical stability margins).

Note that these criteria include the usual nominal and robust stability and performance topics. But they also address controller complexity and so-called "practical considerations."

Complexity is important because it influences design costs and eventual flight computer hardware needs. It is measured not only by controller order, but also by the ease with

which controllers can be scheduled over flight envelopes and the ease with which they can be iterated (redesigned) when inevitable mission/configuration changes occur during an aircraft's overall design program.

Practical considerations often tip the scales between success and failure. For example, design methods should provide direct ways to change inner-loop bandwidths, as this is the only practical remedy for limit cycles due to rate or position limits of actuators. Also, there should be ways to introduce notch-filtering actions to alleviate excessive elastic interactions. Integral actions or boundary limits must be easy to identify in the controller structure, so that appropriate antiwindup logic and boundary switches can be added to the design. There are also certain nonlinear modifications for large angles (e.g., feedback signal $g\phi$ replaced by $g \sin\phi \cos\theta$) that must be included in linear designs to make them flightworthy, even on piloted simulators. (Pilots are sure to try barrel rolls, inducements to the contrary notwithstanding.)

Finally, even for multivariable systems, control designs must satisfy traditional classical SISO stability margins since detailed characteristics of individual actuators or sensors may change before the aircraft's design is complete. Controllers should tolerate reasonable loop variations due to such changes.

1.3 Evaluation of Design Methods

Table 1.2 provides an evaluation of some of the major design methods from Table 1.1, as measured against the criteria described above. Each row of the table corresponds to an evaluation criterion and each column corresponds to one of the design methods. The table entries themselves contain short statements about how well the method addresses the criterion or brief descriptions of the explicit way in which the method's design parameters can be used to satisfy the criterion. If no explicit way exists, the entries are empty, or they contain the symbol *E*. The latter indicates that an easy work-around exists to satisfy the criterion, even though there is no explicit way to do so. (It is recognized that the term *easy* represents a judgment call about which reasonable people may disagree.)

To illustrate this evaluation format, consider column 1 in the table, which corresponds to the linear-quadratic-Gaussian (LQG) design method. It shows that this method guarantees nominal stability, but provides no direct way to guarantee robust stability for real or unstructured uncertainties (recall the famous Doyle counter example [Doyle 1978].) For nominal performance, model-following performance indices can address decoupling/handling quality requirements, while the quadratic criterion (H_2 norm) itself can address gust/noise response requirements. However, there is no direct way to address robust performance. The controller's complexity is n states, with moderate scheduling difficulty and low iteration difficulty via two standard Riccati equations. Finally, although there is no direct way to address practical actuator limit considerations, an easy work-around exists — namely, adjusting bandwidths with the control weighting

Table 1.2: Evaluation of Major Design Methods.

	LQG	Eigen Structure Assignment	Singular-Value-Based Loop Shaping		H_∞	μ-Synthesis	Mixed H_2/H_∞
			LQG/LTR	Dyn Inv			
Nominal Stability	guaranteed	E [full state]	guaranteed	E [min phase]	guaranteed	guaranteed	guaranteed
Robust Stability for Unstructured Δ's			via loop shapes	via loop shapes	H_∞ norm	H_∞ norm	H_∞ norm
Real parameter variations							
Nominal Performance HQ/Decoupling	via model following	via poles and eigenvectors	via model following	via model following	via model following	via model following	via model following
Gust/noise response	H_2 norm		H_2 norm				H_2 norm
Robust Performance for Unstructured Δ's						μ "norm"	
Real parameter variations							
Controller Complexity Compensator dimensions	n	$\leq n$	n	$\leq n$	n	$\geq n$	n
Scheduling difficulty	moderate	moderate	moderate	low	moderate	high	moderate
Iteration difficulty	low	low	moderate	low	moderate	high	moderate
Practical Considerations Actuation limits	E [R,θ]	E [poles]	E [loop shapes]	E [loop shapes]	E [$W\Delta$]	E [$W\Delta$]	E [$W\Delta$]
Elastic interactions	E [$W(j\omega)$]		E [loop shapes]	E [loop shapes]	E [$W\Delta$]	E [$W\Delta$]	E [$W\Delta$]
antiwindup logic, etc	E	E	E	E	E		
Necessary nonlinearities				automatic			
Classical SISO margins			via loop shapes	via loop shapes	H_∞ norm	H_∞ norm	H_∞ norm

and sensor noise intensity matrices, R and θ. Similarly, an easy way exists to alleviate elastic interactions with frequency-dependent weights used to introduce notch filters. Appended integrators explicitly exhibit integration action, so adding antiwindup logic is also easy. However, there is no direct way to include other necessary nonlinearities or to satisfy standard SISO stability margins (Doyle's counter example again).

Similar evaluations for other major design methods appear in the remaining columns of Table 1.2. The columns were arranged in rough chronological order to highlight the historical evolution of multivariable methods toward increasingly more powerful capabilities for practical design. For instance, the empty cells in column 1 clearly show that the basic LQG method has significant shortcomings in its robustness capabilities for both stability and performance. The eigenstructure assignment method in column 2 shares these same weaknesses, and, in addition, lacks direct ways or easy work-arounds for gust/noise response requirements and elastic interactions.

Columns 3 and 4 illustrate the first attempts in the early 1970s to alleviate the robustness shortcomings of LQG and other then-existing design methods. Such attempts include introducing singular-value diagrams as true multivariable generalizations of Bode diagrams [Doyle 1981]. Applied to loop transfer functions of feedback systems, these singular-value diagrams provide reliable measures for nominal performance and robust stability guarantees against unstructured uncertainties. The design problem then reduces to specifying desirable shapes for the diagrams and finding controllers to produce those shapes. The latter step can be accomplished in one of several ways:

Direct Inversion	$K(s)$	$=$	$L(s)G^{-1}(s)$
LQG/LTR	$K(s)$	$\cong$	$L(s)G^{-1}(s)$ but retains nice LQ properties
H_∞	$K(s)$	$\cong$	$L(s)G^{-1}(s)$ but retains nice H_∞ properties
Dynamic Inversion	$u(t)$	$=$	$-B^\dagger Ax(t) + L\dot{x}(t)$ (A, B in Brunovski form)

(1.1)

Of these, the LQG/LTR and dynamic inversion approaches are compared in the table. Robust stability for unstructured Δ's is now handled directly in both cases via the selected loop shapes. But the two approaches differ in other ways. The most important differences appear in scheduling difficulty and in adding necessary nonlinearities. For both these criteria, dynamic inversion is superior because it handles them simply by replacing the dynamic inversion terms, $B^\dagger Ax(t)$, with their nonlinear counterparts.

Column 5 of Table 1.2 represents the next step in the evolution of multivariable design. Having accepted singular-value-based robustness measures, it became evident to researchers that formal optimization problems could be posed to maximize these measures. After all, the maximum value across frequency of singular-value plots corresponds to the induced L_2-norm (the maximum gain) of the associated system, and this, in turn, corresponds to the H_∞ norm of its transfer function. Thus, a frequency-weighted H_∞ optimization problem can be used to achieve robust multivariable designs. The fact that such problems have nice state space solutions (just two iterated Riccati equations) counts as one of the major theoretical discoveries of the controls field [Doyle 1989]. For serious designers, however, H_∞ per se is not such a big deal. Comparing columns 3 and 5 shows that the new theory actually provides no more design power than the basic loop-shaping paradigm.

The μ-synthesis method, highlighted in column 6, generalizes the H_∞ norm to the structured singular value μ, which provides robust stability with respect to multiple unstructured uncertainty blocks [Doyle 1982a] and robust performance whenever performance is measured in the H_∞ norm. (The trick is to require stability with respect to the given uncertainty blocks plus an additional "fictitious" block for the performance variables.)

Arguably μ-synthesis provides the most powerful methodology available for multivariable design today. The drawback, of course, is that high controller state dimensions can potentially result from multiple $D-K$ iterations. This increases hardware requirements and scheduling difficulties, or it necessitates a model reduction step as part of the design cycle. In either case, iteration difficulties are greater. The tradeoff between raw design power and resulting complexity is important.

The last method listed in Table 1.2 is an alternative generalization of the basic H_∞ problem. This generalization seeks to combine H_∞ norm optimization, which is useful for robust stability, with quadratic (H_2 norm) optimization, which is useful for nominal stochastic performance requirements such as gust and noise responses. (Note in the table that neither H_∞ nor μ-synthesis address gust/noise requirements directly.) Useful design equations for these "mixed criteria" problems have been published over the last few years.

Their utility for flight control design has been explored in a few cases to date and shows good promise [Sparks 1990].

1.4 Unsolved Problems

A few words must be said about a completely empty row in Table 1.2 — namely, the row that requires robustness with respect to real parameter variations. No multivariable design method available today provides a direct way to address this requirement. Moreover, it would be misleading to claim that there is even an easy work-around for this requirement in any method. The only reliable way to design for robustness with respect to real variations appears to be to cover them with a larger set of complex uncertainties and then to use one of the available methods to provide robustness with respect to the larger set. (Lyapunov-based methods [Barmish 1985], Peterson-Hollot bounds [Peterson 1986], surrogate systems with extra disturbances [Banda 1991], and various other schemes all fall into this category.) This covering approach has worked well in many design applications, and it is tempting to claim it as an easy work-around. However, there are no guarantees that it will succeed in any specific application, and there are numerous counter examples to show that it can be arbitrarily conservative (i.e., no robust controllers exist for the larger set, even when good ones exist for the true variations).

Given that the synthesis problem for real parameter uncertainty is unsolved, it is necessary for designers to come up with approximations. The approximations will be iterative ones, basically the same for all methods. Real variations will be replaced with complex uncertainty sets whose sizes are initially chosen to cover the real variations but are then iteratively reduced until the true real-parameter robustness of controllers fails to meet spec.

True real-parameter robustness can be analyzed numerically with mixed-μ algorithms that are currently available. Mixed-μ refers to the presence of both real variations and complex ones. The real variations correspond to parametric uncertainties, whereas the complex ones correspond to unmodeled dynamics and to "fictitious" robust performance blocks. Analysis algorithms for mixed-μ setups are currently experiencing rapid research progress. Some alternatives include: determinantal methods [deGaston 1988], power methods [Packard 1988], polynomial methods [Chang 1991], branch-and-bound searches [DeMarco 1990], and other more sophisticated combinations [Young 1992a, Boyd 1992]. (Unfortunately, many other Kharitanov-inspired methods do not apply because aircraft parameterizations lack the needed special structure.)

These numerical tools for mixed-μ have proved to be effective in many flight control problems, and no practical difficulties are expected. Still, it is important to remember that analysis problems for real-parameter variations are fundamentally hard (e.g., NP-complete and not necessarily solvable in polynomial time). There are even issues about the continuity of design objectives on real parameter sets. Stability, for example, is not

always continuous on such sets [Packard 1993]. Such parameter sets can and do contain isolated unstable points [Ackerman 1992]. For these unusual cases, there is no choice but to expand the uncertainty set with small amounts of complex variation to get well-defined analysis problems.

1.5 Selection of Three Design Methods

It is beyond the scope of these design guidelines to cover all of the design methods in Table 1.1, so only three methods will be covered. The techniques to be covered are:

- Eigenstructure snthesis.
- Dynamic inversion, and
- μ-synthesis.

The rationale for selecting these particular three techniques is as follows:

- The techniques are capable of addressing the flight control requirements identified in Section 1.2.
- No more than one technique is to be selected from each category in Section 1.1.
- The software for implementation of the synthesis techniques is sufficiently mature for realistic application to flight control law design.
- The designers have adequate experience with the techniques in flight control applications.

We selected eigenstructure synthesis because it is less closely tied to a model of desired handling qualities, the computations involved are quite straightforward, and there is a fair amount of experience with its application to flight control law design. The method addresses handling qualities and decoupling performance aspects through eigenvalues and eigenvectors and has very low controller complexity. There are concerns about its ability to deal with robust stability and robust performance, and easy work-arounds are not readily available. The computations involve fairly standard matrix operations that are available in many commercial and in-house software packages. Our experience with this technique is limited and more dated compared to the other two techniques selected.

We selected dynamic inversion because it uniquely applies to an entire flight envelope and not just point conditions, reusable software exists, and we have much experience with it. Design complexity is low, practical considerations are easily addressed, and resulting control laws satisfy stability and performance requirements as demonstrated in

several different applications. There are concerns regarding stability (especially robust stability) and robust performance, which are discussed in [Enns 1994], although these same concerns apply to any of the loop-shaping methods. This technique belongs in the frequency domain category because it can be interpreted as a loop-shaping method. Reusable software for implementation of this control law was developed at Honeywell and we have seven years of experience applying it to the F-18 High Angle-of-Attack Research Vehicle (HARV) and other fighter aircraft.

We selected μ-synthesis because it is the only technique that directly addresses the robust performance problem, commercial software is readily available for its implementation, and we have much experience with it. It is the most powerful multivariable approach with regard to stability and performance. However, the design complexity is high in terms of design synthesis and practical considerations regarding its implementation. It is a technique from the numerical optimization category because it minimizes the μ "norm" with an iterative numerical scheme. Software to implement the design iterations is available from two different major software vendors and has been developed in cooperation with Professor John Doyle, one of the authors of these design guidelines. John Doyle invented μ-synthesis while working at Honeywell, and we have gained much experience since in its application to numerous aircraft flight control problems.

Section 2

Fundamentals of Multivariable Control

The fundamental principles of multivariable control are largely the fundamentals of feedback. The basic requirement of feedback systems is to achieve certain desired levels of performance and also to be tolerant of uncertainties. Performance levels concern such things as command following and disturbance rejection. Uncertainty tolerances deal with stability and sensitivity of the closed-loop system in the presence of inevitable differences that exist between a physical system and its mathematical model. As discussed in various textbooks [Bode 1945, James 1947, Horowitz 1963, Doyle 1992] and other references, these two aspects of feedback lead to fundamental tradeoffs and compromises that motivate the entire body of feedback theory.

This section contains an overview of fundamental feedback concepts and some mathematical fundamentals necessary for control design and analysis.

2.1 General Discussion

Control system design is driven by a set of control-law requirements, which are derived from a specified set of vehicle requirements. Vehicle requirements are themselves derived from a set of mission requirements that underlie integrated vehicle design. The process of translating mission requirements into a finished set of control laws is a complex task, requiring an understanding of:

- Flight envelope and mission requirements,
- Vehicle configuration and physical properties,
- Control design methodologies, and
- Control law standards for piloted vehicles.

From an understanding of flight envelope and mission requirements, the control designer selects a set of flight conditions and control modes for design. From the vehicle configuration and physical properties (i.e., vehicle geometry, aerodynamics, mass properties, propulsion, control surfaces), the control designer derives mathematical models of the vehicle dynamics. Then, using one or more control-design methodologies, we produce a set of control laws that are evaluated against a set of control law standards for piloted

vehicles. These control laws are also evaluated for performance to ensure that mission requirements are met.

Our focus in these guidelines is a specific set of multivariable, control-design methodologies that have proven to be effective for practical control of military aircraft. For this reason we will discuss primarily design methodologies, but the reader should be aware of the link between these methodologies and the larger set of issues that the control designer must resolve. Practical concerns impose a set of tacit assumptions about the methodologies — they must apply to vehicle models in industry-standard form, they must be implementable, and they must be amenable to evaluation by standard techniques. Practical concerns also impose a need for flexibility in the design methodology. The methodology must have features (i.e., knobs) that enable the designer to realize a range of design solutions.

The need for flexibility in the control-design methodology is often seen during the process of control-law evaluation. Ideally, there is no need for adjustment of the control laws as evaluation proceeds through the following sequential steps:

- Linear system analyses
- Batch nonlinear simulation
- Piloted simulation
- Flight test

In practice, design iterations are often required. At each step there are numerous criteria to evaluate, and preliminary control designs often fail to meet one or more of them. Upon failing some test, the design must be adjusted. Some adjustment may also be required if vehicle requirements, models, or flight envelope should change during vehicle design or the subsequent life cycle. For these reasons, the control-design methodologies must have some flexibility to allow adjustments.

In evaluating control laws, there are many criteria to consider. Roughly speaking, these criteria fall into three categories:

- Stability
- Performance
- Flying qualities

These categories are not strictly disjoint, but we can distinguish among them roughly as follows. Stability criteria concern the basic ability of the piloted aircraft to return to an equilibrium condition. Performance criteria concern the capability to execute maneuvers

or other tasks in a controlled fashion to within some specified level and accuracy. Flying qualities criteria concern the suitability of the vehicle response to pilot commands. All three types of criteria are important, although stability is the most critical in the event of subsystem failure or battle damage.

The art of flight control design is to realize a solution that achieves an acceptable compromise among the evaluation criteria. A variety of system concepts and mathematical tools (techniques) are used by the designer to create the design. Each methodology has its own set of techniques for representing design criteria in mathematical terms. Some basic techniques are common to different methodologies; these will be discussed in this section and the next. Methodology-specific techniques will be discussed in the third part of this document, starting with Section 9.

2.2 Models

All design methodologies we will discuss are model-based. The vehicle model gives a mathematical description of the relation between control and disturbance inputs and vehicle response. In this model, control and disturbance inputs and vehicle response are represented by functions of time that are called signals.

2.2.1 Signals

A signal is a real-valued (or sometimes complex-valued) function of time. We use signals to represent all time-varying quantities associated with the control system. We will be working with signals of the following types:

- Control input signals
- Vehicle state signals
- Vehicle response (output) signals
- Pilot command signals
- Disturbance and noise input signals
- Internal state signals
- Fictitious signals used to model uncertainty

Examples of control input signals include deflection angles of control surfaces (elevator, aileron, rudder) and variables that control engine thrust. Usually, by vehicle state we mean velocity of the vehicle c.g. ((u, v, w) in body-axis coordinates), angular rates (roll

rate p, pitch rate q, yaw rate r), bank angle ϕ, and pitch angle θ. Sometimes Mach and either altitude or heading are also included in the list of states. In addition, states associated with actuator dynamics and/or elastic modes are also sometimes included in the (extended) vehicle state-space model.

Vehicle response signals may include any of the internal states in addition to quantities measured by onboard sensors (e.g., acceleration at a sensor station, air-data pitot static and total pressure) or quantities computed from states and other model parameters (e.g., acceleration at the pilot station). Pilot command signals typically include pilot stick, rudder pedal, and throttle settings. There are a variety of possible disturbance and noise input signals — wind-gust disturbances and rate-gyro noise are two examples. Feedback-control laws also have internal states, which are signals generated and stored in the flight control computer. Finally, there may be fictitious signals used in the model to represent uncertainty.

All of these signals except the fictitious ones have concrete physical interpretations and could, in principle, be measured directly by some physical device.

2.2.2 Operators

A physical system may be represented mathematically by an operator. In general, an operator is a relationship between two sets of signals. For example, we have the operator relating input and output signals of our vehicle. Another example is the operator relating pilot commands to vehicle state. Operators like these that are associated with physical quantities are called physical operators.

In most applications, physical operators are assumed to define a functional relationship between input and output signals. That is, the operator has a domain (the set of input signals) and a range (the set of output signals), and the outputs can be computed from the inputs once the state of the system at some initial time is given. Implicit in this framework, at least from a mathematical viewpoint, is the assumption that input signals may be chosen independently, at will. From a practical viewpoint, there may be constraints on input signals imposed by physical devices that generate them (e.g., actuator rate limits). The assumption of a functional relationship between inputs and outputs leads to standard block-diagram representations for systems.

Another assumption about physical operators is *causality.* Causality means that the output at time t is determined by the initial state of the system and by the input signals for time less than or equal to t. In other words, the output at time t can in no way depend on the input at a future time $t_1 > t$. Causality implies a sense of cause and effect between input signals and output signals.

Of the two assumptions just made, the functional relationship between inputs and outputs is often only approximately true. For example, the response of an actuator depends on the load, although a block-diagram model might ignore this dependence. The causality

assumption, on the other hand, is considered accurate to arbitrary precision.

2.2.3 Differential Equations

Mathematical operators can be represented in a wide variety of forms. One of the most common representations is a set of nonlinear ordinary differential equations. These typically provide the highest-fidelity vehicle models used for design and analysis of flight control systems. The nominal equations are of the form:

$$\begin{aligned} \dot{x} &= F(x,u) \\ y &= H(x,u) \end{aligned} \tag{2.1}$$

In these equations, x represents a vector of vehicle-state signals, u a vector of control-input signals, and y a vector of output signals. Several different coordinate systems are used to represent the state vector x. Specific forms of these equations will be discussed in Section 4.1.

The operator associated with this system is obtained by integration of the differential equation for the state $x(t)$ as a functional of the control-input vector, $u(\tau)\ \tau \leq t$, and the initial state $x(0)$. The output vector $y(t) = H(x(t), u(t))$ is defined in a causal, functional way on the control-input vector.

An important feature of these nonlinear equations is the set of equilibrium solutions, otherwise known as trim conditions. The trim conditions are defined to be the set of pairs x_0, u_0 such that $F(x_0, u_0) = 0$. If the control input is held fixed at the value u_0 of a trim condition, the state vector will remain at the constant (steady-state) condition x_0. The output at an equilibrium point is also constant: $y_0 = H(x_0, u_0)$.

Except during highly dynamic maneuvers, the vehicle remains close to the trim set. Basic performance numbers (e.g., range, maximum payload, maximum speed) can be computed from analysis of the trim set. For this reason, the trim conditions are examined carefully by the aircraft design team. Control designers also use the trim conditions to derive the linear aircraft models used for design of control laws. We discuss the process of deriving those linear models next.

2.2.4 Linearization

Linearization is the process of computing linear dynamic models that approximate nonlinear equations (2.1) near trim conditions. From a trim condition (x_0, u_0, y_0), a linear model is computed as follows. Let x_1 denote a small perturbation vector in the state space, u_1 a small control perturbation vector, and y_1 a small output perturbation vector.

Now express the signals x, u, and y as the sum of trim values plus perturbations:

$$\begin{aligned} x(t) &= x_0 + x_1(t) \\ u(t) &= u_0 + u_1(t) \\ y(t) &= y_0 + y_1(t) \end{aligned} \tag{2.2}$$

and substitute into the nonlinear state-space equations. After some standard manipulations, we obtain the linear equations (good to first order):

$$\dot{x}_1 = \frac{\partial F(x_0, u_0)}{\partial x} x_1 + \frac{\partial F(x_0, u_0)}{\partial u} u_1 \tag{2.3}$$

$$y_1 = \frac{\partial H(x_0, u_0)}{\partial x} x_1 + \frac{\partial H(x_0, u_0)}{\partial u} u_1 \tag{2.4}$$

Once linearization has been performed, the notation of equations (2.3) and (2.4) is usually simplified by replacing the perturbation variable names x_1, u_1, and y_1 with the original variable names x, u, and y and letting A, B, C, and D denote the partial-derivative matrices, yielding the standard state-space description:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.5}$$

The traditional (linear) control design process invariably begins with equations of this form. Equations (2.5) can be integrated explicitly. The output $y(t)$ is

$$y(t) = Du(t) + C \int_0^t e^{A(t-\tau)} B u(\tau) d\tau \tag{2.6}$$

By writing the solution in this form, we clearly see the causal relation between the input signal $u(t)$ and the output signal $y(t)$. The causal operator relating inputs to outputs is called a convolution operator and has the form

$$y(t) = \int_0^t \mathcal{K}(t-\tau) u(\tau) d\tau \tag{2.7}$$

The function $\mathcal{K}(t)$ in equation (2.7) is called the convolution kernel. The mathematical theory of convolution operators is extensive but finds little use in practical control design. For our purposes, it suffices to say that the convolution representation allows efficient numerical solution of the system equations by computer. For purposes of control design and analysis, there is another representation of the system operator, the transfer function, which we describe next.

2.2.5 Laplace Transform, Frequency Response, and Transfer Functions

Given a time signal $w(t)$, we may associate with it the Laplace Transform function $\hat{w}(s)$ defined by:

$$\hat{w}(s) = \int_0^\infty e^{-st} w(t) dt \tag{2.8}$$

The Laplace transform is a powerful mathematical concept that lets us convert the time-domain convolution operator described in the previous subsection into a frequency-domain multiplicative operator, called the transfer function. The construction of the transfer function is as follows.

First we note that the Laplace transform of the time derivative of a signal $w(t)$ vanishing at $t = 0, \infty$ has a simple form:

$$\hat{\dot{w}}(s) = \int_0^\infty e^{-st} \dot{w}(t) dt = s\hat{w}(s) \tag{2.9}$$

as can be shown using integration by parts. Transforming the system of equations (2.5), we find:

$$\begin{aligned} s\hat{x}(s) &= A\hat{x}(s) + B\hat{u}(s) \\ \hat{y}(s) &= C\hat{x}(s) + D\hat{u}(s) \end{aligned} \tag{2.10}$$

These two equations can be manipulated by linear algebra, eliminating the variable $\hat{x}(s)$, to obtain a simple multiplicative relationship between $\hat{y}$ and $\hat{u}$:

$$\hat{y}(s) = [D + C(sI - A)^{-1}B]\hat{u}(s) \tag{2.11}$$

The matrix $G(s)$ defined by

$$G(s) = D + C(sI - A)^{-1}B \tag{2.12}$$

that appears in equation (2.11) is called the transfer function of the system. The transfer function will serve as our primary means of representing system operators for control analysis and design. Three basic features of the transfer function make it useful in practical applications.

First, it is easier, computationally, to work with multiplication operators than with convolution operators. The information contained in the transfer function $G(s)$ is the same as that in the convolution kernel $\mathcal{K}(t)$. In fact, it can be shown that

$$G(s) = \int_0^\infty e^{-st} \mathcal{K}(t) dt = \hat{\mathcal{K}}(s) \tag{2.13}$$

Second, the transfer function has a clear interpretation in terms of the system's frequency response. For simplicity, consider a single-input single-output system. If we choose a sinusoidal control input

$$u(t) = v \sin(\omega t)$$

we find that, after an initial transient, the output looks like

$$y(t) \approx vMsin(\omega t + \phi)$$

where

$$G(j\omega) = Me^{j\phi},\ j = \sqrt{-1}$$

In other words, the steady-state system response to a sinusoidal input of frequency ω is completely determined by the value of the transfer function at the point $j\omega$. For this reason, the imaginary axis of the s-domain is called the frequency axis and the value of the transfer function along the imaginary axis is called the frequency response. There is practical significance in this observation because the frequency response of a real, physical system can be measured by practical means (e.g., perturb the input signal by small sinusoidal deviations and measure the output deviations). This test can be performed for frequencies within the specified frequency response of the system. (In theory, a similar physical test can be used to recover the convolution kernel. However, the input must then be an impulse, which cannot be delivered to a system without violating its limits. Even approximate impulses can drive the system outside of its linear range and might even cause damage.)

Third, the frequency domain is the natural realm in which to represent many uncertainties in system models (e.g., noise, uncertain actuator dynamics). Practical models are usually valid only for signals of sufficiently low frequency. As the frequency of the input increases, more and more high-frequency modes of the true system become excited, which are not properly represented in the model. It becomes increasingly futile to build models that represent systems predictably for signals of arbitrarily high frequency.

2.3 Basic Properties of Models

The linear models described in the previous section have mathematical properties that we use in design and analysis. In this section we introduce a few basic properties.

First, we discuss norms. A norm is a measure of size, and we can define norms for constants, matrices, signals, and operators. In the process of defining norms, we introduce the notion of singular values, which plays a significant role in modern multivariable system theory. Finally, we discuss the basic properties of poles and zeros of transfer functions.

2.3.1 Norms of Constants

There are several norms of constants to consider:

- Absolute value of a scalar
- Euclidean norm of a column vector

- Maximum singular value of a matrix

The absolute value or magnitude, $|z|$, of a complex scalar, z, is $\sqrt{z\bar{z}}$, where $\bar{z}$ is the complex conjugate of z.

The Euclidean norm, or vector norm, $\|x\|$, of a complex $n \times 1$ column vector, x, is $\sqrt{x^*x}$ where x^* is the complex conjugate transpose of x.

The maximum singular value of a complex $n \times m$ matrix A is

$$\bar{\sigma}[A] = \max_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} \tag{2.14}$$

Note that if A is a scalar, then $\bar{\sigma}[A] = \|A\| = |A|$, and if A is a column vector, then $\bar{\sigma}[A] = \|A\|$.

The maximum singular value can be regarded as a gain of a multivariable system. Consider

$$y = Ax \tag{2.15}$$

where x is regarded as the input and y is regarded as the output. The gain is the ratio of the size of the output to the size of the input, but x and y are vectors, so their size is measured with the Euclidean norm. So gain can be defined as

$$\text{"gain of } A\text{"} = \frac{\|y\|}{\|x\|} = \frac{\|Ax\|}{\|x\|} \tag{2.16}$$

This "gain" clearly depends on the matrix A, but it also depends on which x is used in forming the ratio. Using equation (2.14), we see that the gain corresponding to the value of x that produces the largest ratio of $\|Ax\|$ to $\|x\|$ is just the maximum singular value of A. It is only the direction of x that determines this gain, because a scalar multiplier of x will appear in the numerator and denominator and thus cancel out. Similarly, the gain corresponding to the value of x that produces the smallest ratio of $\|Ax\|$ to $\|x\|$ is known as the minimum singular value of A. If the matrix is singular, or less than full rank, the minimum singular value is zero.

There are actually gains for each of m mutually orthogonal directions of the input space, and all these gains and directions are defined by the singular value decomposition, as described in Appendix ??.

2.3.2 Norms of Signals

To analyze signals, we often make the assumption that they lie in the function space $L_2(0, \infty)$ defined by:

$$L_2(0, \infty) = \{f \mid \int_0^\infty f^2(t)dt < \infty\} \tag{2.17}$$

In words, this is just the set of all square-integrable functions. The norm $\|f\|_2$ of the signal f in $L_2(0,\infty)$ is defined by:

$$\|f\|_2 = \sqrt{\int_0^\infty f^2(t)dt} \tag{2.18}$$

Other function spaces and norms are sometimes used to analyze signals, but for our purposes, the space $L_2(0,\infty)$ will be sufficient. In practice, all signals we can analyze are available only for a finite period, so it might seem unnatural to consider signals defined for all time. To model signals of finite duration T, extend the domain of the function to ∞ and give it the value 0 for time greater than T. This artifice of domain extension simplifies both notation and analysis.

We will usually be concerned with the space of L_2 functions of time (i.e., signals), but at times we also consider the space of L_2 functions of frequency. Such functions arise when we look at the Laplace transform of a signal, considered as a function of s-values on the imaginary axis, and arise in the study of covariance response of a system to stochastic inputs.

2.3.3 Norms of Operators

We will usually be concerned with linear time-invariant operators M. When such an operator is stable (see Section 2.6), it maps L_2 functions to L_2 functions. The norm of such an operator is defined by

$$\|M\|_{L_2-L_2} = \sup_{\|u\|_2 \neq 0} \frac{\|Mu\|_2}{\|u\|_2} \tag{2.19}$$

and one can show that

$$\|M\|_{L_2-L_2} = \|M(s)\|_\infty = \sup_\omega \bar{\sigma}[M(j\omega)] \tag{2.20}$$

where $M(s)$ is the operator's transfer function matrix. The quantity, $\|M(s)\|_\infty$, is called the infinity norm of $M(s)$. It is the largest value of the maximum singular value of the frequency response, $M(j\omega)$. In mathematics, the space of all stable transfer functions is called a Hardy space. Hence, $\|M(s)\|_\infty$ is also known as the H-infinity norm.

2.3.4 Poles and Zeros of Transfer Functions

Poles of a transfer function $M(s)$ are the values of s for which $M(s)$ is infinite, while zeros are the values of s for which $M(s)$ loses rank. For the state-space description (2.5), the transfer function is given by

$$M(s) = C(sI-A)^{-1}B + D = C\frac{\text{Adjoint}(sI-A)}{\det(sI-A)}B + D \equiv \frac{\psi(s)}{\phi(s)} \tag{2.21}$$

where $\phi(s)$ is a scalar polynomial of order equal to the number of system states (n) and $\psi(s)$ is a matrix polynomial with dimensions consistent with the number of system outputs and inputs and of order less than or equal to n. The poles of the system are the roots of $\phi(s) = \det(sI - A) = 0$, which are just the eigenvalues of A. For single-input/single-output systems, the polynomial matrix, $\psi(s)$, in equation (2.21) is scalar, and its roots correspond to the system's zeros, i.e.,

$$\psi(s)|_{s=z_i} = 0 \quad i = 1, 2, \ldots, r \tag{2.22}$$

There are at most $r = n$ such values of s ($n - 1$ when $D = 0$), and there can be as few as none. These values can be located anywhere in the complex plane, and we will see shortly that locations in the right-half plane are particularly undesirable for feedback systems. Note that a scalar $M(s) = \psi(s)/\phi(s)$ vanishes at these zeros. Physically, this means that certain inputs, namely $u(t) = \sum \alpha_i \exp(z_i t)$, are completely blocked by the system. If we try to apply such inputs, no forced output responses result, only unforced initial condition responses.

This signal blocking characteristic provides a way to generalize the concept of zeros to multivariable systems. We here consider only cases where $M(s)$ is square, with m inputs and outputs, and generically full rank (i.e., invertible at almost all values of s)[1]. Then there exist at most n values $s = z_i$, called *transmission zeros*, at which $M(s)$ becomes singular (at most $n - m$ values when $D = 0$). Because $M(s)$ is singular at each z_i, there are corresponding input vectors ν_i such that

$$M(s)\nu_i|_{s=z_i} = 0 \quad i = 1, 2, \ldots, r \tag{2.23}$$

Note that this condition again means that certain inputs, this time $u(t) = \sum \alpha_i \nu_i \exp(z_i t)$, are completely blocked by the system, and we will see that this too causes difficulties when the transmission zeros are located in the right-half plane.

Equations (2.5) and (2.23) together lead to a nice computational procedure for calculating transmission zeros. Specifically, if $M(z_i)\nu_i = 0$, then it follows from (2.5) that there exist corresponding vectors ξ_i such that

$$\begin{aligned} (A - z_i I)\xi_i + B\nu_i &= 0 \\ C\xi_i + D\nu_i &= 0 \end{aligned} \tag{2.24}$$

Finding z_i, ξ_i, and ν_i for this equation is a well-known generalized eigenvalue problem with high-quality numerical routines available from various sources.

[1]The nonsquare case can be treated analogously, by defining zeros as values of s at which $M(s)$ loses rank. Unfortunately, zeros defined this way are very sensitive to small errors in the representations of M and can appear and disappear at will. Also, since $M(s)$ is not invertible, there always exist nonzero vectors, ν, either on the right or left of M, which satisfy equation (2.23). In a sense, then, non-square systems have zeros and block signals everywhere in the complex plane.

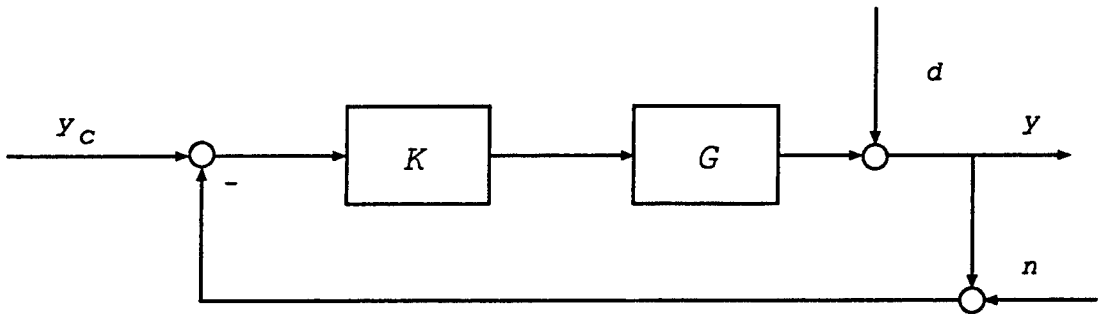


Figure 2.1: Block Diagram of Closed-Loop System.

2.4 Feedback Loops and Basic Functions

Consider the feedback system shown in Figure 2.1, where G represents the system to be controlled and K is the feedback compensator. Here y is the output signal, y_c the desired value (or command) for this variable, d the disturbance, n the sensor noise, and u the actuator input. Since the system is multivariable, G and K are matrices and all signals are vectors.

2.4.1 Loop Transfer Function

The loop transfer function of this feedback system is a matrix that defines the loop transmission around the feedback loop under the assumption that the loop is broken at some point. For a loop broken at the output, the loop transfer function is given by $L_o(s) = G(s)K(s)$. Similarly, for a loop broken at the input, the loop transfer function is given by $L_i(s) = K(s)G(s)$. For SISO systems, these two loop transfer functions are equivalent because G and K are scalars and multiplication is commutative. The loop transfer function for this case, $L(s)$, is often displayed in a Bode plot, which shows both magnitude and phase of $L(s)$ as a function of frequency. The frequencies at which loop magnitude crosses 1 (or 0 dB) are called *crossover frequencies.* In general, there are several such frequencies (e.g., if the loop includes phase stabilized flexure modes). Most often, however, there is only one, and this frequency then corresponds closely to the usual 3-dB bandwidth of the closed-loop system. For this reason, we will use the terms bandwidth and crossover frequency interchangeably. Bandwidth gives an indication of the speed of response or aggressiveness of the closed loop. A high-bandwidth loop responds more quickly to inputs, and responds more accurately to inputs of higher frequency.

For multivariable systems, the two loop transfer functions, GK and KG, are matrices and are, in general, different from each another. Here a conventional Bode plot of magnitude and phase for each element of these matrices is of little practical use. Rather, we must employ the *matrix generalization* of Bode plots, which are plots of the magnitudes of all singular values of the loop transfer function as a function of frequency, as illustrated in Figure 2.2. Here loop crossover rarely occurs at a single frequency but is distributed over

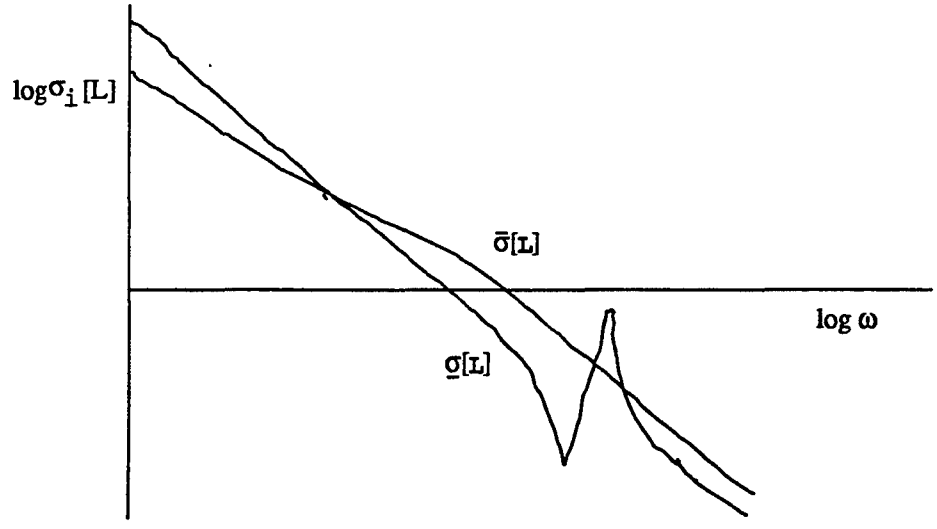


Figure 2.2: Singular Values of Loop Transfer Function.

a range of frequencies determined by the range in magnitudes of the individual singular values. Thus, bandwidth also refers, not to a single frequency, but to a frequency range determined by the singular-value crossovers. Note that the notion of phase angle has no direct matrix generalization for multivariable systems.

There are other implications associated with bandwidth. For example, below the crossover frequency where magnitude is high, feedback control is effective in suppressing disturbances and improving command tracking, while in low-gain regions, usually well beyond the crossover frequency, open-loop and closed-loop responses are nearly the same. There are also monetary cost implications. Higher bandwidth systems generally cost more. Because the system is expected to respond to higher frequency signals, wide-band sensors and actuators are necessary, as well as computers with greater throughput. All these hardware elements are generally more costly to build.

2.4.2 Closed-Loop Response Functions

Returning to the feedback system shown in Figure 2.1, we next derive transfer functions for the closed-loop system response and error response. Algebraic manipulations applied to the block diagram give the following results:

Output Response:

$$\begin{aligned} y &= d + GK\,[y_c - (y + n)] \\ &= (I + GK)^{-1} d + (I + GK)^{-1} GK(y_c - n) \end{aligned} \tag{2.25}$$

Error Response:

$$\begin{aligned} e = y_c - y &= [I - (I + GK)^{-1}GK]\, y_c - (I + GK)^{-1}(d - GKn) \\ &= (I + GK)^{-1}(y_c - d) + (I + GK)^{-1}GKn \end{aligned} \tag{2.26}$$

The *closed-loop transfer function* from command input y_c to output y, then, is given by

$$T = (I + GK)^{-1}GK \tag{2.27}$$

Besides $T(s)$, a second important transfer function that appears in equations (2.25) and (2.26) is the sensitivity function

$$S = (I + GK)^{-1} \tag{2.28}$$

This function derives its name from the fact that feedback control reduces the sensitivity of the open-loop system's response to uncertainty in the plant model, $G(s)$, by the ratio S (see Appendix ??). Simple matrix algebra shows that for the two functions T and S,

$$T + S = I \tag{2.29}$$

That is, T and S add up to I and can be considered complements of each other. For this reason, $T(s)$ is also often called the *complementary sensitivity function.*

Referring back to equations (2.25) and (2.26), we see that T and S are essential in defining closed-loop responses. We will also see shortly that they are essential in analyzing robust stability of feedback systems. Hence S and T play a fundamental role in feedback theory and in the design and analysis of feedback systems.

2.5 Sets of Models

In Section 2.2 we explained how the basic linear models used in flight-control design are generated. From a set of nonlinear differential equations, we first compute the trim set, then we generate linear models at selected trim points. In practical applications, we must recognize that models are not exactly known — all our equations are merely approximations of the dynamics of real-world systems. Despite these uncertainties, our control design methodologies must provide solutions that work.

To address this issue of uncertainty, we focus our design efforts, not on a single model, but on a specified set of models. The idea is to first prescribe a set of models, then design a single controller that works for every model in that set. This approach is called *robust control design*, and we will have much to say about robust control in the subsections that follow. In this subsection we concentrate on sets of models that often arise in practice. To understand the significance of these sets of models, we first review sources of uncertainty.

2.5.1 Sources of Uncertainty

As discussed in the previous subsection, we approximate the real system with finite-dimensional linear time-invariant (FDLTI) models in the form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.30}$$

where x is the state-vector, u is the control input vector, y is the output vector, and A, B, C, D are constant matrices.

The technique of representing the system by the matrices A, B, C, D is convenient from a mathematical viewpoint, but its limitations must be recognized.

First, we must recognize that there are approximations in the nonlinear equations from which the linear equations are derived. Distributed system phenomena are approximated by lumped-parameter models. In some cases, this approximation is accomplished by building sets of table-lookup functions (e.g., aerodynamic functions). In others, the approximation is based on finite-element techniques from structural dynamics, followed by model reduction (e.g., vehicle elasticity). In addition, many high-frequency dynamic phenomena are neglected (e.g., unmodeled actuator and sensor dynamics). The set of models selected for control design should represent uncertainties associated with these approximations in aerodynamics, structural dynamics, and actuator and sensor dynamics.

Second, there are parameters in the nonlinear model that vary and have a significant effect on the linearized models. For example, mass properties of a vehicle vary from flight to flight (due to payload changes) and during flight as a function of fuel use. Air density is another parameter that is often modeled as a fixed function of altitude, although it also depends on weather conditions. The set of models selected for control design should include the effects of realistic ranges of vehicle mass and atmospheric properties.

Third, there are disturbances to be taken into account. Some are external in origin (e.g., wind gusts, weapon delivery), whereas others are internal to our system (e.g., sensor noise). The effects of disturbances on vehicle and control system dynamics should be modeled and accounted for in the set of models.

Finally, there are implementation issues. One significant factor in this category is time delay. The effective delay in the feedback path is due to a combination of sensor-data filtering, data-communication delays, and computation time. The set of models should include these effects as well.

2.5.2 Constructing Sets of Models

Building sets of models that reflect all the various sources of uncertainty above remains an engineering art. In practice, one of the most effective ways to do so relies on what mathematicians call *bounded sets.* These consist of a nominal *center* surrounded by a

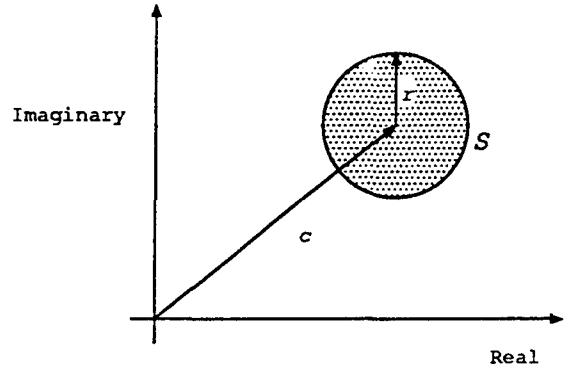


Figure 2.3: Representation of a Set of Complex Models.

collection of possible variations that are bounded in magnitude by a *radius*. The center will be referred to as the nominal model. It might be the highest fidelity model available (i.e., the "best guess"), or it might be a reduced-order model used in design iterations. The radius then describes how far an actual system can deviate from this nominal.

To illustrate this concept, consider the simplest types of models consisting of real numbers. A bounded set for these models is just a segment of the real number line

$$a_\ell < a < a_u \tag{2.31}$$

where a_ℓ is the lowest possible value for a and a_u is the largest possible value for a. In this case, the center is

$$c = \frac{a_\ell + a_u}{2} \tag{2.32}$$

and the radius is

$$r = \frac{a_u - a_\ell}{2} \quad . \tag{2.33}$$

The model set, then, is all real numbers, a, such that

$$a = c + \delta \quad \text{where } |\delta| < r \tag{2.34}$$

which in set notation is given by

$$\mathcal{S} = \{a \mid a = c + \delta,\ |\delta| < r\} \quad . \tag{2.35}$$

In aircraft applications, the real number, a, may be an aerodynamic coefficient, a stability derivative, a mass, or some other parameter of a model.

More generally, for models given by complex numbers, the set (2.35) is just a disk in the complex plane, with c corresponding to the center of the disk and r corresponding to the radius of the disk (see Figure 2.3). In aircraft applications, a may then be a scalar transfer function evaluated at $s = j\omega$ (i.e., a frequency response at one value of ω).

Still more generally, the center and radius of a model set can themselves be functions of frequency. Thus, the set

$$\mathcal{S} = \{\tilde{g}(s) \mid \tilde{g}(s) = g(s) + \delta(s),\ |\delta(s)| < r(s)\} \tag{2.36}$$

describes a whole collection of scalar transfer functions with nominal value, $g(s)$, and possible deviations bounded by $r(s)$. This generalizes still further to describe a whole collection of multivariable transfer function matrices

$$\mathcal{S} = \{\tilde{G}(s) \mid \tilde{G}(s) = G(s) + \Delta(s),\ \bar{\sigma}[\Delta(s)] < r(s)\} \tag{2.37}$$

Whether the model set is real, complex, or frequency dependent, it can be represented in many ways. Different representations imply different centers and bounds for the radii. The most common representations are additive, multiplicative, and divisive. The additive representation corresponds to the one just shown above. Multiplicative representations take the form

$$\mathcal{S} = \{\tilde{G}(s) \mid \tilde{G}(s) = G(s)[I + \Delta_m(s)],\ \bar{\sigma}[\Delta_m(s)] < r_m(s)\} \tag{2.38}$$

and divisive ones take the form

$$\mathcal{S} = \{\tilde{G}(s) \mid \tilde{G}(s) = G(s)[I + \Delta_d(s)]^{-1},\ \bar{\sigma}[\Delta_d(s)] < r_d(s)\} \tag{2.39}$$

Note that, in general, each of these expressions describes a different set of models even when the radii are adjusted to try to match them. (Exact matches are possible in the scalar case.) For this reason, we will find different uses for each of these forms.

Finally, sets of models can involve more than one representation of uncertainty. One example is the set

$$\mathcal{S} = \{\tilde{G} \mid \tilde{G} = (I + \Delta_d)^{-1}G(I + \Delta_m),\ \bar{\sigma}[\Delta_d] < r_d,\ \bar{\sigma}[\Delta_m] < r_m\} \tag{2.40}$$

in which both multiplicative and divisive representations are employed. In this case, the divisive perturbation is said to be an output perturbation because it appears to the left of G and thus modifies the outputs, and the multiplicative perturbation is said to be an input perturbation because it appears to the right of G and thus modifies the inputs. Note that either representation can appear at the input or output.

These and other representations of uncertainty are useful in flight control analyses.

2.6 Stability

We return now to the feedback loop in Figure 2.1 and consider $G(s)$ to be any element of a specified bounded set of models. Then the first job of controller K is to ensure stability of the closed-loop system for all of these G's. If all G's and K are finite-dimensional

linear time-invariant systems, the basic concept of stability is easy to understand: A system is considered stable if its state vector asymptotically approaches zero when the input signal is held fixed at zero. A simple computational criterion is described below for verifying this property when G consists of a single model.

When G is a model set, however, it is not as simple to verify stability, especially when the set contains structured uncertainty. The small-gain theorem is the most basic result along these lines, and we will discuss that result shortly.

To analyze general sets of systems subject to structured uncertainty we require the structured singular value, which is also discussed below.

2.6.1 Nominal Stability

Nominal stability refers to the property that the closed-loop system is stable for one model, namely $\tilde{G}(s) = G(s)$, the center of the model set. It is assessed by constructing a state-space model for the nominal closed-loop system and examining the eigenvalues of its A matrix. If the real parts of all eigenvalues are negative, the system is stable. Otherwise, the system is unstable. (Technically, the system is said to be neutrally stable if the real parts of all eigenvalues are *less than or equal* to zero and all eigenvalues with zero real parts are simple.)

The state-space descriptions needed for these analyses can be easily derived from the individual state-space descriptions of G and K. Most modern control system CAD packages include high-level commands that carry out the closed-loop construction and eigenvalue calculations painlessly.

2.6.2 Robust Stability

Robust stability refers to the property that the closed-loop system is stable for all G's in the set. In principle, we can verify this property by repeating the above eigenvalue analysis for all models. For large sets, of course, this exhaustive evaluation quickly becomes prohibitive, and for infinite sets it fails completely. Fortunately, control engineers have long ago solved this problem with the concept of *stability margins*. By computing gain and phase margins of SISO feedback loops, they have been able to verify robust stability for infinitely many models, provided only that the models' gain and phase deviations do not exceed the margins. This basic idea generalizes nicely to multivariable systems using the bounded set uncertainty representations introduced earlier.

To illustrate this, consider the divisive uncertainty set for which models are described by the equation

$$\tilde{G} = (I + \Delta_d)^{-1} G \quad \text{with } \bar{\sigma}[\Delta_d] < r_d \quad . \tag{2.41}$$

A block diagram of this uncertainty representation is shown in Figure 2.4. The first part of

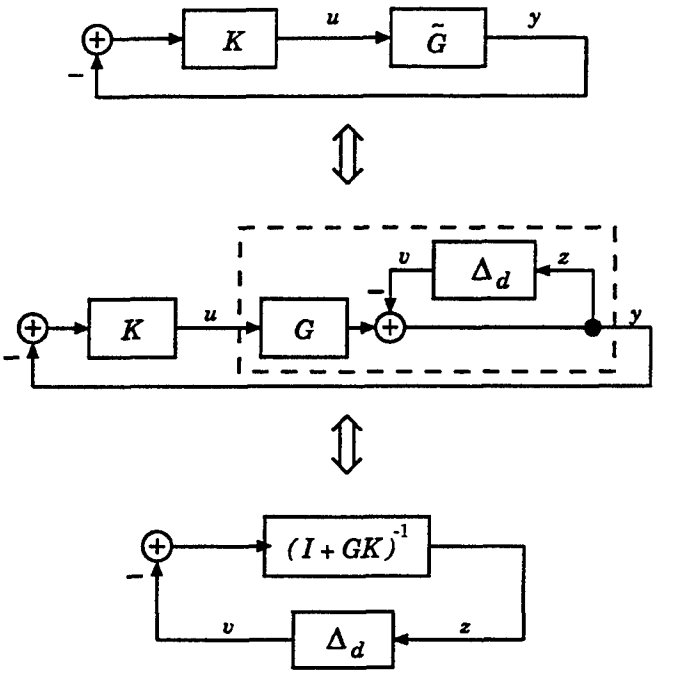


Figure 2.4: Equivalent Closed-Loop System With Divisive Perturbation.

this figure is the original feedback loop. The second part is the loop with equation (2.41) substituted for $\tilde{G}$, and the third part is derived from simple block diagram manipulations. The important point is that all three parts of the figure are equivalent. Hence, robust stability of the original loop is the same as robust stability of an alternative feedback loop consisting of the nominal sensitivity function, $S(s)$, with an unknown but bounded element, $\Delta_d(s)$, connected in feedback around it.

A similar result applies for the multiplicative uncertainty representation, where

$$\tilde{G} = (I + \Delta_m)G \quad \text{with } \bar{\sigma}[\Delta_m] < r_m \tag{2.42}$$

A block diagram for this representation is shown in Figure 2.5, which demonstrates that the original feedback loop in this case is equivalent to an alternative loop consisting of the nominal complementary sensitivity function, $T(s)$, with unknown but bounded element, $\Delta_m(s)$, connected in feedback around it.

It turns out that robust stability of both of these alternative feedback loops can be easily verified using a basic theorem from mathematical systems theory called the *small-gain theorem* [Desoer 1975]. This theorem applies to general feedback loops of the form shown in Figure 2.6, where M is a known operator (corresponding to S or T in the two cases above) and Δ is an unknown but bounded operator. The theorem states that this loop is robustly stable for all Δs such that

$$\|\Delta\|_{L_2-L_2} < \frac{1}{\|M\|_{L_2-L_2}} \tag{2.43}$$

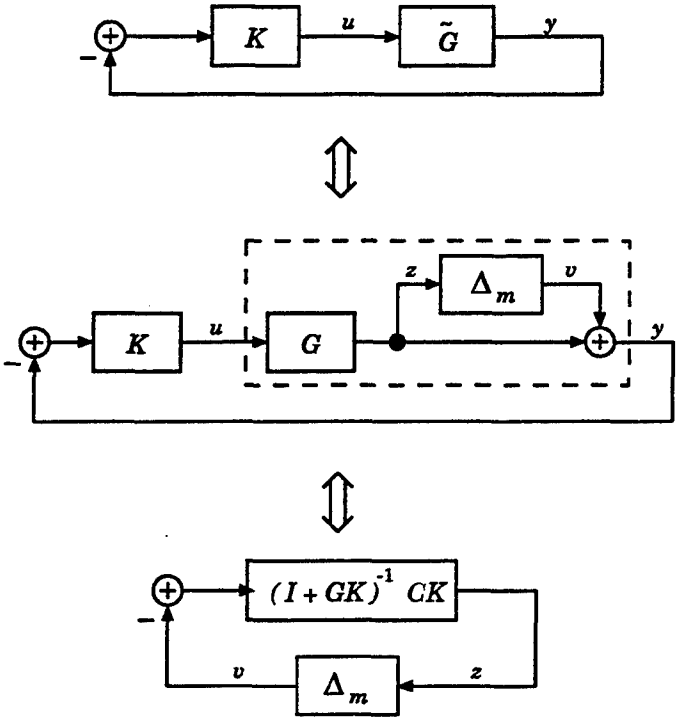


Figure 2.5: Equivalent Closed-Loop System With Multiplicative Perturbation.

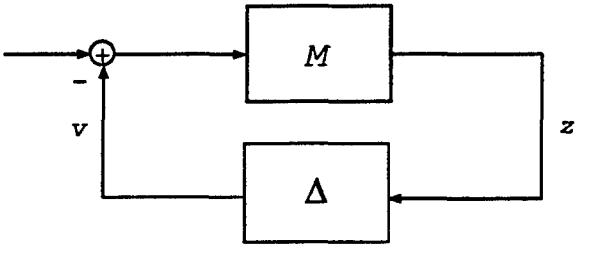


Figure 2.6: Robust Stability Setup.

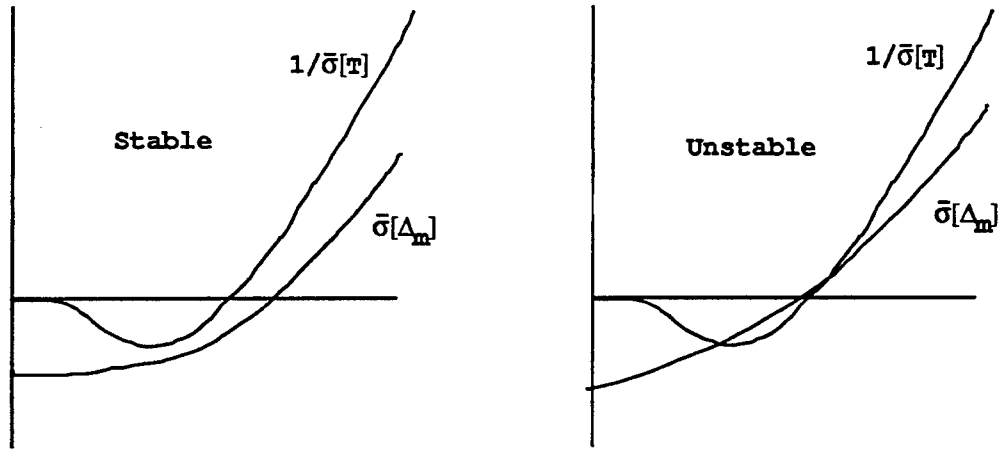


Figure 2.7: Robust Stability Verification.

In other words, the loop is stable if the operator norm of Δ is strictly less than the reciprocal of the norm of M. *So the stability margin of M, with respect to uncertain feedback around it, is just the reciprocal of its operator norm.* This statement applies to very general operators, not just linear time-invariant ones. If we know that M and Δ are, in fact, linear time invariant, then an even stronger version of the theorem applies. Namely, the feedback loop in Figure 2.6 is robustly stable if and only if it is nominally stable and

$$\bar{\sigma}[\Delta(j\omega)] < \frac{1}{\bar{\sigma}[M(j\omega)]} \quad \text{for all } \omega \tag{2.44}$$

Note that this statement guarantees stability if the norm of $\Delta(s)$ is strictly less than the reciprocal of the norm of $M(s)$ pointwise in frequency. Moreover, it also says that the loop is unstable (its responses diverge) if the condition is violated.

Evidently, equation (2.44) provides a very simple graphical test for robust stability of the divisive and multiplicative uncertainty cases above. Simply compare the generalized Bode plots of $1/\bar{\sigma}[S]$ and $\bar{\sigma}[\Delta_d]$ and $1/\bar{\sigma}[T]$ and $\bar{\sigma}[\Delta_m]$, respectively, to see whether the robust stability condition is satisfied. Some illustrations of such comparisons are shown in Figure 2.7.

2.6.3 Robust Stability for Structured Uncertainties

As we have just seen, the small-gain theorem provides a powerful tool to verify robust stability with simple bounded model sets. Unfortunately, this tool becomes much less effective when more than one uncertainty block appears in the set. To illustrate this, consider the two-block uncertainty example from equation (2.40), where we included a divisive uncertainty block at the output of the nominal model as well as a multiplicative

block at its input. The resulting uncertainty set was given by

$$\mathcal{S} = \{\tilde{G} \mid \tilde{G} = (I + \Delta_d)^{-1} G (I + \Delta_m),\ \bar{\sigma}[\Delta_d] < r_d,\ \bar{\sigma}[\Delta_m] < r_m\} \tag{2.45}$$

Using block diagram manipulations similar to those in Figures 2.4 and 2.5, it is still possible to reduce a feedback loop with this model set to the general small-gain form (i.e., to make it look like Figure 2.6). However, in this case we get a larger 2×2 block-structured $M(s)$ matrix connected in feedback with a larger 2×2 block-diagonal $\Delta(s)$ matrix. These larger M and Δ matrices are given by

$$M = \begin{bmatrix} S & -SG \\ -KS & (I + KG)^{-1} KG \end{bmatrix} \quad \Delta = \begin{bmatrix} \Delta_d & 0 \\ 0 & \Delta_m \end{bmatrix} \tag{2.46}$$

We can, of course, still use the small-gain theorem to conclude that the closed-loop system will be robustly stable if and only if $\bar{\sigma}[\Delta(j\omega)] < 1/\bar{\sigma}[M(j\omega)]$ for all ω. Note, however, that this condition applies for any 2×2 Δ, not just for the block-diagonal one we actually have. So it includes many more models in the model set than are actually present. This leads to *conservatism*: the small-gain test can fail for the larger set, even when the true set remains robustly stable.

This conservatism becomes progressively more severe as additional uncertainty blocks are included in the true model set. In general, we get an $m \times m$ block-structured M matrix that must be tested in feedback with a block-diagonal Δ matrix (m blocks), and we cannot afford the conservatism introduced by ignoring the diagonal nature of this Δ. Fortunately, an alternative stability robustness measure, called the *structured singular value*, is available to solve this problem. This alternative measure is based on the mathematical fact that the feedback loop in Figure 2.6, with linear time-invariant M and Δ, is robustly stable for all Δs in some structured set $\mathcal{D}$ if and only if it is nominally stable and

$$\det[I + M(j\omega)\Delta(j\omega)] \neq 0 \text{ for all } \Delta \epsilon \mathcal{D} \text{ and all } \omega \tag{2.47}$$

The structured singular value is a function of M and $\mathcal{D}$, defined specifically to let us test this determinant condition. Its formal definition is

$$\mu_{\mathcal{D}}[M] = \left[\inf_{\Delta \epsilon \mathcal{D}} \{\bar{\sigma}[\Delta] \mid \det(I + M\Delta) = 0\}\right]^{-1} \tag{2.48}$$

Note that this function is just the reciprocal of the magnitude of the smallest perturbation in set $\mathcal{D}$ that makes the determinant zero. So, if destabilizing perturbations exist in $\mathcal{D}$, then $1/\mu$ is the size of the smallest of these, and robust stability is guaranteed if and only if this smallest size is bigger than any perturbation allowed.[2] That is,

$$\bar{\sigma}[\Delta(j\omega)] < \frac{1}{\mu_{\mathcal{D}}[M(j\omega)]} \text{ for all } \omega \tag{2.49}$$

[2]If no destabilizing perturbations exist in $\mathcal{D}$, then $\mu = 0$.

This condition is a direct generalization of the small-gain condition (2.44), except that it now accounts for the structure of Δ and does not suffer from conservatism. It is used just like (2.44) in Figure 2.7 to verify the robust stability of feedback systems.

It turns out that structured singular values, μ, are much more difficult to compute that ordinary singular values, σ. However, modern control system CAD packages provide algorithms and commands to calculate bounds for them. These commands typically handle block-diagonal Δ structures. They also accommodate structures where some of the blocks are complex matrices and others are real scalars, and where some of the blocks are repeated. Their underlying algorithms generally use upper and lower bound calculations based on certain fundamental structured-singular-value inequalities, namely

$$\sup_{U} \rho(UM) \leq \mu[M] \leq \inf_{D} \bar{\sigma}[DMD^{-1}] \tag{2.50}$$

where $\rho(A)$ is the spectral radius of matrix A (i.e., the magnitude of the largest eigenvalue), the Us are block-diagonal unitary matrices, and the Ds are block-diagonal scaling matrices matched to the structure of Δ. The right-hand upper bound in this expression provides the mathematical basis for one of the multivariable design methods used later in these guidelines, namely the μ-Synthesis method using DK-Iteration.

One of the most important properties of the structured singular value is its dependence on the structure, $\mathcal{D}$, as well as its dependence on M. Formally, it should always be referred to as $\mu_{\mathcal{D}}[M]$. However, as a common abuse of notation, we often drop the subscript. To emphasize the structural dependence at least once, we illustrate it with two extremes — the *scalar-times-identity* case and the *unstructured* case.

2.6.3.1 Scalar-Times-Identity Structure

Consider uncertainties such as dynamic pressure that scale more than one parameter in aircraft models (e.g., aileron and rudder contributions to roll and yaw moments). Such uncertainty can be represented using a special scalar-times-identity structure

$$\mathbf{\Delta} = \{\Delta \mid \Delta = \delta I,\ |\delta| < 1\} \tag{2.51}$$

Then, the condition, $\det(I + M\Delta) \neq 0$, can be assessed using the eigenvalue decomposition of M

$$M = V\Lambda V^{-1} \tag{2.52}$$

where Λ is the quasi-diagonal eigenvalue matrix in Jordan form and V is the corresponding generalized eigenvector matrix discussed in many textbooks [Stewart 1973, Strang 1976, Kailath 1980]. Then

$$\begin{aligned} \det[I + M\Delta] &= \det[I + V\Lambda V^{-1}\delta] \\ &= \det[I + \Lambda\delta] \\ &= \textstyle\prod_{i=1}^{n}(1 + \lambda_i\delta) \\ &\neq 0 \text{ for all } |\delta| < \frac{1}{\max|\lambda_i|} \end{aligned} \tag{2.53}$$

Thus, by the definition of μ in equation (2.48),

$$\mu_{\delta I}[M] = \rho[M] \tag{2.54}$$

2.6.3.2 Unstructured

At the other extreme, many uncertainties such as neglected dynamics for high-frequency elasticity, actuators, sensors, multi-rate asynchronous digital implementations, and other sources are often best represented as completely unstructured. The condition, $\det(I + M\Delta) \neq 0$, for this case reduces to

$$\bar{\sigma}[\Delta] < \frac{1}{\bar{\sigma}[M]} \tag{2.55}$$

and hence

$$\mu_U[M] = \bar{\sigma}[M] \tag{2.56}$$

Note that this case reduces to the small-gain theorem.

For these two cases, it is clear that

$$\mu_{\delta I}[M] \leq \mu_U[M] \tag{2.57}$$

and more generally, whenever we have two structured model sets, $\mathcal{D}_1$ and $\mathcal{D}_2$ with $\mathcal{D}_1 \subseteq \mathcal{D}_2$, then

$$\mu_{\mathcal{D}_1}[M] \leq \mu_{\mathcal{D}_2}[M] \tag{2.58}$$

2.7 Performance

Returning once more to the feedback loop in Figure 2.1 with model set G, the second job of controller K is to achieve specified levels of performance for a number of criteria and again do this for all G's in the model set. Performance criteria usually include some objectives expressed in the time domain, others expressed as random response statistics, and still others expressed as frequency domain requirements. As a result, performance analyses of feedback systems generally include all three approaches — time domain simulation, covariance analyses, and frequency domain analyses.

Each of these approaches is discussed below in the context of a generic feedback diagram, which is introduced next.

2.7.1 A More General Feedback Diagram

We have already seen how the basic feedback loop in Figure 2.1 can be redrawn as a generic block diagram, Figure 2.6, consisting of an operator M (derived from G and

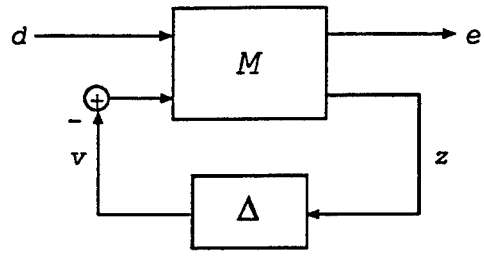


Figure 2.8: Robust Performance Analysis Setup.

K) with diagonally structured bounded uncertain elements in feedback around it. This simple generic form proves effective for robust stability analyses using the small-gain theorem and/or structured singular values. It is inadequate, however, when we want to analyze performance of the feedback system. To do the latter, we introduce the more general diagram in Figure 2.8. This new diagram consists of a larger operator, still called M and still derived from G and K, with uncertain elements in feedback around it, but also with selected external inputs and selected error outputs. The external inputs comprise a vector of all disturbances and/or commands that drive the closed-loop system, whereas the errors comprise a vector of all signals that are intended to be small. In fact, all performance criteria will be expressed in terms of making selected error signals small enough in the presence of specified external inputs.

To illustrate this, consider the basic feedback loop from Figure 2.1 with G being the multiplicative model set from equation (2.42). Let commands, y_c, be the only external inputs (i.e., $d = 0$, $n = 0$). Suppose we want the command-following error, e, to be small in the presence of these inputs. Then, using (2.25) and (2.26), it is easy to verify that the generic diagram in Figure 2.8 describes this situation with $\Delta = \Delta_m$ and M given by

$$\begin{bmatrix} e \\ z \end{bmatrix} = M \begin{bmatrix} y_c \\ v \end{bmatrix} \quad M = \begin{bmatrix} S & -S \\ T & -T \end{bmatrix} \tag{2.59}$$

We will see later that this general setup can be used quite broadly to model performance objectives. External inputs can be chosen to represent commands, sensor noise, gusts, and so on. Errors can be chosen to measure command following, handling quality, ride quality, gust loads, and so on. These inputs and outputs can be scaled to adjust their relative magnitudes, they can be frequency-shaped to adjust their spectra, and they can be normalized such that unit magnitudes represent performance within specs. Much more will be said about this later in connection with specific design methods.

2.7.2 Time Domain Simulation

One of the easiest ways to assess the performance of the closed-loop system in Figure 2.8 is to integrate its differential equations. Such simulations may be carried out with models of varying complexity. A strictly linear model might suffice for checking command-following

capability or responsiveness. A more complex model might include a linear vehicle model with linear actuator dynamics, but nonlinear control effector position and rate limits. A highly nonlinear model might include nonlinear aircraft equations of motion, a nonlinear aerodynamic database, nonlinear sensor and actuator models, and modeled delays.

Whatever the complexity of the model used in a simulation, the differential equations are integrated subject to external forcing functions that effect responses. These external inputs can take a variety of forms, but a few standard ones — steps, pulses, and doublets — are commonly used throughout the controls community — flight controls or otherwise. The step is the most general of these, since the others can be represented as linear combinations of delayed and scaled steps. For strictly linear systems, the principle of superposition ensures that pulse or doublet inputs provide no more performance information than a simple step input.

Step responses for linear systems give performance information that allows us to quantify time constants, natural frequencies, dampings, overshoots, and steady-state errors.

For nonlinear systems, the size and shape of the input is more critical. Responses to small enough inputs should be in close agreement with those from linear simulations. Larger inputs and inputs that change directions abruptly are more likely to show signal-limiting effects, push the range of validity of linear approximations, or cause coupling between aircraft axes. Indeed, the principal strength of time domain simulation is its ability to assess performance in strongly nonlinear regimes. For this reason, high-fidelity simulations, often including pilots and hardware in the loop, are almost always the last verification steps before control systems are cleared for flight. It is important to remember, however, that simulations alone have limited value when evaluating robust performance over large model sets. Each simulation run represents only a single model. Extensive Monte Carlo techniques are required to explore the model set, and infinite sets can never be completely verified.

2.7.3 Covariance Analysis

One of the simplest examples of an infinite model set that can be explored only approximately by time domain simulation is the set

$$\mathcal{S} = \{e(s) \mid e = M(s)d;\ d = \text{random process}\} \tag{2.60}$$

This is just a known linear system, with a known linear controller, forced by a stochastic process. We can evaluate the performance of this system by simulating the model with random number generators used to approximate the stochastics. In fact, by running the simulation often enough, we can assess performance criteria such as RMS output or RMS control effort to any prescribed accuracy. Technically, however, it takes infinitely many time domain runs to compute the statistics precisely.

It is well known, of course, that statistics of linear models driven by stochastic processes can be computed directly using a method called covariance analysis. In particular, the

covariance matrices of the states and outputs, P_x and P_e, respectively, of a linear system

$$\begin{aligned} \dot{x} &= Ax + Bd \\ e &= Cx \end{aligned} \tag{2.61}$$

driven by a zero-mean, white noise random process, d, are given by

$$\begin{aligned} \dot{P}_x &= AP_x + P_xA^T + B\Gamma_dB^T \\ P_e &= CP_xC^T \end{aligned} \tag{2.62}$$

where Γ_d is the intensity of the process input d. It often suffices to examine only the steady-state covariance. In this case, the differential equation for P_x above is replaced by the steady-state Lyapunov equation:

$$0 = AP_x + P_xA^T + B\Gamma_dB^T \tag{2.63}$$

Root-mean-square (RMS) performance of the states and outputs is determined by examining the square roots of the diagonal elements of P_x and P_e.

Coloring of white noise input is accomplished by augmenting the state-space with the dynamics associated with the coloring process (e.g., the standard Dryden spectra for gust inputs). For flight control applications, covariance analysis is most commonly used for evaluating performance with respect to gusts and sensor noise.

Even though covariance analyses provide precise statistics over infinitely many random signals, it is important to note that they still apply to only one linear model from the model set (i.e., for one Δ in Figure 2.8 at a time). To assess robust performance, for example, worst-case statistics over the full model set, we must repeat the analyses for many candidate models and pick the worst result. Algorithms that can compute such worst-case statistics directly are currently being developed in the research literature.

2.7.4 Frequency Domain Analysis

Besides time domain and statistical performance objectives, most feedback systems also have objectives that are most conveniently expressed in the frequency domain. Examples are disturbance rejection requirements for periodic signals, command tracking requirements over specified bandwidths, bounds on worst-case signal amplification, and model matching to equivalent systems. For appropriate choices of signal vectors d and e, these various requirements can all be expressed as magnitude specifications on the closed-loop transfer function from d to e in Figure 2.8,

$$\bar{\sigma}[M_{cl}(\Delta, j\omega)] < \text{PerfSpec}(\omega) \text{ for all } \omega \tag{2.64}$$

Again, this condition must be satisfied for the nominal system ($\Delta = 0$) as well as for all perturbations $\{\Delta \mid \Delta \epsilon \mathcal{D},\ \bar{\sigma}[\Delta] < r\}$. Evidently, the nominal condition can be confirmed by simply comparing singular-value plots of the nominal transfer matrix, the upper left

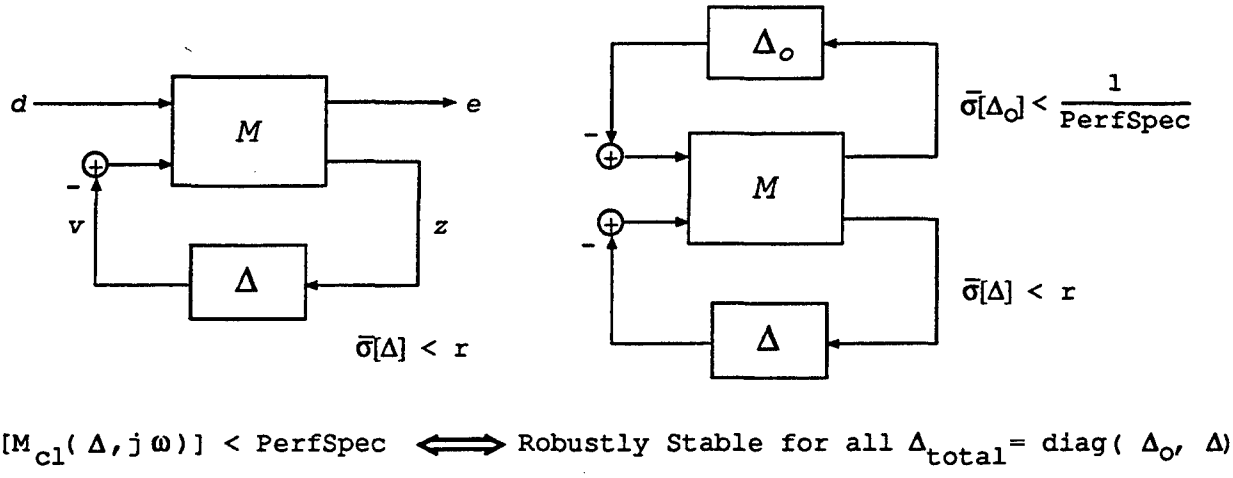


Figure 2.9: Performance/Stability Equivalence.

block M_{ed} of M, against the specification. Like time domain and covariance objectives, therefore, nominal performance verifications are easy. Unlike these other objectives, however, robust performance verifications are easy as well.

Frequency domain robust performance verifications turn out to be easy because the magnitude condition in equation (2.64) is equivalent to a stability condition that can be readily tested with structured singular values. To see this, consider the small-gain robust stability test introduced earlier in equation (2.44). This test ensures stability if and only if the magnitude of an uncertainty block connected across the terminals of stable transfer matrix, M, is smaller than $1/\bar{\sigma}[M]$. Hence, we know that if (2.44) is satisfied, we can connect such a block of size strictly less than 1/PerfSpec(ω) from the e-terminals of Figure 2.8 to the d-terminals without destroying the system's stability. On the other hand, if (2.44) is not satisfied, then the block would destabilize. So the robust performance test for Figure 2.8 is exactly the same as a robust stability test with respect to the original Δ and a new block of size 1/PerfSpec connected from e to d. This equivalence is summarized in Figure 2.9.

Note that the total uncertainty structure, including the added block, has a block-diagonal form, $\mathcal{D}_{total} = \{\Delta_{total} \mid \Delta_{total} = \text{diag}(\Delta_o, \Delta)\}$, which can be easily tested with structured singular values. The added uncertainty block is often called a *fictitious perturbation*, Δ_o, because it is not actually present in the feedback loop. Robust performance analyses based on this performance/stability equivalence find many uses in Section 11 of these guidelines.

2.8 Fundamental Limitations on Control Designs

In the next section, we begin discussing the control system design process — how to go about designing controllers K for the feedback loop in Figure 2.1 that achieve robust stability and robust performance for the various objectives of flight control. Before delving into these topics, however, it is important to understand that feedback loops have limitations. We cannot always achieve all objectives, no matter how omnipotent we are as designers or how powerful our design tools are. Instead, feedback design is an engineering art of compromise, trading off some objectives against others to achieve an overall most pleasing design.

Feedback limitations come in two flavors. First, there are hardware limits of various components in the loop — actuators that are too weak or too slow to oppose the disturbances, sensors that are too noisy to read signals accurately, computers that are too slow or small to execute our algorithms, and so on. Provided there is a favorable case of programmatic cost/benefit, such limits can often be alleviated by hardware changes.

There is a second class of limitations, however, that cannot be alleviated and are not always well understood by designers. We call these *fundamental* limitations. They include three major sources: (1) irreducible algebraic tradeoffs, (2) adverse multivariable condition numbers, and (3) analytic consequences of stability and causality. These three sources are discussed further below.

2.8.1 Irreducible Algebraic Tradeoffs

In the performance analysis section above, we introduced Figure 2.8 as a generic block diagram to analyze robust performance of feedback systems. For frequency domain objectives, in particular, the operator, M, in this figure was replaced by its transfer matrix, $M(s)$, and the structured singular value, $\mu_{\mathcal{D}_{total}}[M(j\omega)]$, was used to verify robust performance. It is clear from this setup that robust performance improves as we make the entire matrix $M(j\omega)$ small (e.g., the smaller we make the matrix, the larger the tolerable perturbations, Δ_o and Δ). Recall that M is a function of the nominal plant model, G, and the controller, K. So the obvious question is: "Can we choose K to make M as small as we wish?, and if not, why not?".

For most interesting control problems, the answer to this question is no, and one reason is that M includes *algebraic constraints* that prevent all blocks of M from being small simultaneously. A simple example is given by the M matrix in (2.59). Recall that this matrix describes the objective of achieving small command-following errors, e/y_c, in the presence of multiplicative uncertainties, Δ_m. We see that the matrix is determined entirely by the sensitivity, S, and complementary sensitivity, T. If we make T small, we have good stability robustness with respect to Δ_m, and if we make S small, we have good nominal performance (stability robustness with respect to a fictitious perturbation, Δ_o). However, we cannot make both matrices small simultaneously because, as shown

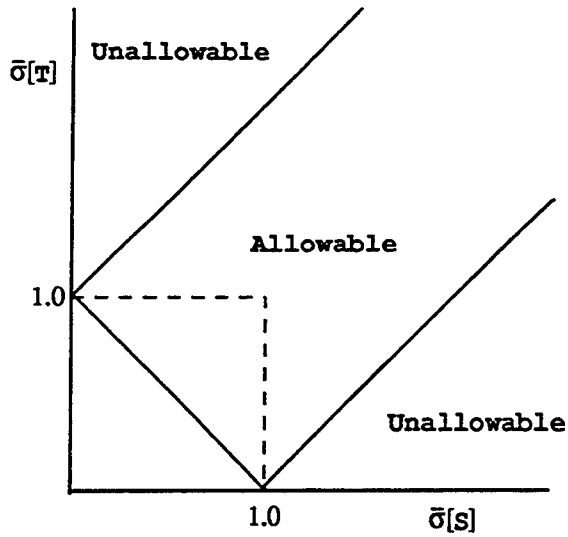


Figure 2.10: S and T Tradeoffs.

earlier, they are complements: $S(s) + T(s) = I$. It is easy to verify that complementary property restricts $\bar{\sigma}[S]$ and $\bar{\sigma}[T]$ to the region shown in Figure 2.10.

Evidently, to make $\bar{\sigma}[S]$ very small (compared with unity), we must let $\bar{\sigma}[T]$ approach unity, and vice versa. So there is a fundamental tradeoff between robust stability and nominal performance, and every controller necessarily represents a compromise between the two.

2.8.2 Multivariable Condition Numbers

Multivariable systems suffer additional algebraic limitations associated with the condition number, $\kappa[G]$, of their transfer matrix. This number is the ratio between the largest singular value of G and its smallest singular value.

$$\kappa[G] = \frac{\bar{\sigma}[G]}{\underline{\sigma}[G]} \tag{2.65}$$

In numerical analysis, the condition number is often used to measure the degree of singularity of matrices. A large number is undesirable because the matrix is close to being singular. In feedback loops, the condition number is important because it bounds differences between sensitivity and complementary sensitivity matrices defined for the outputs and inputs of the plant model. Note that the S and T matrices we have used so far are defined in terms of the loop transfer function, $L_o = GK$, for the loop broken at the output. Hence, they are more correctly called output sensitivity, S_o, and output complementary sensitivity, T_o, respectively. Analogous functions defined in terms of the input

loop transfer function, $L_i = KG$, are

$$\begin{array}{lrcl} \textit{Input sensitivity}: & S_i & = & (I+KG)^{-1} \\ \textit{Input complementary sensitivity}: & T_i & = & (I+KG)^{-1}KG \end{array} \tag{2.66}$$

These functions arise, for example, in stability robustness tests for uncertainties at the plant input (e.g., M in (2.46)) and for performance requirements placed on magnitudes of control signals. The important point is that, unlike SISO feedback loops, the S_i and T_i matrices of multivariable systems can differ substantially from S_o and T_o. In fact, using singular-value inequality results from Appendix ??, we can show that

$$\begin{array}{rcl} \bar{\sigma}[S_i] & \leq & \kappa[G]\bar{\sigma}[S_o] \\ \bar{\sigma}[T_i] & \leq & \kappa[G]\bar{\sigma}[T_o] \end{array} \tag{2.67}$$

So whenever $\kappa[G]$ is large, a good compromise achieved between sensitivity and complementary sensitivity at the output does not necessarily translate into a good compromise between these matrices at the input. As a result, the overall design can be quite poor. For this reason, it is important to include all uncertainty blocks, at both plant inputs and outputs, and all performance error signals, including controls, in the overall robust performance setup of Figure 2.8. In that way, all algebraic limitations, including those caused by poor plant condition numbers, are reflected in the overall matrix $M(s)$.

2.8.3 Consequences of Stability and Causality

Note that the limitations discussed so far are due to algebraic constraints built into the matrix $M(s)$ pointwise at each fixed frequency, $s = j\omega$. There are also *functional constraints* on $M(s)$ at $j\omega$, imposed by values M takes on at other frequencies. These functional limitations are intimately tied to the fact that our feedback systems are stable and causal. Our systems are not allowed to diverge for any bounded input, and they cannot look ahead into the future to determine current control actions. These physical constraints have the mathematical consequence that all closed-loop transfer functions are analytic everywhere in the right-half plane. This is a very special property. Technically, it means that the closed-loop transfer functions are bounded and infinitely differentiable everywhere in the right half plane. Practically, it means that their values at any one point, $s = s_1$, are closely tied to their values everywhere else in the right half plane.

2.8.3.1 Bode Gain/Phase Relations

Among the consequences of this special property are the *Bode gain/phase relations* [Horowitz 1963]. These show explicit connections between the rate of change of magnitude and phase of stable scalar transfer functions. In general, these connections take the form of integral expressions, but they are simple and very familiar to all classical designers for special rates of change: For example, -n × 20 dB/decade magnitude attenuation corresponds to -n × $\pi/2$ radians of constant phase. These connections are

fundamental in SISO design. We cannot, for example, rapidly attenuate from high loop magnitudes to low ones and still achieve stable crossover. There would simply be too much phase lag. Instead, the better part of a decade is required to rolloff stably through the crossover region.

Multivariable loops have similar limitations. If the loop gain, $\bar{\sigma}[GK]$, is large below crossover ($\bar{\sigma}[S]$ small for good nominal performance), it cannot be attenuated rapidly and still achieve stable crossovers. Again, the better part of a decade is required to attenuate through the crossover region, and often several decades are needed if the condition number of the loop is large.

2.8.3.2 Bode Integral

A more precise way to express the impact of the Bode gain/phase relations is through a conservation law of feedback control, known as the *Bode integral*:

$$\int_0^\infty \ln|S(j\omega)|d\omega = 0 \tag{2.68}$$

This integral applies to all SISO feedback systems with stable loop transfer functions satisfying $|GK| \leq k/\omega^2$ at high frequencies. In words, the integral states that the natural logarithm of the sensitivity function of such systems, integrated over frequency, must be zero. This holds for every stabilizing controller K, no matter how it was designed or who designed it. Since performance improvements correspond to $|S| < 1$ ($\ln|S| < 0$) and deteriorations correspond to $|S| > 1$ ($\ln|S| > 0$), it implies that performance improvements in one frequency range must necessarily come at the expense of deteriorations in another range. Again, the control design is just a specific compromise between these regions.

2.8.3.3 Right-Half-Plane Singularities

Other consequences of analyticity have to do with right-half-plane singularities of open-loop transfer functions (e.g., poles and zeros of GK in the half plane). Such singularities impose specific values on all closed-loop transfer functions. For example, suppose that the scalar function G has poles, p_i, $i = 1, \ldots, N_p$, with $\mathrm{Re}(p_i) > 0$, and zeros, z_j, $j = 1, \ldots, N_z$, with $\mathrm{Re}(z_j) > 0$. Then it follows that the sensitivity function must satisfy

$$\begin{array}{rcll} S(p_i) & = & (1 + G(p_i)K(p_i))^{-1} = 0 & \text{for all } i \text{ and all stabilizing } K\text{'s} \\ S(z_j) & = & (1 + G(z_j)K(z_j))^{-1} = 1 & \text{for all } j \text{ and all stabilizing } K\text{'s} \end{array} \tag{2.69}$$

The first result holds because G is infinite at p_i, and the second holds because G is zero at z_j. $K(s)$ cannot cancel the poles or zeros in either case without violating stability. The multivariable generalizations of these constraints are that the matrix $S(s)$ must be singular at right-half-plane poles of G, and it must have a unit eigenvalue at transmission zeros. Similar constraints also apply to various other closed-loop transfer matrices.

Because closed-loop transfer functions are analytic and take on specific values in the half plane, their values elsewhere are also constrained. For example, the presence of right-half-plane poles modifies the Bode integral to the following form:

$$\int_0^\infty \ln|S(j\omega)|d\omega = \pi \sum_{i=1}^{N_p} \mathrm{Re}(p_i) \tag{2.70}$$

Note that this again shows conservation of $\ln|S|$, but the net total sensitivity level over all frequencies is higher because of right-half-plane poles.

In addition, systems with right-half-plane zeros — so called *nonminimum phase* systems — satisfy one *Poisson integral* constraint for each z_j [Freudenberg 1985]:

$$\int_{-\infty}^\infty \ln|S(j\omega)|\theta_{z_j}(\omega)d\omega = \begin{cases} 0 & N_p = 0 \\ \pi \ln[B_p(z_j)] & N_p > 0 \end{cases} \tag{2.71}$$

where

$$B_p(z_j) = \prod_{k=1}^{N_p} \frac{p_k - s}{\bar{p}_k + s}\bigg|_{s=z_j} \tag{2.72}$$

is an all-pass product of right-half-plane poles evaluated at the zero in question (often called a *Blashke product*), and

$$\theta_{z_j}(\omega) = \frac{\mathrm{Re}(z_j)}{[\omega - \mathrm{Im}(z_j)]^2 + \mathrm{Re}(z_j)^2} \tag{2.73}$$

is a *constraint weighting function.* Note that the weighting function is nonnegative for all ω but drops quickly to zero at frequencies greater than $|z_j|$. In effect, this makes (2.71) behave just like a Bode integral except over a shortened finite frequency range, approximately $|\omega| < |z_j|$. Sensitivity improvements in some part of this range, say $|\omega| < \Omega$, must now be paid for with deteriorations in the remainder of the range, $\Omega < |\omega| < |z_j|$. So the ratio of deterioration to improvement is given approximately by the ratio $1/(|z_j|/\Omega - 1)$. Evidently, the smaller $|z_j|$ is relative to Ω, the greater the performance deteriorations must be.[3] This fundamental balance between sensitivity improvement and deterioration is illustrated in Figure 2.11.

Constraints like the Bode and Poisson integrals are much more complicated for multivariable systems. However, we can state bounds for them by simply replacing magnitudes with singular values. For example, the Poisson integral for multivariable systems can be written as:

$$\int_{-\infty}^\infty \ln \bar{\sigma}[S(j\omega)]\theta_{z_j}(\omega)d\omega \geq \begin{cases} 0 & N_p = 0 \\ f(z_j, p) \geq 0 & N_p > 0 \end{cases} \tag{2.74}$$

Although this inequality is not as sharp as (2.71), it clearly shows that multivariable systems suffer from right-half-plane singularities in essentially the same way.

[3]In contrast, note that the Bode integral always leaves an infinite range over which to distribute the deteriorations. Theoretically they can be made arbitrarily small. In practice, of course, we have only finite bandwidths to work with, and similar ratios apply with $|z_j|$ replaced by the bandwidth of the control hardware.

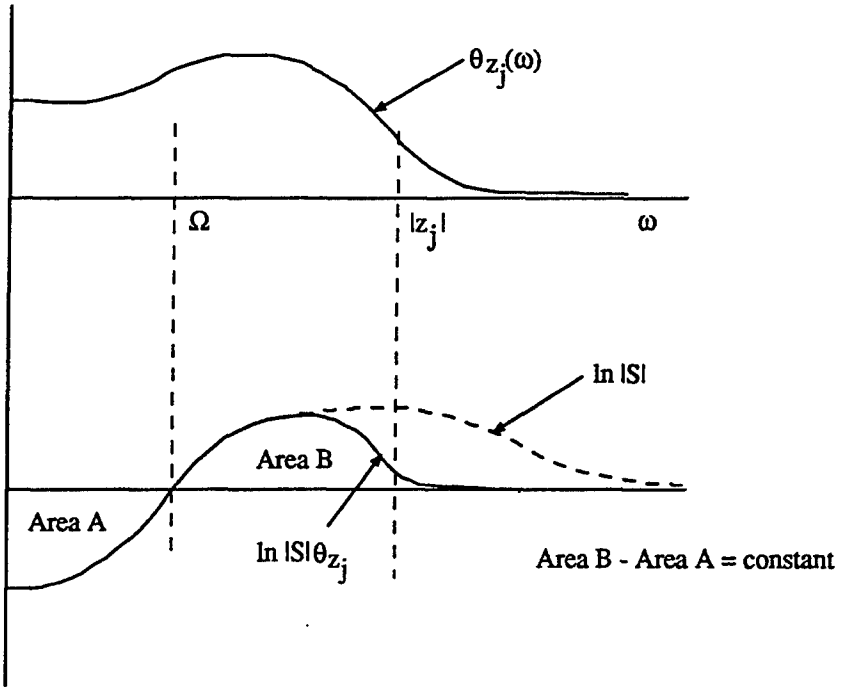


Figure 2.11: Impact of Right-Half-Plane Zeros on Sensitivity.

Section 3
The Control Design Process

The control design process can be decomposed into the following broad tasks/steps:

- Requirements Definition
- Modeling
- Control Design
- Control Analysis and Simulation
- Control Implementation

These steps are carried out in an iterative fashion until requirements are satisfied. When requirements are unachievable, tradeoffs or negotiations become unavoidable. Intermediate tradeoffs may sometimes result in aircraft configuration modifications, but more often can be satisfied by simpler changes such as in a control law gain. A flowchart that depicts the formal control design process is shown in Figure 3.1.

Modern computers, together with control design and analysis tools available today, allow much of the design process to be automated. If a requirement can be expressed in terms of linear tests, the associated design and analysis algorithms can usually be automated. In many cases, computer tools today generate results so rapidly that interactive use of them has become a practical reality. Indeed, the speed at which they produce results often exceeds the control designer's ability to absorb them. This capability was unheard of by earlier generations of control system designers.

Good control designers today still do, however, share much in common with previous generations of control designers. It is imperative that they know: (1) whether the models they are using are reasonable, (2) how to translate design requirements into assumptions for the design process, (3) how to interpret results that tools are producing, and (4) how to exploit previous design experience from other applications. Experience and analytical skills (without the aid of computer tools) are as important today as they were in the past.

The remainder of this section describes the five major steps in the design process.

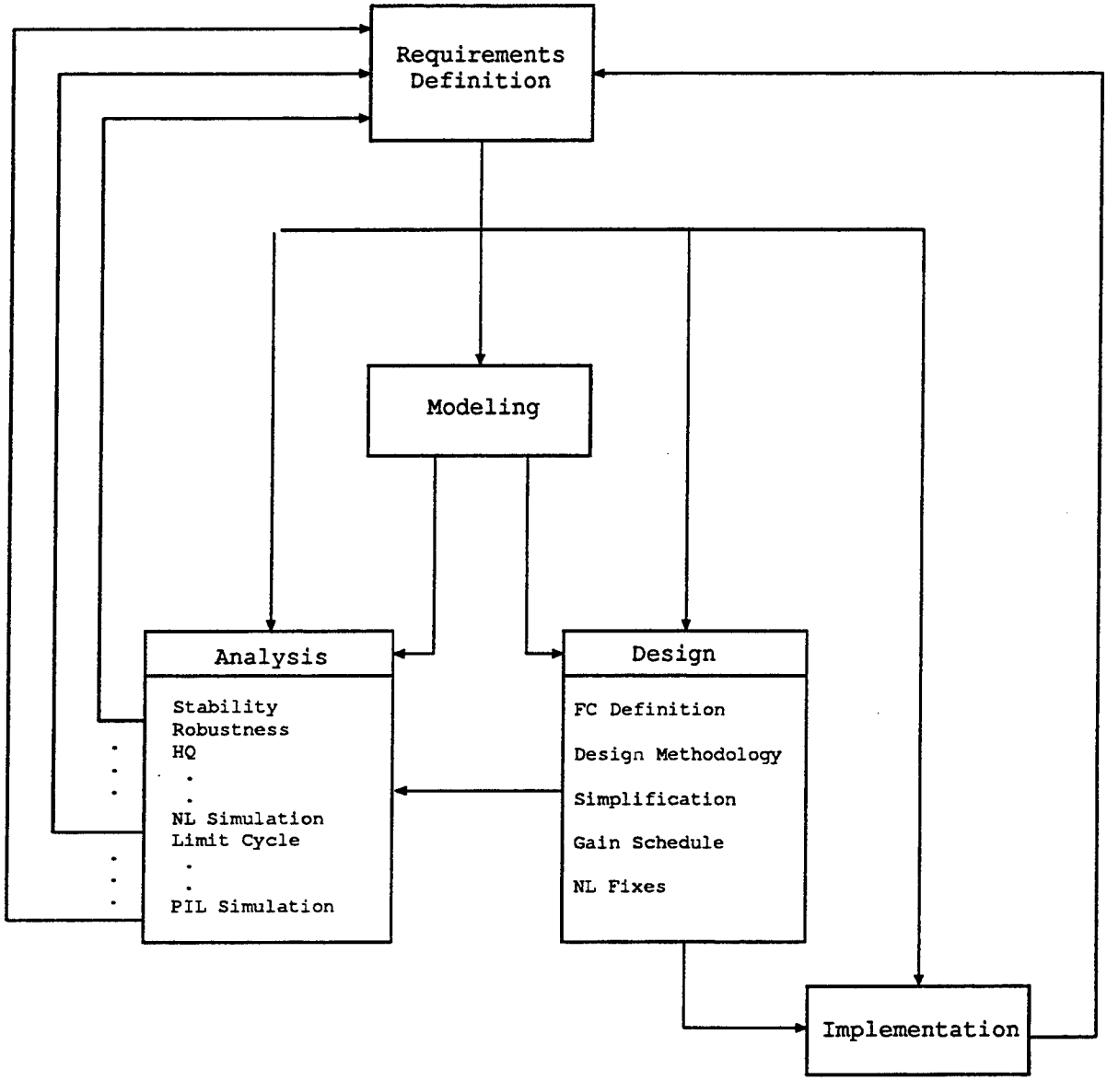


Figure 3.1: Flowchart Describing the Control Design Process.

3.1 Requirements Definition

Requirements define the problem. Either implicitly or explicitly, design iterations imply tradeoffs in requirements. An example of an explicit tradeoff is a design requirement that specifies a given level of attenuation for a bending mode. Design iterations in the associated notch filter design focus on achieving the specified attenuation. An example of a design iteration that is less explicit is the adjustment of a pilot stick gain during a simulation/tuning exercise. In such cases, it is the responsibility of the design engineer to understand the design trade that is being made and what requirements are affected. It is unfortunate that in many instances, such trades are not documented and their impact is not fully understood.

Although in practice many design iterations proceed first by intuition (like the stick gain adjustment mentioned above), the formal process demands that all design iterations be first cast as a requirements trade before design iterations can proceed.

Note that requirements feed the control design steps for modeling, design, analyses, and implementation. A simple example demonstrates this dependence. The requirement for appropriate phase margin at the control input necessitates that an appropriate actuator model exist, that the control design account for the phase margin, that an analysis test can be constructed to ensure that the desired margin is achieved, and that the software implementation update rate is high enough that it does not interfere with these margins.

In these guidelines, requirements are discussed briefly in Section 1.2 and more thoroughly in Sections 4.2, 4.3, and 4.4. Flight control requirements are discussed extensively in two references [MIL-STD-1797 1987, Lockheed Martin 1994].

3.2 Modeling

The control design and analyses steps that follow require suitable models. For the first time through the design process, significant effort is usually required to develop a working model. And this model must be checked, scrutinized, and refined throughout the course of the control law development process. The control design model forms the basis for all designs and analyses. Generally, models for flight control imply a high-fidelity nonlinear rigid-body model, but sometimes they also include dynamics associated with elasticity. For many design methodologies, linear models must be extracted from nonlinear models. Subsequent modeling steps often include the addition of elastic modes (if they were not included at the outset) to address additional requirements for attenuation of high-frequency dynamics.

The model development step is not strictly devoted to aircraft dynamics. Actuator and sensor dynamics must also be addressed and modeled. Well-known gust and turbulence models need to be parameterized for the particular application as well. Other distur-

bances, such as weapons release and discretization effects, must also be understood and modeled.

In these guidelines, models are discussed in very general terms in Sections 2.2, 2.3, and 2.5. Aircraft models in general are discussed in Section 4.1, while models specific to the F-16, F-117, and YF-22 aircraft are described elsewhere [Lockheed Martin 1994].

3.3 Control Design

The control design step begins with the selection of flight conditions at which the control design is to be accomplished. Flight conditions are selected to cover a specified flight envelope. The flight envelope is usually defined in terms of altitude and velocity but may also include angle-of-attack, power settings, turn rate, and landing versus up-and-away configuration variations. Typically there are tens of flight conditions that are of particular interest, although the actual number depends on the size of the flight envelope and mission requirements.

Next, a particular control design methodology is applied. For aircraft flight control, much of the design work and capture of design requirements is embedded in the choice of appropriate controlled variables and in the blending of control effectors. These are discussed in Sections 5 and 6. Much is said about three specific design methodologies in Part III of these guidelines. For linear designs (e.g., the eigenstructure and μ-Synthesis methodologies described in Sections 9 and 11, respectively), control designs are carried out at each flight condition. In contrast with this, the dynamic inversion-based design methodology produces a controller that covers the full flight envelope, at least to the extent that its internal database covers it. But even here, flight conditions at which the design is to be *evaluated* must still be defined.

As the control design matures and passes most of the analysis tests, the designer may wish to simplify either the aircraft model or the controller to facilitate more specialized control analyses or in preparation for implementation. This generally requires some type of model reduction. There are many ways to simplify a model (whether it is for the aircraft or the control law), but none are ideally suited for all purposes. The designer must ultimately rely on a variety of techniques using the one best suited to his/her specific needs. Several model reduction techniques are described in Appendix B.1.

Regardless of whether the control design needs to be simplified, the next step in the design process is the development of a gain schedule. With linear point designs, this generally requires approximating the control law gains and filter coefficients from each point design with continuous functions that span the flight envelope. Even for the dynamic inversion-based technique, there will inevitably be some handling-qualities-related variables (e.g., stick gains) that need to be scheduled. The three main steps in the gain scheduling process are the: (1) decision regarding which gains to schedule and which to treat as functions of the others, (2) determination of which variables to use as scheduling parameters, and (3)

actual definition of the fit function. Like model reduction, the gain scheduling process is not highly formalized and relies to a great extent on ad hoc steps pieced together by the designer. Some gain scheduling considerations and common scheduling techniques are presented in Appendix B.2.

For some control laws, the gain scheduling process may be the first step in developing a practical nonlinear control law. Another potentially significant nonlinear design issue is that of limits. The impact of control limits is often insignificant, but occasionally it can be very significant such that it must be accounted for in the design process. SISO examples that illustrate these behaviors appear in Figure 3.2, where pitch rate response due to pilot control stick is compared for two cases — limited versus unlimited surface motion. In each case, a proportional-plus-integral control law is used to control a short period approximation of an unstable aircraft. The pilot input is a doublet starting at 1 second, reversing at 3 seconds, and returning to zero after 5 seconds. The amplitude of the doublet is varied for the three cases. In the top two plots of the figure, the impact is not significant, whereas in the middle two plots, the limit makes the difference between stability and instability. In the bottom two plots, an integrator anti-windup scheme like that discussed in Section 8.1 is employed to reduce the impact of surface limits.

The multivariable case is even more difficult, because limiting of one effector may have a significant impact on more than one control performance objective. For special cases, namely SISO and/or low-order state dimension, there exist limited theoretical results. Some can be generalized to the multivariable case, but only a few. Approaches used successfully in multivariable flight control designs include: (1) optimization, subject to constraints, (2) describing function analysis for stability and limit-cycle analysis, (3) phase-plane analysis, and (4) a variety of analyses based on much practical experience. Appendix B.3, presents a number of examples where design and analysis in the presence of control limits is considered.

3.4 Control Analysis

Intimately related to the control design step is the control analysis step in the design process. For a requirement to be valid, it must be testable. Thus, as a requirement is incorporated into a design, the analysis test provides an immediate check on whether the requirement is met. As such, the analysis test forms the basis for the design iteration decision. Any failure of an analysis test ultimately leads to a design iteration or a redefinition of requirements. For conventional linear point designs, issues of stability, robustness, handling qualities, and the like are easily tested. Much of the design iteration at this level of the design/analysis steps can be automated. But the designer is still obliged to oversee the design process and take responsibility for the design iteration. The next level of analysis as the design matures includes nonlinearities. Typically, this demands closed-loop nonlinear simulation. It can also include describing function analysis, which is described in Appendix B.4, to assess the potential for limit cycles. Another level of

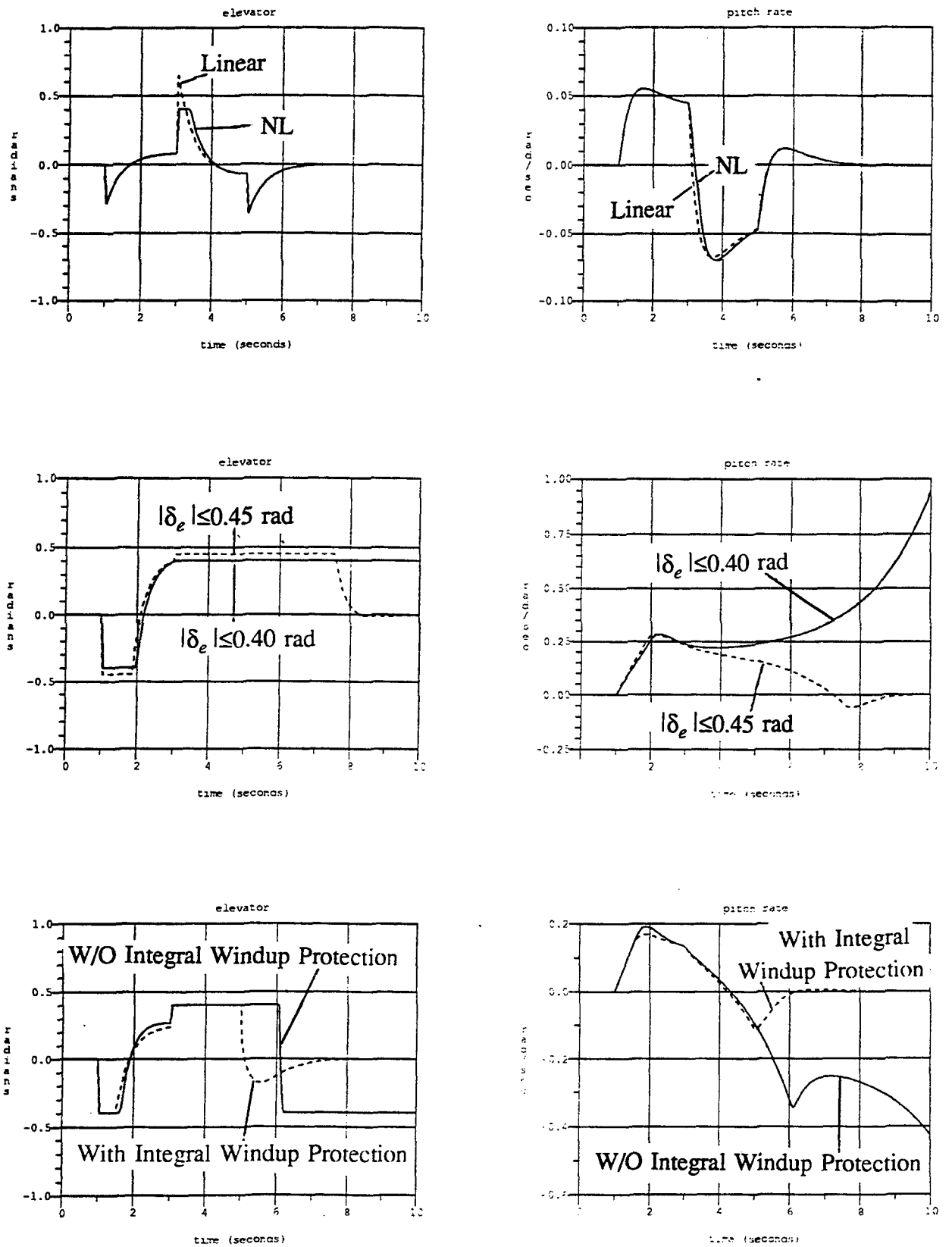


Figure 3.2: Time Histories From a Short-Period Simulation With Saturation.

nonlinear analysis is pilot-in-the-loop simulation. As with any analysis test, the success of such analyses must be judged by the control designer and design iterations taken as deemed appropriate.

These guidelines present analysis-related topics in subsections 2.6, 2.7, and 2.8.

3.5 Control Implementation

Once the design has passed all of its analysis gates, considerable effort is still required to take the algorithm to a flightworthy state. While it is not the intent of these guidelines to address these implementation issues, there are some design implications that control designers should consider during the design process. Among them are the protection of integrators and the effects of digital control. If these issues are not addressed well before implementation, it can be very costly to redo the control design and associated analyses. Section 8 addresses some of these important practical considerations.

THIS PAGE LEFT INTENTIONALLY BLANK

Part II

Flight Control Specific

Part II Overview

In this part of the guidelines, we address certain flight control-specific topics that are independent of the particular multivariable design methodology chosen for a design. These topics include aircraft modeling and requirements, selection of variables to be controlled, use of multiple control effectors, and issues associated with implementation of control laws.

All flight control problems begin with a definition of the aircraft and specifications for desired vehicle performance. Accordingly, the first section of Part II deals with defining the aircraft in terms of mathematical models. Since each phase of the control law design often calls for different modeling assumptions, and consequently different models, we discuss two types of aircraft models. In addition, we discuss typical performance requirements and models used to evaluate performance with respect to the specifications.

In any control application, we must select the quantities to be controlled and determine how they are related to the commands that drive the system. Since most aircraft are designed with moment-producing control effectors, it is natural to choose control variables consistent with pilot desires for these rotational motions. The second section discusses appropriate choices for controlled variables for general aircraft applications. The choices made are applicable to all flight regimes but can be simplified to familiar classical feedback variables for "typical" fighter flight envelopes.

Because we usually want to control three rotational motions, a variety of aircraft control effectors can often be used to produce the desired responses. The control designer must not only decide which control effectors to use but how they are to be used. Most control design methodologies, including the three chosen for these guidelines, can accommodate multiple control effectors. Ultimately, the control designer must "tell" the methodology how to use the effectors via its particular knobs. Ideally, we should make these choices up-front to separate them from the rest of the design process. The third section describes several methods for accomplishing this.

In the fourth section, we discuss some analyses that are particular to flight control law performance assessment.

Finally, the fifth section discusses several issues common to the implementation of control laws for real-world application.

Section 4

Aircraft Flight Control Problem and Requirements

The flight control problem definition begins with an aircraft that has a defined mission. For the aircraft to fly this mission, dynamic motion variables must behave in a particular way. Sensors must measure some of these variables, as well as the pilot's intended actions, and actuators must produce changes in them. The control design problem is to develop an algorithm that closes the loop between the sensors and actuators such that the aircraft accomplishes its mission. This development includes analysis and simulation to assess how well the aircraft performs its mission.

Whether the aircraft already exists or is under development, it usually has some unique dynamic characteristics. It may be aerodynamically unstable or nearly so, or there may be complex couplings between the various motions. The aircraft may have unique configuration characteristics (e.g., stealth-driven shape, tailless, forward-swept wings, oblique wing). It may also have a unique flight envelope (e.g., high speed, high altitude, high angle of attack, low speed). Finally, there may be special issues associated with other systems (e.g., propulsion, fuel management, thermal). In any case, these dynamic characteristics must be modeled, analyzed, and understood for successful flight control law development.

Mission aspects of interest to the flight control law designer require that the aircraft perform some specified maneuvers. As discussed in Section 1, the control law designer is particularly concerned with maneuvers that take place in a few seconds as opposed to minutes. These maneuvers involve rigid-body degrees of freedom of the aircraft. Although rotational motions are the most important, such rotations also induce translations. Common maneuver requirements include minimum and maximum rates of change of dynamic motion variables; that is, to achieve mission goals, certain motions must be sufficiently fast. For example, roll rates, pitch acceleration, time to bank, and other motions are constrained by aircraft or human limitations (e.g., load-factor or angle-of-attack limits).

Pilots execute maneuvers using longitudinal and lateral stick and rudder pedal motions and can also trim out stick and pedal forces with trim buttons and a trim wheel. Although pilots can affect aircraft motion through other means (e.g., throttle, speed brake, flaps, landing gear, leading edge surfaces), stick and pedal inputs have traditionally been the primary ones of interest in flight control law development. Based on past experience, pilots expect the aircraft to respond or handle in a desirable and predictable manner with a minimum of effort. Handling-qualities requirements are concerned with dynamic response characteristics of the aircraft following pilot inputs (e.g., time delays, natural frequencies, time constants, damping ratios).

Atmospheric and other disturbances also affect aircraft motion and play a significant role in flight control law development. The flight control law must attenuate the response of the aircraft to such disturbances, which in turn must be modeled, analyzed, and understood for a successful flight control system development.

4.1 Models

Different types of models are used for different purposes in flight control development. Much of the development can usually accomplished using rigid-body models, although elasticity does play a significant role in some cases. Models can be either linear or nonlinear, and both are sometimes used in flight control law development. Wherever possible, a special or simplified reduced-order model is used even when a high-fidelity model is available.

Two standard models forms will be used in this document: one for nonlinear equations and another for linear equations.

The standard form for nonlinear models is

$$\dot{x} = f(x, u) \tag{4.1}$$

$$y = h(x, u) \tag{4.2}$$

where x is the state, u is the control, and y is the output of interest. The functions f and h are arbitrary and often involve table lookup.

The standard form for the linear models is

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{4.3}$$

where x is the state, u is the control, and y is the output of interest. The matrices A, B, C, and D are constant.

4.1.1 Rigid-Body Models

Modeling of rigid-body aircraft motion is discussed in many textbooks [Etkin 1959, Etkin 1972, Etkin 1982, Blakelock 1965, Blakelock 1991, Roskam 1972, McRuer 1973]. Rigid-body models are described by six degrees of freedom and include forces and moments caused by gravity, aerodynamics, and propulsion. These equations of motion are nonlinear differential equations.

The major assumptions are that the:

- Aircraft is a rigid body,

Table 4.1: Choices for State Variables

Body Components		Flight Path Components	
Variable	Symbol	Variable	Symbol
Roll Rate	p	Roll Rate	p
Pitch Rate	q	Pitch Rate	q
Yaw Rate	r	Yaw Rate	r
Longitudinal Velocity	u	Velocity	V
Lateral Velocity	v	Sideslip Angle	β
Normal Velocity	w	Angle of Attack	α
Euler Roll Angle	ϕ	Bank Angle (about velocity vector)	μ
Euler Pitch Angle	θ	Flight-Path Angle	γ
Euler Yaw Angle	ψ	Heading Angle	χ
North Position	ξ	North Position	ξ
East Position	η	East Position	η
Altitude	h	Altitude	h

- Earth is an inertial reference frame,
- Aircraft mass properties are constant,
- Aircraft has a plane of symmetry.

Clearly other assumptions are possible (see the above textbooks) and sometimes these assumptions do not apply (e.g., hypersonic vehicles, oblique wing configurations).

Aircraft motion can be expressed in either of two coordinate systems:

- Aircraft system with coordinates $\hat{b}_x$ (positive forward), $\hat{b}_y$ (positive out the right wing), $\hat{b}_z$ (positive down) that are fixed in the body of the aircraft.
- Inertial system with coordinates $\hat{I}$ (positive north), $\hat{J}$ (positive east), $\hat{K}$ (positive down) that are fixed in the earth.

The dynamic motion of the aircraft can be expressed in terms of a system of state variables. Two choices for state variables will be considered here: The first is body components and is commonly used throughout the aircraft community; the second is flight-path components, from which point-mass approximations are easily derived. These state variable choices are summarized in Table 4.1.

The flight-path state variables can be derived from the body state variables as follows:

$$V = \sqrt{u^2 + v^2 + w^2} \tag{4.4}$$

$$\beta = \sin^{-1}\left(\frac{v}{\sqrt{u^2+v^2+w^2}}\right) \tag{4.5}$$

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{4.6}$$

$$\mu = \tan^{-1}\left(\frac{uv\sin\theta + (u^2+w^2)\sin\phi\cos\theta - vw\cos\phi\cos\theta}{\sqrt{u^2+v^2+w^2}(w\sin\theta + u\cos\phi\cos\theta)}\right) \tag{4.7}$$

$$\gamma = \sin^{-1}\left(\frac{u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\cos\theta}{\sqrt{u^2+v^2+w^2}}\right) \tag{4.8}$$

$$\chi = \tan^{-1}\left(\frac{u\cos\theta\sin\psi + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) + w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)}{u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)}\right) \tag{4.9}$$

The body state variables can be derived from the flight-path state variables as follows:

$$u = V\cos\beta\cos\alpha \tag{4.10}$$

$$v = V\sin\beta \tag{4.11}$$

$$w = V\cos\beta\sin\alpha \tag{4.12}$$

$$\phi = \tan^{-1}\left(\frac{\cos\gamma\sin\mu\cos\beta - \sin\gamma\sin\beta}{-\cos\gamma\sin\mu\sin\alpha\sin\beta + \cos\gamma\cos\alpha\cos\mu - \sin\gamma\sin\alpha\cos\beta}\right) \tag{4.13}$$

$$\theta = \sin^{-1}(\cos\gamma\sin\mu\cos\alpha\sin\beta + \cos\gamma\cos\mu\sin\alpha + \sin\gamma\cos\alpha\cos\beta) \tag{4.14}$$

$$\psi = \tan^{-1}\left\{\frac{(\sin\mu\sin\alpha - \cos\alpha\cos\mu\sin\beta)\cos\chi + [\cos\gamma\cos\alpha\cos\beta - \sin\gamma(\sin\alpha\cos\mu + \sin\beta\cos\alpha\sin\mu)]\sin\chi}{-(\sin\mu\sin\alpha - \cos\alpha\cos\mu\sin\beta)\sin\chi + [\cos\gamma\cos\alpha\cos\beta - \sin\gamma(\sin\alpha\cos\mu + \sin\beta\cos\alpha\sin\mu)]\cos\chi}\right\} \tag{4.15}$$

In the development that follows, we let T denote the propulsion force along the $\hat{b}_x$ direction and ℓ_p^c, m_p^c, n_p^c propulsion moments for roll, pitch, and yaw, respectively. Note that propulsion forces in the $\hat{b}_y$ and $\hat{b}_z$ directions have been neglected. Similarly, we let m denote the mass of the aircraft, I_{xx}, I_{yy}, I_{zz} the moments of inertia, I_{xz} the xz product of inertia, and g the gravitational acceleration. Note that products of inertia I_{xy} and I_{yz} are zero, due to our assumption of aircraft symmetry about the xz plane.

4.1.1.1 Atmospheric Motion

We model atmospheric motion relative to the earth with three translational velocities (v_x^a, v_y^a, v_z^a for wind from the North, East and down, respectively) and three rotational velocities (ω_x^a, ω_y^a, ω_z^a for wind about axes opposite to the unit vectors for North, East, and down, respectively). These components are rotated through the body Euler angles to obtain equivalent body components of atmospheric motion as follows:

$$\begin{bmatrix} u^a \\ v^a \\ w^a \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\sin\psi\cos\phi + \sin\theta\cos\psi\sin\phi & \cos\psi\cos\phi + \sin\theta\sin\psi\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \sin\theta\cos\psi\cos\phi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} v_x^a \\ v_y^a \\ v_z^a \end{bmatrix} \tag{4.16}$$

$$\begin{bmatrix} p^a \\ q^a \\ r^a \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\sin\psi\cos\phi + \sin\theta\cos\psi\sin\phi & \cos\psi\cos\phi + \sin\theta\sin\psi\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \sin\theta\cos\psi\cos\phi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \omega_x^a \\ \omega_y^a \\ \omega_z^a \end{bmatrix} . \tag{4.17}$$

Dynamic pressure is given by

$$\bar{q} = \frac{1}{2}\rho\left[(u+u^a)^2 + (v+v^a)^2 + (w+w^a)^2\right] \tag{4.18}$$

where ρ is air density at the aircraft's altitude.

Aerodynamic coefficients are functions of the net angle of attack,

$$\tilde{\alpha} = \tan^{-1}\left(\frac{w+w^a}{u+u^a}\right) \tag{4.19}$$

net sideslip angle,

$$\tilde{\beta} = \sin^{-1}\left(\frac{v+v^a}{\sqrt{(u+u^a)^2 + (v+v^a)^2 + (w+w^a)^2}}\right) \tag{4.20}$$

and net Mach number,

$$\tilde{M} = \frac{\sqrt{(u+u^a)^2 + (v+v^a)^2 + (w+w^a)^2}}{v_{sos}} \tag{4.21}$$

net angular rates

$$\tilde{p} = p + p^a \tag{4.22}$$

$$\tilde{q} = q + q^a \tag{4.23}$$

$$\tilde{r} = r + r^a \tag{4.24}$$

and all of the aerodynamic control surfaces (δ). Here v_{sos} is the speed of sound at the aircraft's altitude. Thus, the aerodynamic coefficients can be expressed as

$$\tilde{C}_k = C_k(\tilde{\alpha},\ \tilde{\beta},\ \tilde{M},\ \tilde{p},\ \tilde{q},\ \tilde{r},\ \delta) \quad \text{for } k = L,\ D,\ Y,\ \ell,\ m,\ n \tag{4.25}$$

Note that the functions indicated on the right hand side of equation 4.25 are wind tunnel or analytical estimates, typically in table lookup format. In any case, these functions are independent of atmospheric motion, which is introduced through the inputs to these functions denoted by the tilde above (e.g., $\tilde{\alpha}$, $\tilde{\beta}$). Likewise, we reference output aerodynamic coefficients on the left hand side of equation 4.25 to the effective free-stream motion, which includes both the atmospheric motion and the aircraft motion with respect to the inertial reference frame.

The equations of motion in sections 4.1.1.2 and 4.1.1.3 assume that lift is perpendicular to inertial velocity and drag is parallel to inertial velocity. Thus, we need a rotation from the effective free-stream directions to the inertial directions, which is given by

$$\begin{bmatrix} C_D \\ C_Y \\ C_L \end{bmatrix} = \begin{bmatrix} \cos(\tilde{\alpha}-\alpha) & 0 & -\sin(\tilde{\alpha}-\alpha) \\ 0 & 1 & 0 \\ \sin(\tilde{\alpha}-\alpha) & 0 & \cos(\tilde{\alpha}-\alpha) \end{bmatrix} \begin{bmatrix} \tilde{C}_D \\ \tilde{C}_Y \\ \tilde{C}_L \end{bmatrix} \tag{4.26}$$

Here aerodynamic forces and moments about the mass center are denoted by L for lift force, D for drag force, Y for side force, ℓ_a^c for rolling moment, m_a^c for pitching moment, and n_a^c for yawing moment. Lift, drag, and side force are mutually perpendicular, with lift and drag acting within the plane of symmetry and side force acting outside of this plane (i.e., $\hat{b}_y$ direction). We model these forces and moments with nondimensional coefficients, dynamic pressure ($\bar{q}$), wing reference area (S), mean aerodynamic chord ($\bar{c}$), and wing span (b) as follows:

$$L = \bar{q}SC_L \tag{4.27}$$

$$D = \bar{q}SC_D \tag{4.28}$$

$$Y = \bar{q}SC_y \tag{4.29}$$

$$\ell_a^c = \bar{q}SbC_\ell \tag{4.30}$$

$$m_a^c = \bar{q}S\bar{c}C_m \tag{4.31}$$

$$n_a^c = \bar{q}SbC_n \tag{4.32}$$

The aerodynamic coefficients are functions of aircraft state, control surface deflections, and atmospheric motion relative to the earth. We define state variables for aircraft motion relative to the earth but aerodynamic forces and moments relative to the atmosphere.

4.1.1.2 Body Components

Using body components, linear and angular velocity vectors are

$$\vec{V} = u\hat{b}_x + v\hat{b}_y + w\hat{b}_z \tag{4.33}$$

$$\vec{\omega} = p\hat{b}_x + q\hat{b}_y + r\hat{b}_z \tag{4.34}$$

where u, v, w are the body components of linear velocity, and p, q, r are the body components of angular velocity.

If we denote position and angular orientation of the aircraft relative to the earth by the Cartesian position coordinates, ξ, η, h, and the usual Euler angles, ϕ, θ, ψ, equations of motion for these state variables are

$$I_{xx}\dot{p} - I_{xz}\dot{r} = \ell_a^c + \ell_p^c + I_{yy}r\iota + I_{xz}pq - I_{zz}qr \tag{4.35}$$

$$I_{yy}\dot{q} = m_a^c + m_p^c - I_{xz}p^2 + I_{zz}pr - I_{xx}rp + I_{xz}r^2 \tag{4.36}$$

$$-I_{xz}\dot{p} + I_{zz}\dot{r} = n_a^c + n_p^c + I_{xx}pq - I_{xz}qr - I_{yy}pq \tag{4.37}$$

$$\dot{u} = rv - qw - g\sin\theta + \frac{(T - D\cos\alpha + L\sin\alpha)}{m} \tag{4.38}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi + \frac{Y}{m} \tag{4.39}$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi - \frac{(D\sin\alpha + L\cos\alpha)}{m} \tag{4.40}$$

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{4.41}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{4.42}$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta} \tag{4.43}$$

$$\dot{\xi} = u\cos\theta\cos\psi + v(-\sin\psi\cos\phi + \sin\theta\cos\psi\sin\phi) + w(\sin\psi\sin\phi + \sin\theta\cos\psi\cos\phi) \tag{4.44}$$

$$\dot{\eta} = u\cos\theta\sin\psi + v(\cos\psi\cos\phi + \sin\theta\sin\psi\sin\phi) + w(-\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi) \tag{4.45}$$

$$\dot{h} = u\sin\theta - v\cos\theta\sin\phi - w\cos\theta\cos\phi \tag{4.46}$$

4.1.1.3 Flight-Path Components

Using flight-path components, linear and angular velocity vectors are

$$\vec{V} = V(\cos\chi\cos\gamma\hat{I} + \sin\chi\cos\gamma\hat{J} - \sin\gamma\hat{K}) \tag{4.47}$$

$$\vec{\omega} = p\hat{b}_x + q\hat{b}_y + r\hat{b}_z \tag{4.48}$$

where V, χ, γ are the magnitude of the velocity vector, heading angle, and flight-path angle, respectively, and p, q, r are the body components of angular velocity.

If we let ξ, η, h denote position of the aircraft relative to the earth in Cartesian coordinates and μ, β, α denote body attitudes relative to the velocity vector, equations of motion for these state variables are

$$I_{xx}\dot{p} - I_{xz}\dot{r} = \ell_a^c + \ell_p^c + I_{yy}rq + I_{xz}pq - I_{zz}qr \tag{4.49}$$

$$I_{yy}\dot{q} = m_a^c + m_p^c - I_{xz}p^2 + I_{zz}pr - I_{xx}rp + I_{xz}r^2 \tag{4.50}$$

$$-I_{xz}\dot{p} + I_{zz}\dot{r} = n_a^c + n_p^c + I_{xx}pq - I_{xz}qr - I_{yy}pq \tag{4.51}$$

$$\dot{V} = \frac{1}{m}(-D\cos\beta + Y\sin\beta + T\cos\beta\cos\alpha) - g\sin\gamma \tag{4.52}$$

$$\dot{\chi} = \frac{1}{mV\cos\gamma}[D\sin\beta\cos\mu + Y\cos\mu\cos\beta + L\sin\mu + T(\sin\mu\sin\alpha - \cos\mu\sin\beta\cos\alpha)] \tag{4.53}$$

$$\dot{\gamma} = \frac{1}{mV}[-D\sin\beta\sin\mu - Y\sin\mu\cos\beta + L\cos\mu + T(\cos\mu\sin\alpha + \sin\mu\sin\beta\cos\alpha)] - \frac{g\cos\gamma}{V} \tag{4.54}$$

$$\dot{\mu} = \frac{p\cos\alpha + r\sin\alpha}{\cos\beta} + \frac{1}{mV}[D\sin\beta\cos\mu\tan\gamma + Y\tan\gamma\cos\mu\cos\beta + L(\tan\beta + \tan\gamma\sin\mu) + T(\sin\alpha\tan\gamma\sin\mu + \sin\alpha\tan\beta - \cos\alpha\tan\gamma\cos\mu\sin\beta)] - \frac{g\cos\gamma\cos\mu\tan\beta}{V} \tag{4.55}$$

$$\dot{\alpha} = q - \tan\beta(p\cos\alpha + r\sin\alpha) - \frac{1}{mV\cos\beta}(L + T\sin\alpha) + \frac{g\cos\gamma\cos\mu}{V\cos\beta} \tag{4.56}$$

$$\dot{\beta} = -r\cos\alpha + p\sin\alpha + \frac{1}{mV}(D\sin\beta + Y\cos\beta - T\sin\beta\cos\alpha) + \frac{g\cos\gamma\sin\mu}{V} \tag{4.57}$$

$$\dot{\xi} = V\cos\gamma\cos\chi \tag{4.58}$$

$$\dot{\eta} = V\cos\gamma\sin\chi \tag{4.59}$$

$$\dot{h} = V\sin\gamma \tag{4.60}$$

4.1.2 Elastic Models

Elastic models have additional degrees of freedom associated with elasticity and unsteady aerodynamics. If the elastic deformations are assumed to be small, the equations of motion are linear.

The next sections describe two approaches for generating a model for a flexible aircraft. Section 4.1.2.1 discusses the simpler (and less accurate) of the two, which appends bending modes to a rigid-body model and modifies the sensor equations accordingly. It requires a minimum of data and is useful in early efforts before additional data are available from the aerodynamics and structures specialists. The second method, discussed in Section 4.1.2.2, is based on an accurate finite-element structural model combined with appropriate distributed aerodynamic data.

4.1.2.1 Simple Flexible Aircraft Models

During the initial stages of aircraft design, detailed aeroservoelastic models of the aircraft are not available. Nevertheless, the control designer must be aware of the impact of flexibility on the control design problem. Often simple models of aeroservoelastic effects are useful at this stage. As the aircraft and control design becomes more mature, detailed models like those discussed in Section 4.1.2.2 replace these simplified models.

Simplified models of aeroservoelastic behavior of aircraft vary in complexity and accuracy. The simplest are standard rigid-body models with stability derivatives modified by flexibility-to-rigid ratios obtained from steady-state structural/aerodynamic analyses. More complicated models result from using strip theory aerodynamics. Even more

complex models include one or more second-order elastic modes with additional stability derivatives obtained from steady-state aerodynamics. Further improvements to this model add quasi-steady aerodynamics; that is, add contributions due to apparent mass (s^2 terms) and aerodynamic damping (s terms) but neglect (irrational or distributed) contributions due to wake vortices except for downwash on the tail. Clearly, s^2 terms have less effect at lower frequencies than s^1 and s^0 terms. Additional refinements add finite-dimensional approximations to the remaining circulatory contributions, which are known as Theodorsen's circulation function for incompressible, two-dimensional flows [Theodorsen 1935]. For two-dimensional flows with nonzero Mach numbers, the circulation function is even more complex [Jordan 1953]. The "Cadillac" of strip theory models employs modifications for sweep, finite wing span, and compressibility and has given satisfactory results for aspect ratios as low as 1, reduced frequencies as high as 0.3, and Mach numbers from 0 to 15 [Yates 1966]. The well-known book by Bisplinghoff, Ashley, and Halfman [Bisplinghoff 1955] contains most of the theory needed for developing these models.

Generally, the accuracy of models increases with the cost of generating them. This cost includes both manpower and data requirements. Thus, tradeoffs between accuracy and cost force crude models in some cases and detailed models in others.

Flex/Rigid-Corrected Rigid-Body Models: The form and order (number of state variables) of flex/rigid-corrected models is exactly the same as that for rigid-body models. The difference between the two is that stability derivatives are modified by factors called flex/rigid ratios, or F/R factors, to account for the low-frequency effects of flexure on the dynamics. The calculated F/R ratios use aerodynamic influence coefficients, tuned to wind tunnel data if it exists, and structural influence coefficients. Typically they are significant for stability derivatives corresponding to lift and rolling and pitching moments. These F/R ratios are close to unity for low dynamic pressure.

Rigid-Body-Plus-Elastic-Modes Models: The order of this model is larger than the previous two by twice the number of flex modes modeled. This model does not include any unsteady aerodynamics. The number of flex modes is typically one or two because this model is not very accurate at high frequencies without unsteady aerodynamics. It gives a first cut at flexure contributions at the sensors and actuators. This model requires only limited information on modal frequencies and mode shapes. Usually, we know the first modal frequency and can estimate the mode shapes. This model also requires information on span-wise loading. For example, an elliptical load distribution for Mach < 1 is often a good approximation, except for low-aspect ratio wings where the outboard distribution is higher than that resulting from an elliptical assumption. In any case, we require mass and inertia properties, as well as geometric dimensions and angles. The computational requirements for generating differential equation models require evaluating various integrals involving the mode shapes, as was done for the F-4 lateral-directional dynamics [Stein 1971]. These details can also be carried out for the longitudinal axis.

Quasi-Steady Models: Quasi-steady models takes into account noncirculatory contributions to forces and moments (apparent mass and aerodynamic damping) and circulatory downwash contributions, which do not require increasing the state dimension. Thus, quasi-steady models take the form of rigid-body-plus-flex-modes models but also take into account some unsteady aerodynamics effects. To maintain the same order (i.e., eight rigid-body states plus two states per flex mode), they neglect circulatory contributions to forces and moments that are irrational or distributed with respect to frequency. "Irrational or distributed with respect to frequency" refers to the fact that transfer functions from wing motion to aerodynamic loading are transcendental rather than rational. Transcendental transfer functions cannot be reduced to a finite set of differential equations in the time domain. Neglecting this transcendental contribution in the quasi-steady model amounts to ignoring time delays associated with changing forces and moments on the aircraft. We can also apply the quasi-steady approximation to a swept wing in compressible flow [Yates 1966]. Similarly, we can also account for finite-span effects by employing lift-curve slopes and aerodynamic-center locations obtained from any three-dimensional analytical or wind-tunnel source.

Strip Theory Distributed Model: The "Cadillac" of the simple models includes all circulation contributions to unsteady aerodynamics. This results in a frequency-domain model that can be approximated by a finite number of differential equations, as discussed in Section 4.1.2.2. We employed this procedure for the X-29 pitch axis before a detailed aeroservoelastic model was available to assess the impact of flexibility on the control of this highly unstable vehicle [Enns 1982]. Here lift and moment equations employ a compressibility correction to Theodorsen's circulation function [Yates 1966].

4.1.2.2 Detailed Flex-and-Aerodynamic-Data Models

"Stone's Diagram" shown in Figure 4.1 captures the essence of this subsection. This figure is taken from Dick Stone's presentation at the 1984 Aeroservoelastic Specialists Meeting at Wright Laboratory. At the top of the figure, data enters as it might be received from structures and aerodynamic groups. At the bottom of the figure, is a suitable design model for the controls engineer. The center portion of the figure shows the following major steps:

- Low-frequency "correction" of theoretical data with wind tunnel data
- Euler inertial-to-body-axis coordinate transformation
- Rational approximation of aerodynamic matrix
- Appending force actuator model (if desired)
- Conversion to state space
- Model-order reduction

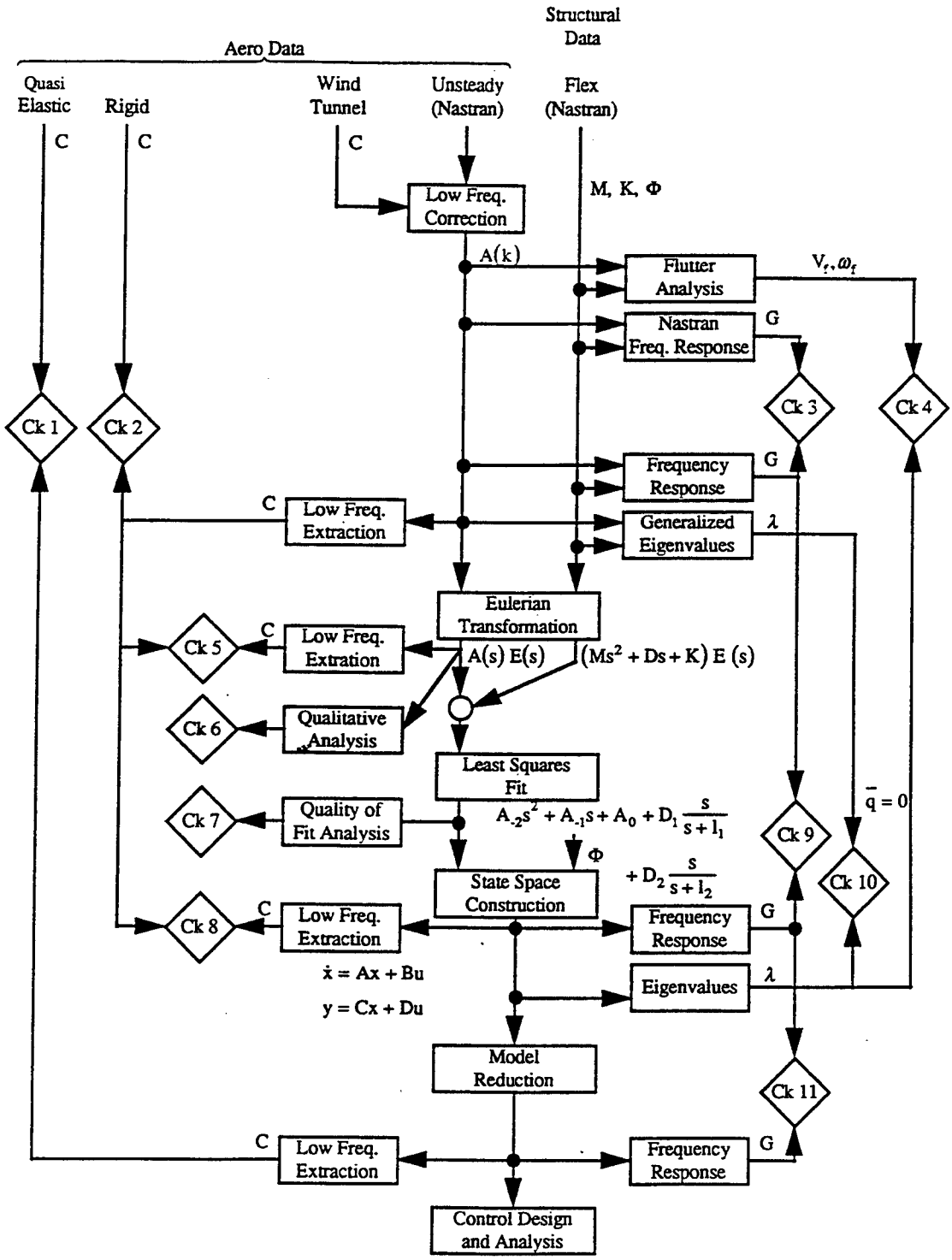


Figure 4.1: "Stone's Diagram" for Aeroservoelastic Model Development.

We address each topic in the following paragraphs. In the remainder of the section, we describe the various "checks" used to maintain the integrity of data and enforce consistency. We emphasize that this figure is not just an idealized approach, but corresponds to existing tools that have been used successfully on real-world flight control problems.

Distributed Model: The starting point for the distributed model is a generalized force equation of the form

$$[Ms^2 + (1 + gj)K - \bar{q}A_{in}(k)]q_{in} = A_{in-gust}(k)w_{gust} \tag{4.61}$$

where

$M = n \times n$ mass matrix

$K = n \times n$ stiffness matrix

$A_{in}(k) = n \times n$ aerodynamic matrix at a number of reduced frequencies

$A_{in-gust}(k) = n \times 1$ gust vector at the same reduced frequencies

w_{gust} = vertical gust velocity at a reference point

$k = \frac{\omega \bar{c}}{2V}$ = reduced frequency

V = velocity

$\bar{c}$ = mean aerodynamic chord

$\bar{q}$ = dynamic pressure

g = structural damping

$j = \sqrt{-1}$

and

$$q_{in} = \left[\begin{array}{cccccccccc} z & \theta & \eta_1 & \eta_2 & \ldots & \eta_{n_f} & \delta_1 & \delta_2 & \ldots & \delta_{n_u} \end{array} \right]^T \tag{4.62}$$

is a column vector of generalized coordinates, and the subscript in denotes inertial coordinates.

The generalized force equation is linearized about the trim condition, which could be straight and level or in a turn, for example. Detailed derivations of the equations of motion for the unrestrained vehicle can be found in Chapter 9 of the Bisplinghoff and Ashley textbook [Bisplinghoff 1962].

Note that we have allowed a slight abuse of notation in the equation in that both s terms (for Laplace transform) and k and gj terms (for simple harmonic motion) are present in the same equation. Soon, we will replace s with $j\omega$ for simple harmonic motion.

Later, we will derive a state-space approximation that applies to nonharmonic as well as harmonic motion using frequency response data.

We can replace the gjK term that accounts for structural damping, which has been used since World War II, with more physical models of structural damping. Although we could also represent structural damping as pure modal damping, this would require transforming to modal coordinates and introducing damping ratios for each of the modal degrees of freedom. In any case, the structural damping terms amount to less than 2 percent of critical damping.

The generalized force equation is applicable to symmetric (or pitch) motion of an aircraft, as well as anti-symmetric (or lateral-directional) motion. We will use the pitch-axis model to illustrate the following discussion.

The response vector associated with the generalized force equation is

$$y = \Phi q_{in} \tag{4.63}$$

where Φ = 'the matrix of influence coefficients.' For the pitch axis, q_{in} contains:

z = rigid-body linear displacement (positive up)

θ = rigid-body pitch displacement (positive nose up)

η_j = jth flex-mode displacement (positive up)

δ_k = kth control-surface deflection (positive trailing edge down)

Examination Of Aerodynamic Data: Part of the modeling process involves plotting individual elements of the $A_{in}(k)$ matrix in polar (imaginary vs. real) form to check qualitative characteristics. The plots are examined for reasonableness. Qualitatively, the (i,j) elements of A_{in} should be real numbers at zero frequency and continuous with increasing frequency. If wind tunnel estimates are available, we can attempt to reconcile theoretical data with them [Giesing 1976].

Coordinate Transformations: When pitch axis equations are expressed in inertial coordinates (z and θ) rather than body coordinates (w and $q = \dot{\theta}$), we can apply a frequency dependent "Eulerian" transformation:

$$\begin{bmatrix} z \\ \theta \end{bmatrix} = \frac{1}{s} \begin{bmatrix} 1 & \frac{-V}{s} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} \tag{4.64}$$

This must also be applied to the response equations. Thus, the force equation retains its second-order matrix form but now has coordinate vector

$$q_e = \begin{bmatrix} w & q & \eta_1 & \eta_2 & \dots & \eta_{n_f} & \delta_1 & \delta_2 & \dots & \delta_{n_u} \end{bmatrix}^T \tag{4.65}$$

where the subscript e denotes Eulerian coordinates. We next approximate $A_{in}(k_j)$ at points k_j with a rational function.

Rational Approximation of Aerodynamic Matrix: At this step, we approximate $A_{in}(k_j)$ for a given set of frequencies (typically, ten or less) with a set of continuous functions $f_j(s)$.

Typically, the fit function takes the form

$$\hat{A}_{in}(k) = A_1 s^2 + A_2 s + A_3 + A_4 \frac{s}{s+\lambda_1} + A_5 \frac{s}{s+\lambda_2} \tag{4.66}$$

where

$$s = j\frac{2V}{\bar{c}}k \tag{4.67}$$

and the A_i are constant matrices that are often determined by a least squares fit. The highpass break frequencies λ_j may be selected on the basis of velocity and chord length or they may be adjusted as part of the fit process, at the expense of making it a nonlinear least squares problem.

There are other alternative methods for approximating unsteady aerodynamics: One involves generalizing Pade's approximation to a matrix setting [Vepa 1975]; another addresses obtaining approximations involving a minimum number of augmented states [Karpel 1980].

In performing the fit, it is important to force the w term to have the same magnitude as w_{gust} at low frequency.

Ultimately, the proof of the fit is in how well $\hat{A}_{in}(k_j)$ approximates $A_{in}(k_j)$. This comparison is often accomplished by first using a numerical "score" from the fitting procedure to screen fit quality and then relying upon "expert" visual evaluation of polar plots for selected terms.

State Space: At this point in the model development, our model is a high-order differential equation model in *second order* form. We can put it into standard first-order state-space form using simple algebra and careful bookkeeping.

Model-Order Reduction: As the final modeling step, we must generate reduced-order models for control synthesis. Model quality of these control design models must be consistent with their intended purpose (see Figure 4.2). Desired control bandwidth is one of the factors used in determining the number of states to be retained in the design model. The weighted-balancing method for model reduction is well-suited for this step [Enns 1984].

The preceding paragraphs have given a quick synopsis of flexible aircraft modeling. Now, let's review the 11 checks in Figure 4.1 that are performed along the way to complete the modeling process:

- Check 1: Check zero frequency values of transfer functions for the design model with quasi-elastic predictions derived from alternate methods. This is an important

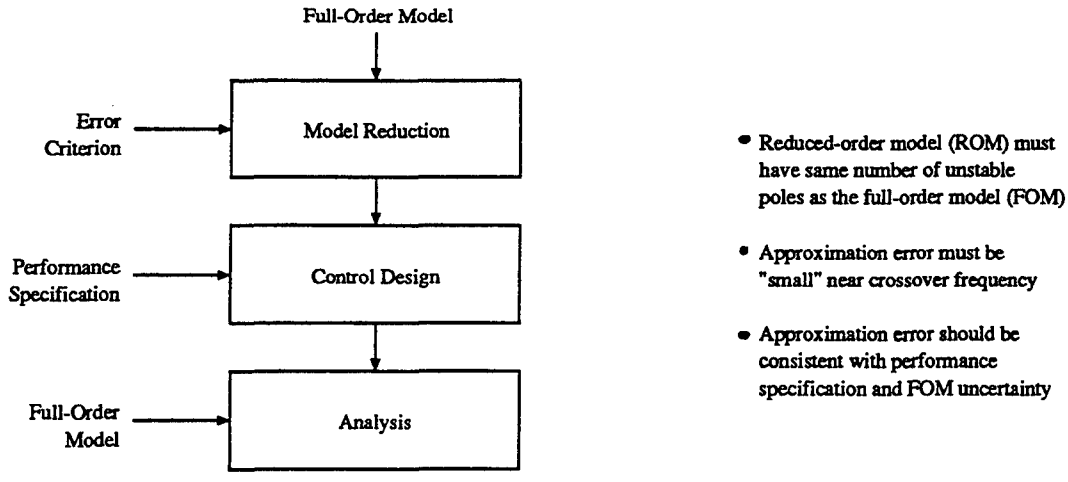


Figure 4.2: Outline of Model-Reduction Procedure.

consistency check on overall flexibility as it impacts low-frequency gain.

- Check 2: Resolve low-frequency values of stability and control derivatives with independent rigid-body predictions and/or wind-tunnel estimates.

- Check 3: Verify integrity of data transfer between programs (and often computers) by generating frequency response of distributed system.

- Check 4: Verify high-frequency fidelity by comparing flutter-speed predictions from the distributed and state-space models.

- Check 5: Verify correctness of Euler transformation.

- Check 6: Verify shape and smoothness of aerodynamic functions

- Check 7: Verify fit quality.

- Check 8: Check stability and control derivatives derived from state-space and distributed models.

- Check 9: Compare state-space and distributed frequency response.

- Check 10: Check mass and stiffness characteristics (set dynamic pressure to zero) of state-space with eigenvalues from structures program.

- Check 11: Compare design model frequency response with full-order state space to ensure compatibility with control objectives.

4.2 Maneuvers

Fighter aircraft maneuvers are significantly more demanding than maneuvers for other manned aircraft and hence pose unique control challenges. Fighter aircraft can change certain states variables rapidly while simultaneously minimizing excursions in other state variables (e.g., rapid bank at high angle of attack during deceleration). Aircraft possessing these characteristics are said to highly maneuverable and agile. For example, roll rate, normal acceleration, longitudinal acceleration can be changed rapidly while simultaneously keeping lateral acceleration and sideslip angle sufficiently small.

Requirements on the control system are much less demanding for slow changes in flight condition than for rapid changes between equilibrium conditions. Higher levels of roll rate, roll acceleration, g-onset rate associated with fighter maneuvers make it increasingly more difficult to maintain coordinated flight. In particular, transient lateral acceleration and/or sideslip angle during aggressive maneuvers drives up the control system requirements for decoupling and control system bandwidth.

In addition, aggressive fighter maneuvers often imply control power limitations in terms of actuator rate and position limits. This is especially true for low-speed flight conditions (assuming thrust vectoring is not available). Control power limitations associated with multiple actuation devices with individual rate and position limits presents a huge challenge to the multivariable control law designer.

Control laws for these highly maneuverable aircraft often include command limiters to prevent exceeding certain angle of attack limitations, such as on the F-16 and F-117 where control power is insufficient to prevent departure beyond a known angle of attack that is flight-condition dependent. Roll-rate-command limiters and normal-acceleration limiters are also commonly employed based on high-maneuverability goals for the control law.

Post-stall maneuvers subject the closed-loop system to rapidly changing nonlinear aerodynamic coupling and inertial coupling phenomena. The utility of such maneuvers and aircraft characteristics of interest have been discussed elsewhere [Herbst 1980, Well 1982, Herbst 1983]. Time-optimal maneuvers have been developed for fighter aircraft with various initial and final conditions, assumptions, and constraints [Well 1982]. The dynamic inversion control laws of Section 10 were developed and tested with these types of maneuvers.

Several standard maneuvers for evaluation of control laws have been documented in the literature [Wilson 1993]. These maneuvers are suggested for control law testing based on both engineering and pilot points of view.

Further discussion about air combat maneuvers from a pilot perspective can be found elsewhere (e.g., [Shaw 1985]).

4.3 Handling Qualities

Much research has gone into quantifying what makes an aircraft possess good handling qualities. Results exist in terms of either frequency or time response, most of which is restricted to the pitch axis.

Military specifications for handling qualities take the form of a standard and a handbook [MIL-STD-1797 1987].

Several different approaches exist:

- Low-order equivalent systems;
- Numerator time constant
- Bandwidth and phase sensitivity
- Time-domain criteria
- Neal-Smith criterion
- Step target tracking
- Paper pilot and similar optimal pilot models
- Heading control criteria
- Ralph Smith's criteria

All were summarized in a conference proceedings [Moorhouse 1982] and further elaborated upon by the original authors in these same proceedings.

4.4 Disturbance Models

4.4.1 Gusts

Two main gust model forms are widely accepted today: Von Karman and Dryden. The Dryden form is much simpler to use and is described below. The three components of gust (axial, lateral, and normal) have the power spectra

$$\Phi_{u-gust}(\omega) = \frac{\sigma_u^2}{V}\frac{2L_u}{\pi}\frac{1}{1+(L_u\frac{\omega}{V})^2} \tag{4.68}$$

$$\Phi_{v-gust}(\omega) = \frac{\sigma_v^2}{V}\frac{L_v}{\pi}\frac{1+3(L_v\frac{\omega}{V})^2}{[1+(L_v\frac{\omega}{V})^2]^2} \tag{4.69}$$

$$\Phi_{w-gust}(\omega) = \frac{\sigma_w^2}{V}\frac{L_w}{\pi}\frac{1+3(L_w\frac{\omega}{V})^2}{[1+(L_w\frac{\omega}{V})^2]^2} \tag{4.70}$$

Each of these spectra are implemented using a linear dynamic filter with matching spectrum for each direction. Since the spectrum is defined by

$$\Phi(\omega) = \frac{1}{\pi}F(j\omega)F(-j\omega) \tag{4.71}$$

the resulting filters, $F(s)$, for each gust component are

$$F_{u-gust}(s) = \sigma_u\sqrt{\frac{2V}{L_u}}\frac{1}{s+\frac{V}{L_u}} \tag{4.72}$$

$$F_{v-gust}(s) = \sigma_v\sqrt{\frac{V}{L_v}}\frac{\sqrt{3}s+\frac{V}{L_v}}{s^2+2\frac{V}{L_v}s+(\frac{V}{L_v})^2} \tag{4.73}$$

$$F_{w-gust}(s) = \sigma_w\sqrt{\frac{V}{L_w}}\frac{\sqrt{3}s+\frac{V}{L_w}}{s^2+2\frac{V}{L_w}s+(\frac{V}{L_w})^2} \tag{4.74}$$

The scales (L_u, L_v, L_w) and intensities (σ_u, σ_v, σ_w) for various levels of gust can be found in MIL-STD-1797. Independent unit-intensity white-noise signals drive each of these filters to produce the desired turbulence model.

Section 5
Choice of Controlled Variables

Three variables are important in aircraft flight control: one for each rotational degree of freedom. Each of these variables may actually be a combination of signals to be controlled by the pilot. All must be compatible with flying qualities and mission objectives. In this section, we describe one possible set of such controlled variables. But first, we briefly discuss a common and necessary aspect of controlled variables.

5.1 Minimum-Phase Zero Constraint

Transfer functions from appropriate control inputs to controlled variables must be minimum phase. A *minimum phase zero* is a zero that lies in the left half of the complex plane. Thus, our constraint ensures that the transfer function of the open-loop plant from independent (or effective) control inputs to controlled variable outputs has stable zeros. These zeros become closed-loop poles for dynamic inversion designs [Morton 1994] and impact the design with any synthesis technique. Large magnitude open-loop zeros, even though stable or minimum phase, often lead to loop gains that are unacceptably large. This is particularly true with the dynamic inversion design methodology whenever zero magnitudes are larger than the intended loop bandwidth.

Today, the only way to ensure that controlled variables satisfy the minimum-phase constraint is to evaluate them at the time they are chosen using the simple iterative sequence outlined in Figure 5.1. If the controlled variables selected turn out to have nonminimum phase zeros, then we must either make a new selection or modify the current selection. In general, to stabilize unstable zeros in a controlled variable selection, the designer can either increase the relative contribution of angular rate or decrease the contribution of other variables. This is true because transfer functions from aerodynamic and thrust vectoring control effectors to angular rates are usually minimum phase.

The choice of controlled variable is obvious if there is only one response and one control input (e.g., longitudinal axis with pitch rate response to horizontal tail). In the special case of two responses and two independent control inputs (e.g., roll and yaw rate response to aileron and rudder), the corresponding transfer function matrix and its multivariable *transmission zeroes* are of interest. In cases with redundant controls, the choice is less obvious and depends on the control allocation approach.

Suppose that the controls are ganged, as discussed in Section 6, such that

$$u = G\delta \tag{5.1}$$

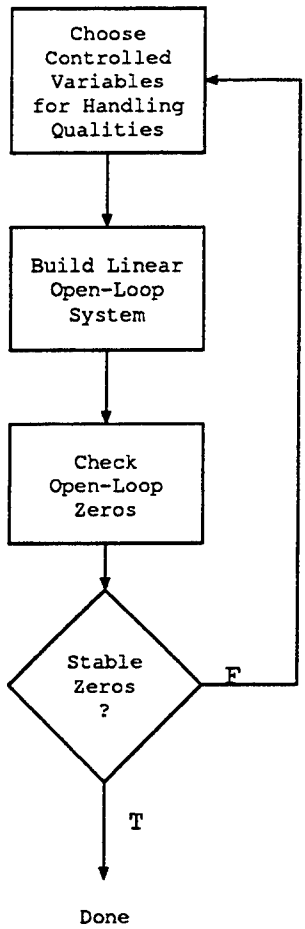


Figure 5.1: Procedure for Choosing Controlled Variables.

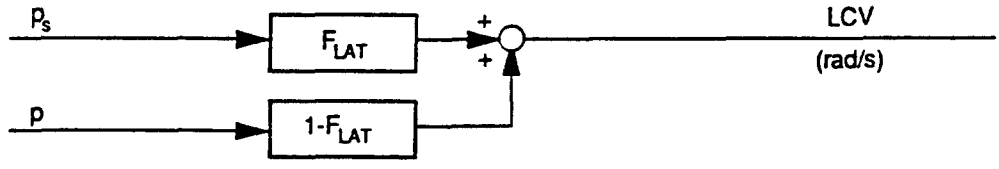


Figure 5.2: Roll-Axis Controlled Variable: LCV.

where δ is the internal effective actuator commands, u the actuator commands, and G the ganging matrix. If the number of effective commands (dimension of δ) equals the number of controlled variables, which is less than the number of controls (dimension of u), then the square transfer function matrix of interest is the controlled variable response to δ.

5.2 Roll Axis: LCV

The roll-axis controlled variable, LCV, is defined as (see Figure 5.2):

$$LCV = (1 - F_{LAT})LCV_{lo\bar{q}} + F_{LAT}LCV_{hi\bar{q}} \tag{5.2}$$

where

$$\begin{aligned} LCV_{lo\bar{q}} &= p \\ LCV_{hi\bar{q}} &= p_s \end{aligned} \tag{5.3}$$

At normal and high dynamic pressures, the controlled variable is the stability-axis roll rate, p_s. This reflects the desire to roll about the velocity vector, which is consistent with keeping the aircraft near its current angle of attack and in coordinated flight (i.e., β small). For low dynamic pressures, the controlled variable is body-axis roll rate, p. The rationale here is that control with respect to the air mass becomes undesirable when the velocity of the aircraft is low or approaches zero (e.g., a VTOL aircraft near hover or a space vehicle). Moreover, quantities measured with respect to the air mass become more difficult to measure and less well defined or even indeterminate near zero airspeed (e.g., sideslip and angle of attack). The only sensible option under these conditions is to

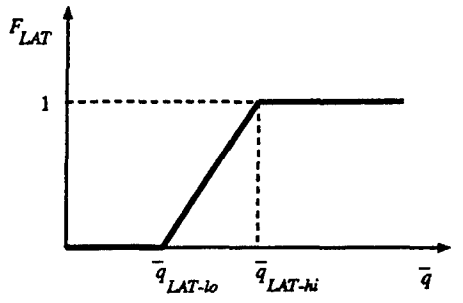


Figure 5.3: Lateral/Directional Control Transition Function.

directly relate cockpit control to the body-axis response (i.e., lateral stick and body-axis roll rate).

As Figure 5.3 shows, the F_{LAT} function transitions lateral/directional controlled variables from normal to low-dynamic-pressure/high-angle-of-attack flight regimes. It transitions from 0 to 1 as $\bar{q}$ goes from $\bar{q}_{LAT-lo} = 5$ psf to $\bar{q}_{LAT-hi} = 10$ psf and is unnecessary for applications where flight is restricted to low-α/high-$\bar{q}$ regimes.

5.3 Pitch Axis: MCV

In this subsection, we discuss three choices for the pitch-axis controlled variable, MCV. Each reduces to q for the low-$\bar{q}$ regime, and each includes the term $q + K_\alpha \alpha$ for the normal flight regime. The choices are illustrated in Figure 5.4 and are defined as a function of the switch SW_{cv} as follows:

$$MCV = (1 - F_{LON})MCV_{lo\bar{q}} + F_{LON}MCV_{hi\bar{q}} \tag{5.4}$$

where

$$\begin{aligned} MCV_{lo\bar{q}} &= q \\ MCV_{hi\bar{q}} &= (q + K_\alpha \alpha) \\ &+ \begin{cases} \frac{g}{u}\cos\phi\cos\theta - \frac{v}{u}p - g(\frac{1}{u} + \frac{1}{V_{co}}) & \text{if } SW_{cv} = 0 \\ \frac{g}{u}\cos\phi\cos\theta - \frac{v}{u}p - g\cos\theta(\frac{1}{u} + \frac{1}{V_{co}}) & \text{if } SW_{cv} = 1 \\ \frac{g}{V\cos\beta}\cos\gamma\cos\mu - p_s\tan\beta - K_v V & \text{if } SW_{cv} = 2 \end{cases} \end{aligned} \tag{5.5}$$

As Figure 5.5 illustrates, the F_{LON} function transitions longitudinal controlled variables from normal to low-dynamic-pressure/high-angle-of-attack flight regimes. It transitions from 0 to 1 as $\bar{q}$ goes from $\bar{q}_{LON-lo} = 12$ psf to $\bar{q}_{LON-hi} = 50$ psf. F_{LON} could be set to 1 for aircraft applications whose flight is restricted to low-α/high-$\bar{q}$ regimes.

For those terms that are common to all three choices of controlled variable,

$$K_\alpha = \frac{\bar{q}S(C_{L_\alpha})_{ref}}{mV_{co}}$$

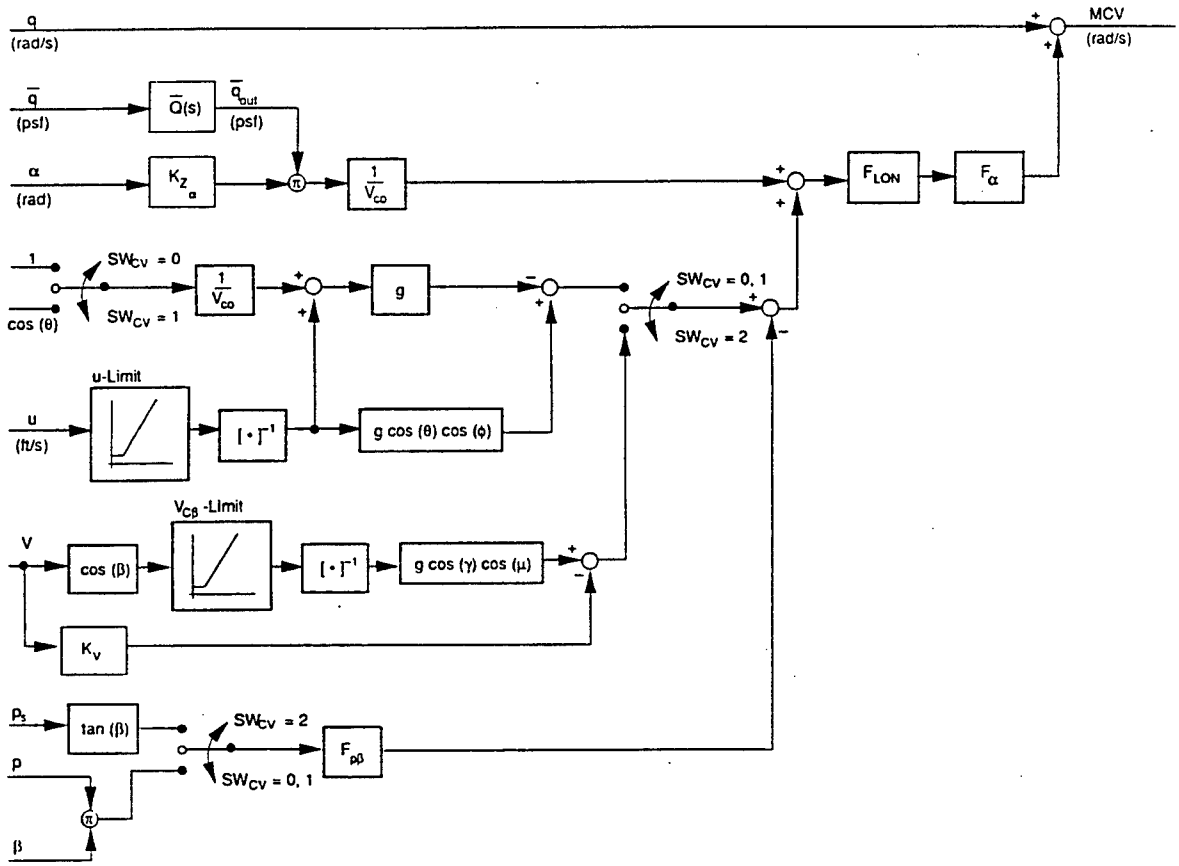


Figure 5.4: Pitch-Axis Controlled Variable: MCV.

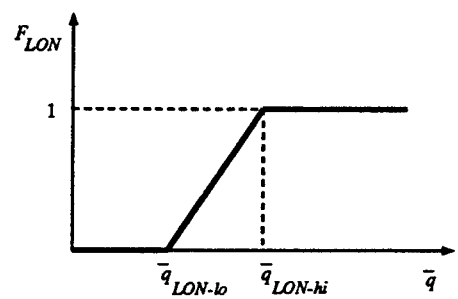


Figure 5.5: Longitudinal-Axis Control Transition Function.

where $(C_{L_\alpha})_{ref}$ is the nominal lift-curve slope (assumed constant) for subsonic flight at low α. For angles of attack below stall, the term $K_\alpha\alpha$ is approximately nz/V_{co}, where nz is the normal acceleration in ft/sec^2. Hence $q + K_\alpha\alpha$ is approximately equal $q + nz/V_{co}$ for normal flight conditions. This combination is sometimes called "Cstar" in the literature, where the "crossover velocity," V_{co}, is the airspeed at which pitch rate and normal acceleration (at constant α) make equal contributions to the controlled variable. We use a value of 400 ft/sec for V_{co}, as is used in some existing flight control systems. The rationale for using this combination as the pitch-axis controlled variable is that it produces a natural transition in control priority from rotation to translation as airspeed increases. Some type of priority system is necessary because two degrees of freedom are being controlled by essentially one control effector.

For angles of attack above stall, nz feedback is either ineffective or destabilizing due to the lift-curve slope reversal. We avoid this problem by using the $K_\alpha\alpha$ form which is continuous through and beyond stall. The $q+nz/V_{co}$ form would be preferable below stall for reasons of sensor reliability and acceleration control accuracy, but the complication of implementing a sensor transition scheme is not warranted for early design study phases. The preferred value for V_{co} (and hence K_α) in the stalled flight regime may require some tuning, however, since a value optimized for unstalled flight is unlikely to be optimal for higher angles of attack. The ideal value of K_α varies greatly with application. In some cases, it is so large that it dominates the q term (producing effectively an α controlled variable), whereas in others it is small enough to be negligible (producing a q controller).

5.3.1 Option 1: $SW_{cv} = 0$ or 1

The second pair of terms in the high-$\bar{q}$ part of MCV, $\frac{g}{u}\cos\phi\cos\theta$ and $-\frac{v}{u}p$, are provided for pitch- and lateral/directional-axis decoupling and avoid large changes in α due to lateral as well as longitudinal maneuvers. The first term, which includes the direction cosine between the gravity vector and the body z axis, represents the change in pitch rate necessary to avoid a change in normal acceleration (and, in turn, α) due to gravity orientation. Thus, flight will be coordinated if the yaw axis is effective *and* rudder pedals are not being used. If neither of these conditions hold, significant sideslip may occur, making the second term, $\frac{v}{u}p$, significant. This term corresponds to the change in pitch rate necessary to avoid a change in normal acceleration due to sideslip and is nonlinear, involving perturbations in roll rate and sideslip. Therefore, it tends to contain high-frequency components that are difficult to regulate with achievable control bandwidths.

The final components of MCV are intended to produce a combination of good low-frequency pitch stability (per MIL-F-1797, Appendix A) and a desirable zero-stick-force flight characteristic. The latter, as a minimum, avoids manual trim changes for straight-and-level, clean flight conditions, sometimes called "apparent neutral speed stability." Other trim properties are debatable. Two candidates are:

- $SW_{cv} = 0$: $g(\frac{1}{u} + \frac{1}{V_{co}})$

- $SW_{cv} = 1$: $g\cos\theta(\frac{1}{u} + \frac{1}{V_{co}})$

Note that at small pitch attitudes, both options are equal. Both maintain the controlled variable near zero at $nz = 1g$, $q = 0$, and $\phi = 0$; both maintain trim α during roll attitude changes at small θ.

Differences are evident at significant pitch attitudes. The first option tends to hold trim α for constant stick input at all attitudes if speed changes are slow. This property tends to reduce α changes during rolls at elevated pitch attitudes, but requires a pilot trim input change for zero-stick-force, wings-level climbs and descents. It also has a nose-up drift tendency and a destabilizing influence on the long period for positive pitch attitudes. The second option eliminates these disadvantages at the expense of some increase (perhaps small) in roll-induced α changes at larger pitch attitudes. Its primary benefit is the constant pitch-rate trim property ($q = 0$ for zero stick input) at small roll attitudes; a convenient feature for general flying.

5.3.2 Option 2: $SW_{cv} = 2$

The motivation for this option is the desire for stable closed-loop phugoid or low-frequency characteristics and thus more well-behaved α dynamics. Comparison of the angle of attack dynamic equation in Section 4.1.1.3 under straight-and-level flight for the high-$\bar{q}$ part of MCV reveals that

$$MCV_{hi\bar{q}} = \dot{\alpha} + (K_\alpha \alpha - \frac{L + T\sin\alpha}{mV}) - K_v V$$

For sufficiently small V_{co}, $K_\alpha \alpha - \frac{L+T\sin\alpha}{mV}$ has a positive slope with respect to α for all α. Thus, the α response is stable if we neglect the velocity term.

We include the velocity-feedback term for stabilization of the phugoid dynamics. It was earlier noted that open-loop zeros become closed-loop poles when closing a dynamic inversion feedback loop around a linear system. The nonlinear generalization of this concept is that of *zero dynamics*, which refer to the dynamics of the system when the controlled variables are zero (or a constant). Because our controlled variable was chosen for the short-period frequency range, slightly unstable phugoid zero dynamics result in unstable closed-loop responses in the phugoid frequency range. The feedback of velocity stabilizes these zero dynamics and hence the closed-loop responses [Enns 1994]. Currently, fixed gain K_v is used for the velocity feedback gain. Although this works, a generalized gain is desirable. For the HARV program, true airspeed was used as the feedback velocity, but it may be advantageous to use calibrated airspeed instead.

Simulation experience has shown that the controlled variable for Option 2 is preferable to either of those examined under Option 1. The only drawback to Option 2 is its use of feedback signals that are relative to the airmass, and thus not accurately measured (e.g., the $\cos\gamma$, $\cos\mu$ terms). In contrast with this, all inertial quantities for Option 1 are easily measured (e.g., $\cos\theta$, $\cos\phi$ terms).

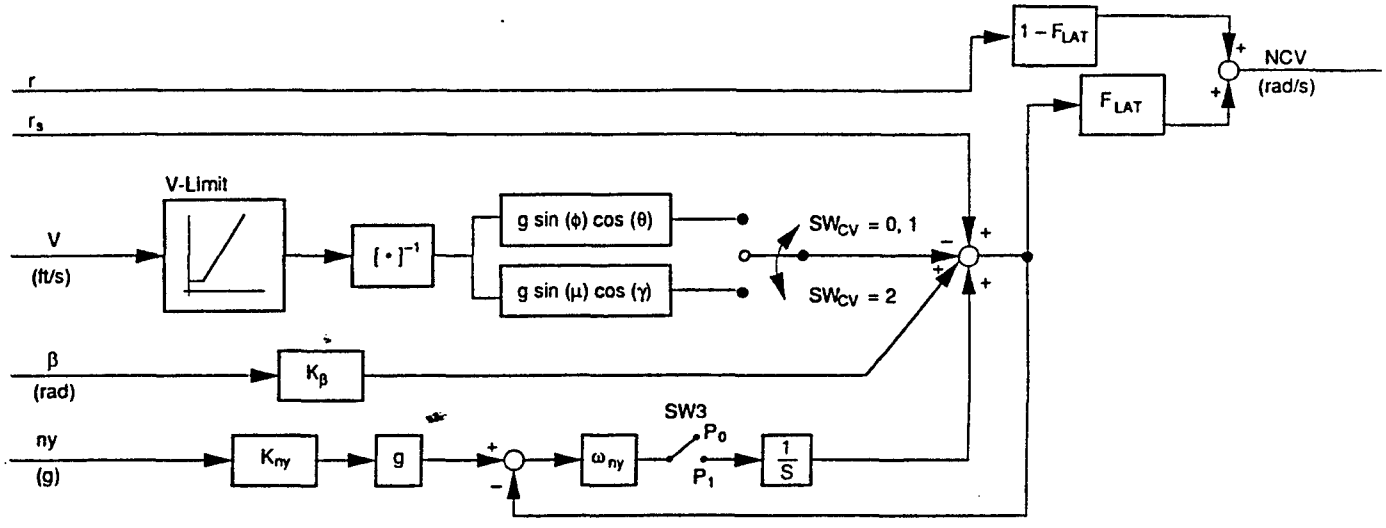


Figure 5.6: Yaw-Axis Controlled Variable: NCV.

5.4 Yaw Axis: NCV

Here we present two choices for the yaw-axis controlled variable, NCV. Each reduces to r in the low-$\bar{q}$ regime. The choices are illustrated in Figure 5.6 and are defined as a function of the switch SW_{cv} as follows:

$$NCV = (1 - F_{LAT})NCV_{lo\bar{q}} + F_{LAT}NCV_{hi\bar{q}} \tag{5.6}$$

where

$$\begin{array}{rcl} NCV_{lo\bar{q}} & = & r \\ NCV_{hi\bar{q}} & = & (r_s + K_\beta \beta + trim) \\ & - & \begin{cases} \frac{g}{V}\cos\theta\sin\phi & \text{if } SW_{cv} = 0 \text{ or } 1 \\ \frac{g}{V}\cos\gamma\sin\mu & \text{if } SW_{cv} = 2 \end{cases} \end{array} \tag{5.7}$$

with

$$trim = \frac{\omega_{ny}}{s} \times \begin{cases} 0 & \text{for pedal input} \neq 0 \\ (K_{ny}ny - NCV_{hi\bar{q}}) & \text{for pedal input} = 0 \end{cases} \tag{5.8}$$

Note that with pedal input (i.e., pedal force exceeds a preset threshold), the $trim$ value remains constant. Without pedal input, the high-frequency (above ω_{ny} rad/sec) content of the controlled variable equals $NCV_{hi\bar{q}} - trim$ (see equation 5.7), whereas the low-frequency content equals $K_{ny}ny$. This arrangement is designed to meet the following objectives:

1. Without pedal input, we want zero steady-state lateral acceleration at the center of gravity (CG) with no manual trim required. This is the most accurate measure of

long-term directional trim. At higher frequencies (above ω_{ny} and up to the NCV control bandwidth) we want zero sideslip.

2. With pedal input, we want a satisfactory response of sideslip and stability-axis yaw rate. The sideslip response should satisfy current Level 1 military criteria for frequency and damping.

In the following paragraphs, we describe how the indicated controlled variable terms produce the above objectives.

The high-frequency component, with or without pedal input, is the combination $r_s - g/V \cos\theta \sin\phi + K_\beta\beta$. We provide zero-sideslip turns under steady-wind conditions without pedal input by zeroing $r_s - g/V \cos\theta \sin\phi$, and any sideslip that does occur is attenuated by the added sideslip angle feedback. The term $r_s + K_\beta\beta$ largely determines pedal response in that roll angle change is attenuated by the lateral axis. This is a parallel form to the $q + K_\alpha\alpha$ combination used in the pitch axis. To produce a comparable weighting between translation and rotation (a linear sum of yaw rate and lateral acceleration due to sideslip), we select the K_β term to be a linear function of $\bar{q}$ (as in K_α). We select the magnitude of K_β so that sideslip response to pedal inputs approximates a second-order critically damped system. The resulting second-order system should have a natural frequency and damping above the 1 rad/sec and 0.4, respectively, specified by Level 1 minimums of MIL-F-1797, Appendix A.

The final parameter we explain is the acceleration gain K_{ny}. We select it to approximate the same steady-state effectiveness as the $K_\beta\beta$ feedback term. For a constant value for C_{Y_β}, a constant aircraft weight, and assuming $ny = C_{Y_\beta}\bar{q}S\beta/m$, then $K_{ny} = K_\beta m/C_{Y_\beta}/\bar{q}/S$ sec/ft.

THIS PAGE LEFT INTENTIONALLY BLANK

Section 6
Reduction of Control Dimension

As aircraft become more complex, redundant and/or novel control effectors are becoming a common addition to the traditional aileron, rudder, and elevator complement. The most common approach to using multiple control effectors is by a priori choice.

All multivariable control design methods can accommodate multiple effectors, but design procedures for using them are often indirect and do not produce controllers that achieve desired results. For example, the eigenstructure design method takes full advantage of all available control inputs to achieve the desired eigenstructure but often produces high-gain controllers with poor stability margins with one control "fighting" another.

Because we are primarily concerned with rotational dynamics in the design of aircraft inner-loop control laws, we must often combine multiple effectors such that they can be reduced to a single *effective* control input in each axis that produces a body moment in that axis. For the longitudinal axis, our single effective control must produce primarily pitching moment, whereas for the lateral-directional axes, these effectors must produce primarily rolling and yawing moments.

In the following subsections, we discuss several methods for reducing the dimension of the control inputs:

- Explicit Ganging
- Pseudo Control
- Pseudo Inverse
- Daisy Chain

For the first three methods, the combination of control effectors can be described by

$$u = G\,\delta \tag{6.1}$$

where u is the full set of control effectors, G the ganging matrix, and δ the effective control input. The daisy-chain method is nonlinear and cannot be described in this linear fashion.

6.1 Explicit Ganging

In many applications, the easiest method of reducing the control dimension is to develop a predetermined scheme for ganging multiple control effectors to produce a single effective control. Simple examples of this include using left and right horizontal tail to produce an effective pitching-moment device or using left and right ailerons to produce a rolling-moment device. Such schemes for combining controls produce the G ganging matrix directly. For example, the ganging law

$$\delta_e = 0.5(\delta_{el} + \delta_{er}) \tag{6.2}$$

implies a ganging matrix of

$$G = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{6.3}$$

6.2 Pseudo Control

When it is less obvious how to gang the control inputs together, the pseudo-control method can often be used. This method (derived from [Lallman 1985, Sobel/Lallman 1988]) uses a singular value decomposition of the B matrix in the usual aircraft state-space description to determine which combinations of controls are most effective. For best results, we first scale the B matrix such that the size of the elements are in some way consistent.

The pseudo-control method works as follows. First, we perform a singular value decomposition on the scaled B matrix.

$$B_{scaled} = S_l B S_r = U \Sigma V^T \tag{6.4}$$

where S_l and S_r are scaling matrices. This decomposes the control action into orthogonal directions or pseudo-control inputs.

Next, we partition the singular value decomposition as follows:

$$[U_{01}\, U_{02}\, U_1] \begin{bmatrix} \Sigma_{01} & 0 & 0 \\ 0 & \Sigma_{02} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{01}^T \\ V_{02}^T \\ V_1^T \end{bmatrix} \tag{6.5}$$

where Σ_{01} contains the largest singular values of the dimension desired.

We now approximate B using only contributions due to the largest singular values to get

$$B = S_l^{-1} U \Sigma V^T S_r^{-1} \cong S_l^{-1} U_{01} \Sigma_{01} V_{01}^T S_r^{-1} \tag{6.6}$$

Letting δ denote the pseudo controls, $\tilde{B}$ the corresponding influence matrix, and G the ganging matrix from the pseudo controls to the actual controls, we see that

$$Bu = BG\delta \equiv \tilde{B}\delta \tag{6.7}$$

Now letting

$$\begin{aligned} \tilde{B} &= S_l^{-1} U_{01} \Sigma_{01} \\ G &= S_r V_{01} \end{aligned} \tag{6.8}$$

and subtituting for B, G, and $\tilde{B}$ gives

$$BG \cong \left(S_l^{-1} U_{01} \Sigma_{01} V_{01}^T S_r^{-1}\right)(S_r V_{01}) = S_l^{-1} U_{01} \Sigma_{01} = \tilde{B} \tag{6.9}$$

The reduced-input-dimension state space is now represented as

$$\dot{x} = Ax + \tilde{B}\delta \tag{6.10}$$

and is used for the control design. The ganging matrix to convert the pseudo controls back to the actual controls is

$$G = S_r V_{01} \tag{6.11}$$

which can be used to convert the control law produced with any control design methodology to match the actual control effectors available.

6.3 Pseudo Inverse

Another method for allocating control among multiple control effectors employs a pseudo inverse. Here we assume in the following differential equations that the number of controls, u, is greater than the number of states, x:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{6.12}$$

Assuming a constrained optimization approach, the formal problem statement is:

$$\min_u J = \min_u \frac{1}{2} u^T W_u u \tag{6.13}$$

$$\text{subject to } \dot{y} = CAx + CBu \text{ for specified } x \tag{6.14}$$

This optimization minimizes the weighted square deflections of the controls, u, subject to the modeled physical system differential equation constraint. We choose the weights W_u such that the controls are used in proportion to their total range of motion:

$$W_u = \text{diag}\left\{ \frac{1}{(u_{1_{max}} - u_{1_{min}})^2} , \frac{1}{(u_{2_{max}} - u_{2_{min}})^2} , \cdots \right\} \tag{6.15}$$

Using a Lagrange multiplier formulation, the problem becomes

$$\min_u \{ \frac{1}{2} u^T W_u u + \lambda^T (\dot{y} - CAx - CBu) \} \tag{6.16}$$

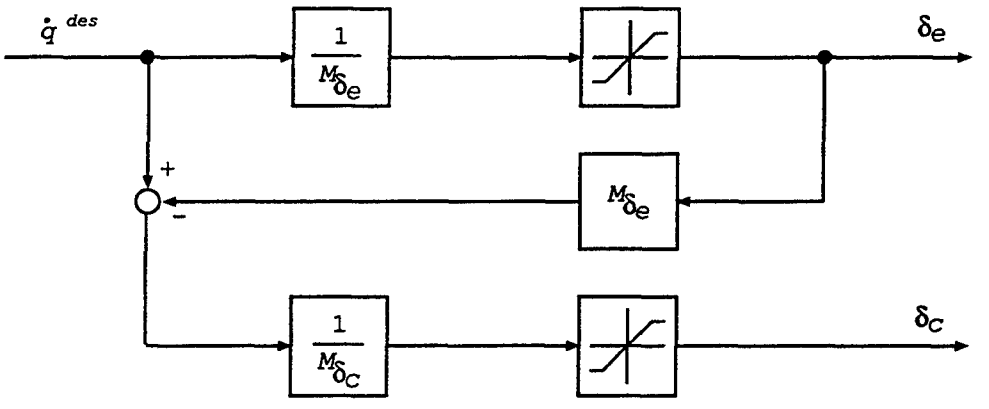


Figure 6.1: Pitch-Axis Daisy Chain Example.

where λ is the Lagrange multiplier. Solving for u gives

$$u = G(\dot{y} - CAx) \tag{6.17}$$

where the ganging matrix is

$$G = W_u^{-1}(CB)^T[(CB)W_u^{-1}(CB)^T]^{-1} \tag{6.18}$$

The effect of this ganging is that the control effectiveness matrix from the effective controls δ to $\dot{y}$ is an identity. That is, $\frac{\partial \dot{y}}{\partial \delta} = I$ where $u = G\delta$.

6.4 Daisy Chain

The daisy chain is a fairly straightforward idea. It assumes a hierarchy of control effectors such that when one control (or a group of controls) saturates, another becomes active to supply the remaining control forces or moments. For example, consider the following pitch-axis model

$$\dot{q} = M_{\delta_e}\delta_e + M_{\delta_c}\delta_c \tag{6.19}$$

where δ_e and δ_c are symmetric elevator and canard, respectively. The daisy chain configuration shown in Figure 6.1 uses elevator exclusively until it limits or loses effectiveness, at which point the canard takes over.

Section 7
Aircraft-Specific Analyses

Section 2.7 described analyses that are common to a wide variety of multivariable control designs. This section describes analyses that are *specific* to aircraft flight control.

7.1 Pilot Inputs

Pilot inputs can be broadly classified into two categories: flight test inputs and maneuver inputs. Flight test inputs relate to baseline control system performance and are typically directed toward one or two variables (e.g., 360° roll or $2g$ pullup). Maneuver inputs relate to mission performance and involve multiple axes and variables (e.g., acquisition and tracking tasks and windup turns).

7.2 Automatic Maneuvers

Most flight control systems employ a number of automatic modes. Examples include heading hold, Mach or velocity hold, vertical speed hold, and altitude hold. To accomplish such maneuvers, additional control loops must be closed around the inner stability and control loops. For stability reasons, these outer-loop controls must be sufficiently slower (i.e., have lower bandwidth) than the inner loops.

A general configuration of an outer-loop control structure is given in Figure 7.1. The

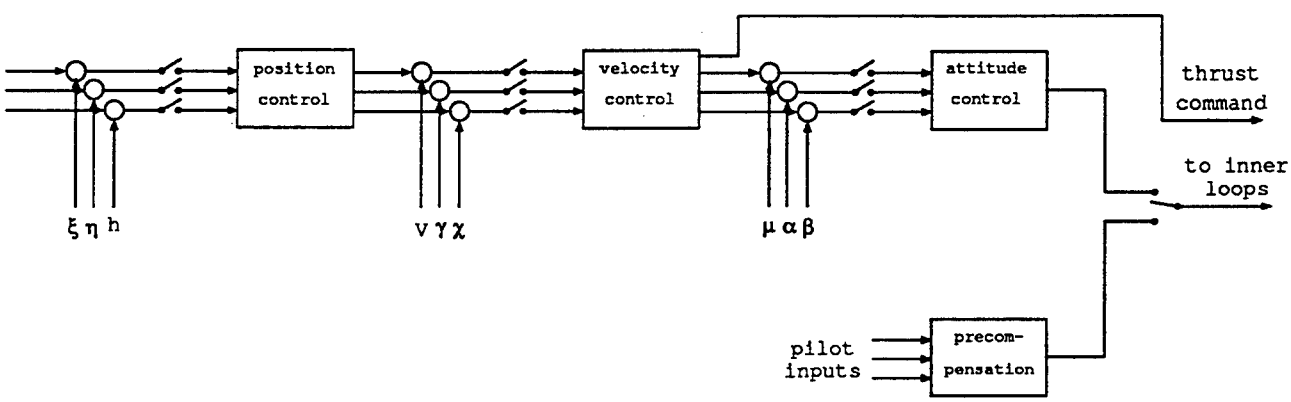


Figure 7.1: General Outer-Loop Control Architecture.

Table 7.1: Low-Order Transfer-Function Fits.

Description	Low-order Transfer Function
$\frac{p}{lateral\ stick}$	$\frac{K_p e^{-\tau_p s}}{T_R s+1}$
$\frac{ny_{cr}}{rudder\ pedal}$	$\frac{K_{ny} e^{-\tau_{ny} s}}{s^2+2\zeta_{dr_{ny}}\omega_{dr_{ny}} s+\omega^2_{dr_{ny}}}$
$\frac{\beta}{rudder\ pedal}$	$\frac{K_\beta e^{-\tau_\beta s}}{s^2+2\zeta_{dr_\beta}\omega_{dr_\beta} s+\omega^2_{dr_\beta}}$
$\frac{nz_{cr}}{longitudinal\ stick}$	$\frac{K_{nz} e^{-\tau_{nz} s}}{s^2+2\zeta_{sp}\omega_{sp} s+\omega^2_{sp}}$
$\frac{q}{longitudinal\ stick}$	$\frac{K_q(s+z) e^{-\tau_q s}}{s^2+2\zeta_{sp}\omega_{sp} s+\omega^2_{sp}}$

speed of these loops is such that the natural dynamic response of each set of feedback variables becomes slower as we move outward (i.e., toward the left) in the figure. The switch closings depend upon particular mode selections.

7.3 Handling-Qualities

We judge handling qualities by comparing closed-loop-system frequency responses with those of low-order transfer functions whose parameters are given by military specifications. Because there are no multivariable specifications for handling qualities, we only consider those for single input/single output systems.

Table 7.1 defines the transfer functions addressed by the specifications and the related fit functions. Accelerations are measured at an appropriate center of rotation. Acceptable ranges for the frequencies, dampings, and delays are given in the MIL-STD-1797.

7.4 Turbulence

Figure 7.2 shows the Dryden gust model described in Section 4.4.1 concatenated with an aircraft model. We use covariance analysis, described in Section 2.7.3, applied to this model to quantify the impact of gusts on aircraft performance.

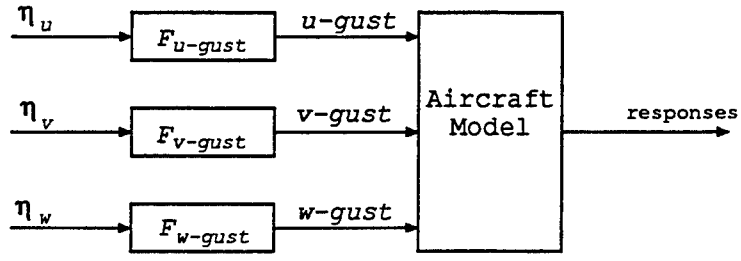


Figure 7.2: Connection of Gust Filter to Aircraft Model.

THIS PAGE LEFT INTENTIONALLY BLANK

Section 8

Implementation Issues

In this section, we examine a variety of issues associated with preparing a control law for practical application. Although the development is by no means exhaustive, it does touch on some of the more important practical considerations. Appendix B provides additional information on practical implementation issues: model reduction, gain scheduling, nonlinearities and limits, and describing function analysis.

8.1 Integrator Antiwindup

Integrators in feedback loops "wind up" in the presence of actuator rate or position limiting. This windup phenomenon is due to the integration of feedback errors that cannot be driven to zero in response to large actuator commands. Because of physical limits, actuators cannot respond to very large actuator commands. Limited actuator responses cause limited aircraft responses, which, in turn, cause large errors. Degraded performance, especially during pilot command reversal, is the result. In this section, we examine the limitations of actuator limits and describe methods to minimize their impact.

Figure 8.1 presents a very simplified aircraft model with a PI feedback control law. Note the pure integrator in the feedback loop. Figure 8.2 presents results of a linear simulation of this closed-loop system in response to a pulse command. A 50 rad/sec actuator was also inserted, but not shown in the figure, causing a slight discrepancy between the actuator command and the actuator response in the center plot of the figure. This system achieves good command following and well-behaved integrator response.

Figure 8.3 shows results for a simulation of the same system with a position-limited actuator. Note the poor command following and larger values for the integrator state. These results would be even worse if the pilot attempted to change his command to offset the slow response of the aircraft.

Figure 8.4 presents a simple scheme to prevent integrator windup. By feeding back the difference between actuator command and actuator position (or predicted position), the feedback loop drives the integrator state to the value of the actuator, which is at its limited value. As a result, the integrator does not move far from the value corresponding to the actuator limit. Applying this scheme to the example problem gives the results shown in Figure 8.5. Although these results are obviously nonlinear, qualitative performance is close to that for the linear simulation.

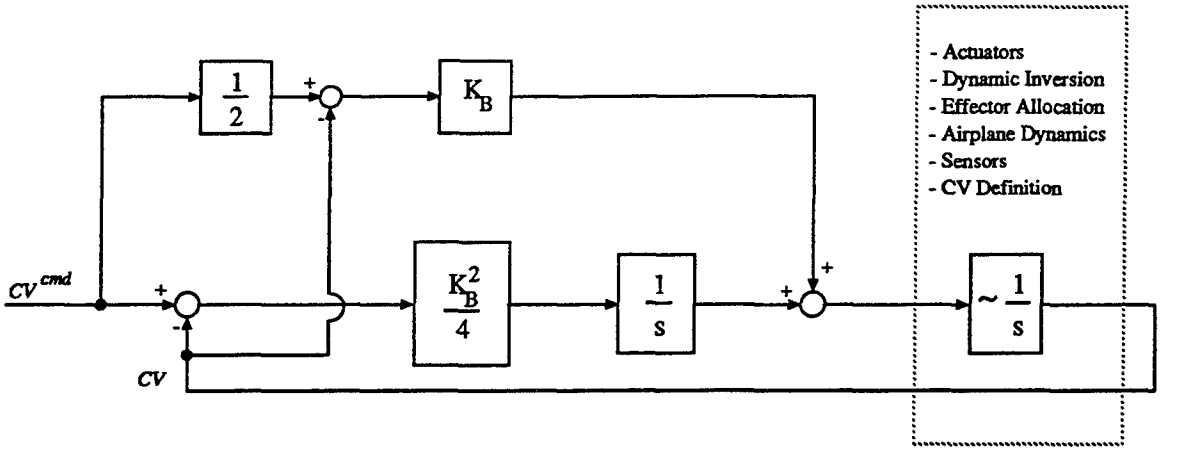


Figure 8.1: Block Diagram Describing Desired Dynamics.

8.2 Discretization

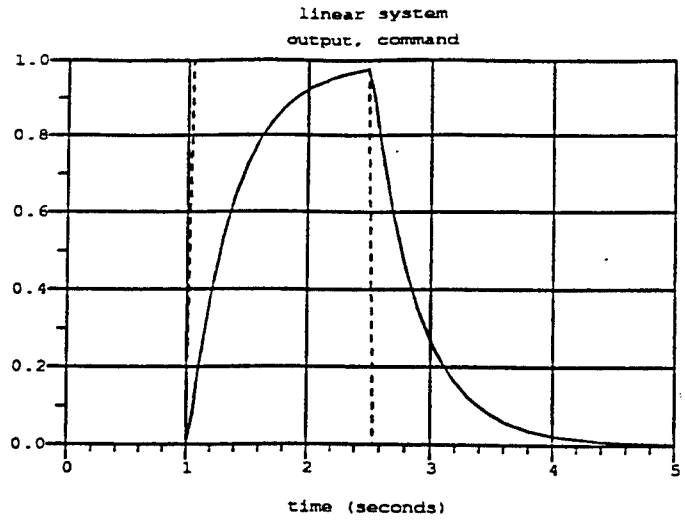
Because control laws are typically implemented in a digital flight control computer, all dynamic analyses must account for the discrete nature of the implementation. Since multivariable design is often carried out in the continuous (Laplace transform) domain, we must discretize the continuous control law. Several discretizations options are discussed in textbooks [Franklin 1990, Astrom 1990, Ackerman 1985].

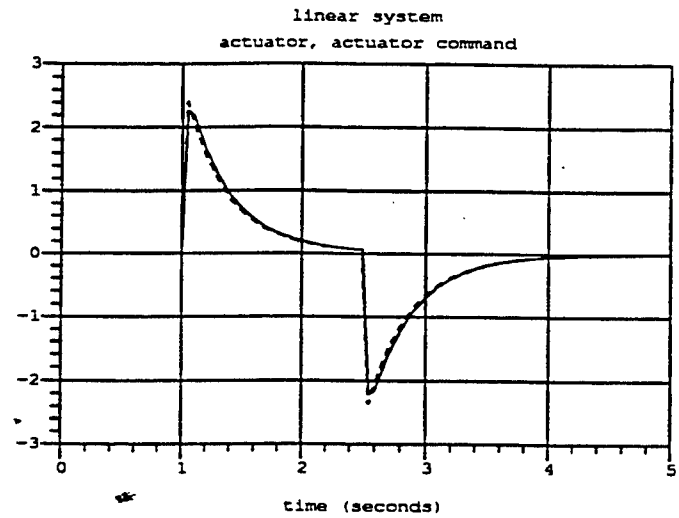
Some control design methodologies are carried out in the discrete domain, making the discretization step unnecessary. This is one area where discrete control law designs have an advantage over their continuous counterparts. Another is when the sample rate is low relative to control bandwidth. In most cases, however, sample rate is much higher than the bandwidth of the flight control system, in which case this advantage disappears. In any case, many control designers have a much better *intuitive feel* for continuous control designs than discrete ones.

8.3 Multichannel Control System

Most flight control laws are implemented with a redundant set of computers (typically 3 or 4) in such a manner that the control system is fault tolerant. Fault tolerance implies that one or more of the channels can fail with no degradation in control system performance and is accomplished as follows: Sensed data is passed to multiple channels (or multiple computers). Each channel processes the control law and derives commands for the actuators. Actuators are then commanded with a mid value (or mean value) or some other appropriate command derived from the multiple channels. When actuator commands disagree (across the different channels), a failure has occurred.

Sensors may also be redundant and inputs to each of the control law channels may be





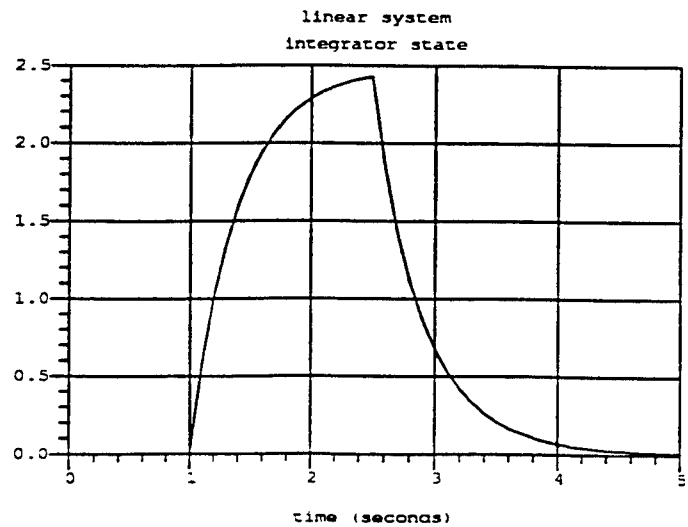
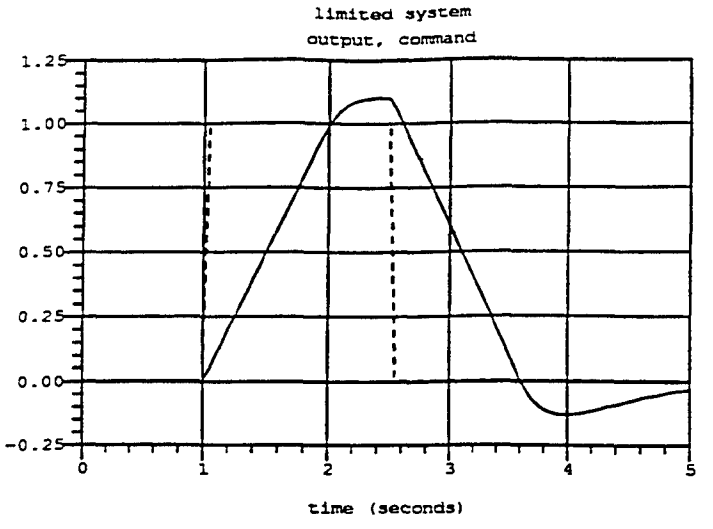
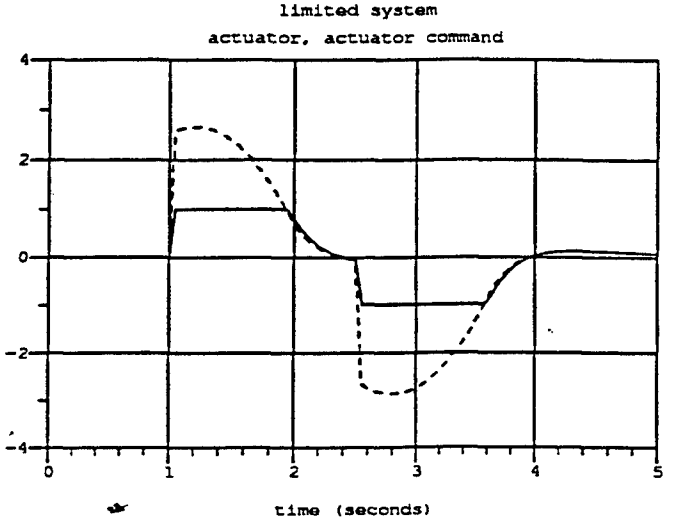


Figure 8.2: Time Histories for Linear Simulation.





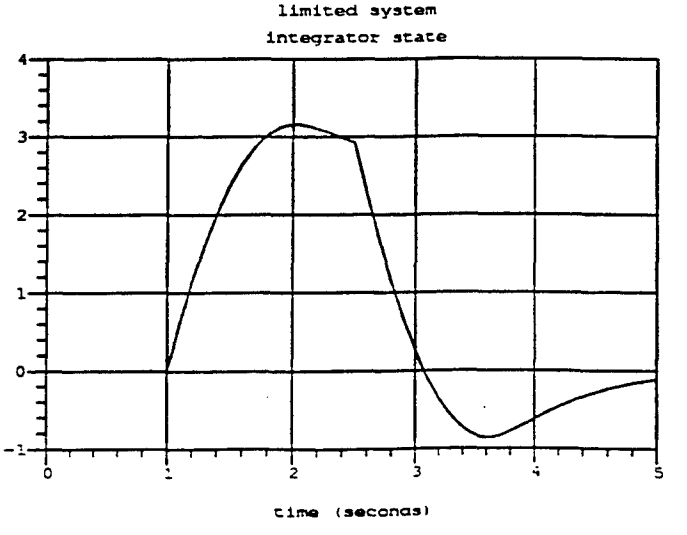


Figure 8.3: Time Histories for Simulation With Limited Actuator.

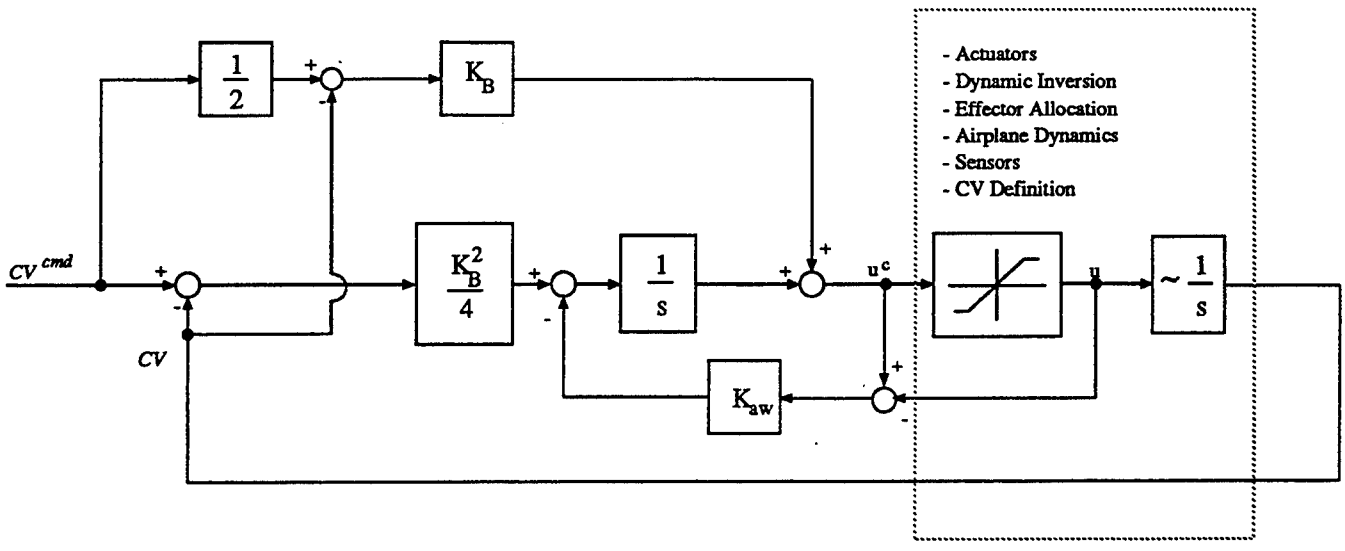


Figure 8.4: Block Diagram of Ideal System With Actuator Limit.

identical if some selection logic is employed (again, average or mid value). Or if each channel only reads one particular sensor, inputs may still be different because of sensor noise or timing differences.

At one extreme, if all sensed values are shared identically by all computers (same hardware), and all computers use the same operational flight program to implement the control law, and actuator commands disagree (by even the least significant bit), it is assumed that a failure has occurred.

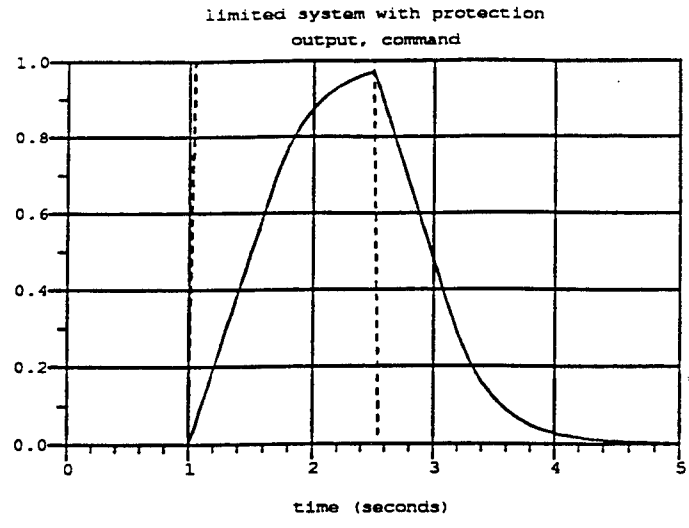
At the other extreme, computers may process sensor data with slightly different timing or may actually sample different noisy sensors. In this case, actuator commands will not agree exactly, even without failures. In this case, if the actuator commands disagree by some preset threshold, it is again assumed that failure has occurred.

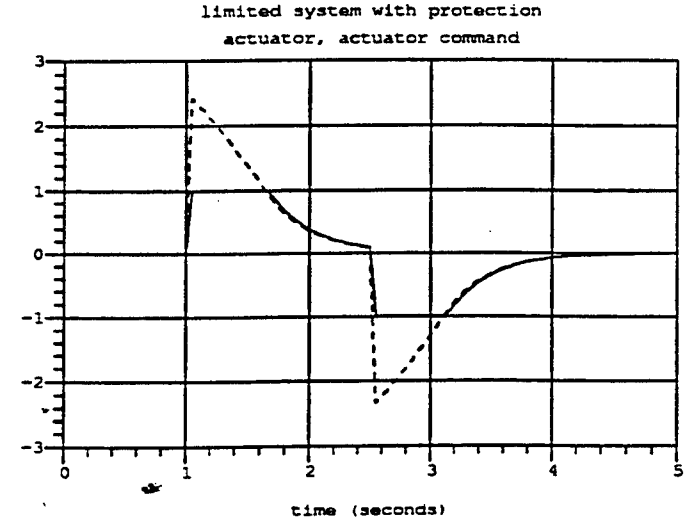
In either case, additional algorithms are used for isolating the failure to one computer or hardware component, which are beyond the scope of this discussion.

8.4 Need for Cross-Channel Data Link

From an analytical point of view, a multichannel implementation of a control law can be thought of as a nonminimal realization of the control law. Consider the diagram in Figure 8.6, where, for convenience, we show only two channels. Assume that the control system is linear, the inputs to each channel are different because of sensor bias, noise, round-off, and/or timing mismatch denoted by n_1, n_2. Assume further, that we use the average of the two control law outputs to command the actuators.

In this special case, it is possible to consider a linear state-space realization of the imple-





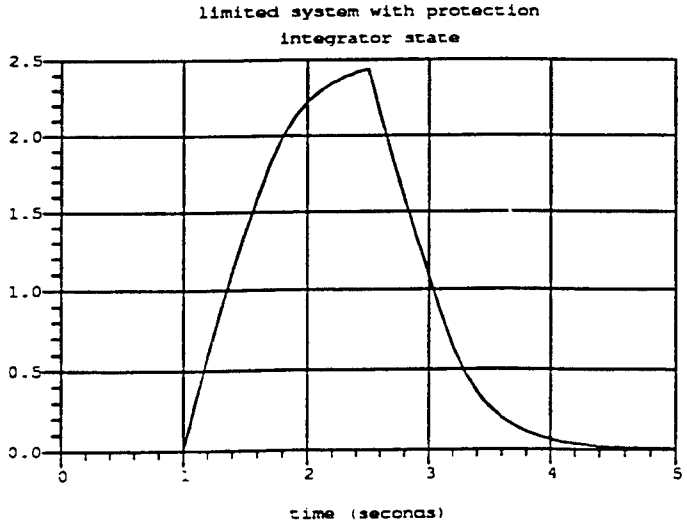


Figure 8.5: Time Histories for Simulation With Antiwindup Protection.

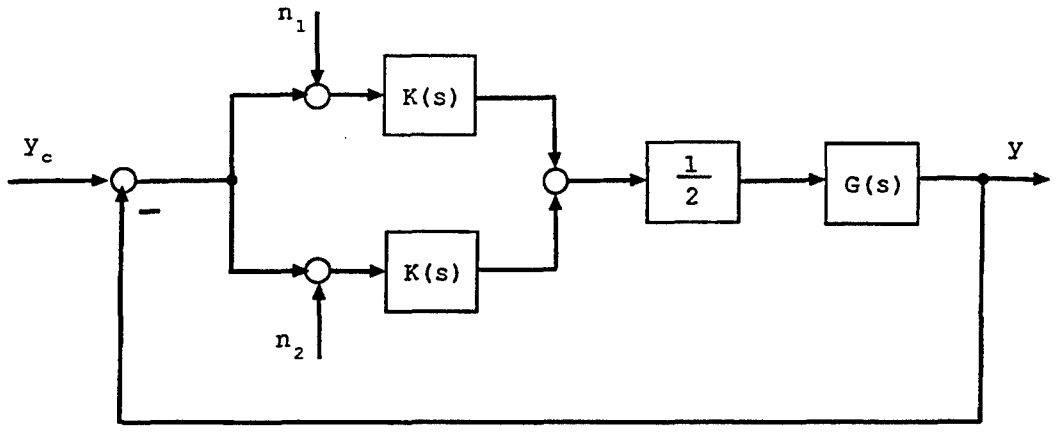


Figure 8.6: Two-Channel Implementation.

mentation, which is just the parallel combination of two identical systems. It is known that such a combination will be a nonminimal realization (either uncontrollable or unobservable, c.f., [Kailath 1980]). Since the parallel combination is not minimal, some internal dynamics may or may not grow without bound, depending on the poles of the compensator transfer function. If a pole of the compensator lies in the closed right half plane, it is possible for an internal variable to grow without bound.

A proportional-plus-integral compensator is of particular interest here since it has a pole at the origin. Any small disturbance can potentially lead to significant mismatches between the two channels, which could then be misinterpreted as a failure.

Classical designers have known for years that integrators in multiple channels must be equalized to prevent problems of the type just described. Figure 8.7 illustrates one approach to equalization. The two integrator outputs are differenced, multiplied by a gain k, and passed back to the integrator inputs. This gain does not change the input/output behavior (i.e., the output is still the integral of the input). This new realization is still nonminimal, but instead of a nonminimal mode at the origin, the nonminimal mode is now at location $-k+j0$. This implies that k must be positive for internal stability. Because this equalization requires communication between the outputs of the two channels and additional inputs to the integrators, it is called a cross-channel data link.

Note that if the compensator poles are in the left half plane, then no cross-channel data link is necessary because the nonminimal modes are stable and will decay on their own without a cross channel data link. On the other hand, compensator poles in the right half plane would require equalization and cross-channel data links to achieve internal stability.

It is also necessary to provide the equalization for every compensator mode that is not sufficiently well damped. Thus, a small pole in the left half plane may warrant equalization even if it is stable and its time constant is long. Even integrators that are not a part of the feedback loop require equalization, which includes integrators associated with

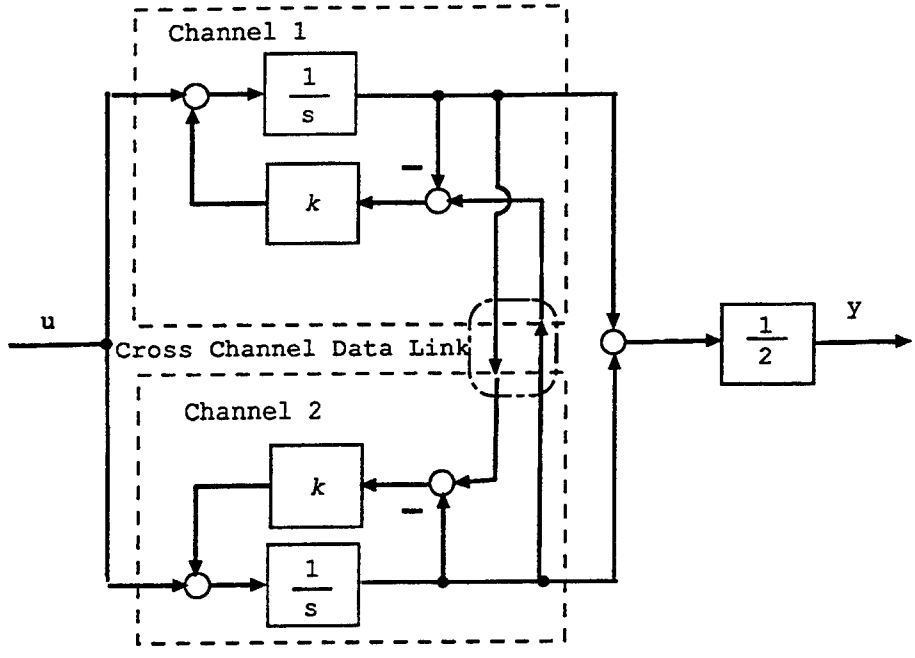


Figure 8.7: Integrator Equalization.

trim or coolie-hat buttons.

There are many other ways to accomplish equalization, and this is an important part of any practical design.

Part III

Methodology Specific

Part III Overview

In this part of the guidelines, the specifics of three multivariable control methodologies are detailed.

The first is eigenstructure synthesis, the second is dynamic inversion, and the third is μ-synthesis.

Section 9

Eigenstructure Assignment Control

In this section, we present a general method for designing controllers using eigenstructure assignment and the framework shown in Figure 9.1. We first select the desired closed-loop eigenstructure to meet MIL-STD-1797 flying qualities requirements for closed-loop frequency and damping of the various dynamic modes. We next design an output feedback gain matrix to meet the desired eigenstructure requirements. We then add integral control to improve robustness to modelling errors and disturbances. Next, we prescribe tracking requirements in terms of three controlled variables that correspond to the desired response from lateral and longitudinal stick and rudder pedals. Finally, we design a dynamic feedforward compensator to achieve the desired controlled variable response.

We used several eigenstructure assignment references in developing the methods described in this section [Sobel 1985a, Sobel/Lallman 1988, Lallman 1985, Srinathkumar 1979, Smith 1991, Sobel 1987, Garrard 1989].

Controller development takes the following steps:

1. **Choose Controlled Variables and Desired Response:** For this first step, we choose a variable for each piloted control input. Conventionally, this ties lateral stick, longitudinal stick, and rudder pedals to a variable (or blend of variables) for the roll, pitch, and yaw axes, respectively. The controlled variable must reflect the pilot's control desires for handling qualities in each axis. Eigenstructure assignment assumes the existence of a precompensator to develop the controlled variable commands, as described in Section 10.4.

2. **Develop Model:** In this step, we define an aircraft model using the equations of motion given in Section 4.1. The eigenstructure design method requires a simple linear model of the dynamics. Thus, we separate longitudinal and lateral/directional models and ignore high-frequency dynamics such as actuator and sensor dynamics.

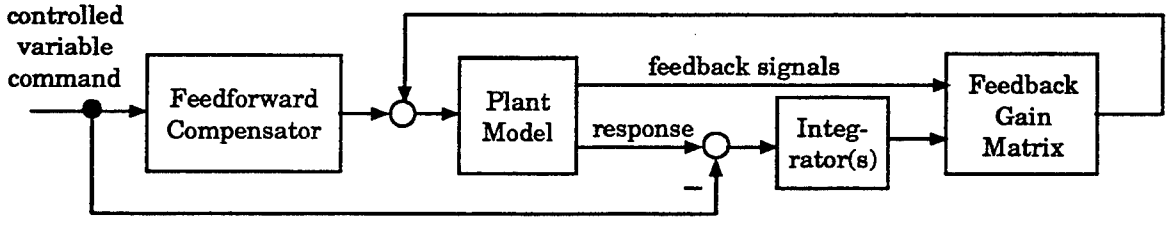


Figure 9.1: Eigenstructure Control Law Framework.

Finally, we reduce the dimension of the model input and develop a model of the controlled variable response for incorporating integral control.

3. **Design Eigenstructure Feedback:** The desired closed-loop eigenstructure consists of the desired modes (or eigenvalues) for the closed-loop aircraft and controller, along with the eigenvectors associated with these modes. We first choose the desired eigenvalues to meet requirements for the specific aircraft and flight condition. Desired eigenvectors associated with these modes need not change for different aircraft or flight conditions because these eigenvectors are simply representations of the basic aircraft modes. Next, we choose an eigenvalue for the pole associated with integral control. Once the desired eigenstructure is defined, we compute the feedback gain matrix using the eigenstructure synthesis algorithm.

4. **Design Feedforward Compensator:** Once the feedback control is designed, we compute the closed-loop transfer function. We design the feedforward compensator by performing a state-space division of the achieved closed-loop transfer function into the desired closed-loop transfer function. Finally, we simplify the high-order feedforward compensator that results from this design using model reduction.

We now elaborate on each of these basic steps.

9.1 Controlled Variables and Desired Response

We discussed controlled variable development at length in Section 5 of the Flight Control Specific part of this document. To develop controlled variable commands, the eigenstructure assignment method assumes that a precompensator exists, as will be described in Section 10.4.

The desired response of the controlled variables (LCV, MCV, NCV) to their respective commands (LCV_{cmd}, MCV_{cmd}, NCV_{cmd}) will be needed later in Section 9.4.1. For simplicity, we assume that a simple lag with a break frequency of 2.5 rad/sec is a good candidate for desired tracking response for each of the three variables; that is:

$$\left(\frac{CV}{CV_{cmd}}\right)_{desired} = H(s) = \frac{2.5}{s+2.5} \tag{9.1}$$

9.2 Model Development

This section describes aircraft models necessary for the eigenstructure controller design. We discuss the longitudinal and lateral-directional models separately.

9.2.1 Linear State-Space Models

Linear state-space models must assume the form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{9.2}$$

where $x \in \Re^n$, $u \in \Re^m$, $A \in \Re^{nxn}$, $B \in \Re^{nxm}$, $C \in \Re^{rxn}$, r is the number of outputs, and m the number of control inputs. We separate models for longitudinal and lateral-directional dynamics. These models should include only the outputs that are available to the controller for feedback, and these outputs must be linearly independent. (See Section 4.1 for a development of linear state-space equations of motion.)

Note that the eigenstructure synthesis method requires that there be no D matrix in the state-space equations. We know of no references that describe an algorithm for non-zero D, and we have been unable to derive one that works. When using a sensor, such as an accelerometer, that includes a D term in the model, a simple work around that eliminates this term is to pass the sensor signal through a low-pass filter with a break frequency that is well beyond desired loop bandwidth. The eigenvalue associated with this filter should be left unassigned in the eigenstructure.

A common state vector for the longitudinal axis is

$$x_{long} = [q \;\; \alpha \;\; V \;\; \gamma]^T \tag{9.3}$$

which was discussed in some depth in Section 4.1.1. We can eliminate the last two states associated with phugoid motion for the design since they have little impact on controlled variable response or short-period stabilization of the aircraft.

A common state vector for the lateral-directional axes is

$$x_{latdir} = [p \;\; r \;\; \beta \;\; \mu]^T \tag{9.4}$$

which, once again, was discussed at length in Section 4.1.1.

9.2.2 Augmenting With Integrators

Integral action on the controlled variable can help to reduce tracking errors and provide additional robustness to modelling errors and disturbances. To achieve this using eigenstructure assignment, we augment the *aircraft model* with an integrator, or multiple integrators when we have more than one controlled variable. As Figure 9.2 shows, the input to each integrator is a controlled variable. For the lateral-directional case, which has two controlled variables, we add two integrators.

For the final implementation, we again add back these same integrators to the *feedback controller*. The inputs to the integrators in the controller are the controlled variable errors, not just the controlled variables themselves, as illustrated in Figure 9.3.

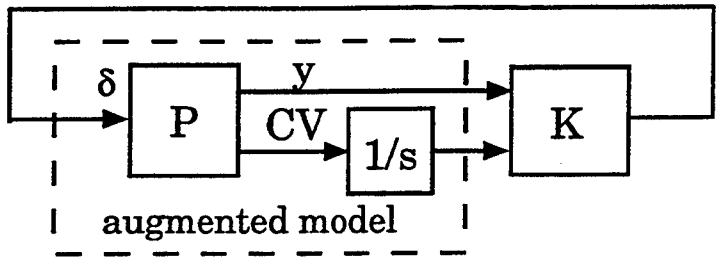


Figure 9.2: Augmented Model for Feedback Control Design.

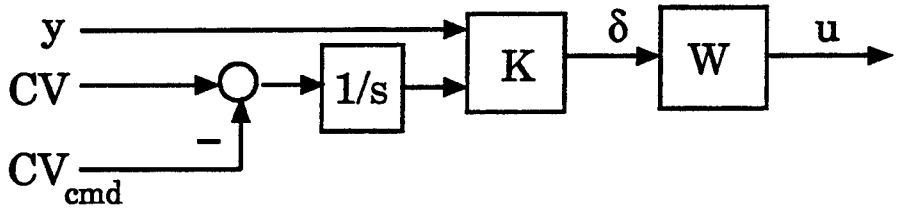


Figure 9.3: Augmented Controller for Implementation.

9.3 Eigenstructure Assignment

Eigenstructure assignment allows the control designer to place closed-loop modes of the aircraft in desired "directions" and frequencies. Standard aircraft requirements easily lend themselves to the desired eigenstructure framework because they commonly specify frequency and damping for basic modes. The feedback gain matrix is computed in a straightforward manner from the aircraft model and the desired eigenstructure.

9.3.1 Desired Eigenstructure

The first step in designing an feedback controller using eigenstructure assignment is to choose the desired eigenstructure, which we accomplish by choosing a set of desired eigenvalues and a matching set of eigenvectors. In Section 9.3.3.1, we will show that the eigenvector corresponding to a given eigenvalue can be chosen from an m dimensional subspace, where m is the number of independent control inputs. In general, arbitrary selection of an eigenvector is not possible. Two basic approaches can be used to resolve this: (1) partial specification of the desired eigenvector, or 2) projecting the desired eigenvector onto the achievable subspace. In practice, we often mix these two methods. There is no need to specify an eigenvector for every eigenvalue. Leaving some eigenvectors (or some of their elements) free allows the projection of the desired eigenvector onto the achievable subspace to give an achievable eigenvector that better meets requirements.

The basic modes of an aircraft are standard: longitudinal short period, phugoid, dutch roll, roll subsidence, and spiral. Thus, desired *eigenvectors* need not be modified for dif-

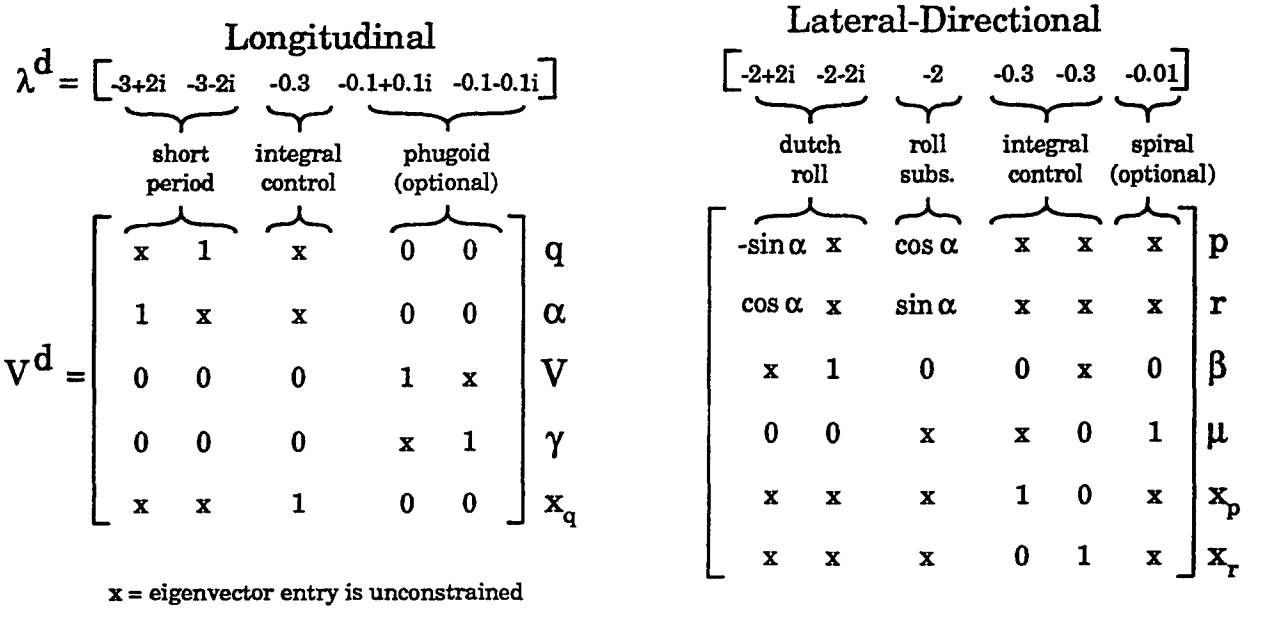


Figure 9.4: Desired Eigenvectors and Candidate Eigenvalues.

ferent applications. Desired frequencies (i.e., *eigenvalues*) for these modes, on the other hand, may well vary across different aircraft and/or flight conditions. Figure 9.4 illustrates (candidate) desired eigenvalues λ^d along with desired eigenvectors V^d for both the longitudinal and lateral-directional axes. Here "x" denotes an entry that is unconstrained (i.e., arbitrary), whereas all other values (including "0") denote hard eigenvector constraints. The states x_p, x_q, x_r are those associated with integral control of LCV, MCV, NCV, respectively.

The specific format chosen for specifying the desired eigenstructure matrix for computer implementation is slightly different from that suggested in Figure 9.4. To avoid nonnumeric input, all arbitrary elements are actually entered as zeros rather than the x's shown. Unfortunately, this introduces an ambiguity between arbitrary eigenvector elements and those that are to be explicitly assigned the value 0. To avoid this ambiguity, we introduce two additional quantities. We first define a row vector N_s, whose individual elements N_{s_i} specify the number of entries in each eigenvector V_i^d that are to be explicitly assigned, as Figure 9.5 shows. Next, we define another matrix N whose columns N_i specify the state indices corresponding to the entries in V_i^d that are to be assigned, also shown in Figure 9.5. With these definitions, the algorithm to be described in Section 9.3.3 carries out the design.

In the subsections that follow, we describe the standard aircraft modes listed above in the context of desired eigenvalues and eigenvectors.

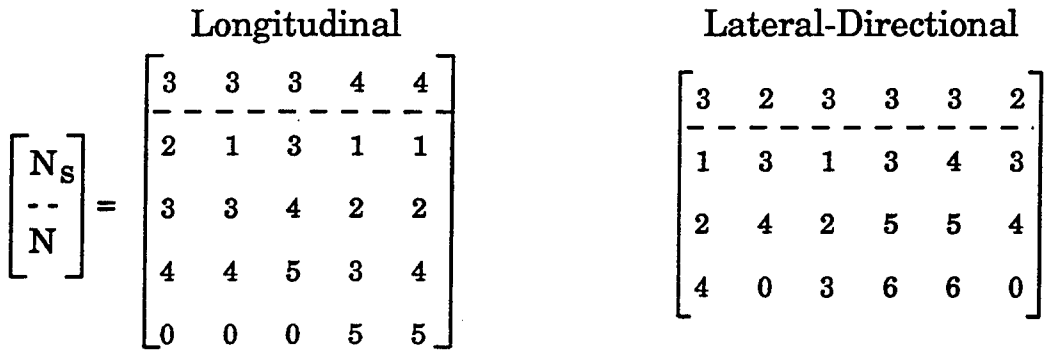


Figure 9.5: Number and Indices of Eigenvector Entries.

9.3.1.1 Short Period

Short-period motion of an aircraft involves an oscillation between the pitch rate q and the angle of attack α. Velocity and flight-path angle should change very little during this motion. The two eigenvectors for the short period are chosen to place the motion into the q and α states, and keep the V and γ response minimal. The selection of one eigenvector with ($q = 1, \alpha =$ arbitrary) and the second with ($q =$ arbitrary, $\alpha = 1$) couples the motion for this eigenvalue pair with the states q and α. We assign the V and γ elements of both eigenvectors to zero to reflect our desire that the short period mode not affect these states.

9.3.1.2 Phugoid

Phugoid motion involves an oscillation between velocity V and flight-path angle γ. This mode is usually slow enough that the pilot can control it. The motions of q and α due to this mode should remain small. We choose the two eigenvectors for this mode so as to couple the V and γ states together while minimizing q and α motions.

9.3.1.3 Dutch Roll

Dutch roll motion is an oscillation between yaw rate r and sideslip angle β. At high angles of attack, it is desirable that the oscillation be relative to stability-axis yaw rate r_s rather than body-axis yaw rate r. We choose the two eigenvectors to place the mode into r_s and β while holding the response in the bank angle μ to a minimum. We use equations 9.5 and 9.6, which define p_s and r_s, to form the desired eigenvectors in terms of p and r to provide motion in the directions of the stability axes. In equation 9.6, with $p_s = 0$, the desired quantities for p and r can be read from the second column of the matrix.

$$\begin{bmatrix} p_s \\ r_s \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} \tag{9.5}$$

$$\begin{bmatrix} p \\ r \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} p_s \\ r_s \end{bmatrix} \tag{9.6}$$

9.3.1.4 Roll Subsidence

The roll subsidence mode represents the decay of rolling velocity. It is typically measured in the body axes as the first-order decay of roll rate p. At high angles of attack, it is desirable that this motion be relative to stability-axis roll rate p_s, rather than body axes roll rate p. To do so, we use equation 9.6, with $r_s = 0$, to form the desired eigenvectors, much like was done for dutch roll. The roll subsidence mode should not contribute to β, but the roll angle μ can be arbitrary.

9.3.1.5 Spiral

The spiral mode represents the decay of roll angle. This mode is usually slow enough that the pilot can control it. We choose the eigenvector with p and r entries arbitrary to allow motions necessary to roll the vehicle about the stability roll axis. This mode should not affect the sideslip angle β, so that entry is set to zero.

9.3.1.6 Integrator Modes

It is not obvious how to pick eigenvectors associated with integral action. The controlled variables contribute primarily to high-frequency motions, such as p, q, r, but also include terms with α, β, μ, and even the phugoid states. Candidate selections for these eigenvectors appear in Figure 9.4.

9.3.2 Basics of Eigenstructure Assignment

Consider the state-space system of equations given in 9.2 with feedback control law

$$u = Ky \tag{9.7}$$

Assume that the system described by (A, B, C) is controllable and observable and that B and C are full rank. Using output feedback, the achievable eigenstructure is limited by the dimensions of the control inputs (m) and the sensor outputs (r). A total of $\max(m, r)$ closed-loop eigenvalues can be assigned, and the associated eigenvectors can be selected from a $\min(m, r)$-dimensional subspace.

In typical flight control applications, $r > m$, so that r eigenvalues can be assigned and the eigenvectors can be arbitrarily chosen from an m-dimensional subspace. Note, in particular, that the longitudinal axis typically has only one effective control, so eigenvector directions cannot be assigned for each eigenvalue chosen because the achievable subspace is only one-dimensional. The desired eigenvectors in this case do not affect the feedback gain matrix computed.

9.3.3 Eigenstructure Synthesis

Once the model and the desired eigenstructure have been defined, we can compute the feedback gain matrix using a straightforward algorithm described below (derived from [Sobel 1985a]).

Output feedback eigenstructure assignment allows the placement of r eigenvalues of the closed-loop system with little control over the remaining $n-r$ eigenvalues, which may well be unstable. We must be careful in selecting eigenvalues and eigenvectors and diligent in analyzing the resulting system to ensure that stability and performance requirements are met. If available, additional sensor outputs may aid in achieving a more complete assignment of closed-loop eigenvalues.

For complex eigenvalues, the computations below require complex arithmetic. Alternative computer algorithms using real arithmetic are also possible when complex arithmetic is not available [Sobel 1985a].

9.3.3.1 Step 1: Achievable Eigenvectors

In this first step, we determine the achievable eigenvectors. For the system described by equation 9.2, the achievable m-dimensional eigenvector subspace for the i-th desired eigenvalue λ_i^d is

$$L_i = (\lambda_i^d I - A)^{-1} B \tag{9.8}$$

We first select the rows from both the achievable and desired eigenvector subspaces that correspond to the entries in the i-th column of the matrix N (recall Figure 9.5):

$$\tilde{L}_i = \text{selection}(L_i, N_i) \tag{9.9}$$

$$l_i = \text{selection}(V_i^d, N_i) \tag{9.10}$$

For example, if the fifth eigenvector contains only two constrained entries, say the second and fourth entries, then $\tilde{L}_5$ consists of the second and fourth rows of L_5, and l_5 consists of the second and fourth entries of V_5^d (recall V^d from Figure 9.4).

We next compute the achievable eigenvector V_{A_i} by projecting the desired eigenvector onto the achievable eigenvector subspace:

$$V_{A_i} = L_i \, p_i \tag{9.11}$$

where p_i is a projection vector that is determined by a least squares error fit:

$$p_i = \begin{cases} \tilde{L}_i^T(\tilde{L}_i\tilde{L}_i^T)^{-1}l_i & N_{s_i} < m \\ (\tilde{L}_i^T\tilde{L}_i)^{-1}\tilde{L}_i^T l_i & N_{s_i} \geq m \end{cases} \tag{9.12}$$

Repeating the above procedure for each eigenvalue/eigenvector pair, we assemble the achievable eigenvectors into a matrix:

$$V = [V_{A_1}\ V_{A_2}\ \ldots\ V_{A_r}] \tag{9.13}$$

9.3.3.2 Step 2: Similarity Transformation

In this second step, we apply a similarity transformation to simplify computations. First, we perform a singular value decomposition on the B matrix:

$$B = U\Sigma V^T \tag{9.14}$$

Next, we define a transformation matrix T by augmenting the B matrix with the last $n - m$ columns of its left singular vectors:

$$T = [B\ U_{m+1}\ \ldots\ U_n] \tag{9.15}$$

This choice makes T invertable, assuming B is full rank, and ensures that

$$T\begin{bmatrix} I_m \\ 0 \end{bmatrix} = B \tag{9.16}$$

Finally, we compute the transformed state-space matrices:

$$\tilde{A} = T^{-1}AT \tag{9.17}$$

$$\tilde{B} = T^{-1}B = \begin{bmatrix} I_m \\ 0 \end{bmatrix} \tag{9.18}$$

$$\tilde{C} = CT \tag{9.19}$$

so that the transformed achievable eigenvectors become

$$\tilde{V} = T^{-1}V \tag{9.20}$$

9.3.3.3 Step 3: Feedback Gain Matrix

In this last step, we evaluate the feedback gain matrix. We first scale the first m rows of each transformed eigenvector by its associated eigenvalue and assemble them into a matrix:

$$Z = [\lambda_1^d z_1\ \ldots\ \lambda_r^d z_r] \tag{9.21}$$

where the z_i are the first m elements of the transformed eigenvector $\tilde{V}_i$. Next, we select the first m rows of the transformed A matrix:

$$A_1 = \text{first } m \text{ rows of } \tilde{A} \tag{9.22}$$

Finally, we compute the feedback gain matrix using

$$K = \text{real}\left[(Z - A_1\tilde{V})\,(\tilde{C}\tilde{V})^{-1}\right] \tag{9.23}$$

9.4 Feedforward Compensation

Recall that the *feedback* compensator is designed to meet the desired natural frequencies (or modes) of the system and to ensure good stability margins. The *feedforward* compensator, on the other hand, is designed to provide good tracking response to pilot commands.

The method to be described in this section assumes that the feedforward compensator requires dynamics to meet tracking requirements. If the inner-loop control system is designed with tracking requirements in mind, it may suffice to design a constant gain feedforward compensator. One method for designing such a feedforward gain matrix [Broussard 1978] has been used by other investigators [Sobel 1985b, Smith 1991]. We do not recommend this method because it is generally better to design the feedback controller for desired closed-loop properties and achieve desired tracking response with precompensation.

The first step in designing the feedforward compensator is to develop the closed-loop transfer functions from the $\Delta\delta$ and CV_{cmd} inputs to the CV output with the feedforward compensator removed or set to zero. These two transfer functions, denoted by G_1 and G_2 in Figure 9.6, should not include unnecessary high frequency dynamics, such as actuator and sensor dynamics. Moreover, they should have first- or second-order rolloff at high frequency to allow computation of the feedforward compensator using state-space division (to be described in Section 9.4.2).

9.4.1 Desired Tracking Response

Let H denote the desired tracking response from the controlled variable command CV_cmd to the controlled variable response CV in Figure 9.6. Then, for F to be realizable, H must have a high-frequency rolloff rate at least as great as that for G_1 and G_2.

A simple lag with break frequency at 2.5 rad/sec was suggested earlier as a good candidate for the desired tracking response:

$$H(s) = \frac{2.5}{s + 2.5} \tag{9.24}$$

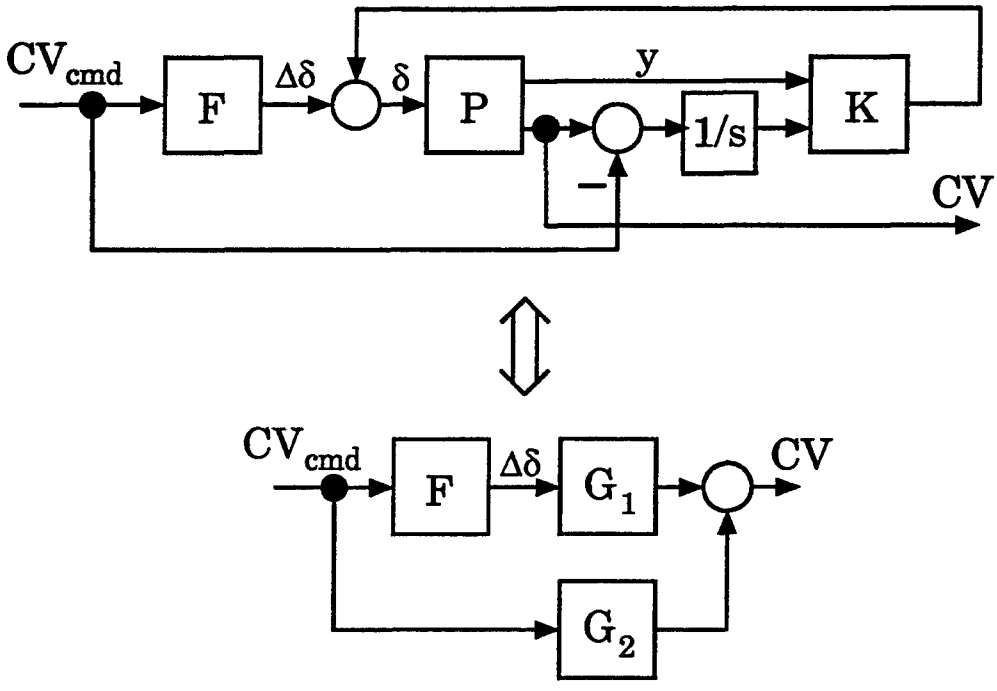


Figure 9.6: Framework for Design of Feedforward Compensator.

9.4.2 Feedforward Compensator Computation

We choose the feedforward compensator such that the closed-loop-command response matches the desired response. From Figure 9.6, the transfer function from CV_{cmd} to CV is:

$$\frac{CV}{CV_{cmd}}(s) = G_1F + G_2 = H \tag{9.25}$$

Solving for F gives

$$F = G_1^{-1}(H - G_2) \tag{9.26}$$

provided G_1 is invertable. When G_1 is not invertable, F is not directly computable. But, if $(H - G_2)$ has a rolloff rate greater than or equal to that for G_1, we can still compute F using state-space division [Enns 1984].

9.4.3 Model Reduction

The feedforward compensator F computed by state-space division is typically high order having many internal pole/zero cancellations and should be reduced using model reduction techniques. One method for obtaining a minimal realization using internal balancing [Enns 1984] is discussed in Appendix B, Section B.1.

Figure 9.7 shows the complete controller implementation.

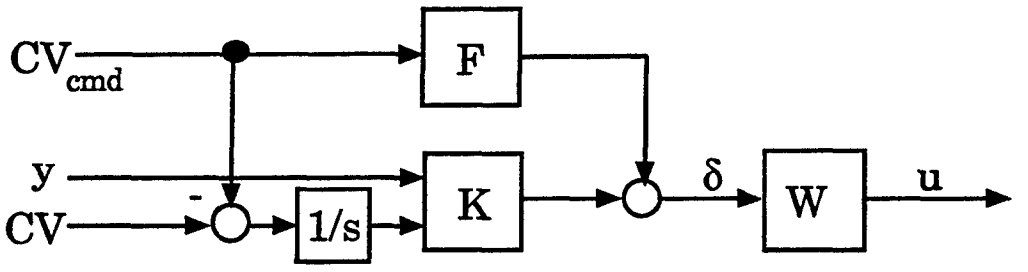


Figure 9.7: Feedforward/Feedback Controller Implementation.

9.5 Example

We here present a numerical example to help clarify the steps in the eigenstructure assignment method just described. The aircraft model is for an F-16 MATV model and the flight condition is for Mach 0.375, 40,000 feet altitude, and 45 deg angle of attack.

We develop control designs separately for the longitudinal and lateral-directional axes.

9.5.1 Longitudinal Example

The linear model for longitudinal-axis dynamics is

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \\ \dot{V} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} -0.2461 & 0.6399 & .000376 & 0.0 \\ 0.9838 & 0.0356 & -.000347 & 0.05327 \\ -5.647 & -33.754 & -0.1195 & -25.56 \\ 0.0162 & -0.0356 & .000347 & -.05327 \end{bmatrix} \begin{bmatrix} q \\ \alpha \\ V \\ \gamma \end{bmatrix}$$

$$+ \begin{bmatrix} -0.2086 & -0.2086 & -0.9413 \\ -0.00569 & -0.00569 & -.008169 \\ -4.772 & -4.772 & -3.426 \\ 0.00569 & 0.00569 & 0.008169 \end{bmatrix} \begin{bmatrix} \delta_{Ht_l} \\ \delta_{Ht_r} \\ \delta_{TVC_z} \end{bmatrix} \tag{9.27}$$

$$\begin{bmatrix} q \\ \alpha \\ V \\ MCV \\ nz_{cg} \\ nz_{accel} \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 1.0 & 0.1068 & -.000046 & 0.02561 \\ 0.2492 & 0.3221 & 0.005415 & 0.0 \\ 0.1387 & 0.6094 & 0.005585 & 0.0 \end{bmatrix} \begin{bmatrix} q \\ \alpha \\ V \\ \gamma \end{bmatrix}$$

$$+\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.1508 & 0.1508 & 0.1410 \\ 0.0572 & 0.0572 & -.2816 \end{bmatrix} \begin{bmatrix} \delta_{Ht_l} \\ \delta_{Ht_r} \\ \delta_{TVC_z} \end{bmatrix} \tag{9.28}$$

Control effectors for the longitudinal axis include left and right horizontal tail (δ_{Ht_l}, δ_{Ht_r}) and longitudinal thrust vectoring (δ_{TVC_z}). Outputs of interest include all of the states except γ plus normal acceleration at the cg and accelerometer locations.

Additionally, the model includes actuator dynamics approximated by simple lag filters for each control input, with a break frequency of 30 rad/sec.

The desired controlled variable response is given by a simple lag.

$$MCV_{desired} = \left(\frac{2.5}{s+2.5}\right) MCV_{cmd} \equiv H(s)\, MCV_{cmd} \tag{9.29}$$

We reduce the control dimension by a simple ganging of the two horizontal tails and the thrust vectoring as follows:

$$\begin{bmatrix} \delta_{Ht_l} \\ \delta_{Ht_r} \\ \delta_{TVC_z} \end{bmatrix} = \begin{bmatrix} 30.0 \\ 30.0 \\ 20.0 \end{bmatrix} \delta \equiv W\,\delta \tag{9.30}$$

These ganging coefficients were chosen to allow all three effectors to reach their maximum deflection limit when δ is one, while assisting one another in providing pitch moment.

Outputs appropriate for feedback include q, α, V, and $\int MCV$.

Thus, the state-space model used for the eigenstructure assignment is

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \\ \dot{V} \\ \dot{\gamma} \\ \int \dot{MCV} \end{bmatrix} = \begin{bmatrix} -0.2461 & 0.6399 & .000376 & 0.0 & 0.0 \\ 0.9838 & 0.0356 & -.000347 & 0.05327 & 0.0 \\ -5.647 & -33.754 & -0.1195 & -25.56 & 0.0 \\ 0.0162 & -0.0356 & .000347 & -.05327 & 0.0 \\ 1.0 & 0.1068 & -.000046 & 0.02561 & 0.0 \end{bmatrix} \begin{bmatrix} q \\ \alpha \\ V \\ \gamma \\ \int MCV \end{bmatrix} + \begin{bmatrix} -31.34 \\ -0.5049 \\ -354.8 \\ 0.5049 \\ 0.0 \end{bmatrix} \delta \tag{9.31}$$

$$\begin{bmatrix} q \\ \alpha \\ V \\ \int MCV \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} q \\ \alpha \\ V \\ \gamma \\ \int MCV \end{bmatrix} \tag{9.32}$$

Section 9.3.1 described the desired eigenstructure. Because there are only four outputs in the model, only four eigenvalues can be assigned. Therefore, we choose to ignore the

second half of the phugoid mode, which means that the closed-loop phugoid mode will not be precisely placed where it is desired.

The eigenstructure synthesis algorithm produces the following feedback controller for the model given in equations 9.31 and 9.32:

$$K = \begin{bmatrix} 0.2032 & 0.2214 & -0.001401 & 0.2518 \end{bmatrix} \tag{9.33}$$

In terms of the original model given by equations 9.27 and 9.28, the controller is

$$K_{orig} = \begin{bmatrix} 30.0 \\ 30.0 \\ 20.0 \end{bmatrix} K = \begin{bmatrix} 6.098 & 6.644 & -0.04204 & 7.556 \\ 6.098 & 6.644 & -0.04204 & 7.556 \\ 4.065 & 4.429 & -0.02803 & 5.037 \end{bmatrix} \tag{9.34}$$

We next develop a state-space description for the closed-loop transfer functions G_1 and G_2 using the methods presented in Section 9.4 from the open-loop state-space description (equations 9.31 and 9.32) and the controller (equation 9.33):

$$MCV = \left[\begin{array}{ccccc|cc} -6.617 & -6.302 & 0.04431 & 0 & -7.895 & -31.34 & 0 \\ 0.8811 & -.07624 & 0.00036 & 0.05327 & -.1271 & -.5049 & 0 \\ -77.78 & -112.3 & 0.3779 & -25.56 & -89.38 & -354.8 & 0 \\ 0.1188 & 0.0762 & -0.00036 & -.05327 & 0.1271 & 0.5049 & 0 \\ 1 & 0.1068 & -.000046 & 0.02561 & 0 & 0 & -1 \\ \hline 1 & 0.1068 & -.000046 & 0.02561 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} \Delta\delta \\ MCV_{cmd} \end{bmatrix} \tag{9.35}$$

We generate the feedforward compensator using state-space division, as described in Section 9.4, followed by pole/zero cancellation, to give

$$F(s) = (-0.07968)\frac{(s+0.06682)(s+0.4835)(s^2+2(0.5927)(2.243)s+(2.243)^2)}{(s+0.1989)(s+2.5)(s^2+2(0.3633)(0.08048)s+(0.08048)^2)} \tag{9.36}$$

Finally, we construct the complete compensator by assembling the dynamic feedforward compensator $F(s)$, the static feedback gain K, and the ganging matrix W, as Figure 9.7 illustrates.

Figure 9.8 shows closed-loop system response to a step command in MCV.

9.5.2 Lateral-Directional Example

The linear model for lateral-directional axes dynamics is

$$\begin{bmatrix} \dot{p} \\ \dot{r} \\ \dot{\beta} \\ \dot{\mu} \\ \dot{x_{NCV}} \end{bmatrix} = \begin{bmatrix} -0.7437 & 0.5440 & -2.749 & 0.0 & 0.0 \\ 0.1584 & -0.2160 & -1.149 & 0.0 & 0.0 \\ 0.7070 & -0.7070 & -0.01705 & 0.0875 & 0.0 \\ 1.0 & 0.1386 & 0.0 & 0.0 & 0.0 \\ 0.7071 & -0.7071 & -0.007479 & 0.07067 & -0.1 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \mu \\ x_{NCV} \end{bmatrix}$$

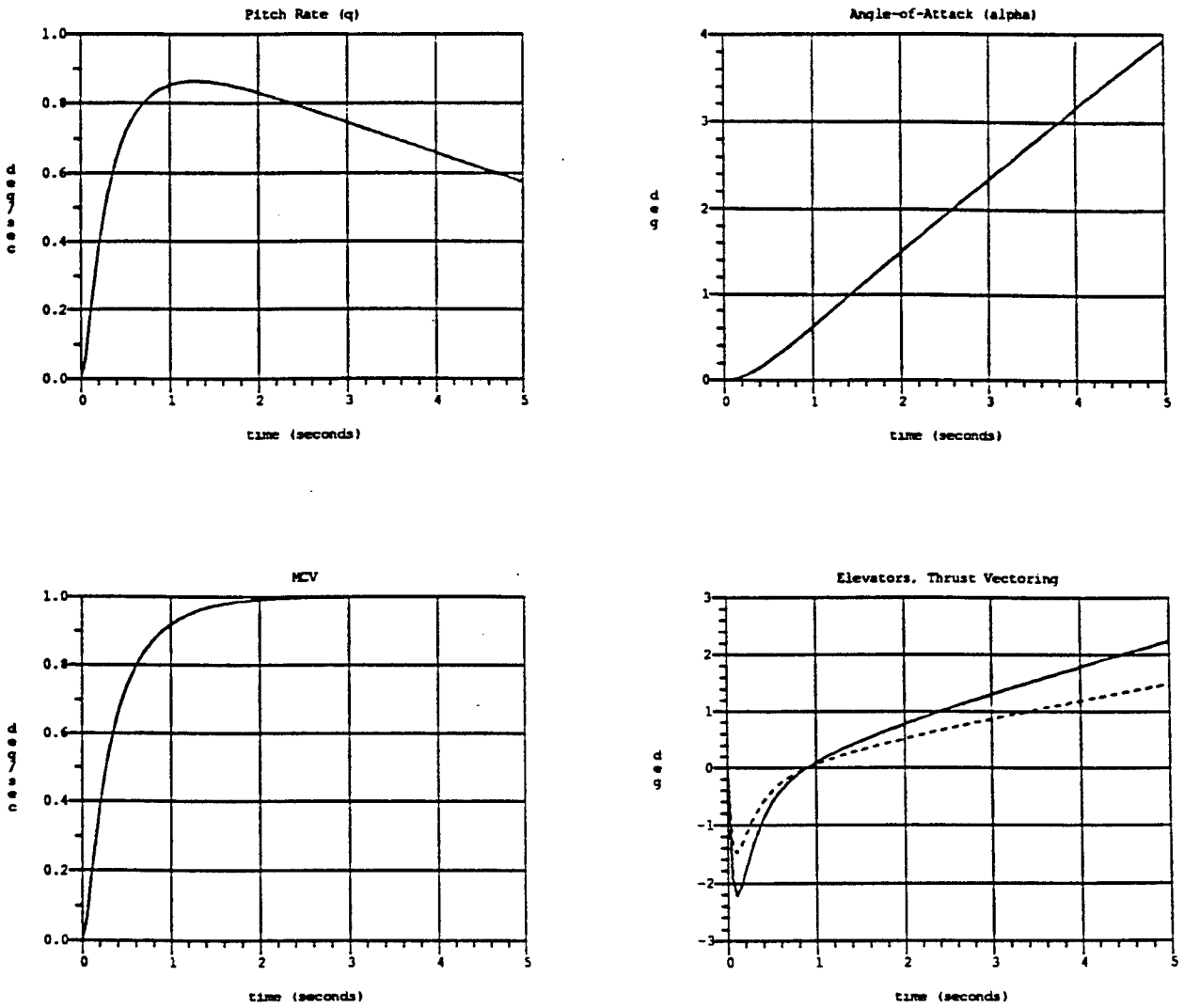


Figure 9.8: Closed-Loop Response to Longitudinal (MCV) Step Input.

$$+\begin{bmatrix} 0.8153 & -0.0003738 & 0.3234 & -0.004691 \\ -0.1595 & -0.06637 & -0.1655 & -0.8330 \\ 0.0 & 0.0 & 0.002820 & 0.01249 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.4118 & 1.824 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_f \\ \delta_r \\ \delta_{TVC_y} \end{bmatrix} \tag{9.37}$$

$$\begin{bmatrix} p \\ r \\ \beta \\ LCV \\ NCV \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.7071 & 0.7071 & 0.0 & 0.0 & 0.0 \\ -0.7071 & 0.7071 & -2.482 & -0.07067 & 0.1 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \mu \\ x_{NCV} \end{bmatrix} \tag{9.38}$$

where x_{NCV} is a state associated with the computation of NCV. Additionally, the model includes actuator dynamics approximated by simple lag filters for each control input, with a break frequency at 30 rad/sec. Note that the control inputs have already been partially ganged to form two effective ailerons: differential tail (δ_a) and differential wing flaps (δ_f). The other control inputs include rudder (δ_r), and lateral thrust vectoring (δ_{TVC_y}). Outputs of interest include all of the states except μ plus LCV and NCV.

The desired controlled variable responses are given by simple lags:

$$\begin{bmatrix} LCV_{desired} \\ NCV_{desired} \end{bmatrix} = \begin{bmatrix} \frac{2.5}{s+2.5} & 0 \\ 0 & \frac{2.5}{s+2.5} \end{bmatrix} \begin{bmatrix} LCV_{cmd} \\ NCV_{cmd} \end{bmatrix} \equiv H(s) \begin{bmatrix} LCV_{cmd} \\ NCV_{cmd} \end{bmatrix} \tag{9.39}$$

We reduce the control dimension using the pseudo-control method described in Section 6.2. We choose scaling matrices to stress rolling and yawing moments (the first and second rows of B) and to scale control inputs by their maximum values.

$$S_l = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \varepsilon & 0 \\ 0 & 0 & 0 & \varepsilon \end{bmatrix} \tag{9.40}$$

$$S_r = \Delta_{max} = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 15 \end{bmatrix} \tag{9.41}$$

where $\varepsilon = 10^{-6}$. This results in the control ganging law:

$$\begin{bmatrix} \delta_a \\ \delta_{ep} \\ \delta_r \\ \delta_{TVC_y} \end{bmatrix} = \begin{bmatrix} -48.35 & -5.911 \\ -0.03424 & 0.5148 \\ -7.391 & 6.289 \\ -0.9671 & 14.53 \end{bmatrix} \equiv W\delta \tag{9.42}$$

Outputs appropriate for feedback here include p, r, β, $\int LCV$, and $\int NCV$.

Thus, the state-space model used for the eigenstructure assignment is

$$\begin{bmatrix} \dot{p} \\ \dot{r} \\ \dot{\beta} \\ \dot{\mu} \\ \int \dot{LCV} \\ \int \dot{NCV} \\ \dot{x}_{NCV} \end{bmatrix} = \begin{bmatrix} -0.7437 & 0.5440 & -2.749 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.1584 & -0.2160 & -1.149 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.7070 & -0.7070 & -0.01705 & 0.08752 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.1386 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.7071 & 0.7071 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.7071 & 0.7071 & -2.482 & -0.07067 & 0.0 & 0.0 & 0.1 \\ 0.7071 & -0.7071 & -0.007479 & 0.07067 & 0.0 & 0.0 & -0.1 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \mu \\ \int LCV \\ \int NCV \\ x_{NCV} \end{bmatrix}$$

$$+ \begin{bmatrix} -41.806 & -2.853 \\ 9.744 & -12.24 \\ -0.03292 & 0.1993 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ -4.808 & 29.11 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_z \end{bmatrix} \tag{9.43}$$

$$\begin{bmatrix} p \\ r \\ \beta \\ \int LCV \\ \int NCV \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \mu \\ \int LCV \\ \int NCV \\ x_{NCV} \end{bmatrix} \tag{9.44}$$

Section 9.3.1 described the desired eigenstructure. Because there are only five outputs in the model, only five eigenvalues can be assigned. Thus, the spiral mode is ignored.

Here again, we used the eigenstructure synthesis algorithm to produce the following feedback controller for the model given in equations 9.43 and 9.44:

$$K = \begin{bmatrix} 0.08044 & -0.05174 & 0.1092 & -0.03204 & -0.1784 \\ -0.1671 & 0.4164 & -0.8528 & 0.2256 & 0.7992 \end{bmatrix} \tag{9.45}$$

In terms of the original model given by equations 9.37 and 9.38, the controller is

$$K_{orig} = W\,K = \begin{bmatrix} -2.901 & 0.04063 & -0.2405 & 0.2152 & 3.904 \\ -0.0888 & 0.2161 & -0.4428 & 0.1172 & 0.4175 \\ -1.645 & 3.001 & -6.171 & 1.656 & 6.345 \\ -2.507 & 6.104 & -12.50 & 3.312 & 11.79 \end{bmatrix} \tag{9.46}$$

We next develop a state-space description for the closed-loop transfer functions G_1 and G_2 again using the methods presented in Section 9.4 from the open-loop state-space description (equations 9.43 and 9.44) and the controller (equation 9.45):

Once again, we generate the feedforward compensator using state-space division, as described in Section 9.4, and simplify the compensator using internal balancing.

Finally, we assemble the complete compensator from the elements just defined according to Figure 9.7.

Figures 9.9 and 9.10 show closed-loop system response to step commands in LCV and NCV.

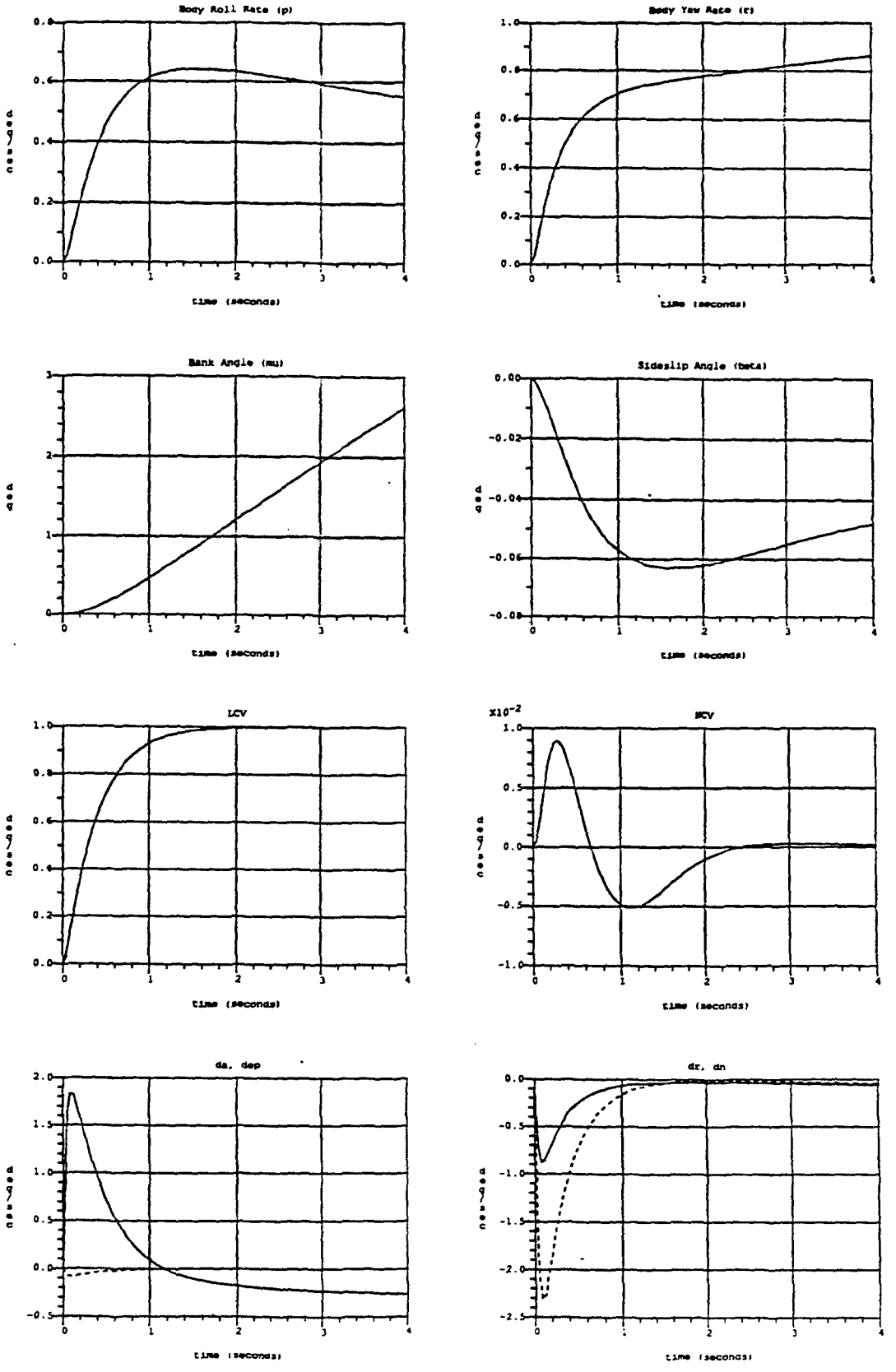


Figure 9.9: Closed-Loop Response to Lateral (LCV) Step Input.

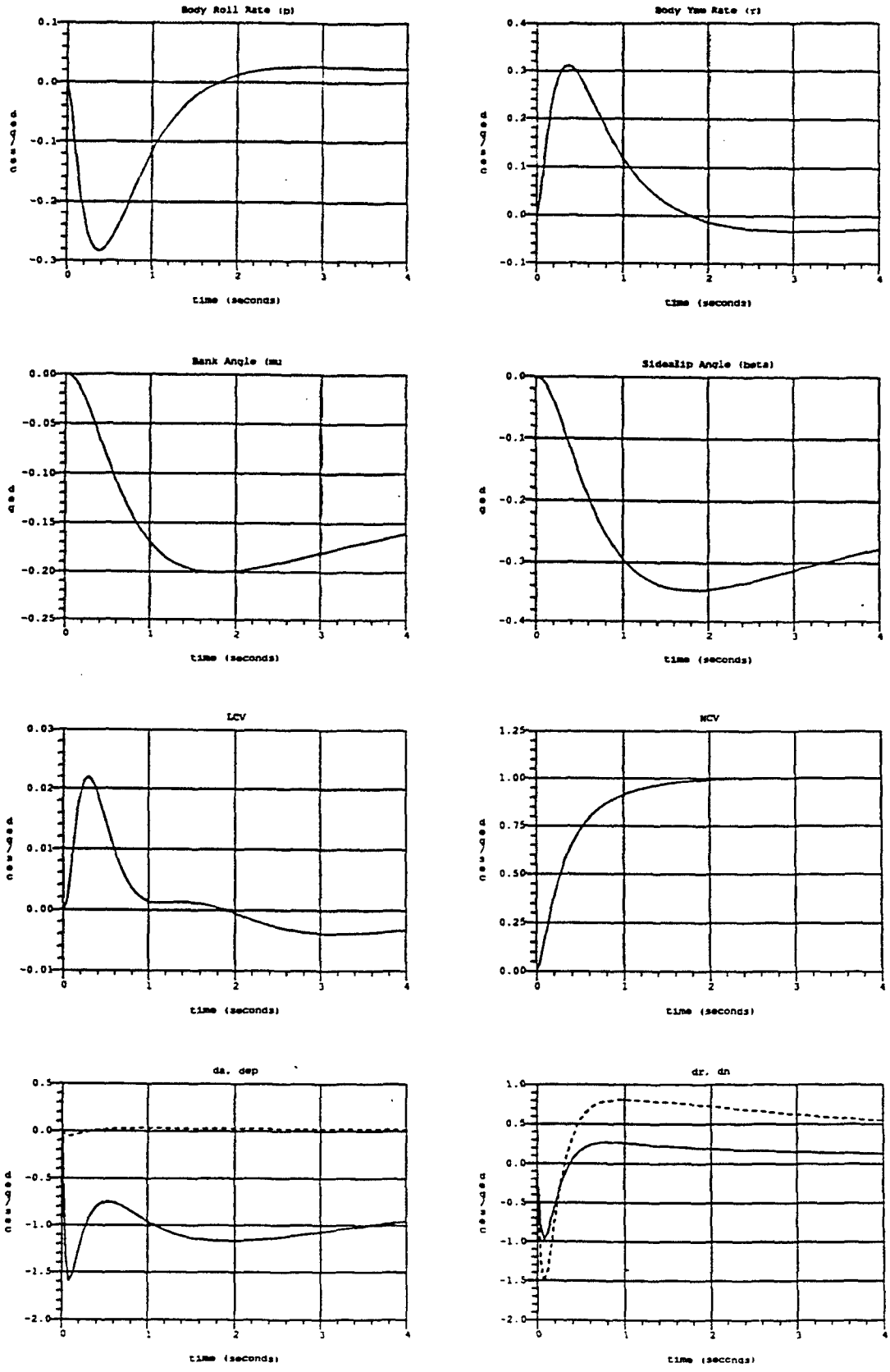


Figure 9.10: Closed-Loop Response to Directional (NCV) Step Input.

Section 10

Dynamic Inversion Control

In this section, we present a dynamic inversion control law. Figure 10.1 illustrates the general control law structure. The *Precompensation* block takes pilot inputs and creates commands, LCV^{cmd}, MCV^{cmd}, NCV^{cmd}, for the feedback loop. The three controlled variables signals, LCV, MCV, NCV, represent the variables we wish to control in the roll, pitch, and yaw axes, respectively. They are a blend of various sensed quantities and are formed in the *Controlled Variable Definition* block. In the *Signal Shaping* block, we specify the desired characteristics of the controlled variables. The outputs of this shaping block are desired values for the derivatives of the controlled variables, $\dot{LCV}^{des}$, $\dot{MCV}^{des}$, $\dot{NCV}^{des}$. These desired signals are fed to the *Dynamic Inversion/Effector Allocation* block, whose purpose is to generate actuator commands that achieve the desired aircraft responses. The net effect of the combination of dynamic inversion and aircraft dynamics is that the transfer functions from the desired responses ($\dot{LCV}^{des}$, $\dot{MCV}^{des}$, $\dot{NCV}^{des}$) to the controlled variables (LCV, MCV, NCV) reduce to three pure integrations, as Figure 10.2 illustrates. That is, the actual controlled variable rate equals the desired rate within bandwidth and control-power constraints and airframe modeling accuracies.

A general step-by-step approach to implement a dynamic inversion control law of the form described above is:

1. **Choose Controlled Variables:** In this first step, we choose a variable for each pilot control input. Conventionally, this ties lateral stick, longitudinal stick, and rudder pedals to a variable or blend of variables in the roll, pitch, and yaw axes, respectively. Two issues are particularly important at this stage: First, controlled variables must reflect the pilot's control desires in each axis to meet handling qualities objectives. Second, open-loop responses to controlled variables must have no right-half-plane zeros. Because open-loop zeros become closed-loop poles with dynamic inversion control laws [Morton 1993], we must select controlled variables so as to avoid unstable open-loop zeros, as was discussed at length in Section 5.

2. **Develop Model:** In this step, we define a practical aircraft model. Equations of motions are straightforward and were presented in Section 4.1.1.3. Because data tables associated with the high-fidelity aerodynamic models are often large and cumbersome, we require simplified models (including thrust characteristics) that can run in real time.

3. **Choose Desired Dynamics:** In this step, we specify desired characteristics for all controlled variables. In our application of the technique, the dynamic inversion part

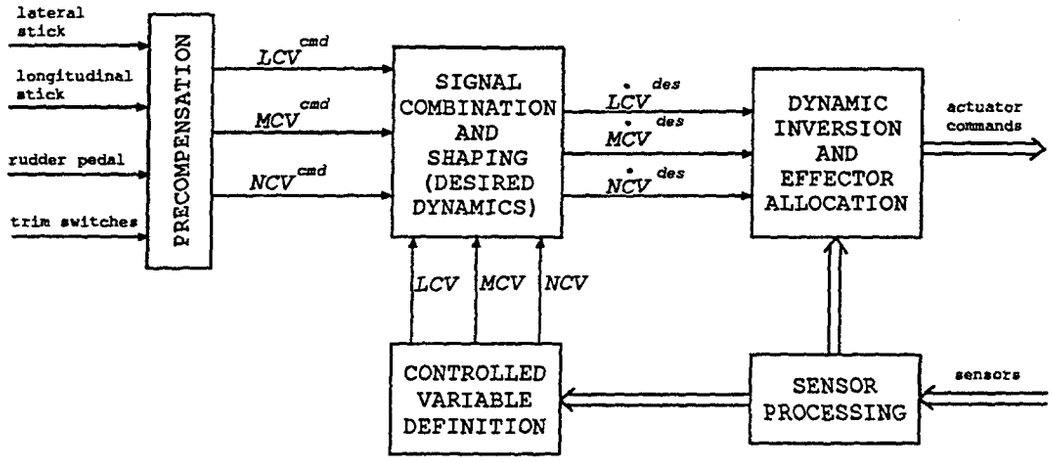


Figure 10.1: General Dynamic Inversion Control Law Structure.

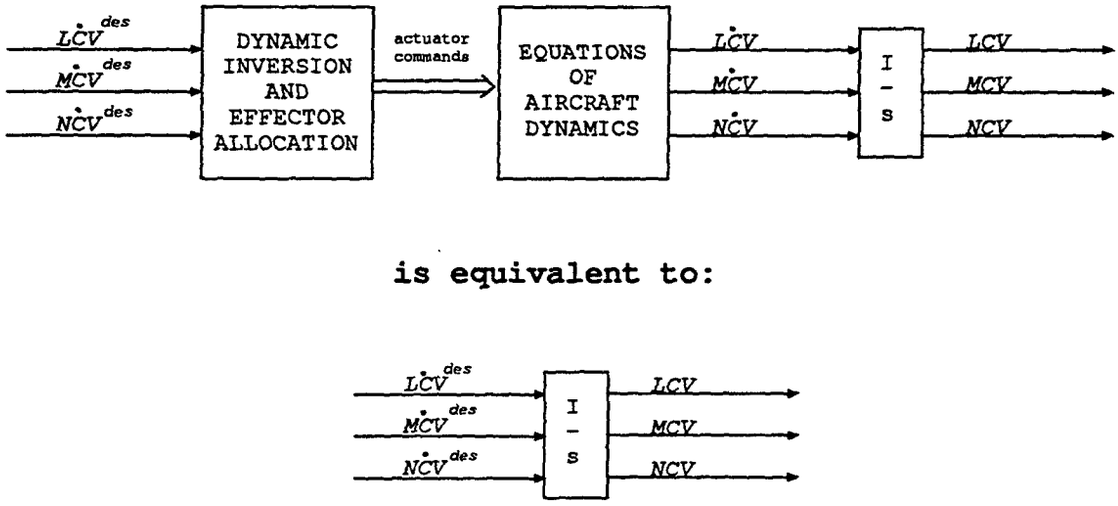


Figure 10.2: Basic Principle of Dynamic Inversion Control.

of the control law, together with the open-loop aircraft dynamics, ideally produces integrations in all axes (see Figure 10.2). Thus, given our discussion of feedback fundamentals in Section 2, the choice of desired dynamics is reasonably simple.

4. **Dynamic Inversion:** Here we model forces and moments corresponding to current aerodynamic, gravitational, and inertial contributions as functions of measured state variables. We then command forces and moments to create desired net forces and moments, taking into account the current forces and moments.

5. **Precompensation:** In this step, we tune the handling qualities of the specific aircraft. For example, we define stick and rudder pedal gains and shapes. This is not a multivariable task *per se* within the dynamic inversion framework. Through an F-18 HARV example, we show how some performance requirements relate to precompensator parameters.

In the remainder of this section, we discuss critical aspects associated with most of these steps.

10.1 Models

The primary modeling task for the dynamic inversion control law is the computation of time derivatives for the three controlled variables: $\mathbf{V} = (LCV, MCV, NCV)$.

As Section 10.3 will show, the dynamic inversion control approach relies on a control effectiveness matrix, which is given by the partial derivatives (with respect to control effectors) of the time derivatives for the three controlled variables. Because closed-form solutions for this control effectiveness matrix are seldom readily available, we often compute them numerically: That is, compute the change in $\dot{\mathbf{V}}$ when each control effector is perturbed from its nominal value. For the remainder of this subsection, we restrict ourselves to the computation of $\dot{\mathbf{V}}$.

Because controlled variables $\mathbf{V}$ are expressed in terms of state variables x, we can express $\dot{\mathbf{V}}$ as

$$\dot{\mathbf{V}} = \frac{\partial \mathbf{V}(x)}{\partial x} \frac{\partial x}{\partial t} \tag{10.1}$$

We derive $\partial \mathbf{V}(x)/\partial x$ analytically and code it in software, whereas the quantity $\partial x/\partial t$ is given by the aircraft six-degree-of-freedom equations of motion presented in Section 4.1.1.3 (the flight path state variables set). (It is also advantageous to have the equations of motion internal to the control law because certain limiters rely on models for $\dot{p}$, $\dot{q}$, $\dot{r}$, or $\dot{\alpha}$.)

We compute aerodynamic forces and moments that drive the equations of motion by interpolation over the full aerodynamic database. Similarly, we compute thrust forces and moments from special engine and thrust models supplied by the engine contractors.

To make real-time control-law implementation practical, we must have simplified models. One method of generating such simplified models is with the following least-squares technique.

In its most general form, we can express the complete aerodynamic database such that aero coefficients are functions of several variables:

$$C_k = f_k(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r, M, h, \dot{\alpha}) \tag{10.2}$$

where $k = D$ (drag), Y (side force), L (lift), ℓ (roll), m (pitch), and n (yaw). For many applications, we can neglect $\dot{\alpha}$ effects. For strictly low-subsonic applications, Mach effects may also be small. In addition, altitude effects are often negligible. For the purpose of illustrating the least-squares technique, let us assume that $\dot{\alpha} = 0$, altitude effects are small, and Mach effects are significant. Then, an approximation for the aero coefficients is

$$C_k = f'_k(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r, M) \tag{10.3}$$

Now, we specify a particular form for the desired model by assuming that it is nonlinear in α and M and linear in the other variables:

$$\begin{aligned} C_k &= C_{k_o}(\alpha, M) + C_{k_\beta}(\alpha, M)\beta + C_{k_p}(\alpha, M)\frac{b}{2V}p + C_{k_q}(\alpha, M)\frac{\bar{c}}{2V}q \\ &+ C_{k_r}(\alpha, M)\frac{b}{2V}r + C_{k_{\delta_a}}(\alpha, M)\delta_a + C_{k_{\delta_e}}(\alpha, M)\delta_e + C_{k_{\delta_r}}(\alpha, M)\delta_r \end{aligned} \tag{10.4}$$

which has the vector form

$$C_k = \begin{bmatrix} C_{k_o}(\alpha, M) \\ C_{k_\beta}(\alpha, M) \\ C_{k_p}(\alpha, M) \\ C_{k_q}(\alpha, M) \\ C_{k_r}(\alpha, M) \\ C_{k_{\delta_a}}(\alpha, M) \\ C_{k_{\delta_e}}(\alpha, M) \\ C_{k_{\delta_r}}(\alpha, M) \end{bmatrix}^T \begin{bmatrix} 1 \\ \beta \\ \frac{b}{2V}p \\ \frac{\bar{c}}{2V}q \\ \frac{b}{2V}r \\ \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \tag{10.5}$$

or, more simply

$$C_k = \mathbf{C}_k(\alpha, M)\eta \tag{10.6}$$

where η is the vector of independent variables.

Now, at every (α_i, M_i) pair required to cover the flight envelope, we choose to minimize the following squared-error function over all N combinations of $\beta, p, q, r, \delta_a, \delta_e, \delta_r$:

$$J_k(\alpha_i, M_i) = \frac{1}{2}\sum^N [C_k - \mathbf{C}_k(\alpha_i, M_i)\eta]^2 \tag{10.7}$$

Minimizing cost function J_k leads to the solution

$$\mathbf{C}_k(\alpha_i, M_i) = \left[\sum^N \eta\eta^T\right]^{-1} \left[\sum^N C_k\eta^T\right] \tag{10.8}$$

which, as the subscripts imply, must be solved once for each of the six linear variables (corresponding to $k = D,\ Y,\ L,\ \ell,\ m,\ n$) and once for each nonlinear (α_i, M_i) pair desired. We interpolate over the array of $\mathbf{C}_k(\alpha_i, M_i)$ values and use the resulting values from equation 10.4 to get the total coefficients. The resulting array of $\mathbf{C}_k(\alpha_i, M_i)$ should be small enough for onboard use and storage.

A further simplification that we commonly use is to assume that the three longitudinal coefficients are functions only of α, Mach, and the other longitudinal independent variables, whereas the three lateral-directional coefficients are functions only of α, Mach, and the lateral-directional independent variables. That is, we assume a least-squares models for aero coefficients of the form:

$$\begin{aligned} C_{k1} &= f'_{k1}(\alpha, M, q, \delta_e) \\ C_{k2} &= f'_{k2}(\alpha, M, \beta, p, r, \delta_a, \delta_r) \end{aligned} \tag{10.9}$$

where $k1 = D$ (drag), L (lift), and m (pitch) and $k2 = Y$ (side force), ℓ (roll), and n (yaw).

In any case, we must check the resulting least-squares model for validity against the full-database model throughout the flight envelope. If linear fits do not accurately model coefficient data, we must modify the model form. For example, in modeling engine drag terms for the F-18 HARV, we used a least-squares fit to a model of the bilinear form:

$$D_{eng} = [D_o\ D_h\ D_m\ D_{hh}\ D_{hm}\ D_{mm}] \begin{bmatrix} 1 \\ h \\ M \\ h^2 \\ hM \\ M^2 \end{bmatrix} \tag{10.10}$$

The flowchart of Figure 10.3 presents an outline of the modeling process for dynamic inversion control law development. Section 10.5 spells out data requirements.

10.2 Desired Dynamics

Two sets of variables feed into the *Signal Shaping* block of Figure 10.1: three controlled variable commands that are the outputs of the precompensation operations (to be discussed in Section 10.4), and three controlled variables that are the outputs of operations described in Section 5. This block defines the dynamic behavior we wish control variables to follow in tracking their commands.

Because we have chosen to use dynamic inversion in a manner such that the resulting transfer functions from the three desired CVs to the three CVs approximate integrators, control variables are decoupled as are the filters in the 3×3 desired dynamics

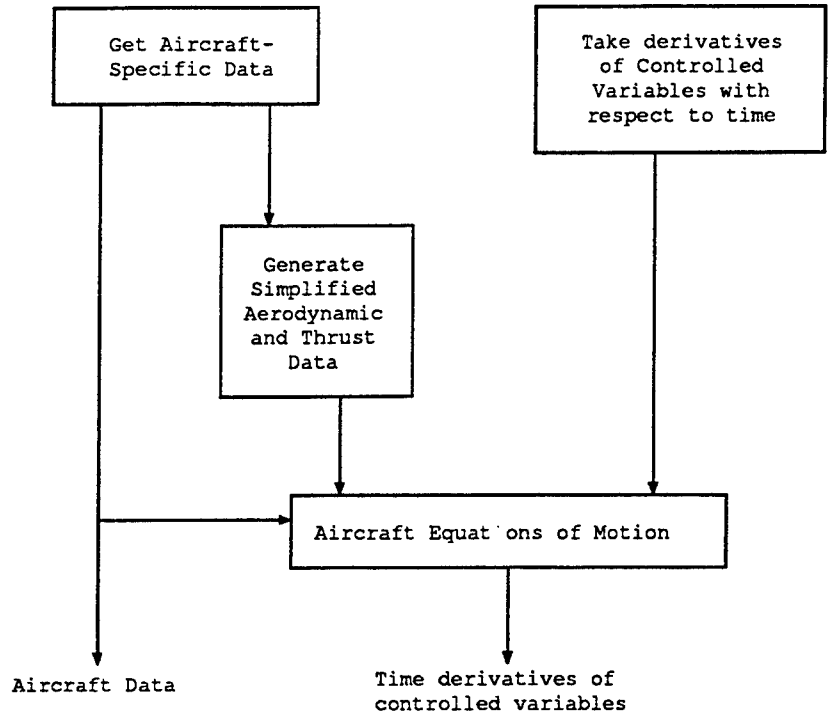


Figure 10.3: Modeling Steps for Dynamic Inversion Control.

block. Thus, the multivariable design is effectively reduced to three single-loop designs. Nevertheless, we must still follow general design principles for multivariable control and perform multivariable control analyses.

Figure 10.4 presents a general flow chart for developing the desired dynamics part of the control law. We suggest the use of a proportional-plus-integral (P+I) filter in each of the controlled variable paths. We will discuss the suggested P+I compensation that was used on the HARV program shortly, but first we we address some issues to consider during development of the Desired Dynamics block.

10.2.1 Bandwidth Reduction

Real-time simulation efforts have shown that there are advantages to reducing the bandwidth gain, K_B, in a particular axis when the control surfaces associated with that axis are limited (either rate or position). Logic to implement decay of K_B to some fraction of its nominal value, as well as first-order growth of K_B back to its nominal value after controls come off their limits, is one possible solution.

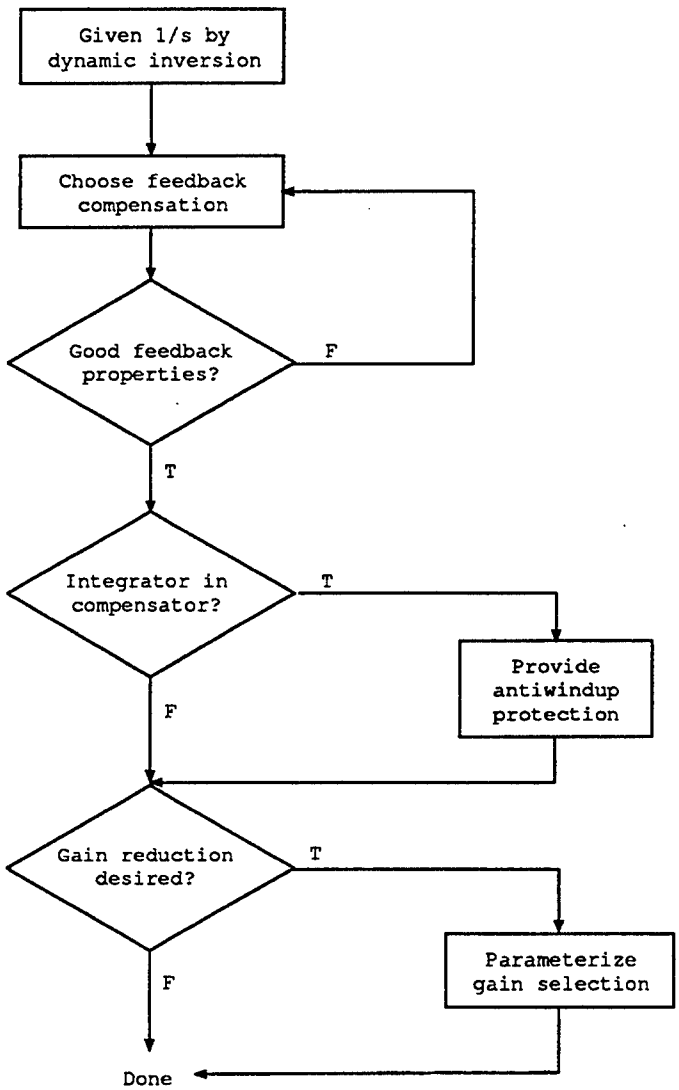


Figure 10.4: Desired Dynamics Development for Dynamic Inversion.

10.2.2 Other Choices

Other options beyond the P+I network are also possible. Note that P+I provides high low-frequency loop gain, good high-frequency attenuation, and a link to classical control ideas. As such, it has the advantage that it is easily tunable. For example, second-order responses are possible by simply changing gains or using a simple precompensation filter.

Using a simple gain for desired dynamics does not necessarily provide integral action in the presence of model errors. Thus, desensitization and disturbance rejection may be compromised.

Some investigators have experimented with design techniques such as H_∞ to provide additional robustness. But, because the feedback control is effectively around an integrator and is single-loop, the additional complexity seems unwarranted.

10.2.3 Suggested Dynamics

Currently, we suggest that desired dynamics be identical for each of the three axes and consist of a first-order pole at 2.5 rad/sec (or time constant of 0.4 sec).

We can achieve these desired dynamics using the architecture of Figure 8.1. Note that we use a proportional-plus-integral network with proportional command path gain of 0.5 to achieve a net first-order response. As the figure suggests, the resulting closed-loop dynamics are critically dependant on the dynamic inversion routine inverting aircraft equations of motion to produce an integration from desired controlled variable rate to the controlled variable. This integration, together with the integrator on the error signal, results in the second-order closed-loop transfer function:

$$\frac{CV}{CV^{cmd}} = \frac{\frac{1}{2}K_B s + \frac{1}{4}K_B^2}{s^2 + K_B s + \frac{1}{4}K_B^2} = \frac{1}{2}K_B \frac{s + \frac{1}{2}K_B}{\left(s + \frac{1}{2}K_B\right)^2} = \frac{\frac{1}{2}K_B}{s + \frac{1}{2}K_B} \tag{10.11}$$

Note that this transfer function has one real zero and two real poles at $-\frac{1}{2}K_B$. The resulting cancellation of one of the real poles by the zero leaves a first-order transfer function.

The loop transfer function for the desired dynamics, together with the pure integrator ideally resulting from dynamic inversion, is $\frac{1}{4}K_B^2/s^2$ at low frequencies and K_B/s above $\omega = K_B/4$ to provide adequate phase margin at the crossover frequency of $\omega = K_B$. Thus, we may interpret the gain K_B as the bandwidth (or crossover frequency) of the loop transfer function. The transfer function continues to rolloff at high frequencies ($\omega > K_B$). We choose $K_B = 5$ to give an open-loop crossover frequency (and approximate bandwidth) of 5 rad/sec. Based on our experience, this choice is a practically achievable value for a fighter-class aircraft flight control system in the presence of higher-order dynamics and associated filtering.

The resulting closed-loop command response pole at $\frac{1}{2}K_B = 2.5$ rad/sec is compatible with accepted handling qualities requirements. As an additional note, although the integrator on the error signal may appear unnecessary given the integration produced by the dynamic inversion, the ability to achieve a "perfect" inversion (and hence a pure integration) is limited by modeling errors, actuator dynamics that are unaccounted for, delays, etc. Thus the integrator is necessary in the presence of these effects.

10.3 Dynamic Inversion

In this section, we describe the nonlinear control technique, referred to as dynamic inversion, that is used here to produce actuator commands from desired controlled variable commands (i.e, output of the *Desired Dynamics* block in Figure 10.1). The logic to incorporate multiple control effectors (e.g., thrust vectoring) within the dynamic inversion framework is a separate issue that we will discuss shortly. In this first subsection, we introduce the dynamic inversion method from a practical level.

10.3.1 Inversion

The purpose of dynamic inversion is to compute proper control inputs that achieve desired responses. The simple case we present next comes from linearized aircraft roll-axis dynamics:

$$\dot{p} = L_p p + L_{\delta_a}\delta_a \tag{10.12}$$

Rewriting (i.e., *inverting*) this dynamic equation for the control input yields

$$\delta_a = \frac{1}{L_{\delta_a}}\left[\dot{p} - L_p p\right]. \tag{10.13}$$

To obtain a control law, we replace δ_a in equation 10.13 with its command δ_a^{cmd}, $\dot{p}$ with its desired dynamics $\dot{p}^{des}$, and p with measured roll rate p^{meas}. Now, assuming that we have a perfect model for L_p and L_{δ_a}, $p^{meas} = p$, and $\delta_a = \delta_a^{cmd}$, substituting equation 10.13 into equation 10.12 gives the result that $\dot{p} = \dot{p}^{des}$. This says that under the given assumptions roll acceleration achieved (by the aircraft) is identical to that specified. In practice, these assumptions never hold. First, $\delta_a \neq \delta_a^{cmd}$ due to actuator dynamics and $p^{meas} \neq p$ due to sensor dynamics and/or errors. In addition, deflection and rate limits on δ_a prevent us from achieving perfect inversion. Moreover, equation 10.13 assumes that a model of the aircraft is available to get L_p and L_{δ_a}. In the presence of inevitable model errors, dynamic inversion is not exact. A final assumption is that L_{δ_a} not be too small because, as L_{δ_a} approaches zero, the commanded control surface becomes unbounded. This is a real concern because for low $\bar{q}$, ailerons become ineffective and L_{δ_a} becomes small, which motivates using an alternative control scheme.

A slightly more complicated example comes from the lateral-directional axes of an aircraft (also linearized), for which the dynamics are (neglecting the μ and β states):

$$\begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L_p & L_r \\ N_p & N_r \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} + \begin{bmatrix} L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{10.14}$$

Here the dynamic inversion control is

$$\begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}^{cmd} = \begin{bmatrix} L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix}^{des} - \begin{bmatrix} L_p & L_r \\ N_p & N_r \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix}^{meas} \right\} \tag{10.15}$$

This is analogous to the earlier roll example but is in matrix form. All of the assumptions above are still relevant, and the analog of $1/L_{\delta_a}$ being finite is that the matrix inverse exist (i.e., the matrix is nonsingular).

For general aircraft flight control, we apply these same principles in a slightly more general form. Instead of specifying the desired dynamics for the derivative of a state variable (like p, q, or r), we specify the dynamics for the derivative of the controlled variables $\mathbf{V}$. The nonlinear equations of motion from Section 4.1.1.3 here assume the form:

$$\dot{x} = F(u, x) \tag{10.16}$$

$$\approx f(x) + g(x)u \tag{10.17}$$

where x is the state vector and u is the control vector. Note that our controlled variables are a function of the state variables:

$$\mathbf{V} = \mathbf{V}(x) \tag{10.18}$$

where $\mathbf{V} = [LCV, MCV, NCV]^T$ is the column vector of controlled variables, as described in Section 5. Then

$$\dot{\mathbf{V}} = \frac{\partial \mathbf{V}}{\partial x}\dot{x} = \frac{\partial \mathbf{V}}{\partial x} f(x) + \frac{\partial \mathbf{V}}{\partial x} g(x)u \tag{10.19}$$

or

$$\dot{\mathbf{V}} = a(x) + b(x)u \tag{10.20}$$

with $a(x) = \frac{\partial \mathbf{V}}{\partial x} f(x)$ and $b(x) = \frac{\partial \mathbf{V}}{\partial x} g(x)$. Thus, under the same assumptions that were detailed earlier, the dynamic inversion part of the control law is

$$\begin{aligned} u^{\text{cmd}} &= \left[\tfrac{\partial \mathbf{V}}{\partial x} g(x)\right]^{-1} \left[\dot{\mathbf{V}}^{\text{des}} - \tfrac{\partial \mathbf{V}}{\partial x} f(x)\right] \\ &\cong [b(x)]^{-1}[\dot{\mathbf{V}}^{\text{des}} - a(x)] \end{aligned} \tag{10.21}$$

The important generalization to notice is that the number of independent control effectors must equal the number of controlled variables: in this case, three. (i.e., the row dimension of $\partial \mathbf{V}/\partial x$ must equal the column dimension of $g(x)$ for the matrix $b(x)$ to be square.) Because our three controlled variables have a strong angular rate content, and

conventional aircraft typically have three moment-producing controls (ailerons, symmetric horizontal tail, and rudder), this requirement is satisfied for conventional aircraft.

In reality of course, not all aircraft have conventional controls. Many aircraft, such as F-18 HARV, F-16/MATV, and YF-22, have multiple control effectors for a single axis (e.g., symmetric elevator and pitch vectoring for the longitudinal axis). This corresponds to the control effectiveness matrix having more columns than rows. In such cases, we require a method to allocate control commands to the different effectors. Section 6 describes several methods for handling cases like these. Similarly, it may happen, due to control effector limiting or ineffectiveness, that the control effectiveness matrix *effectively* has more rows than columns. The following section provides a solution to problems of this type.

10.3.2 Controlled-Variable Priority

This section addresses the issue of the control effectiveness matrix having more rows than columns. Practically, this means that there are more performance desires than there are controls to accomplish them. For example, the lateral-directional axes of a maneuvering conventional aircraft might have: (1) ailerons that are on their rate limits due to pilot commands, (2) a rudder on its rate limit trying to maintain coordinated flight, and (3) available differential horizontal tail that has effectiveness for both roll and yaw control. If the pilot has two desires (i.e., roll and maintain coordinated flight) and only one available control effector (i.e., differential tail), a *decision* must be made regarding which axis to control with the available control effector.

For aircraft where the longitudinal and lateral-directional axes decouple (symmetric aircraft, conventional controls), this is only an issue for the lateral-directional axes. For asymmetric aircraft (oblique wing) and aircraft without conventional controls (tailless aircraft), we must consider pitch- as well as lateral-directional-axes desires. (Generally, though, the pitch axis is given priority.)

One way to address problems of this sort is to solve the following least-squares problem:

$$\min_u J_r = \min_u r^T W_r^T W_r r \tag{10.22}$$

where

$$r = b(x)u - [\dot{\mathbf{V}}^{\text{des}} - a(x)] \tag{10.23}$$

is an error vector that measures closeness to achieving of our desires. The weighting matrix W_r is a diagonal matrix that weights control variables according to a designer-imposed priority. For the lateral-directional axes of the HARV, we weighted the yaw controlled variable five times more heavily than the roll controlled variable. This was done to place priority on turn-coordination performance in lieu of roll performance.

One way to *solve* the above problem is to characterize the line segments that define the edges of the achievable control effector space. We do this by introducing a parameter

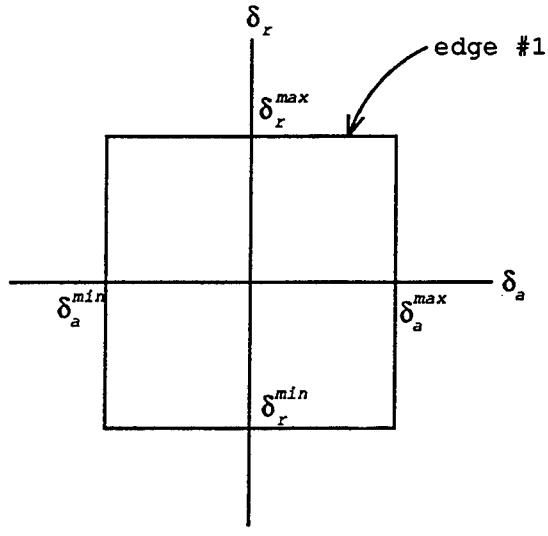


Figure 10.5: Aileron/Rudder Control Space.

t that allows interpolation between adjacent corners of the control space. For example, consider the aileron (δ_a) and rudder (δ_r) control effector space shown in Figure 10.5. Then, edge #1 can be parameterized as follows:

$$\begin{bmatrix} \delta_a(t) \\ \delta_r(t) \end{bmatrix} = \nu_0 + t(\nu_1 - \nu_0) \tag{10.24}$$

where

$$\nu_0 = \begin{bmatrix} \delta_a^{max} \\ \delta_r^{max} \end{bmatrix} \quad \text{and} \quad \nu_1 = \begin{bmatrix} \delta_a^{min} \\ \delta_r^{max} \end{bmatrix} \tag{10.25}$$

Note that these boundaries may be due to physical limits or first-order integration of the rate limit.

Now, we substitute equation 10.24 for u in equation 10.22 and solve for the t that minimizes J_r to get

$$t = \frac{\left([\dot{\mathbf{V}}^{\text{des}} - a(x)] - b(x)\nu_0\right)^T W_r^T W_r b(x)(\nu_1 - \nu_0)}{(\nu_1 - \nu_0)^T [b(x)]^T W_r^T W_r b(x)(\nu_1 - \nu_0)} \tag{10.26}$$

If $t \leq 0$, then $u = \nu_0$ is the solution for edge #1; if $t \geq 1$, then $u = \nu_1$; if $0 < t < 1$, then $u = \nu_0 + t(\Upsilon\nu_1 - \nu_0)$.

These same computations must be carried out for each edge, and the u with the smallest J_r becomes the u^{cmd}. For cases where the number of controls is two, there are four edges that must be checked. In general, for m controls, $2m!$ edges must be checked. Clearly, for $m > 4$, the number of computations starts to become impractical.

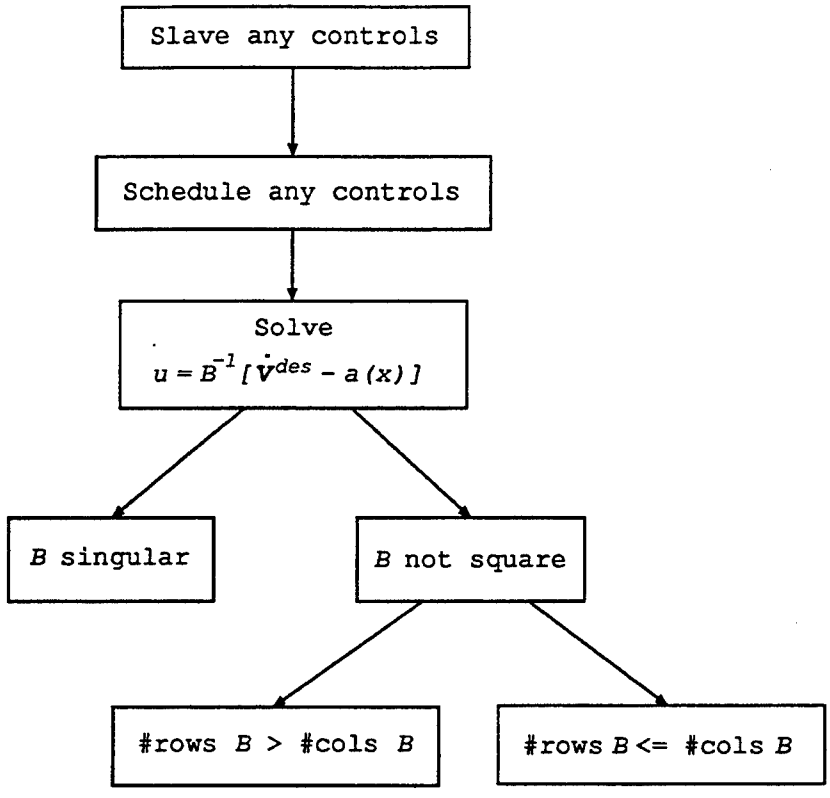


Figure 10.6: Flow Chart for the Inverse.

10.3.3 Summary

This section presented the basics for the *inversion* part of the dynamic inversion methodology. The process can be described at a high level, as the flow chart in Figure 10.6 shows. The figure indicates that we may slave some surfaces together (e.g., left and right ailerons) and schedule others (e.g., leading edge flaps). Next we solve the inverse equations of motion, during which time certain decisions must be made about the matrix to be inverted.

10.4 Precompensation

Precompensation design for dynamic inversion control, is not, strictly speaking, a multivariable task. As such, its design steps do not add to the multivariable control design knowledge that is the intent of these guidelines. For this reason, a detailed description is neither warranted nor presented here. An overview of the precompensation is, however, presented since many of the military specifications are handling-qualities related, and this part of the control law requires translation of these design requirements into design parameters. This being the case, the precompensation overview we present is done using

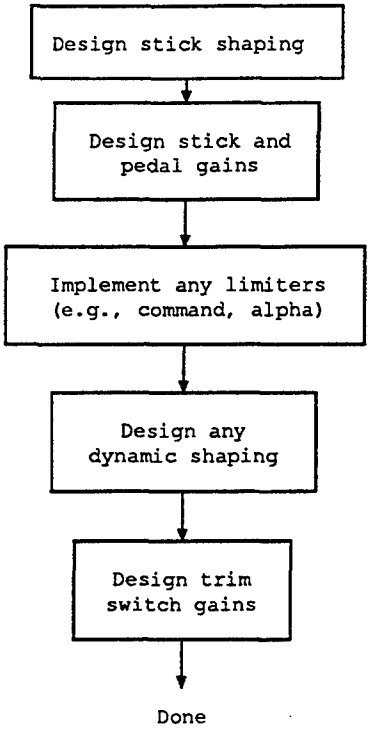


Figure 10.7: Precompensation Design Steps.

the HARV design example. Attention will be given to the general structure and principles for gain selection rather than well-defined design steps. A summary of the steps is given in flow chart form in Figure 10.7.

10.4.1 HARV Roll-Axis Precompensation

The block diagram for the roll-axis precompensation and shaping appears in Figure 10.8. The first block we address is the stick shaping block. This function smoothes the characteristics of the stick around zero deflection. It is particularly important in fine tracking of targets for fighter applications. We use a function that is parabolic around zero deflection and smoothly transitions to linear stick feel at larger deflections.

The command path for the lateral stick next shows two paths: one for very low $\bar{q}$ conditions ($F_{LAT} = 0$), the other for moderate- to high-$\bar{q}$ conditions ($F_{LAT} = 1$). In both cases, stick gain ($K_{roll-lo\bar{q}}$ and $K_{roll-hi\bar{q}}$) reflects the maximum controlled variable command (p for $K_{roll-lo\bar{q}}$, p_s for $K_{roll-hi\bar{q}}$) desired for the aircraft. In the case of the HARV, for normal-$\bar{q}$ ranges (high-$\bar{q}$ path), stick gain is sized:

1. So as not to exceed the roll-rate gyro measurement,

2. To be consistent with the roll-rate capabilities of the aircraft, and

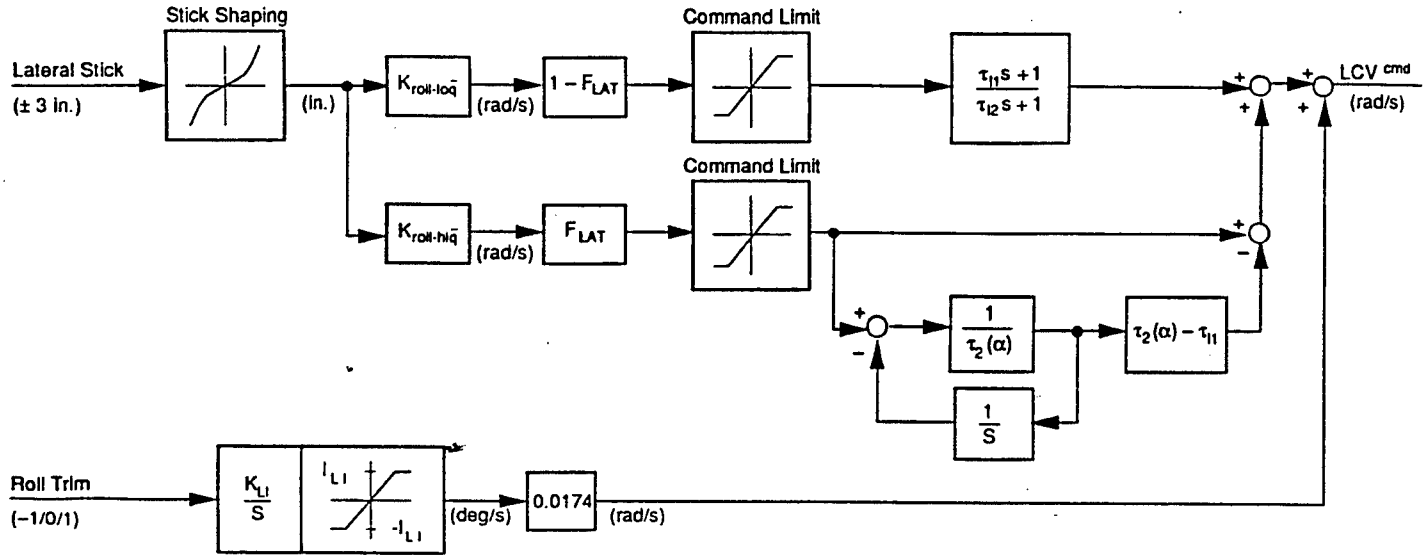


Figure 10.8: HARV Roll-Axis Precompensation.

3. To adhere to the military specification corresponding to stick force.

For example, because full stick deflection is 3 inches and the maximum sensed roll rate deg is 300 deg/sec, $K_{roll-hi\bar{q}}$ for $\alpha < 30$ is set to be 70 (deg/sec)/inch. If the maximum stick force (per military specification) of 20 lb at 3 in stick deflection is in force, then stick sensitivity is

$$70\frac{\text{deg/sec}}{\text{inch}} \times \frac{3 \text{ inch}}{20 \text{ lb}} = 10.5\frac{\text{deg/sec}}{\text{lb}} \tag{10.27}$$

This is within the military specification maximum of 15 (deg/sec)/lb. For very low high-$\bar{q}$ ranges, $K_{roll-lo\bar{q}}$ is 33 (deg/sec)/inch.

We limit the command in both high- and low-$\bar{q}$ paths, and choose the roll-rate limit to be the smaller of the two constraints. The first constraint limits the amount of roll rate commanded without causing uncontrolled motion in the pitch axis. In this way, we avoid uncontrolled inertial coupling of roll motion into the pitch axis. The second command limit is similar in that it prevents inertial coupling of roll rate into the directional (yaw) axis.

We include a first-order lead-lag filter in each of the high- and low-$\bar{q}$ paths to allow changing the first-order dynamics specified by the desired dynamics P+I filter. We set the low-$\bar{q}$ filter to unity and the pole of the high-$\bar{q}$ filter (shown as a function of α) to 10 rad/sec (for all α). The zero cancels the 2.5 rad/sec pole of the feedback loop.

We set roll trim limits to 5% of full roll-rate authority and roll-rate integrator gain to reach the limit in 10 seconds.

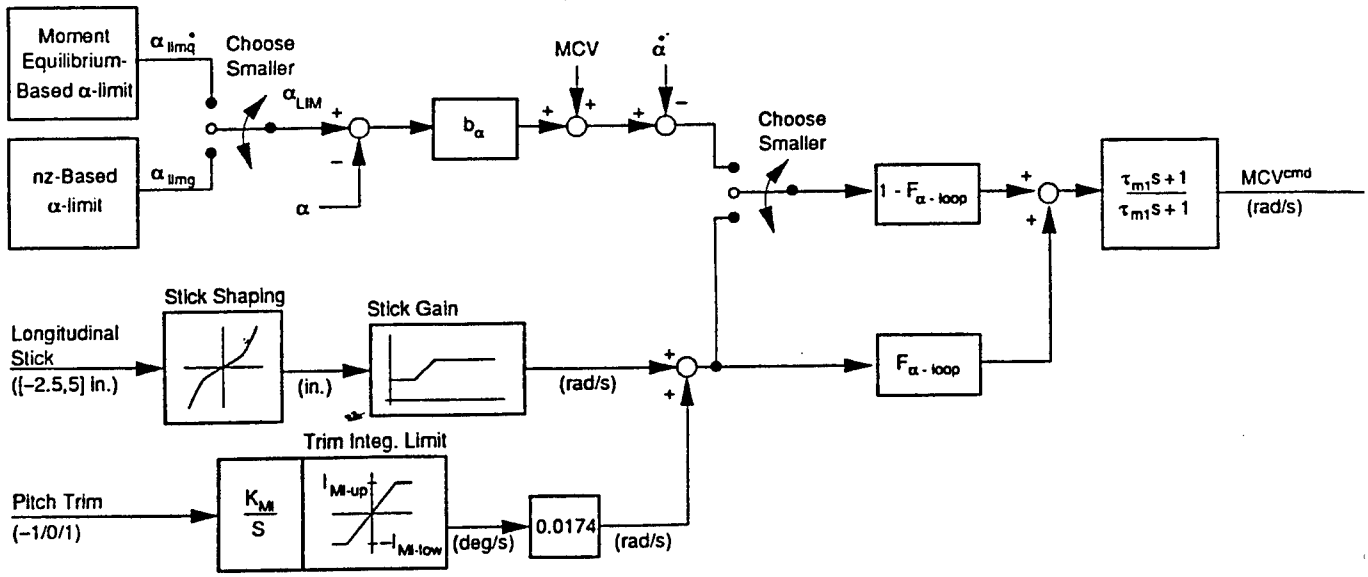


Figure 10.9: HARV Pitch-Axis Precompensation.

10.4.2 HARV Pitch-Axis Precompensation

Figure 10.9 gives the block diagram for pitch-axis precompensation and shaping.

The first block we address is the stick shaping block, which is much like roll-axis shaping.

Stick gain is a function of $\bar{q}$. For very low $\bar{q}$, we choose gain to produce a range of 60 deg/sec nose-up pitch rate to 15 deg/sec nose-down pitch rate. For higher $\bar{q}$ (more normal flight), we choose gain to:

1. Meet pitch-rate and/or α or nz requirements, and

2. Adhere to the military specification corresponding to stick force per g.

Recall that we can interpret the output of the stick gain block as an MCV command; therefore we must translate the above requirements into MCV units.

We select pitch-trim limits to produce incremental changes of $+1$ g and -0.5 g. We translate these to MCV units using the definition of MCV and $K_\alpha \alpha \approx (\Delta nz \times g)/V_{co}$ with Δnz in g's. We choose the speed of the trim integrator so as to reach the positive limit in 10 seconds at some intermediate speed.

The top portion of Figure 10.9 implements an angle-of-attack limiter. For the HARV, there are two α-based limits: a load-factor limit and a moment-equilibrium limit. We translate both to an α limit that is maintained using an outer α feedback loop. We

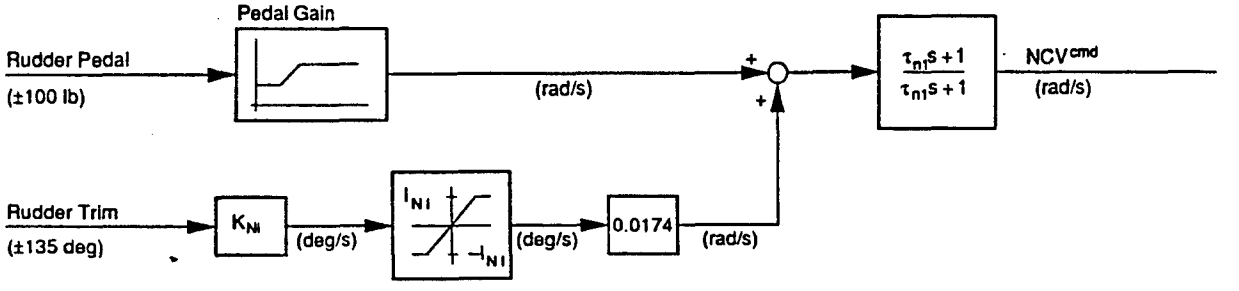


Figure 10.10: HARV Yaw-Axis Precompensation.

implement the load-factor limiter by exploiting the approximation:

$$nz \approx \frac{1}{g}\frac{\bar{q}SC_L(\alpha)}{m} \tag{10.28}$$

Then, if a simple model of $C_L(\alpha)$ is stored, we can find an α that corresponds to the nz limit. The moment-equilibrium limit does not allow angle of attack to rise above a value where it cannot achieve pitch moment trim for the current aircraft state. Like the nz limit, it exploits the equation:

$$\dot{q} \approx \frac{1}{I_y}(\bar{q}S\bar{c}C_m(\alpha) - I_{xz}p^2 + I_zpr - I_xrp + I_{xz}r^2 + tvc) \tag{10.29}$$

Here if a simple model of $C_m(\alpha)$ (with surfaces set for maximum nose-up moment) is stored, we can find an α that corresponds to the limit. We can also implement a "hard" α limit, if necessary. We accomplish α control using an approximate dynamic inversion, which forms an MCV command. The $F_{\alpha-loop}$ function determines whether to feed commands from either the limits or the stick to the feedback loop.

Finally, we implement a first-order lead-lag filter on commanded MCV. Like the roll axis, this allows changing the first-order dynamics specified by the desired dynamics P+I filter.

10.4.3 HARV Yaw-Axis Precompensation

The block diagram for the yaw precompensation and shaping appears in Figure 10.10.

The first block we address is pedal gain, which is a function of $\bar{q}$. For very low-$\bar{q}$, we choose gain to produce 45 deg/sec of yaw rate for full-pedal deflection. For higher $\bar{q}$ (more normal flight), we choose pedal gain to produce an approximately constant lateral-acceleration response for full-pedal deflection at all velocities. We translate lateral-acceleration requirement into an appropriate value of NCV.

The HARV has a thumb wheel for yaw-axis trim. We select trim gain to produce 12 deg/sec of yaw rate at low $\bar{q}$ or 0.2 g of lateral acceleration at higher $\bar{q}$. A limit is added for protection.

Finally, we implement a first-order lead-lag filter on commanded NCV. Like the roll and pitch axes, this allows changing the first-order dynamics specified by the desired dynamics P+I filter.

10.5 Summary

In summary, basic steps for the dynamic inversion control design methodology are:

1. **Controlled variables selection.** Section 5 discussed the selection and rationale for choosing controlled variables. These should suffice for most fighter aircraft applications.

2. **Model development.** Developing a model for a particular aircraft that is of sufficient fidelity and suitable for real-time application represents the bulk of the work in this 5-step process. Section 10.1 presented one method for obtaining a simplified model.

3. **Desired dynamics.** Section 10.2 describes a selection for the desired dynamics. These first-order dynamics should suffice for many applications.

4. **Dynamic inversion/effector priority.** A decision must be made regarding how to use control effectors and how to integrate this decision with the dynamic inversion framework.

5. **Precompensation:** We design precompensation last and often tune it using piloted simulation. It is not a part of feedback design *per se*. We verify command following performance using simulation by commanding controlled variables.

Once these pieces of the control law are completed, we connect them together as shown in Figure 10.1. Table 10.1 summarizes the primary data items that must be gathered for each of the above steps.

10.6 Piloted Simulation Process

High-fidelity, real-time piloted simulation is employed to evaluate and refine control laws using pilots' feedback. These simulations and pilot comments provide insights into the strengths and deficiencies of the control laws that are difficult or impossible to obtain in

Table 10.1: Data Requirements for Dynamic Inversion Control Law.

Description	Use
Mass	MCV^{cmd}, NCV , Models
x-axis inertia	LCV^{cmd}, MCV^{cmd}, Models
y-axis inertia	LCV^{cmd}, MCV^{cmd}, Models
z-axis inertia	LCV^{cmd}, MCV^{cmd}, Models
xz product of inertia	LCV^{cmd}, MCV^{cmd}, Models
Wing reference area	NCV, MCV^{cmd}, Models
Mean aerodynamic chord	MCV^{cmd} , Models
Wing span	Models
x-axis dist. from cg to thrust	MCV^{cmd}, Models
y-axis dist. from thrust to x-axis	Models
Dist. cg is forward of aero ref. point	Models
Dist. cg is below aero ref. point	Models
Nominal bandwidths for LCV, MCV, NCV loops (K_B)	Desired Dynamics
Load factor limit	MCV^{cmd}
$(C_{L_\alpha})_{ref} = C_{L_\alpha}$ at $\alpha \approx 0$	MCV, Models
$(C_{Y_\beta})_{ref} = C_{Y_\beta}$ at $\alpha \approx 0, \beta \approx 0$	NCV
Control effector limits	Dynamic Inversion, LCV^{cmd}
Control effector rate limits	Dynamic Inversion
C_m approximation for moment Equilibrium limit	MCV^{cmd}
C_L approximation for g limit	MCV^{cmd}
C_D	Models
C_Y	Models
C_L	Models
C_ℓ	Models
C_m	Models
C_n	Models
Minimum size of $b(x)$ for aerodynamic controls	Dynamic Inversion
Control effector priority choices: • Number of groups of effectors • Number of effectors in each group • Priority of each group	Dynamic Inversion
Control variable choice (SW_{cv})	MCV, NCV, MCV^{cmd}, Models

offline, non-real-time simulations. Specific handling qualities issues that can be better understood through the use of piloted simulation include: command sensitivities, cross-axis coupling, pilot induced oscillation tendencies, and performance at the "corners" of the envelope (command limiting and integrator windup protection logic). The pilot-vehicle interface (e.g., visual scenes, displays, and pilot inceptor location and characteristics) employed for the piloted simulation can significantly influence the results and must be considered in post-simulation analysis.

10.6.1 Procedures

A general piloted simulation process follows these steps:

1. Test Plan and Run Card Definition
2. Simulator Calibration and Checkout
3. Piloted Evaluation And Ratings
4. Data Reduction And Analysis

10.6.1.1 Test Plan and Run Card Definition

The test plan defines the objectives for the piloted simulation tasks and specifies the particular tasks to be performed to meet these objectives. Specific maneuvers are selected to challenge the control system and aircraft and provide useful feedback to improve the control system. Standard Evaluation Maneuvers (STEMS) are available in the literature (e.g., [Wilson 1993a, Wilson 1993b]). The specific details describing the maneuver; required performance levels, level of aggressiveness, and desired comments are provided to the pilot on a set of run cards. These cards are taken into the simulator with the pilot and are used to ensure that each run is performed in the same manner with the same evaluation criteria.

10.6.1.2 Simulator Calibration and Checkout

Simulator calibration and checkout is performed to ensure that the pilots evaluate the actual control law as defined by the designer. During this phase, the models are transferred to the real-time simulator, compiled, linked, and compared with offline simulation results to verify a proper match. After satisfactory matches are obtained, timing tests are performed to ensure that the models can run in real time at the desired update rates without frame overruns. Any concessions made to the models or in model update rates should be recorded for post-simulation analysis.

10.6.1.3 Piloted Evaluation And Ratings

During the piloted simulation, the pilot proceeds through the planned test run cards and comments on the handling qualities of the aircraft as specified on the cards. Some maneuvers require comments only, while others require Cooper-Harper Ratings (CHRs). A formal process for collecting CHRs is described in the litrature [Wilson 1989, Wilson 1990].

10.6.1.4 Data Reduction And Analysis

Following the piloted simulation, pilot comments, CHRs, and time history data are analyzed to define control law deficiencies and their level of criticality. Also, this data can often be used in conjunction with offline analyses, including the non-real-time simulation, to determine the cause and potential solutions to the control law deficiencies.

10.7 MCT F-16 Piloted Simulation Example

Piloted simulation for the MCT contract was performed to provide a challenging evaluation of a subset of the selected multivariable control law design methodologies, and thus provide for further maturation of the design guidelines. The F-16 Dynamic Inversion control laws and two primary pilot maneuvers were selected for the evaluation. In addition, other standard handling qualities evaluation maneuvers were chosen to ensure a more complete demonstration.

10.7.1 Maneuvers

Maneuver 1 is a flat-turn strafing maneuver. The aircraft is capable of performing flat turns (i.e., change the lateral flight path with little roll or sideslip). This capability requires the use of multiple yaw effectors to provide a direct lateral force on the aircraft, and has been shown to improve the effectiveness of strafing missions. The pedals are used to command the flat-turn response.

The targets for this maneuver are the columns of a bridge. The maneuver begins with the aircraft initialized at an altitude of 10,000 feet and at a 0 deg glideslope aimed several deg off the left end of the bridge. The pilot first acquires the left end of the bridge and then aggressively tracks each bridge column with a simulated 0.5 sec firing burst on each of as many of the columns as possible before passing the bridge. Flat turns are commanded to eliminate small target tracking errors (i.e., keep the pipper on the target) and break off targets and acquire others during strafing.

Maneuver 2 is high angle-of-attack bank captures. The aircraft is able to maneuver at high angle of attack (AOA), defined here as AOA near or greater than Clmax, by

employing pitch and yaw thrust vectoring. Specifically, large, rapid lateral nose pointing capability providing improved air combat effectiveness is evaluated by performing high AOA bank captures. Longitudinal and lateral sidestick inputs command pitch rate and stability-axis roll rate, respectively. Compensation for aerodynamic and inertial coupling between the axes is required for successful accomplishment of this maneuver.

With the aircraft stabilized at 110 KCAS, an altitude of 20,000 feet, and an AOA of 40 deg, the pilot captures, and holds for approximately 5 sec, bank angles (wind axes) of 45 deg, –45 deg, and 45 deg, passing through 0 deg bank on each segment. This maneuver is performed at maximum power and in an aggressive manner such that maximum roll rate is achieved.

10.7.2 MCT Pilot Simulation Test Plan

The primary goal of the MCT piloted simulation test plan was to evaluate the performance of the MCT dynamic inversion control laws for the design maneuvers listed above. The control laws were also evaluated using standard handling qualities maneuvers to more completely assess handling qualities.

The information below is taken from one of the MCT piloted simulation run cards. The card contains the information required for the pilot to perform and evaluate the maneuver.

10.7.2.1 Flat-Turn Strafing Maneuver

Definition: Start at the given initial conditions. Maneuver the aircraft to place the pipper on one end of the target bridge. Aggressively track the end of the bridge to simulate a 0.5 sec strafing burst. Use pedals to acquire and track the next column of the bridge and simulate another 0.5 sec strafing burst. Acquire and strafe as many columns of the bridge as possible before passing the bridge. Use pitch stick and throttle to maintain constant velocity and glideslope.

Conditions: Mach 0.7, 10,000 Feet

Performance: CHR required. See performance criteria.

Comments: Initial flat-turn nose pointing, Flat-turn target tracking, Yaw/sideslip, Coupled pitch motion, Coupled roll motion, Predictability of nose movement, Number of targets acquired, Time Delay, PIO Tendencies, Forces, Feel System

10.7.2.2 Flat-Turn Performance Criteria

Desired:

- Strafe four targets before passing bridge
- Light stick forces required to maintain 0 deg bank angle
- Light stick forces required to maintain pitch attitude and flight path
- No more than one overshoot to acquire target
- No PIO Tendencies

Adequate:

- Strafe three targets before passing bridge
- Bank angle remains within 5 deg
- Coupled pitch motion is controllable
- No more than two overshoots to acquire target
- PIO ceases when roll input relaxed

Controllability:

- Unexpected pitch or yaw motion cannot be stopped with full opposite input
- Divergent PIO
- Bounded PIO which diverges after input is relaxed

10.7.3 Simulator Calibration and Checkout

The initial simulation calibration and checkout went smoothly. The MCT aircraft models and dynamic inversion code were transferred to the real-time simulation facility. The code was then passed through automatic code translators to make the code compatible with the real-time compilers and executive. During this process, several coding changes were required to make the program function properly on the real-time computer. These changes were made in the offline simulation program and then transitioned to the real-time simulation to maintain configuration control.

Once the code was converted and compiled, a matrix of test cases was run to compare the results of the real-time versus the offline simulation. The cases were initially run in non-real-time mode on the real-time computer. After several minor modifications, the results matched within a close tolerance. Next, the test matrix was run in real-time mode, using canned inputs instead of hooking up to the cockpit. During this testing, we saw significant differences in the results of the two simulations that appeared to occur at

random. An investigation showed that these differences were caused by one of the MCT simulation subsystems over-running its allotted frame time. Further investigation showed that the cause of the over-run was that the simulation was running out of memory during a run. This problem was fixed by borrowing a larger capacity memory board from the computer vendor.

10.7.4 Pilot Evaluation and Ratings

The pilot evaluation of the MCT Dynamic Inversion control laws was performed in several steps. First, an initial evaluation was performed to ensure that that system was functioning as expected and identify handling qualities problems. Next a secondary evaluation was performed to evaluate the initial corrections and provide feedback for final revisions to the control laws. During the final evaluation, formal Cooper-Harper ratings were taken to document the control law performance.

For the initial pilot evaluation, fifteen tasks, including the two design maneuvers, were selected to evaluate handling qualities:

Eight Tasks Were Flown at Mach 0.8:

1. Pitch-Attitude (20 deg) Capture — Moderate
2. Pitch-Attitude (20 deg) Capture — Aggressive
3. Bank-Angle (90 deg) Capture — Moderate
4. Bank-Angle (90 deg) Capture — Aggressive
5. Bank-Angle (180 deg) Capture — Moderate
6. Bank-Angle (180 deg) Capture — Aggressive
7. Flat Turn with Flat-Turn Mode On*
8. Flat Turn with Flat-Turn Mode Off

Seven Tasks Were Flown-At Mach 0.2:

1. Throttle Transient
2. AOA (60 deg) Capture
3. AOA (30 deg) Capture
4. Pitch-Attitude (55 deg) Capture
5. Pitch-Attitude (10 deg) Capture

6. Bank-to-Bank With Pitch Stick (hold AOA)*

7. Bank-to-Bank With Minimum Pitch Stick

*Design Maneuvers

Only pilot comments were recorded during these initial tests, no formal ratings were made. The pilot made these general comments after the initial evaluation:

1. It was a successful simulation, the results were good for a first run in piloted simulation.

2. Pilot-vehicle coupling was noticed, particularly during pitch attitude captures.

3. Cross-axis coupling was noticed, particularly at very high angles of attack (above 60 deg AOA).

10.7.5 Intermediate and Final Evaluations

The intermediate piloted simulation evaluations were made with two pilots. During the first pilots (Pilot 1) evaluation, it became apparent that the control laws needed additional modifications, especially at maximum commands. During this session, no formal CHRs were collected, but comments were recorded to use in improving the control laws. These corrections were made prior to the final evaluation of the control laws and are provided in the lessons learned section below. Only Pilot 2 evaluated the control laws with the corrections in these final evaluations.

The maneuvers listed below were used to evaluate control law performance in the intermediate and final evaluation runs. Formal Cooper-Harper ratings were assessed for the maneuvers marked by the asterisk (*).

Maneuvers: High AOA Condition, Mach 0.23, 20,000 Feet

1. Open-Loop Pitch Response

2. Open-Loop Yaw/Roll Response

3. Pitch-Attitude Capture*

4. Angle-of-Attack Capture*

5. Bank-to-Bank Rolls*

Maneuvers: Low AOA Condition, Mach 0.7, 10,000 Feet

1. Open-Loop Pitch Response

2. Open-Loop Yaw/Roll Response

3. Pitch Attitude Capture*

4. Bank-to-Bank Rolls*

5. Pullup

6. Push-Over

7. Heading-Angle Capture — Flat-Turn Mode

8. Windup Turn

9. Loaded Roll*

10. Flat-Turn Strafing Maneuver*

Cooper-Harper ratings assigned during the final evaluation follow.

10.7.6 Cooper-Harper Ratings (CHR)

High-Angle-of-Attack Maneuvers:

Maneuver 3: Pitch Attitude Capture
+10 deg Capture: Pilot 2: CHR 4

Maneuver 4: Angle-of-Attack Capture
50 deg Capture: Pilot 2: CHR 2

Maneuver 5: High AOA Bank-to-Bank Rolls
30 deg Banks: Pilot 2: CHR 4
45 deg Banks: Pilot 2: CHR 4

Low-Angle-of-Attack Maneuvers:

Maneuver 3: Pitch Attitude Capture
20 deg Attitude Change: Pilot 2: CHR 3 (nonaggressive), CHR 4 (aggressive)

Maneuver 4: Bank-to-Bank Rolls
45 deg Banks: Pilot 2: CHR 1 (nonaggressive), CHR 2 (aggressive)

Maneuver 9: Loaded Roll
Three-G Roll: Pilot 2: CHR 4

Maneuver 10: Flat-Turn Strafing Maneuver
Acquire And Strafe Bridge: Pilot 2: CHR 3

10.7.7 Lessons Learned, Corrective Actions

This section contains a description of the control-law related problems noted by the pilots during the evaluations with the corrective actions taken to try to improve performance.

10.7.7.1 Roll Axis PIO At High Angle of Attack

Pilot 1 experienced roll axis PIO at high angles of attack during aggressive bank angle capture maneuvers.

Corrective Action:

1. Increase the design roll-mode time constant. The initial design used a roll-mode time constant of 0.4 at low angles of attack (below 30 deg) and 0.8 at high angles of attack (above 45 deg), with values in-between linearly interpolated. These values are more stringent than the level 1 boundary values for high AOA tracking suggested in the MCT requirements document, so they were increased to 0.5 at low angles of attack (below 20 deg) and 1.6 at high angles of attack (above 60 deg).

2. Adjusted the roll-stick command gradient to reflect actual F-16 roll performance to roll commands. The use of an integrator in the roll axis in the dynamic inversion control law drives the system to the commanded values, which led to a much too sensitive gradient.

3. Investigate software gain reduction function. The MCT dynamic inversion control laws have a function to reduce the system bandwidth when control power saturates in an axis, effectively reducing the gain.

Result:

1. Increasing the roll-mode time constant and decreasing the roll-command gradient improved the roll response at both flight conditions. The roll PIO was less evident during aggressive high-angle-of-attack maneuvers.

2. Same as 1.

3. The software gain reduction function behaved unpredictably. There was little experience or analysis of this function during the MCT program, so it was not investigated further.

Further Corrections Needed: Investigate scheduling some of the desired dynamics parameters to reduce controller bandwidth at low dynamic pressure conditions when operating at actuation limits.

10.7.7.2 Pitch Axis Stability Deteriorated Above Mach 0.8

Pilot 1 noted that the pitch axis response became very lightly damped above Mach 0.8.

Corrective Action: Add more least squares data to better describe the aircraft model in the nonlinear transonic region.

Further Corrections Needed: Perform more complete analysis on the required number and placement of least squares data points to cover the nonlinear transonic region. Investigate scheduling some controller parameters to improve stability margins.

10.7.7.3 Nose Movement Lagged During Flat-Turn Maneuvers

After the modified least squares data was implemented, pilot number one noted that the lateral movement of the aircraft nose lagged behind rudder pedal inputs and jumped abruptly when the pedal was released. Analysis of the real-time and offline data showed that the controller was achieving the desired heading angle rate and lateral acceleration, the body axis yaw rate was very slow to respond and jumped abruptly when the pedal command was released. The analysis also showed that the controller was tracking the YCV command accurately, but not the NCV command.

Corrective Action:

1. Implement a lag filter on the rudder pedal command to slow down heading-rate response and allow the NCV command system to maintain better control of the body-axis yaw rate and sideslip angle.

2. Modify the proportion of yaw rate to slideslip angle in the NCV command system (the K_Beta gain).

3. Reformulate the least squares data to be closer to the original data set.

Results:

1. The lag filter was analyzed in offline simulation: several filter constants were evaluated against proper yaw-rate response. This option was not taken to real-time simulation because the filters that improved the yaw-rate response also slowed the lateral-acceleration response too severely

2. The value of K_Beta was varied in offline simulations between the original value (–1) and the final value (0) without finding a value that successfully provided the desired flat turn, yaw rate, and sideslip response. This option was not evaluated in piloted simulation.

3. The reformulated least squares data corrected the yaw-rate response with some detriment to pitch-axis stability. Test cases were flown successfully below transonic velocities.

Further Corrections Needed: Investigate the compatibility of the command variables, YCV (lateral flight path rate) and NCV (yaw rate and sideslip blend). Determine whether the implementation of the YCV channel in a nonstandard sense (where the effectiveness of the controls is not calculated with respect to the derivative of YCV, but rather to YCV itself) produces control commands that are inconsistent with simultaneously achieving both YCV and NCV.

10.7.7.4 Insufficient Normal Acceleration Command Available

Pilot 1 noted that the normal acceleration was limited to between 3 and 4 g's.

Corrective Action:

1. A limiting function was removed from the controller software. This limiter was set for a different type of stick than the F-16, so that the units on the limiter were not set properly.

2. The stick gain was modified to provide increased command magnitude.

Result:

1. Removing the limiting function provided enough additional command capability to complete the maneuver set

2. Increasing the stick gain boosted the command capability to an acceptable level, but also increased the stick sensitivity to an uncomfortable level (see next pilot comment below). Pilots 1 and 2 noted an oscillation in the g response at maximum load factors, occasionally reaching $\pm$ 1 g. This oscillation was corrected by turning off the software gain reduction function in the dynamic inversion control laws.

10.7.7.5 Too Sensitive Pitch Response, Pitch Dropback

After pitch axis command limit was increased (corrective action 4b), Pilots 1 and 2 noted that pitch command was too sensitive and Pilot 2 noted excessive pitch dropback. These attributes of the control laws were not noted before the pitch command path was modified.

Correction: The pitch-command gradient was fixed to the original values.

Result: Pilot noted good pitch command sensitivity; did not note dropback.

10.7.7.6 Pitch/Roll Coupling

At high angles of attack, pilots noticed that significant roll rates developed during large pitch commands, especially at angles of attack over 60 deg.

Correction: Implement a lag filter in the MCV command path (3 rad/sec).

Result: The pitch/roll coupling was virtually eliminated below 60 deg angle of attack and reduced above 60 deg angle of attack.

Further Corrections Needed: Investigate control allocation and command limiting logic to ensure that roll control power remains available at maximum pitch commands to eliminate pitch/roll coupling above 60 deg angle of attack. The current daisy chain system for control allocation uses symmetric horizontal tail as the first priority pitch controller, pitch thrust vectoring as the secondary pitch controller, asymmetric trailing-edge flap as the primary roll controller, and asymmetric horizontal tail as the secondary roll controller. At high angles of attack, the asymmetric horizontal tails are the most effective roll controller and the pitch thrust vectoring is the most effective pitch controller. However; this daisy chain system often saturates the horizontal tails trying to satisfy pitch commands, eliminating roll-control power.

10.7.7.7 Maximum Roll Rates Too Large

Pilot 2 noticed that roll rates generated with maximum stick commands at the high speed condition were too large, often exceeding 300 deg/sec. These roll rates were accompanied by a large amount of roll coast. Both pilots also noted that the roll commands were too sensitive at high angles of attack.

Correction: Reduce the roll command gradient to 60gradients for the MCT control laws were taken from the F-16 control laws; however only 60achieved because there is no integrator in the F-16 roll axis. The MCT roll-axis control law does contain an integrator, so the control law always tries to achieve the maximum command. Thus, excessive roll rates were commanded and caused control power saturation, leading to poor handling qualities.

Result: The command reduction reduced the maximum roll command to normal levels, which provided adequate roll response while reserving sufficient control power for the other axes.

Section 11

μ-Synthesis

Currently, structured singular value (μ) synthesis is the only multivariable design method that can be used to directly optimize robust performance. It involves H_∞-synthesis and μ-analysis. Performance specifications are weighted transfer functions that describe magnitude and frequency content of external disturbances, pilot commands, atmospheric gusts, and sensor noise, as well as allowable magnitude and frequency content of generalized tracking errors, handling qualities, ride qualities, and actuator activity. The design technique applies to a set of models consisting of a nominal model plus structured perturbations, with magnitude bounds and frequency content again specified using weighted transfer functions. Specifications of performance and definition of the model set over which performance must be achieved are all incorporated into a single standard interconnection structure, upon which existing design algorithms can operate.

This section is organized as follows. First, we summarize the procedure for μ-synthesis controller design. Next, we explain each step of the procedure in some detail. Following that, we provide some examples to illustrate use of the design method. Finally, we summarize the H_∞- and μ-analysis theory that underlies the μ-synthesis procedure.

11.1 μ-Synthesis Steps

Fundamental steps in the μ-synthesis design procedure that are described in this section include:

1. **Interconnection Structure Definition:** The first step in μ-synthesis is to construct an interconnection structure, P. This structure is just a state-space realization of the aircraft dynamics, augmented with handling-qualities models and weighting functions with various inputs and outputs that specify control design goals. All conventional control designer's "knobs" are embodied in the definition of the interconnection structure.

2. **H_∞-Synthesis:** Once the interconnection structure has been defined, we design an H_∞-optimal controller for the structure. This involves solving two Riccati equations [Doyle 1989] that are iterated over a scalar parameter γ in a one-dimensional search. This step results in a control compensator K. The closed-loop system is then formed by connecting the sensors and actuators of P to K to produce the closed-loop interconnection structure, M. (Section 11.4.)

3. **μ-Analysis:** In the next step, we apply μ-analysis to the closed-loop system, M. This involves calculation of the structured singular value, $\mu[M]$, and its associated frequency-dependent D-scales. The structured singular value provides a measure of how close the compensator from Step 2 is to meeting its robust performance goals. Small values are good, large values are bad, and $\mu[M] = 1$ means that the goals are just barely satisfied. (Section 11.5.)

4. **Rational Approximation of D-Scales:** In this step, the D-scales from μ-analysis are approximated by frequency-response magnitudes of rational transfer functions. (Section 11.5.)

5. ***D-K* Iteration:** Next, the rational-transfer-function approximations from Step 4 are incorporated into the interconnection structure P and the H_∞-synthesis, μ-analysis, and D-scale approximation steps (Steps 2–4) are repeated until D and K no longer change. (Section 11.6.)

6. **Changing Weights:** If D and K have converged, but the compensator does not meet its goals (i.e., $\mu > 1$), then the weights in Step 1 must be changed, trading off some goals against others the D-K iteration repeated. M-analysis is used to determine which input/output paths are driving the problem. (Section 11.6.3.)

7. **Compensator Model Reduction:** Once the design is complete, the controller is simplified using model reduction. The μ-analysis step is then repeated with the simplified controller to ensure that design goals are still satisfied. (Section 11.7.)

11.2 Interconnection Structures

The interconnection structure, P, has inputs and outputs arranged as follows:

$$\begin{bmatrix} z \\ e \\ y \end{bmatrix} = P \begin{bmatrix} v \\ d \\ u \end{bmatrix} \tag{11.1}$$

where z and v denote signals associated with model uncertainty or perturbations; e generalized tracking errors; d external commands, disturbances, and sensor noise; y available measurements for control; and u control actuator inputs. Thus, P is just a (large) transfer-function matrix with associated state-space matrices $[A, B, C, D]$ such that

$$P = C(sI - A)^{-1}B + D \tag{11.2}$$

P plays a central role in the general feedback diagram shown in Figure 11.1. As shown in this diagram, P is connected to two other system components: The first component is Δ, which is another transfer-function matrix used to define the model set over which

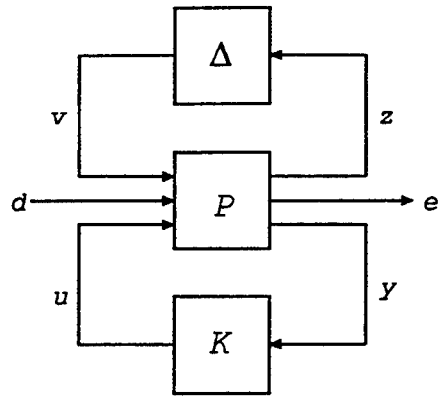


Figure 11.1: General Feedback-Loop Block Diagram.

performance must be achieved. Δ is completely unknown, except for its fixed block-diagonal structure and magnitude bound $\|\Delta\|_\infty \leq 1$. That is, if

$$v = \Delta z \qquad \text{then} \qquad \|v\| \leq \|z\| \tag{11.3}$$

The second component is the control law to be synthesized, K, which is a transfer function that connects y and u:

$$u = Ky \tag{11.4}$$

The final product of μ-synthesis is a realization for K given in terms of its state-space matrices.

Within this general framework, K contains both the feedback and precompensation portions of the control law. Precompensation is that portion of the control law that acts on measured generalized disturbances (e.g., pilot commands), whereas feedback is that portion of the control law that acts on the sensed motion of the aircraft. We include both of these signals in measurement vector y; therefore, K includes both control functions. This form of K is often called a *two-degree-of-freedom* controller.

Various software tools exist that facilitate setting up state-space matrices for the interconnection structure. Some work with algebraic system descriptions, whereas others work with graphical ones. In either case, the basic operations involved are addition, multiplication, and loop closures of state-space realizations.

Subsections 11.2.1 through 11.2.5 describe intermediate steps that are used to construct the interconnection structure. Subsection 11.2.6 then collects the results of these intermediate steps to generate the full P. Sections 11.8 and 11.9 contain detailed examples for longitudinal and lateral-directional aircraft flight control design.

11.2.1 Aircraft Model

The first, and most obvious, part of P is the aircraft model with actuator commands as inputs (e.g., elevator, aileron, rudder actuators, etc.) and measured dynamic signals

as outputs (e.g., accelerations, angular rates, attitudes, airspeed, etc.). As appropriate, some of the dynamics included in the aircraft model may be omitted. For example, low-frequency states that are not important to the design may be truncated. High-frequency states, whose frequencies exceed intended bandwidth of the control loops, may be residualized.

As discussed previously, the choice of control effectors to be used is part of this model definition step, as was discussed Section 6. Often, a first-order model is adequate for actuator dynamics.

The choice of sensed quantities and appropriate sensor models is also part of this step. Rate gyros and accelerometers are usually high-bandwidth devices so that their dynamics can often be neglected. Likewise, inertial navigation systems are high-bandwidth and no dynamic models need be incorporated into P, although we may include a time delay (Pade-approximated) for specific systems. We may also require air-data system dynamics for some applications. This involves air-data computers with relatively slow dynamics caused by signal conditioning and smoothing of the sensed data itself.

In any case, the start of the interconnection structure is shown in Figure 11.2, and this part of the structure defines control signals u and most of the measurement signals y defined in the lower part of the general feedback diagram in Figure 11.1.

Figure 11.2: Aircraft Model: Start of the Interconnection Structure.

11.2.2 Performance Model

The second portion of P to be specified is the performance model. This model defines generalized errors e and generalized disturbances d in the middle part of Figure 11.1. These are the signals that the designer uses to judge the quality of closed-loop performance.

Because the design method is optimization-based, generalized errors must be defined such that small magnitudes represent good performance, whereas large magnitudes represent poor performance. Sometimes, the signals are just simple errors in the feedback loop, but they can also be complex dynamically shaped signals that describe desired closed-loop behavior. The most common generalized errors are:

- Tracking errors
- Actuator deflections
- Actuator rates

11.2.2.1 Tracking Errors

In the classical SISO feedback loop of Figure 11.3, tracking errors correspond to the error signals going into controller K. We also use these error signals for μ-synthesis.

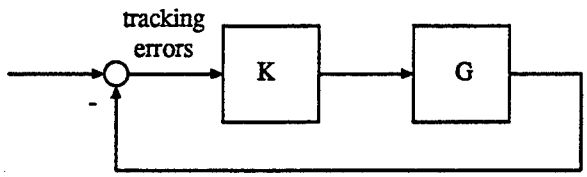


Figure 11.3: Tracking Errors in Classical SISO Feedback Loop.

As already discussed, however, the general feedback block diagram in Figure 11.1 has the two-degree-of-freedom structure of Figure 11.4: For this case, we must separately

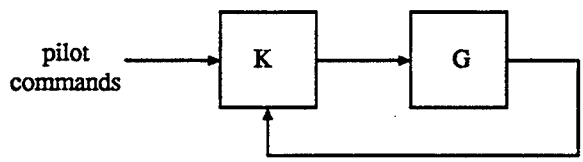


Figure 11.4: Alternate Two-Degree-of-Freedom Structure.

form tracking errors between pilot commands and desired closed-loop responses. This is done by feeding the pilot commands through a dynamic model with desired closed-loop characteristics and subtracting the actual responses from the desired responses, as shown in Figure 11.5. The performance model is typically a first- or second-order filter.

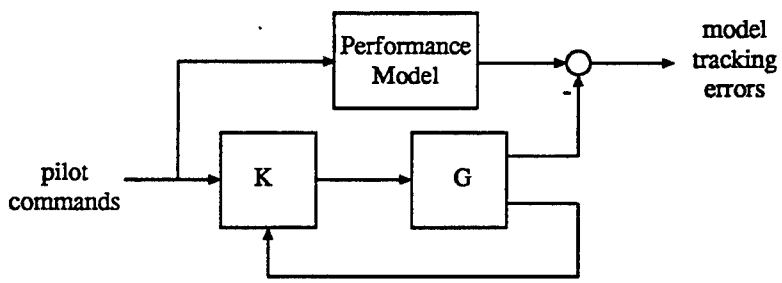


Figure 11.5: Adding an Explicit Model to Form Tracking Errors.

11.2.2.2 Actuator Deflections

All practical control systems must keep actuator deflections below a maximum hardware-imposed bound. We address this aspect of performance by including actuator deflections as error signals in the definition of P. Figure 11.6 shows a model for these signals.

As described in Section 11.4, certain rank conditions for well-posedness of H_∞ synthesis require that control signals u be penalized at high frequency. For this reason, it is often easier to use actuator position *commands* as performance signals rather than actuator positions themselves.

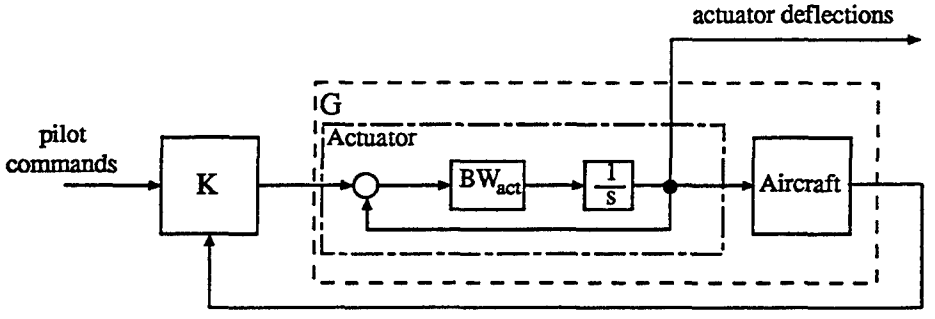


Figure 11.6: Pulling Off Actuator-Deflection Signals.

11.2.2.3 Actuator Rates

Similarly, all practical controllers must keep actuator rates below a maximum hardware bound. We address this aspect of performance by including actuation rates as error signals in the definition of P, as shown in Figure 11.7. As we will show later in the examples, weights on actuator rate signals provide effective "knobs" to tune final bandwidth of the control system.

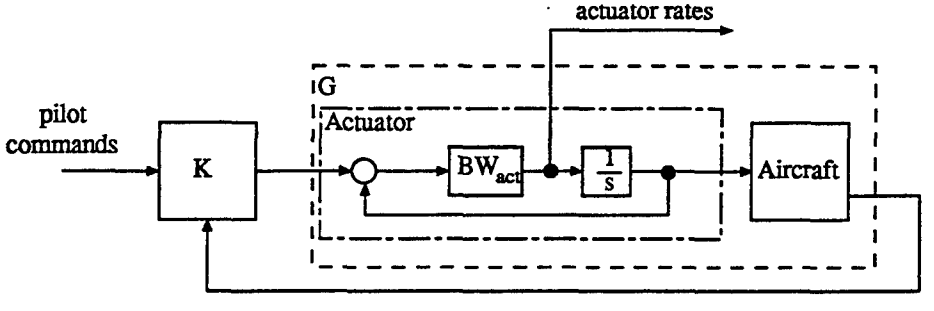


Figure 11.7: Pulling Off Actuator-Deflection-Rate Signals.

11.2.2.4 Generalized Disturbances

The second category of performance signals are the various external inputs that excite the feedback loop and drive errors to potentially large magnitudes. Most commonly, these signals include:

- Sensor noise
- External disturbances (gusts, store drops, gun fire transients, etc.)
- Commands from pilots or outer loops

Sensor noises are typically represented by wide-band disturbances added directly to measured signals. Often they are not optional, but are required to meet the rank conditions described in Section 11.4. External disturbances typically enter the aircraft model directly, whereas pilot or outer-loop commands pass directly through to measurements y to be used as inputs of controller K, as illustrated in Figure 11.8.

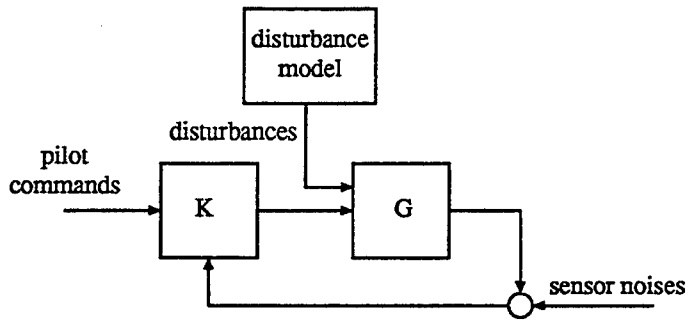


Figure 11.8: Disturbance and Noise Inputs.

11.2.3 Uncertainty Models

The third partition of P to be specified is the uncertainty model. This model defines the set of plants over which performance objectives must be satisfied. It is represented by signals v and z in the top part of Figure 11.1, which connect uncertain components Δ into the feedback loop.

Sections 2.5 and 2.6 have already discussed uncertainty, its characterization, and its importance in control design, whereas Appendix ?? describes methods for bounding it. In this section we describe how the characterizations from those sections are used to incorporate uncertainty into P. We treat both complex-valued perturbations (e.g., perturbations of transfer functions) and real-valued perturbations (e.g., mass and inertia variations) as well as their interactions. Together, these factors define the final structure of Δ.

11.2.3.1 Uncertainty at the Input

Design models used for flight control typically exhibit good fidelity at lower frequencies, say $\omega < 10$–20 rad/sec, but they degrade rapidly at higher frequencies due to such

poorly modeled or neglected effects as aeroelasticity, actuator compliance, servo dynamics, computational/digital effects, and so on. Such modeling errors are well-represented by complex-valued, unstructured, multiplicative perturbations located at plant input, as illustrated in Figure 11.9 for which

$$\Delta_{input} = \begin{bmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1m} \\ \delta_{21} & \delta_{22} & \cdots & \delta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{m1} & \delta_{m2} & \cdots & \delta_{mm} \end{bmatrix} \tag{11.5}$$

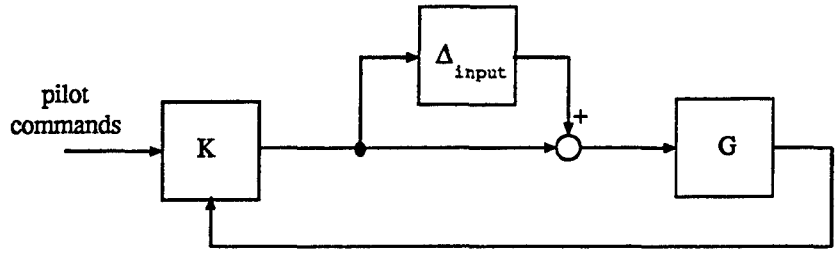


Figure 11.9: Unstructured Uncertainty at the Plant Input.

A complex-valued unstructured representation is appropriate here because magnitude and phase errors and cross-channel-coupling errors are all prominent at higher frequencies. The multiplicative form is chosen for convenience because it permits the intuitive interpretation of uncertainty magnitudes in terms of percent errors relative to the design model.

To incorporate this uncertainty representation into interconnection structure P, we include the outputs of K (i.e., signals u) in the signal set z. Thus, a subset of z is $z_{input} = u$. We also include a summing junction that adds the corresponding signals $v_{input} = \Delta_{input} z_{input}$ to the nominal u's before passing them to the aircraft. P does not explicitly incorporate the uncertainty block Δ_{input} itself, which we assume to be unit-size. Its actual size is determined in the weight-selection step described later.

11.2.3.2 Uncertainty at the Output

Similar high-frequency uncertainties are also associated with sensor hardware for various measured aircraft outputs. It is again reasonable to represent these sensor uncertainties as complex-valued perturbations in multiplicative form. In addition, we often also assume that these perturbations are diagonally structured, as shown in Figure 11.10 where

$$\Delta_{output} = \begin{bmatrix} \delta_{11} & 0 & \cdots & 0 \\ 0 & \delta_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_{mm} \end{bmatrix} \tag{11.6}$$

This implies that errors in one sensor signal will not effect sensed quantities in other channels.

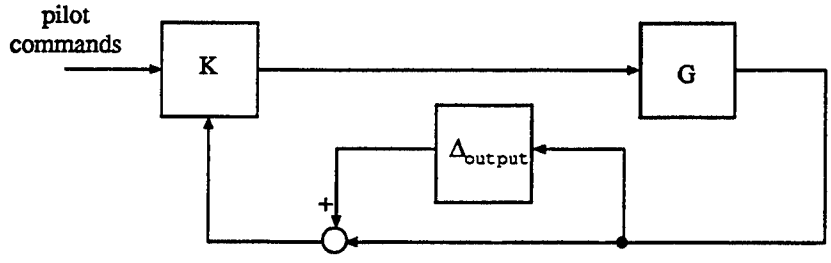


Figure 11.10: Structured Uncertainty at the Plant Output.

To incorporate this output uncertainty representation into interconnection structure P we include aircraft measured outputs as another subset of signal set z and add another summing junction that adds corresponding signals $v_{output} = \Delta_{output} z_{output}$ to the aircraft outputs to produce the final measured signals y. As in the input uncertainty case, P does not explicitly include the output uncertainty block itself, which is taken to be unit-size, with actual size determined in the weight selection step.

11.2.3.3 Real-Parameter Uncertainty

Another important type of uncertainty is internal to the design model and, as the name implies, is characterized by errors in real-valued parameters. Real-parameter uncertainty includes uncertainty associated with such things as mass, inertia and/or aerodynamic coefficients. The incorporation of such uncertainty into P involves modification of the aircraft model itself. As was true for input and output uncertainty descriptions, the net effect of real-parameter uncertainty is to create additional inputs v and outputs z in the upper partition of P.

The method for doing this is well-documented [Morton 1985], but will be described by way of a simple example here.

Roll-Rate Example: Suppose we wish to describe uncertainty in C_{l_p}, which for the sake of this example, is bounded by $C_{l_p}^- \leq C_{l_p} \leq C_{l_p}^+$.

This term appears in the linearized aircraft dynamics as

$$\dot{p} = \frac{\bar{q}Sb}{I_{xx}} C_{l_p} p + \frac{\bar{q}Sb}{I_{xx}} C_{l_{\delta_a}} \delta_a$$

or in state-space form

$$\left[\begin{array}{c} \dot{p} \\ \hline p \end{array}\right] = \left[\begin{array}{c|c} \frac{\bar{q}Sb}{I_{xx}} C_{l_p} & \frac{\bar{q}Sb}{I_{xx}} C_{l_{\delta_a}} \\ \hline 1 & 0 \end{array}\right] \left[\begin{array}{c} p \\ \hline \delta_a \end{array}\right]$$

The uncertain roll damping coefficient can be parameterized, as discussed in Section 2.5, with a center or nominal value given by $C_{l_p}^{nom}$ and a bound for a perturbation that corresponds to the range of possible values between $C_{l_p}^-$ and $C_{l_p}^+$:

$$\dot{p} = \frac{\bar{q}Sb}{I_{xx}}(C_{l_p}^{nom}p + k_1 v) + \frac{\bar{q}Sb}{I_{xx}}C_{l_{\delta_a}}\delta_a \tag{11.7}$$

$$z = p + k_2 v \tag{11.8}$$

It can be verified that if $v = \Delta z$, then

$$\dot{p} = \frac{\bar{q}Sb}{I_{xx}}C_{l_p}p + \frac{\bar{q}Sb}{I_{xx}}C_{l_{\delta_a}}\delta_a \tag{11.9}$$

where

$$C_{l_p} = C_{l_p}^{nom} + k_1\Delta(1 - k_2\Delta)^{-1} \quad . \tag{11.10}$$

with

$$k_1 = \frac{2(C_{l_p}^+ - C_{l_p}^{nom})(C_{l_p}^{nom} - C_{l_p}^-)}{(C_{l_p}^+ - C_{l_p}^-)}$$

$$k_2 = \frac{(C_{l_p}^+ + C_{l_p}^-) - 2C_{l_p}^{nom}}{C_{l_p}^+ - C_{l_p}^-}$$

Simple substitution shows that $C_{l_p} = C_{l_p}^-,\ C_{l_p}^{nom},\ C_{l_p}^+$ for $\Delta = -1,\ 0,\ 1$, respectively.

The perturbed state-space, which has the necessary additional input and output corresponding to the uncertainty, is:

$$\left[\begin{array}{c} \dot{p} \\ \hline z \\ p \end{array}\right] = \left[\begin{array}{c|cc} \frac{\bar{q}Sb}{I_{xx}}C_{l_p}^{nom} & \frac{\bar{q}Sb}{I_{xx}}k_1 & \frac{\bar{q}Sb}{I_{xx}}C_{l_{\delta_a}} \\ \hline 1 & k_2 & 0 \\ 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} p \\ \hline v \\ \delta_a \end{array}\right]$$

with the block diagram representation shown in Figure 11.11 where

$$\hat{G} = \left[\begin{array}{c|cc} \frac{\bar{q}Sb}{I_{xx}}C_{l_p}^{nom} & \frac{\bar{q}Sb}{I_{xx}}k_1 & \frac{\bar{q}Sb}{I_{xx}}C_{l_{\delta_a}} \\ \hline 1 & k_2 & 0 \\ 1 & 0 & 0 \end{array}\right]$$

Closing the Δ-loop in Figure 11.11 with $\Delta = 1$ and $\Delta = -1$ gives

$$\dot{p} = \frac{\bar{q}Sb}{I_{xx}}C_{l_p}^+ p + \frac{\bar{q}Sb}{I_{xx}}C_{l_{\delta_a}}\delta_a$$

and

$$\dot{p} = \frac{\bar{q}Sb}{I_{xx}}C_{l_p}^- p + \frac{\bar{q}Sb}{I_{xx}}C_{l_{\delta_a}}\delta_a$$

respectively, which is what was desired.

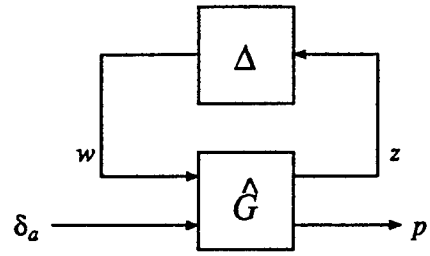


Figure 11.11: Perturbed System.

Note that the nominal C_{l_p} is often at the center of the range of variation: that is, $C_{l_p}^{nom} = 0.5(C_{l_p}^- + C_{l_p}^+)$. In this case, $k_2 = 0$ and $k_1 = 0.5(C_{l_p}^+ - C_{l_p})$. This special case was called a *multiplicative* perturbation in Section 2.5.

Another special case of interest is when the nominal $C_{l_p}^{nom} = \frac{2C_{l_p}^+ C_{l_p}^-}{C_{l_p}^- + C_{l_p}^+}$, $k_1 = \frac{2(C_{l_p}^+ - C_{l_p})C_{l_p}^+ C_{l_p}^-}{(C_{l_p}^- + C_{l_p}^+)^2}$, and $k_2 = \frac{C_{l_p}^+ - C_{l_p}^-}{C_{l_p}^- + C_{l_p}^+}$. This special case was called a *divisive* perturbation in Section 2.5.

Similarly, we can modify the state space to allow for variations in $C_{l_{\delta_a}}$. The state-space for this case is the following:

$$\left[\begin{array}{c} \dot{p} \\ \hline z_2 \\ p \end{array}\right] = \left[\begin{array}{c|cc} \frac{\bar{q}Sb}{I_{xx}} C_{l_p} & \frac{\bar{q}Sb}{I_{xx}} l_1 & \frac{\bar{q}Sb}{I_{xx}} C_{l_{\delta_a}}^{nom} \\ \hline 0 & l_2 & 1 \\ 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} p \\ \hline v_2 \\ \delta_a \end{array}\right]$$

with

$$l_1 = \frac{2(C_{l_{\delta_a}}^+ - C_{l_{\delta_a}}^{nom})(C_{l_{\delta_a}}^{nom} - C_{l_{\delta_a}}^-)}{(C_{l_{\delta_a}}^+ - C_{l_{\delta_a}}^-)}$$

$$l_2 = \frac{(C_{l_{\delta_a}}^+ + C_{l_{\delta_a}}^-) - 2C_{l_{\delta_a}}^{nom}}{C_{l_{\delta_a}}^+ - C_{l_{\delta_a}}^-}$$

Note that this specific uncertainty (i.e., an uncertain element in the B matrix) is a subset of the complex-valued input uncertainties described earlier. Thus, if complex-valued input uncertainties are already included in P, there is no need to include additional uncertainty in the interconnection structure.

Finally, these two real-parameter uncertainties can be included together by adding an input and output for each parameter. For this case, the state space is

$$\left[\begin{array}{c} \dot{p} \\ \hline z_1 \\ z_2 \\ p \end{array}\right] = \left[\begin{array}{c|ccc} \frac{\bar{q}Sb}{I_{xx}} C_{l_p}^{nom} & \frac{\bar{q}Sb}{I_{xx}} k_1 & \frac{\bar{q}Sb}{I_{xx}} l_1 & \frac{\bar{q}Sb}{I_{xx}} C_{l_{\delta_a}}^{nom} \\ \hline 1 & k_2 & 0 & 0 \\ 0 & 0 & l_2 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} p \\ \hline v_1 \\ v_2 \\ \delta_a \end{array}\right]$$

As mentioned earlier, multivariable generalizations of the ideas detailed in this example can be found in the literature [Morton 1985].

11.2.4 Performance Weights

The fourth and final portion of P to be specified is a set of weighting functions for the various signals in P. These functions are transfer functions, whose state-space realizations are embedded in P, with specified frequency response shapes and magnitudes. The functions associated with signals d and e are called *performance weights,* whereas those associated with v and z are called *uncertainty weights.*

The role of weights is to scale the interconnection structure (i.e, *normalize* it) such that control objectives in the unscaled structure are satisfied whenever the closed-loop gain from d to e in the scaled structure is less than unity for all unit-size perturbations Δ with the requisite block-structured form. The μ-synthesis design tools then work with the scaled structure to find a controller K that achieves this unit-norm robust performance goal.

The weights are the specifications that drive the control design. As is true with classical SISO design specifications, they can be used in two ways: to determine achievable performance against fixed specifications (an analysis question) or to trade off some specifications against others. The difference is subtle, but important. For example, a designer may know that 10 dB gain margin is required, that 5-rad/sec bandwidth is desired for performance reasons, and that there is 50% model uncertainty below 20 rad/sec. Incorporating these design objectives into the interconnection structure and completing the *D-K* iterations, we may find that the final μ-analysis gives μ values larger than unity. Hence, we know that desired performance cannot be achieved in the presence of the expected uncertainty. The achievable performance question is thus resolved: "No, it can't be done." Of course, the follow-up question, "Ok, what can be done?," is even more compelling. To answer this question, the designer must alter the weights in a series of trials to effect suitable compromises. For instance, he may find that robust performance can be guaranteed if the bandwidth goal is dropped to 4 rad/sec and model uncertainty level is dropped to 40%. In this manner, all of the classical control design "knobs" are embodied in the size and shape of the weights.

The remainder of this subsection examines typical weight selections for each of the performance and uncertainty signals described above.

11.2.4.1 Tracking-Error Weights

The weights chosen for the tracking errors can be thought of as penalty functions. That is, weights should be large in frequency ranges where small errors are desired and small where larger errors can be tolerated. Another way to think about these weights is that their frequency responses should be chosen so that acceptable unnormalized error signals excited by disturbances d when passed through the weighting transfer function have normalized-error frequency responses that are approximately flat and unit-size. (Note that for classical SISO loops with d entering at the output, this interpretation means

that these weights correspond to the desired frequency-response magnitude of the return difference $I + GK$.)

In terms of the performance model in Figure 11.5, we incorporate the weight W_e as shown in Figure 11.12. The signal, e^1 is the normalized error signal.

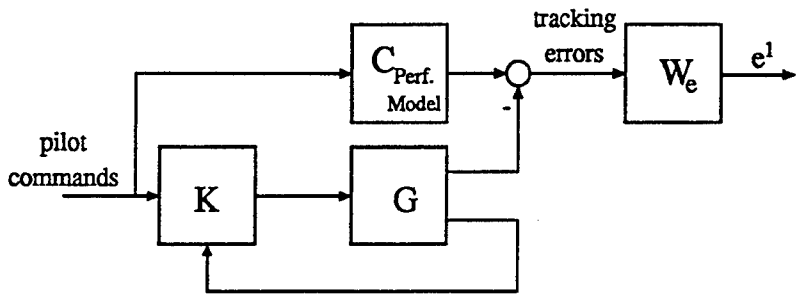


Figure 11.12: Weighted Tracking Errors.

The size and frequency-response shapes of tracking-error weights depend upon several considerations. First, we should recognize that errors at frequencies well beyond feedback loop bandwidth will necessarily be open-loop size. This means that the weights should roll off toward zero magnitude or level off to values less than the inverse of the open-loop d-to-e gain. Second, to achieve integral action (i.e., zero steady-state errors), weights should be large at very low frequency. We caution, however, that most software packages for μ-synthesis do not tolerate pure integrations in the weights, so a $1/(s + \epsilon)$ weight with ϵ sufficiently small must suffice. We can specify similar near-zero error performance for other narrow frequency ranges by including oscillators in the weights, again with a small amount of damping added to satisfy the software. Finally, it is critical to choose appropriate relative sizes for multiple signals within e. The rules for sizing are simple: each weight should be chosen such that the product of acceptable unnormalized signals multiplied by the weight is approximately flat and unit-size. In this way, all performance signals receive comparable emphasis in μ-synthesis optimization. Note that units are very important in this context. For example, weighting two angular signals, one measured in degrees and the other in radians, with identically sized weights implicitly favors the signal measured in degrees by a factor of $180/\pi$. By the same reasoning, a means of design tradeoffs is also apparent. If the performance of one error needs to be improved relative to another error, one need only change the relative weighting of the two.

When choosing performance weights for design tradeoffs, it is also important to keep in mind the range of validity of the design model. For instance, there is nothing to be gained by requiring integral action for an error signal when the design model does not correctly represent low-frequency characteristics. The short-period approximation of longitudinal aircraft dynamics, which neglects phugoid modes, is a case in point. Because low-frequency behavior is not correctly represented in such approximations, weights can be rolled off in that frequency range without compromising the (short-period) design.

Figure 11.13 shows a number of typical shapes for error weighting functions.

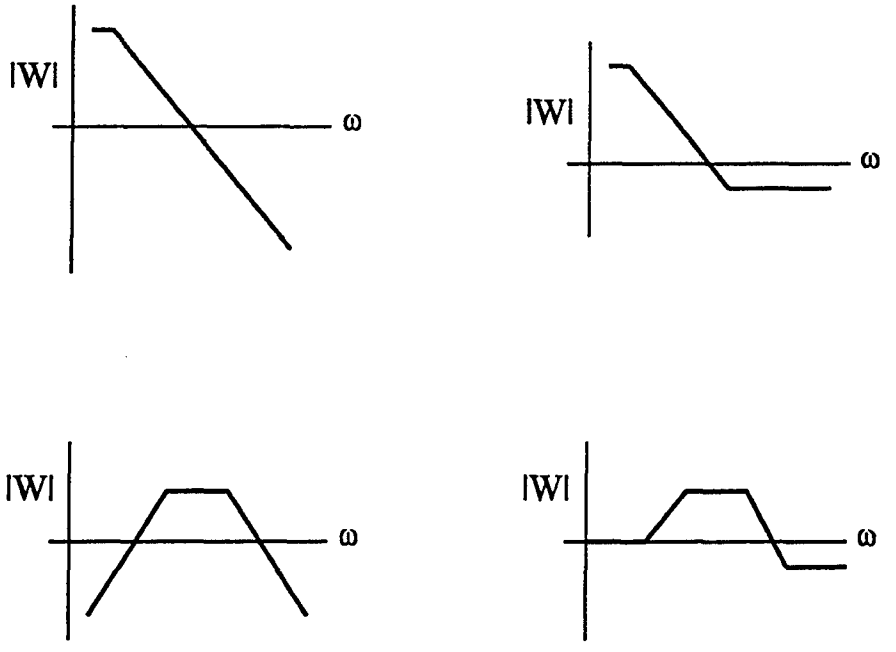


Figure 11.13: Typical Weighting Shapes for Tracking Errors.

11.2.4.2 Actuator-Deflection Weights

By weighting actuator deflections or deflection commands, the designer can penalize larger deflections and thereby minimize control activity. Note that we can again interpret these weights as penalties. We can also apply the notion that their size should be chosen to make the product of acceptable unnormalized deflections multiplied by the weight nearly flat and unit-size. Figure 11.14 illustrates how these weights are incorporated into the interconnection structure.

Following these interpretations, the simplest weight on actuator deflections is constant across frequency and has magnitude equal to the inverse of the deflection limit. For example, if aileron limits are -50 degrees to $+50$ degrees in a lateral-directional design problem, a suitable weight for their deflections would be $1/50$. A recommended alternative is to make the weight equal to the inverse of the size of the deflection range. In the previous instance the weight would become $1/100$. This choice eliminates ambiguities that arise from asymmetric limits.

11.2.4.3 Actuator-Rate Weights

Similarly, weights on actuator rates also minimize control activity. Figure 11.15 shows how they are incorporated into P.

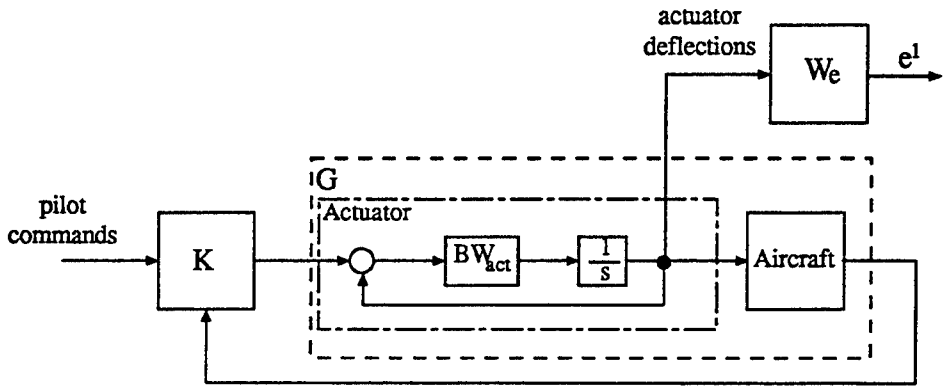


Figure 11.14: Weighted Actuator Deflections.

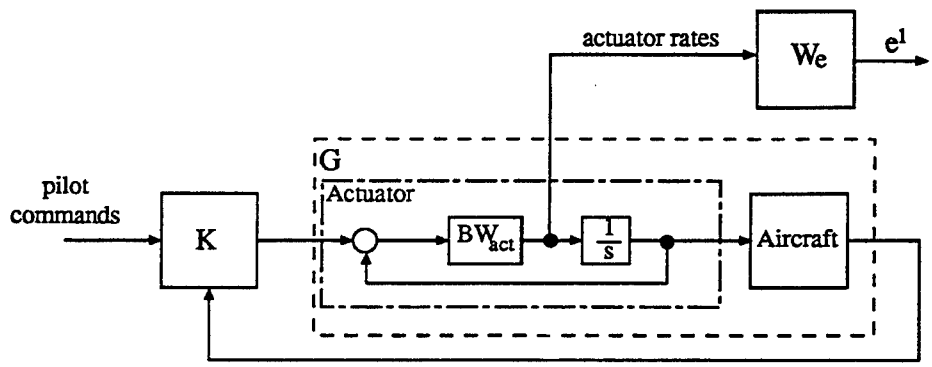


Figure 11.15: Weighted Actuator Rates.

Suitable choices for these weights are again the inverse of maximum acceptable signal values (i.e., the rate limits, held constant for all frequency). In special cases, a frequency-dependent weight may be justified by hinge moment considerations but is usually not necessary.

Because system bandwidth is directly related to system response speed and actuator response rate can limit system response, weights on actuator rates can be used as effective "knobs" to modulate final bandwidth of the closed-loop system: larger weights decrease bandwidth whereas smaller ones increase it.

11.2.4.4 Commands, Disturbances, and Noise Input Weights

We now turn to weight selection for external distu bance signals d in the interconnection structure. Recall that these signals include sensor noise, external disturbances (e.g., gust, transients, etc.), and pilot commands. The role of weights for these signals is basically the opposite of the role of weights for e discussed so far. Rather than taking unscaled signals and normalizing them, the disturbance weights take flat unit-size signals and scale them to produce a specified range of magnitudes and frequencies for d over which the design must insure good performance.

Figure 11.16 illustrates this role of disturbance weights. Inputs to the weights are signals whose frequency responses are flat and unit-size. The weights themselves contain scale factors and frequency shaping that match the size, units and frequency content of the true inputs that the system is expected to see. Typically we have only two categories of disturbance weights. The first category consists of simple constants that are used to model wide-band signals such as sensor noise. The constants simply scale the noise to its correct magnitude. In most flight control designs, sensor noises are small and do not impact performance significantly. Thus, it is tempting to set their weights to zero or omit them from the interconnection structure entirely. However, certain technical conditions of the H_∞ solution usually make small nonzero weights mandatory.

The second category of disturbance weights consists of low-pass filters that are used to model band-limited signals such as wind gusts and pilot or outer-loop commands. Typically these weights are first-order transfer fu.ctions with gains selected to produce the correct signal-levels and time constants selected to match the bandwidth of the signals in question. For example, we might scale a wind gust model to produce 6 ft/sec gusts over a bandwidth of V/L rad/sec, where V is airspeed and L is the gust scale length at the current altitude. (This mimics the popular Dryden gust spectrum.) Similarly, we might scale pilot commands to produce full-stick commands over a bandwidth of 1–2 rad/sec.

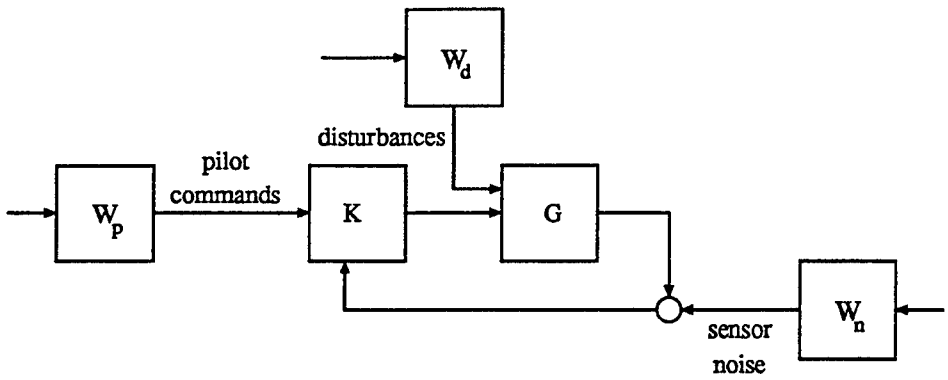


Figure 11.16: Weighted Input Signals.

11.2.5 Uncertainty Weights

The final set of weighting functions to be specified are uncertainty weights. These transform the normalized unit-size perturbations described earlier into perturbations whose magnitudes and frequency content match uncertainty levels in the design model. Recall that perturbations consist of several elements: input uncertainty, output uncertainty, and various real-parameter errors. We must specify weights for each one.

11.2.5.1 Weights for Uncertainty at the Input

As shown earlier in Figure 11.9, uncertainties at the plant inputs are represented by a complex-valued multiplicative perturbation block, Δ_{input}. With the block itself taken to be flat over frequency and unit-size, we specify the actual size of the perturbations by scaling the signals going into the block, as shown in Figure 11.17. As with the other weights discussed above, the weight here is a diagonal transfer-function matrix with one scalar weighting function for each signal.

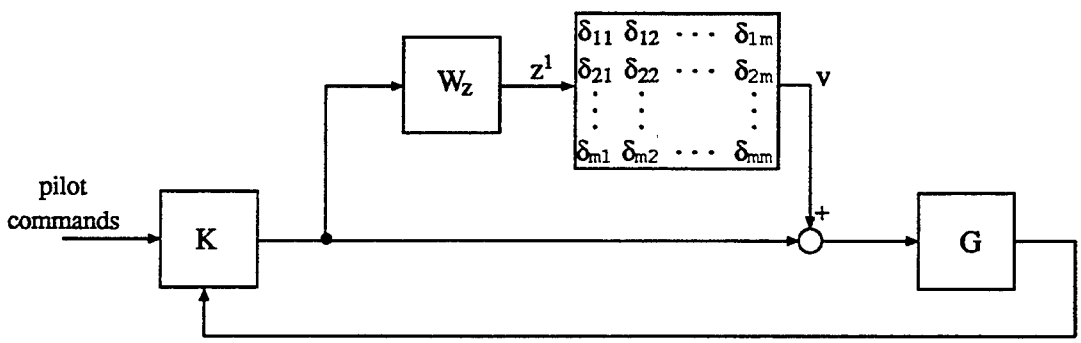


Figure 11.17: Weighted Uncertainty at Input.

The size and shapes of scalar weights across frequency are dictated by the relative level of

uncertainty in the plant model. Weights are large in frequency ranges where the model is very uncertain and are smaller where the model is well known. For most aircraft flight control design work, models are reasonably well known out to the short-period and dutch-roll frequencies. Beyond that, they become progressively less reliable. Uncertainty weights must reflect this, specifying perhaps 50% uncertainty in the short-period range and rising well above 100% at higher frequencies. Many longitudinal designs begin with the short-period approximation, neglecting phugoid and spiral dynamics. In these cases, the weights should also reflect increasing uncertainty at low frequencies. Typical shapes for the resulting uncertainty weights are shown in Figure 11.18. The plot on the the left is representative of models with small low-frequency uncertainty and increasing uncertainty at high frequency, whereas the plot on the right is representative of models based on short-period approximations.

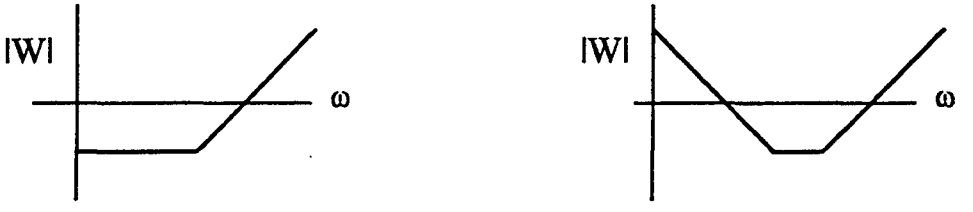


Figure 11.18: Example Input Uncertainty Weights.

Because input uncertainties use multiplicative representations, actual magnitudes of the weights are easily determined. The magnitude at any frequency corresponds to the fraction (relative to the nominal model) of possible model error at that frequency: 0.50 corresponds to errors as large as 50%, 1.00 corresponds to 100%, 2.00 to 200%, and so on. Because perturbations are complex-valued, each level describes possible errors in gain, phase, or both simultaneously. In addition, because the perturbations are unstructured, possible errors also include cross-coupling between channels and various other multivariable interactions.

As a rule of thumb, it is common to assume 50% uncertainty throughout the control bandwidth and to let this level rise with a +1 slope (on a log-log scale) outside that region. This ensures reasonable classical stability margins (6 dB, 30 deg for SISO loops) and adequate rolloff beyond crossover. It is also common to use low-order transfer functions with simple poles and zeros to achieve the frequency-domain shaping of the uncertainty weights. Higher order transfer functions are not usually necessary and lead to higher order final compensators.

It should be pointed out that the weights shown in Figure 11.18 are unrealizable as state-space systems in P because each has an excess of zeros over poles (i.e., their transfer functions are improper). Practical implementations require additional poles at high frequencies to make the transfer functions proper or, alternatively, must utilize existing dynamics already in P to achieve the same end. As an example of the latter approach to weight realization, the following "trick" shows how existing actuator models can be used

to accommodate a zero-pole excess.

Trick: Suppose the desired uncertainty weight ramps up with a +1 slope, and a first-order actuator already exists in the design model. Then, an uncertainty block can be placed around the actuator, rather than ahead of it, to realize the weight. Figure 11.19 illustrates this using two equivalent block diagrams. Note that W in these diagrams can be as simple as a constant, if the uncertainty zero happens to be the same as the actuator pole, or a (proper) lead-lag, $W = (s+z)/(s+a)$, that places the zero at any desired location.

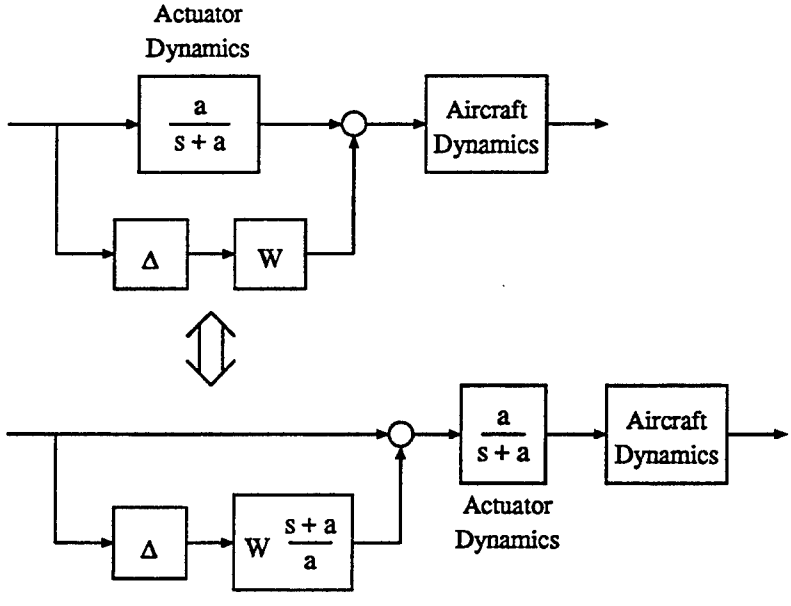


Figure 11.19: Using Actuator Model to Realize Improper Weights.

11.2.5.2 Weights for Uncertainty at the Output

Uncertainty at the output is used to model uncertain gain and phase characteristics of sensors. Again, the magnitudes and frequency response shapes of these weights should express the relative size of possible errors to the nominal sensor model. Usually, constant weights suffice here, but we can sometimes justify frequency-dependent ones to represent specific known sensor errors. The weights then look very similar to those shown in Figure 11.18.

11.2.5.3 Real-Parameter Uncertainty Weighting

Because true uncertainty levels for real parameters were already incorporated as scale factors in the design model when we added additional inputs and outputs in Section

11.2.3, additional weights are usually unnecessary here. Unsteady aerodynamic effects can, in theory, motivate additional frequency-dependent shaping. But these effects usually occur well outside the control bandwidth and are, if necessary, better handled as additional dynamics in the nominal model (e.g., Kussner/Wagner lags, as discussed in Section 4.1.2.2).

11.2.6 Full Interconnection Structure

Figure 11.20 shows an interconnection structure that combines all of the intermediate steps described in the previous subsections. Note that this figure is just one instance of the general feedback diagram in Figure 11.1. The large perturbation block Δ in Figure 11.1 consists of all the individual Δ_x blocks of Figure 11.20 arranged in block-diagonal fashion; that is,

$$\Delta = \begin{bmatrix} \Delta_{input} & 0 & 0 \\ 0 & \Delta_{output} & 0 \\ 0 & 0 & \Delta_{param} \end{bmatrix} \tag{11.11}$$

Correspondingly, Δ's inputs z are all individual block inputs collected as a single vector, and outputs v are all individual block outputs collected similarly. Control compensator K in Figure 11.1 corresponds to the *linear controller* block in 11.20, with inputs collected as y and outputs collected as u. The rest of Figure 11.20 corresponds to P, with all inputs collected as d and all outputs collected as e.

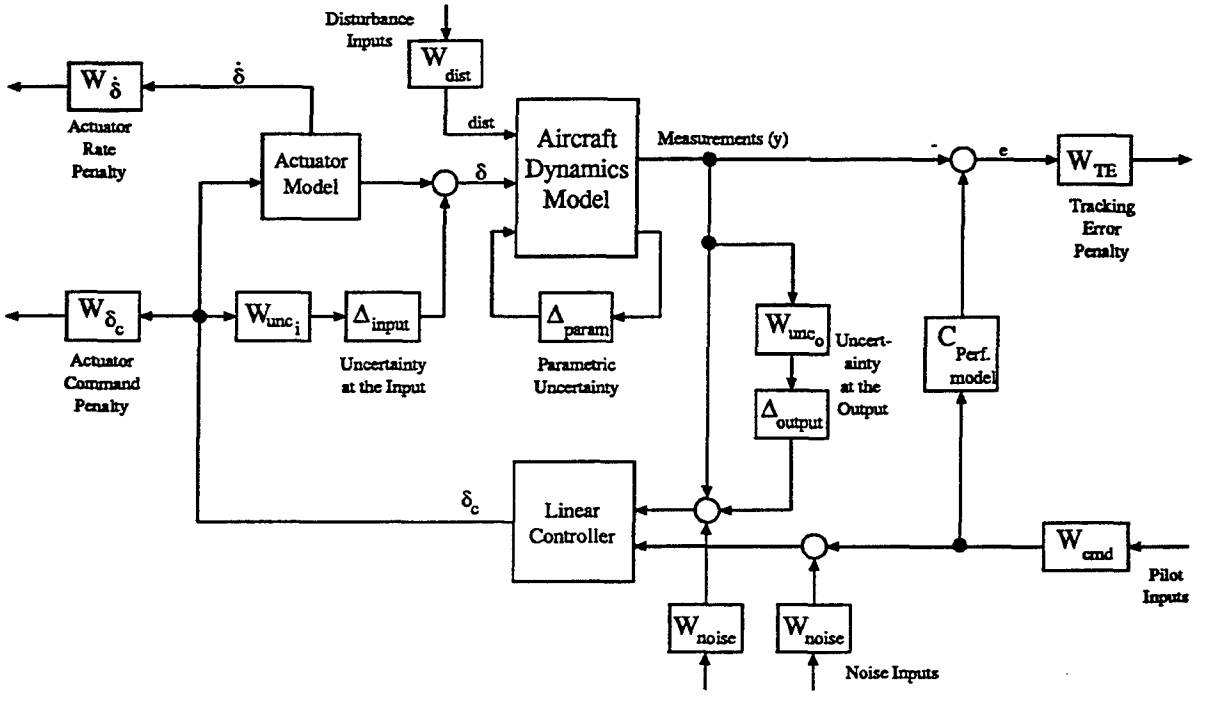


Figure 11.20: Full Interconnection Structure Example.

11.3 Closed-Loop Response

Given a (normalized) interconnection structure such as that in Figure 11.20, our design problem is to find a stabilizing compensator K that makes the maximum singular value of the closed-loop frequency-response matrix from d to e less than unity for all possible unit-size perturbations Δ with the defined block structure. Such a compensator guarantees that performance specifications are robustly satisfied for all models in the uncertainty set. The theory of μ-analysis tells us that this problem has an equivalent alternative interpretation, namely to find a stabilizing K that makes the structured singular value of a larger matrix less than unity. The larger matrix is the closed-loop frequency response from (v, d) to (z, e) with Δ removed. To date, closed-form solutions do not exist for either interpretation of this synthesis problem. However, the second interpretation leads to an approximate iterative solution that works quite effectively. This iterative scheme uses repeated H_∞-solutions with rescalings of all signal sets after each iteration. Details of these steps will be described shortly. For each step, we require the closed-loop version of the interconnection structure, which we describe next.

Consider the general feedback diagram from Figure 11.1 with Δ removed. Then the closed-loop response from (v, d) to (z, e) can be written as

$$\begin{bmatrix} z \\ e \end{bmatrix} = M \begin{bmatrix} v \\ d \end{bmatrix} \tag{11.12}$$

where M is the transfer function consisting of the K feedback loop closed around P, as shown in Figure 11.21. Note that M depends on both P and K.

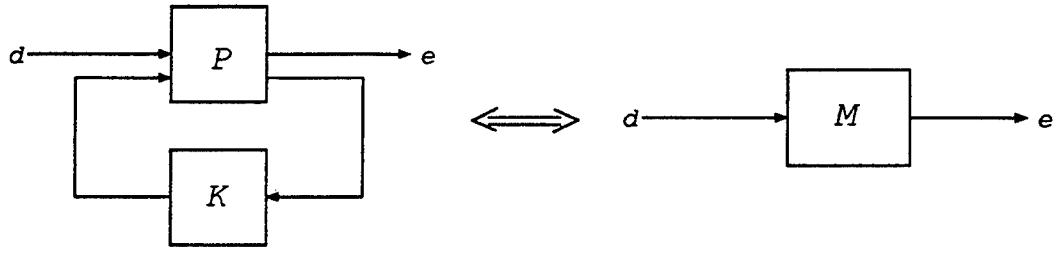


Figure 11.21: General Feedback-Loop Block Diagram.

In terms of block-partitioned transfer functions, we next derive the functional form for M. Let

$$P = \left[\begin{array}{cc|c} P_{zv} & P_{zd} & P_{zu} \\ P_{ev} & P_{ed} & P_{eu} \\ \hline P_{yv} & P_{yd} & P_{yu} \end{array}\right] \tag{11.13}$$

where the partitions divide P into two signal sets at both the input and output such that

$$\begin{bmatrix} z \\ e \end{bmatrix} = \begin{bmatrix} P_{zv} & P_{zd} \\ P_{ev} & P_{ed} \end{bmatrix} \begin{bmatrix} v \\ d \end{bmatrix} + \begin{bmatrix} P_{zu} \\ P_{eu} \end{bmatrix} u \tag{11.14}$$

$$y = \begin{bmatrix} P_{yv} & P_{yd} \end{bmatrix} \begin{bmatrix} v \\ d \end{bmatrix} + P_{yu}u \tag{11.15}$$

Substituting $u = Ky$ into these expressions and eliminating y gives the closed-loop response:

$$M = \begin{bmatrix} M_{zv} & M_{zd} \\ M_{ev} & M_{ed} \end{bmatrix} = \begin{bmatrix} P_{zv} & P_{zd} \\ P_{ev} & P_{ed} \end{bmatrix} + \begin{bmatrix} P_{zu} \\ P_{eu} \end{bmatrix} K(I - P_{yu}K)^{-1} \begin{bmatrix} P_{yv} & P_{yd} \end{bmatrix} \tag{11.16}$$

11.4 H_∞ Synthesis

As mentioned above, the problem of finding compensators that make the structured singular value of M less than unity can be solved by repeated H_∞-solutions alternated with rescalings of the signal sets. The H_∞ part of these iterations use the well-developed theory of H_∞-optimal control (e.g., [Doyle 1989]). This theory provides compensators that minimize the H_∞-norm of M (i.e., minimize the *maximum* singular value, rather than the *structured* singular value, of M over frequency). The required calculations consist of two Riccati equations solved iteratively in a one-dimensional search over a parameter γ that corresponds to the currently achieved value of the H_∞-norm. The objective of the search is to make γ as small as possible. The Riccati equation solutions as well as the γ-search can be accomplished using any of several available software packages.

The input data required for H_∞ synthesis is a state-space representation of P corresponding to the transfer function above:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|ccc} A & B_v & B_d & B_u \\ \hline C_z & D_{zv} & D_{zd} & D_{zu} \\ C_e & D_{ev} & D_{ed} & D_{eu} \\ C_y & D_{yv} & D_{yd} & D_{yu} \end{array}\right] \tag{11.17}$$

where the partitions separate states x from the input signal set (v, d, u) and state derivatives $\dot{x}$ from the output signal set (z, e, y).

Special rank conditions on the state space D matrix must be satisfied for solutions to exist. These conditions can be interpreted as follows:

1. Full rank $D_{zu}^T D_{zu} + D_{eu}^T D_{eu}$ means that control actuation at high frequencies is penalized in either z or e.

2. Full rank $D_{yv} D_{yv}^T + D_{yd} D_{yd}^T$ means that sensors see high-frequency disturbances due to either v or d.

It is also necessary to provide an admissible starting value for the search parameter γ such that

$$\gamma_{opt} = \inf_K \|M\|_\infty < \gamma \tag{11.18}$$

where γ_{opt} is the (initially unknown) smallest achievable value for the H_∞ norm of M over all possible stabilizing K's. For starting values of γ greater than γ_{opt}, we know that the first K from H_∞-synthesis will produce an M with $\|M\|_\infty \leq \gamma$ [Doyle 1989]. We can then select progressively smaller values and repeat the synthesis until no further reductions are possible. If our starting value or any of the intermediate search values fall below γ_{opt}, the H_∞-solution fails to satisfy certain additional eigenvalue conditions [Doyle 1989]. Most software packages test these conditions, report the failure, and accept larger γ values to try again. Some software products even automate the search entirely, usually with bisection methods, requiring only stopping tolerances and (perhaps) search intervals as user input.

After the γ-search is completed, the final compensator K is supplied in state-space form with $[A, B, C, D]$ matrices calculated directly from the last Riccati solutions. These matrices are then used to construct the closed-loop system M and to compute its frequency-response matrix over a frequency range of interest for subsequent μ-analysis. Although the theory guarantees that M is stable and that its maximum singular value is less than or equal to the final γ, subtle numerical difficulties sometimes arise. Therefore, it is a good idea to verify both properties.

11.5 μ-Analysis and D-Scales

The next step in μ-synthesis is a complete point-by-point μ-analysis of closed-loop frequency response $M(j\omega)$. As described in Section 11.5, this involves calculating M's structured singular values at each frequency point and comparing those values against unity. Recall that structured singular values are always defined with respect to a specific uncertainty structure. The structure in this case is given by

$$\Delta_M = \left[\begin{array}{c|c} \Delta & 0 \\ \hline 0 & \Delta_p \end{array}\right] = \left[\begin{array}{ccc|c} \Delta_{input} & 0 & 0 & 0 \\ 0 & \Delta_{output} & 0 & 0 \\ 0 & 0 & \Delta_{param} & 0 \\ \hline 0 & 0 & 0 & \Delta_p \end{array}\right] \tag{11.19}$$

The first block is the plant's original uncertainty set Δ (itself block-structured), with input-output signals z and v. The second block Δ_p, is the so-called "fictitious" perturbation representing our performance requirements, with input/output signals e and d.

For this uncertainty structure, the condition, $\mu[M] < 1 \; \forall \, \omega$, guarantees that closed-loop system M remains stable when Δ is connected from z to v and Δ_p is connected from e to d, simultaneously. The latter condition means precisely that the d-to-e gain is less than unity for the entire model set, which ensures that performance is robust.

As discussed in Section A.2, the exact value of μ at each frequency is not, in general, easy to calculate. Instead, upper and lower bounds are used to bracket the true value. The

upper bound, in particular, is based on a computationally tractable search over the class of scaling matrices D that commute with perturbation Δ_M. Matrices that commute are matched to the structure of the perturbation in the following manner:

$$D = \left[\begin{array}{ccc|c} d_{input}I_{input} & 0 & 0 & 0 \\ 0 & D_{output} & 0 & 0 \\ 0 & 0 & D_{param} & 0 \\ \hline 0 & 0 & 0 & I_p \end{array}\right] \tag{11.20}$$

Here d_{input} is a scalar multiplying an identity matrix with the same dimension as Δ_{input}, and D_{output} and D_{param} are diagonal matrices with the same dimensions as Δ_{output} and Δ_{param}, respectively. In all, there are $1 + n_{output} + n_{param}$ individual scalar parameters in D.

Because these D's commute with the perturbation, any one of them provides an upper bound for the structured singular value, namely $\bar{\sigma}[DMD^{-1}]$. Taking the smallest of these bounds gives

$$\mu[M] \leq \inf_D \bar{\sigma}[DMD^{-1}] \tag{11.21}$$

The specific D's that achieve the infimum in this equation are called D-scales. They are computed automatically as part of μ-analysis calculations.

11.6 D-K Iteration

Once the μ-analysis calculations are complete, we examine test condition $\mu[M] < 1 \, \forall \omega$. If this condition is satisfied, the current H_∞-compensator, K, meets all robust performance goals. Two options to proceed are possible:

1. Accept the compensator and move to the simplification step of μ-synthesis, or
2. Execute another design iteration. We can repeat this option until one or both of the convergence conditions discussed below are satisfied. Each iteration reduces μ further below unity. This means that the design satisfies tighter performance specifications, $\|e\| \leq \mu$, over larger uncertainty sets, $\|\Delta\| \leq \mu^{-1}$.

If the test condition fails, on the other hand, two other options are possible:

1. The iteration process has converged, which is indicated by an approximately flat μ-function across frequency and/or by current D-scales that differ insignificantly from the previous iteration. No further improvement can then be expected. Rather, design tradeoffs that relax some specifications in favor of others are necessary, as guided by features of the current M. We discuss such tradeoffs further under the title *M-Analysis* below.

2. The process is not converged. Execute another design iteration

If another iteration is appropriate, it differs from the current one only in the sense that a modified optimization problem is solved in the H_∞ step. This modified problem is a rescaled version of the original one, using the current D-scales as scaling factors.

11.6.1 Rational D-Approximations

Unfortunately, current D-scales are available only pointwise across frequency, so they cannot be used directly to scale the state-space realization of P. Rather, rational approximations, $\hat{D}$, for D must be found, with state-space realizations of their own. These approximations can then be appended to the original interconnection structure to produce the modified form illustrated in Figure 11.22.

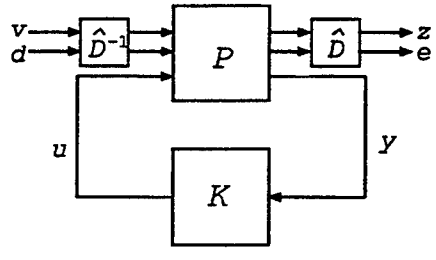


Figure 11.22: Interconnection Structure With Appended D-Scales.

To find the necessary rational approximations, individual diagonal elements of D are interpreted as transfer functions whose Bode magnitude diagrams are given by the computed D-scale values. These transfer functions are further constrained to be stable, minimum phase, and have no singularities on the $j\omega$-axis (including none at $\omega = \infty$). Under these conditions, both $\hat{D}$ and $\hat{D}^{-1}$ can be realized as stable state-space systems.

For simple functions, this constrained-transfer-function approximation step can often be done manually with just a few trial-and-error iterations. Classical designers are particularly skilled at this. Software tools are necessary for more complex functions. These tools use the pointwise computed D-scales as input data as well as information about the desired approximation order and (sometimes) fit accuracy as a function of frequency. The latter is useful if the upper bound is very sensitive to D-scales in certain frequency ranges and less sensitive in others.

11.6.2 Properties of D-K Iteration

From a theoretical perspective, the iteration process described above is only an approximate method for solving the true μ-minimization problem that exploits the upper bound:

$$\inf_{K} \mu[M(P,K)] \leq \inf_{K,D} \bar{\sigma}[DM(P,K)D^{-1}] \tag{11.22}$$

D-K iterations solve the right-hand optimization problem sequentially by first minimizing over K with D fixed, then minimizing over D with K fixed, and so on. It can be shown that this process improves performance monotonically with each iteration, assuming that the rational D-scales are approximated perfectly. Beyond that, few "nice" properties apply. In particular, the process is not guaranteed to find global optimal solutions because the function $\bar{\sigma}[DM(P,K)D^{-1}]$ is not jointly convex in D and K. Thus, it can get stuck at local minima.

It addition, the iterations become progressively less well-conditioned numerically as they converge because the final μ-optimal solution, which is flat across frequency, requires proper compensators (with non-zero state-space D matrices). The Riccati solvers at each iteration, however, produce strictly proper compensators. The necessary D matrices are approximated by wide-band components in K, whose poles tend toward infinity as the iterations proceed. Eventually, the growing pole spread overpowers the numerical capabilities of the Riccati solvers, and we must ultimately terminate the iterations before convergence is complete.

Despite these limitations, however, design experiences with aircraft flight control and other applications show that the process is effective in synthesizing control laws for difficult design situations with tight robust performance requirements.

11.6.3 M-Analysis

Even though the test condition, $\mu[M] < 1 \; \forall \; \omega$, is the principal property of interest for the closed-loop frequency response, $M(j\omega)$, much additional is information embedded in this matrix. This information helps in understanding the current design and is essential in directing performance tradeoffs and compromises in cases where the converged test condition fails.

Further information can be extracted by *decomposing* M into various constituents to determine why it is large in certain frequency ranges. One of the most basic decompositions looks at the contributions of *robust stability* and *nominal performance* requirements on the overall *robust performance* goal. We accomplish this by comparing plots of $\mu[M_{zv}]$ and $\bar{\sigma}[M_{ed}]$ with that of $\mu[M]$, which bounds both of these functions. In frequency ranges where $\mu[M_{zv}] \cong \mu[M]$, the robust stability requirement is the main contributor to the overall goal and, if $\mu[M] > 1$, the uncertainty set must be made smaller. On the other hand, in frequency ranges where $\mu[M_{ed}] \cong \mu[M]$, nominal performance is the main contributor and, if $\mu[M] > 1$, performance requirements must be relaxed (i.e., accept larger

e). At some frequencies, of course, both constituents may contribute significantly to $\mu[M]$ and relaxation of either one will effect the design.

Other contributions to $\mu[M]$ can be assessed with other plots. For example, the contribution of a particular disturbance or group of disturbances can be assessed by selecting out the appropriate columns of M and plotting the maximum singular value of the matrix formed from these columns. Likewise the contribution of a particular tracking error or group of errors can be assessed by selecting out the appropriate rows of M and plotting the maximum singular value of the matrix formed from these rows.

One of the rank conditions on D discussed in Section 11.4 is sometimes satisfied by introducing sensor noise into d. It is then desirable to know whether this noise is in fact negligible in its contributions to the overall design goal. This can be verified by comparing a plot of the maximum singular value of the columns of M corresponding to the sensor noise elements of d with that for $\mu[M]$. If $\bar{\sigma}[M_{sensor\ noise}] \ll \mu[M]$, then the noise makes no significant contribution to the overall design.

These same types of comparisons can be made for other rows or columns of M. For example, the penalty associated with control actuation can be assessed by comparing a plot of the maximum singular value of the rows of M corresponding to the control penalty elements of e with that for that for $\mu[M]$. If $\bar{\sigma}[M_{control\ penalty}] \ll \mu[M]$, then the control penalty makes insignificant contributions to the overall design. In many cases, the contribution of the control penalty will be frequency dependent (e.g., actuation rate penalties may be significant at high frequencies but not at low frequencies).

In yet other analyses, subsets of input/output signals can be deleted from M and the μ-function of the resulting smaller matrix can be compared to the original $\mu[M]$. This shows whether the deleted element (i.e., a specific model perturbation or performance signal) is significant to the overall design. In this way, for example, the effects of uncertain stability derivatives can be studied separately from the effects of high-frequency unmodeled dynamics. Again, the idea is to find a smaller problem where μ for the smaller problem agrees closely with $\mu[M]$.

11.7 Compensator-Order Reduction

As noted above, as D-K-iterations converge, the maximum singular value of $M(j\omega)$ tends toward a function that is flat across frequency, and some poles of the compensator tend toward infinity. These poles prove troublesome in eventual digital implementations of the compensator, so is desirable to remove them. The residualization procedures of Section B.1.2.2 applied to a block-diagonal realization of the compensator, as discussed in Section B.1.3, usually work well for this purpose.

Still further order reduction is often possible because the compensators also include many near pole-zero cancellations. Truncated frequency-weighted balanced realizations

[Enns 1984], Hankel model reduction [Glover 1984], or frequency-weighted Hankel model reduction [Latham 1986, Anderson 1986, Khou 1993] are useful in removing these dynamics.

Of course, after all such simplification steps, the closed-loop system M should be reconstructed using the simplified K, and robust performance reverified.

11.8 Longitudinal Example

11.8.1 Short-Period Design

Very often, longitudinal control designs begin with a two-state aircraft model that represents short-period dynamics. For symmetric aircraft, if a control variable has been chosen and control effectors slaved together, as suggested in earlier sections, the control design problem is essentially single-input/single-output. For the sake of this example, we did not select a controlled variable *a priori*, but rather discuss the mu-synthesis methodology using pitch-rate and normal-accelerometer measurements. However, the multiple controls are slaved together. The aircraft is a version of the F-16 and is briefly discussed below.

11.8.1.1 Interconnection Structure

The interconnection structure for this example appears in Figure 11.23 and is very similar to the generic interconnection structure of Figure 11.20. Three exceptions are the omission of actuator command/rate penalties, noise inputs, and real-parameter uncertainty.

11.8.1.2 Aircraft Model

The aircraft model is for an F-16 modified to add thrust vectoring and vertical canards. The thrust vectoring, along with the existing horizontal tails, make up the pitch-axis control effectors. The flight condition for this example is straight and level at Mach 0.8 and 10,000 ft altitude, with a trim angle of attack of 1.6 deg. At this flight condition, the short-period dynamics are unstable with the unstable pole at 1.25 rad/sec.

The two control effectors are slaved together using the pseudo-inverse method described earlier. A pitch-rate (q) gyro and normal accelerometer (nz) provide sensed quantities used by the controller.

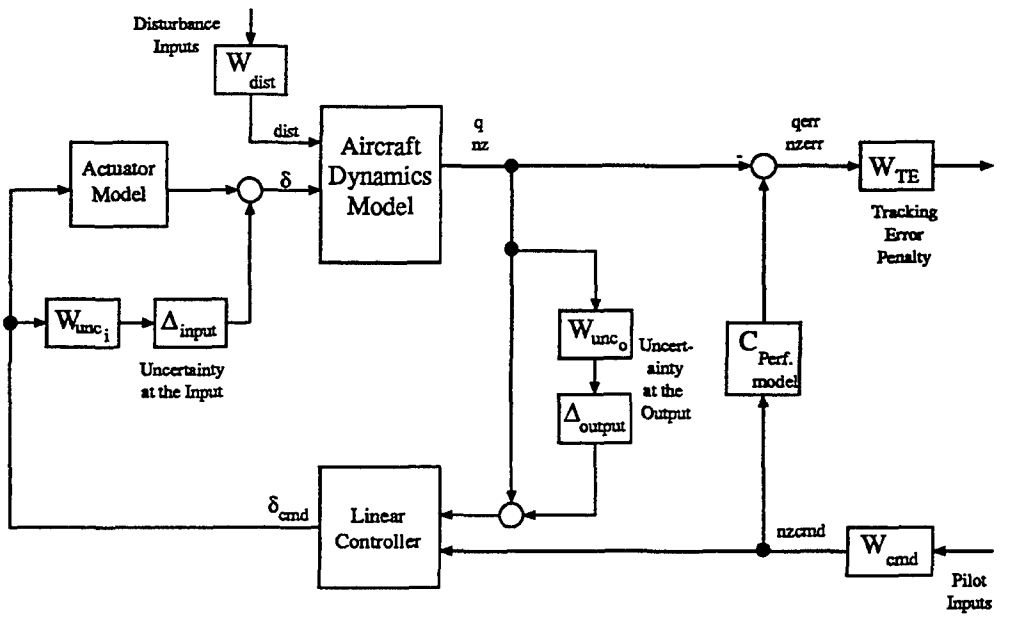


Figure 11.23: Interconnection Structure for Short-Period Design.

11.8.1.3 Performance Model

For this example, optimal performance is measured in terms of model tracking errors shown in Figure 11.5. The pilot input is longitudinal stick and the two signals that make up the error vector are pitch rate and normal acceleration.

11.8.1.4 Weights

W_{cmd}: This weight takes a unit-norm signal as input and produces a signal with size and frequency content consistent with pilot commands.

For the closed-loop system, we chose a normal-acceleration command system, as discussed in Section 11.8.1.3. That is, the pilot stick commands nz at low frequency, and we select the weight

$$W_{cmd} = 0.001\frac{s+100}{s+0.1}$$

Because the DC gain of W_{cmd} is unity, unit-sized signals are translated into unit-sized nz commands, which gives reasonable 1-g load-factor commands. The pole at 0.1 rad/sec indicates that pilot inputs diminish at frequencies above 0.1 rad/sec. Above 100 rad/sec, an acceptable 0.001-g nz command results from our choice of poles, zeros, and DC gain.

W_{TE}: This function weights errors formed by subtraction of the outputs of the closed-loop system from those of the performance model. In our case, the two outputs are q and nz so that W_{TE} is a 2-by-2 diagonal matrix transfer function. Whenever there is more than one performance signal, we must take care to ensure that the relative weighting

functions are appropriate. The selected weights are

$$W_{TE-nz} = \frac{300s}{(s+0.1)(s+10)}$$

$$W_{TE-q} = \frac{V}{g}\frac{z}{(s+z)}W_{TE-nz}$$

The weight for the nz error has a magnitude of 30 between 0.1 and 10 rad/sec. This frequency range is commensurate with the MIL-STD-1797 short-period specification. The magnitude of 30 implies that errors as large as $1/30 = 0.033$ g are acceptable in that frequency range. Letting magnitude roll off at high and low frequencies indicates that larger errors are acceptable in those frequency ranges.

We express the weight for q error relative to that for nz error and include the factor V/g to make the low-frequency scaling between q and nz similar because $nz = qV/g$ (for nz in g's) at low frequencies. The lag at z rad/sec is included to negate the effects of the open-loop zero in the pitch-rate transfer function. If only the V/g scaling were included, then, in the frequency range above the pitch-rate zero, pitch-rate errors would be more heavily penalized than the nz errors. This is due to the magnitude rise associated with the open-loop pitch-rate zero.

W_{dist}: This weight scales unit-norm signals to have proper size and frequency content for disturbance inputs. For this example, the disturbance is a normal gust, with a weight of

$$W_{dist} = \frac{4}{s+0.2}$$

This transfer function is a crude first-order approximation to the second-order filter associated with the Dryden gust model (see Section 4.4.1) using $L_w = 1750$ ft, $V = 400$ ft/sec, $\sigma_w = 10$ ft/sec.

W_{unc_i}: This weight describes model uncertainty at the design model input (i.e., controller output). For this example, the single input is the composite pitch effector (slaved elevator and thrust vectoring), with a weight of

$$W_{unc_i} = 2\frac{s+2}{s+20}$$

This choice indicates that at frequencies below 2 rad/sec, we expect at most $4/20 = 0.2$ or 20% model error. For frequencies above 2 rad/sec, the model uncertainty grows until the pole at 20 rad/sec, which is needed to make the weighting function realizable. It also sets the expected uncertainty level at high frequency to be 2.

Short-period models are generally reliable out to between 5 and 10 rad/sec. But, we chose our weighting to indicate that the model loses fidelity beyond 2 rad/sec because we use this same weight to modulate closed-loop bandwidth. By increasing model uncertainty beyond 2 rad/sec, the optimization keeps bandwidth roughly between 2 and 10 rad/sec. Note that 10 rad/sec is the frequency at which model uncertainty exceeds unity.

W_{unc_o}: This weight describes model uncertainty at the design model output (i.e., controller input). We use these functions to guide blending of the two sensed quantities. The two outputs for this example are q and nz, and the associated weights are

$$W_{unc_o-nz} = \frac{s+2}{s+20}$$

$$W_{unc_o-q} = 0.00001$$

The nz weight indicates that the accelerometer is good to within $2/20 = 0.1$ (or 10%) at frequencies below 2 rad/sec. Allowable accelerometer errors grow to 1 (or 100%) beyond 20 rad/sec. Actual accelerometer bandwidth is greater than 2 rad/sec. By increasing model error beyond 2 rad/sec, we are requiring the μ-synthesis design to avoid using accelerometer beyond 2 rad/sec. (It is, of course, common practice to rolloff accelerometers at 2–4 rad/sec.)

The weight on the pitch-rate gyro indicates that it is accurate to 0.001% for all frequencies. The actual sensor has limited bandwidth, albeit high, and its accuracy is somewhat better than the modeled 0.001%.

11.8.1.5 *D-K* Iteration

We obtain the first H_∞ controller K with all D-scales set equal to unity. The maximum singular value of the frequency response of the resulting closed-loop system, $\bar{\sigma}[M]$, is about 43, whereas the maximum value of $\mu[M]$, assuming diagonal uncertainties and a full Δ_p block for performance, is about 14. This indicates that D-scales should be useful in further optimizing the design. Figure 11.24 shows $\bar{\sigma}[M]$ along with upper and lower bounds for $\mu[M]$. M-analysis shows that at very low frequencies and frequencies beyond 10 rad/sec, robust stability (M_{zv}) is large and hence is the dominant factor that keeps the design from achieving desired robust performance. Figure 11.25 shows the same $\bar{\sigma}[M]$ plot as in Figure 11.24, as well as the robust stability plot, M_{zv} (dashed), and the nominal performance plot, M_{ed} (dash-dot).

One output of current μ-analysis software is a frequency response plot for D-scales, which must be fit with low-order approximations. State-space approximations for D and D^{-1} are then appended to the interconnection structure and another H_∞-optimal controller computed. For this example, D-scales corresponding to the three uncertainties (one at the input and two at the output) are each fit with (at most) fourth-degree transfer functions.

Once the D-scale approximations and their inverses have been determined, we form a new weighted design model by multiplying each of the uncertainty inputs (to the interconnection structure, P), by the appropriate D^{-1}-scaling. Similarly, each of the uncertainty outputs is also multiplied by the appropriate D-scaling. This new weighted design model is then used to compute the next H_∞-optimal controller.

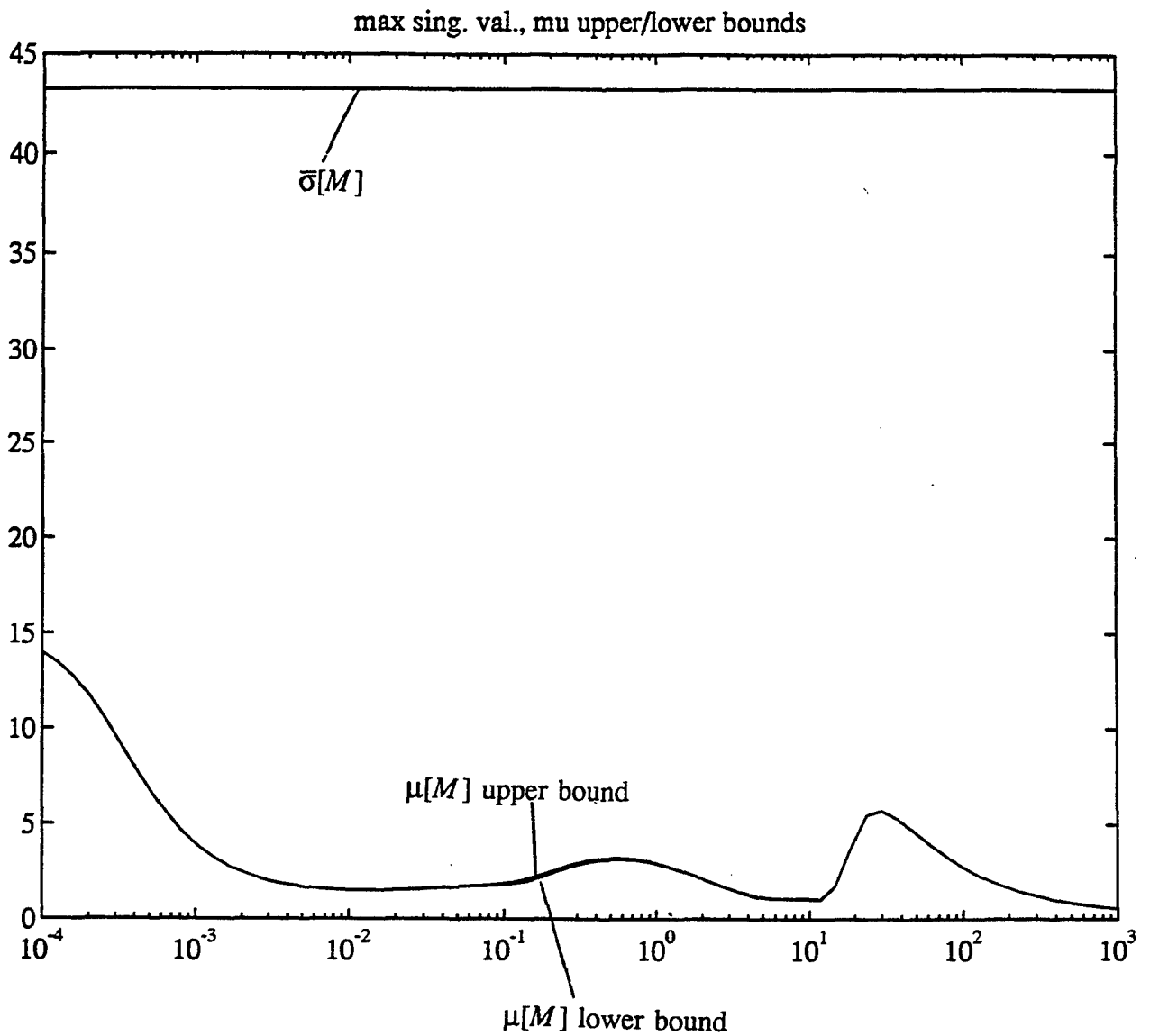


Figure 11.24: Maximum Singular Value and μ Bounds for First Iteration.

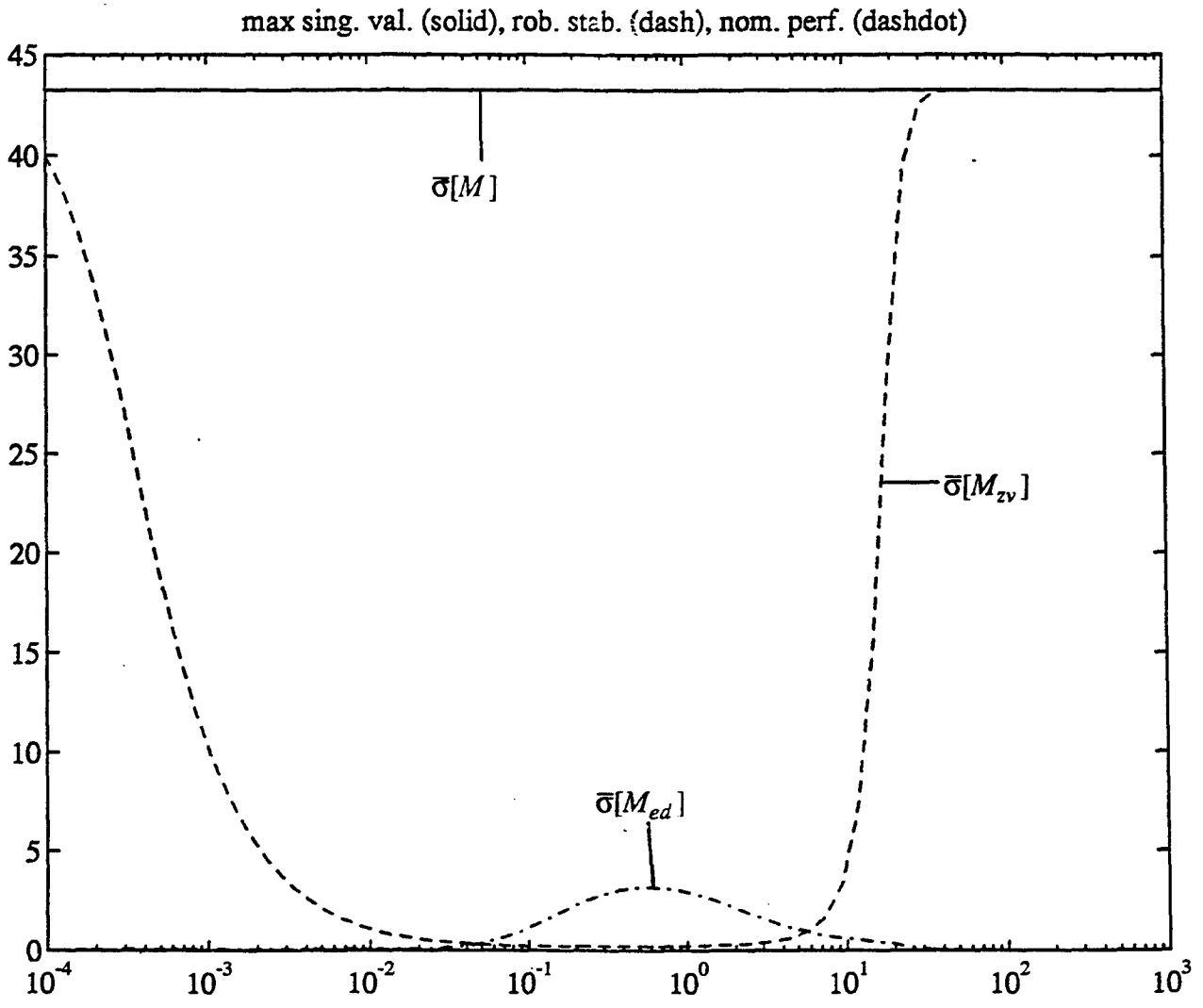


Figure 11.25: M-analysis for First Iteration.

For this example, three iterations were completed before reduction of $\bar{\sigma}[M]$ stopped. At this step, $\bar{\sigma}[M]$ is greater than 2.5, as illustrated in Figure 11.26. This figure also shows bounds for $\mu[M]$. Note that in the frequency range around 1 rad/sec, bounds for μ are nearly identical to the largest singular value (magnitude of about 2.3). This implies that D-scales have accomplished as much as is possible in this region. In fact, further D-scale optimization is unwarranted throughout the entire frequency range. Because we have not yet achieved guaranteed robust performance (i.e., $\mu[M] > 1$), a design trade must be made.

M-analysis shown in Figure 11.27 for this third iteration shows a typical result. At low and high frequencies, model uncertainty errors dominate the problem because the robust stability curve (dashed) is near the $\bar{\sigma}[M]$ curve (solid). In the control bandwidth, nominal performance (dash-dot) requirements drive the problem. Further examination of rows and columns of the frequency response of the nominal performance block, M_{ed}, show that gust input requirements cause nominal performance to be large (which, in turn, causes μ to be large).

At this point, we must relax our gust input specification by changing the weight W_{dist} to

$$W_{dist} = \frac{1}{s + 0.05}$$

This maintains the steady-state size of the gust description but lowers the frequency content between 0.05 and 0.2 rad/sec.

Now we design another H_∞-optimal controller, including the D-scales from the previous iteration in our interconnection structure. We do four more D-K-iterations (again, using at most fourth-order approximations for the D-scales) to arrive at a final design. Figure 11.28 shows $\bar{\sigma}[M]$ and the μ bounds for the last iteration. Note that the μ bounds are now close to $\bar{\sigma}[M]$ for all frequency indicating that further D-K iteration is unnecessary. Also, because μ is less than 1 for all frequency, we guarantee robust performance for those inputs and uncertainties characterized by the weighting functions in the interconnection structure. Finally, Figure 11.29 shows the robust stability and nominal performance parts of M. Note that, although nominal performance still dominates in the control bandwidth frequency region, the peak is much lower than before the design tradeoff.

11.8.1.6 Model Reduction

The controller resulting from the final iteration has 38 states. In light of the original two-state design model, this is clearly impractical.

The method we used to reduce compensator order is as follows. First, the states beyond 100 rad/sec were residualized (see Section B.1.2.2). This removed eight states. We then identified appropriate high-frequency states for residualization by performing a similarity transformation to an eigenvalue form. Figure 11.30 presents frequency-domain comparisons for the three controller inputs (pilot command, q, and nz) to the slaved pitch control

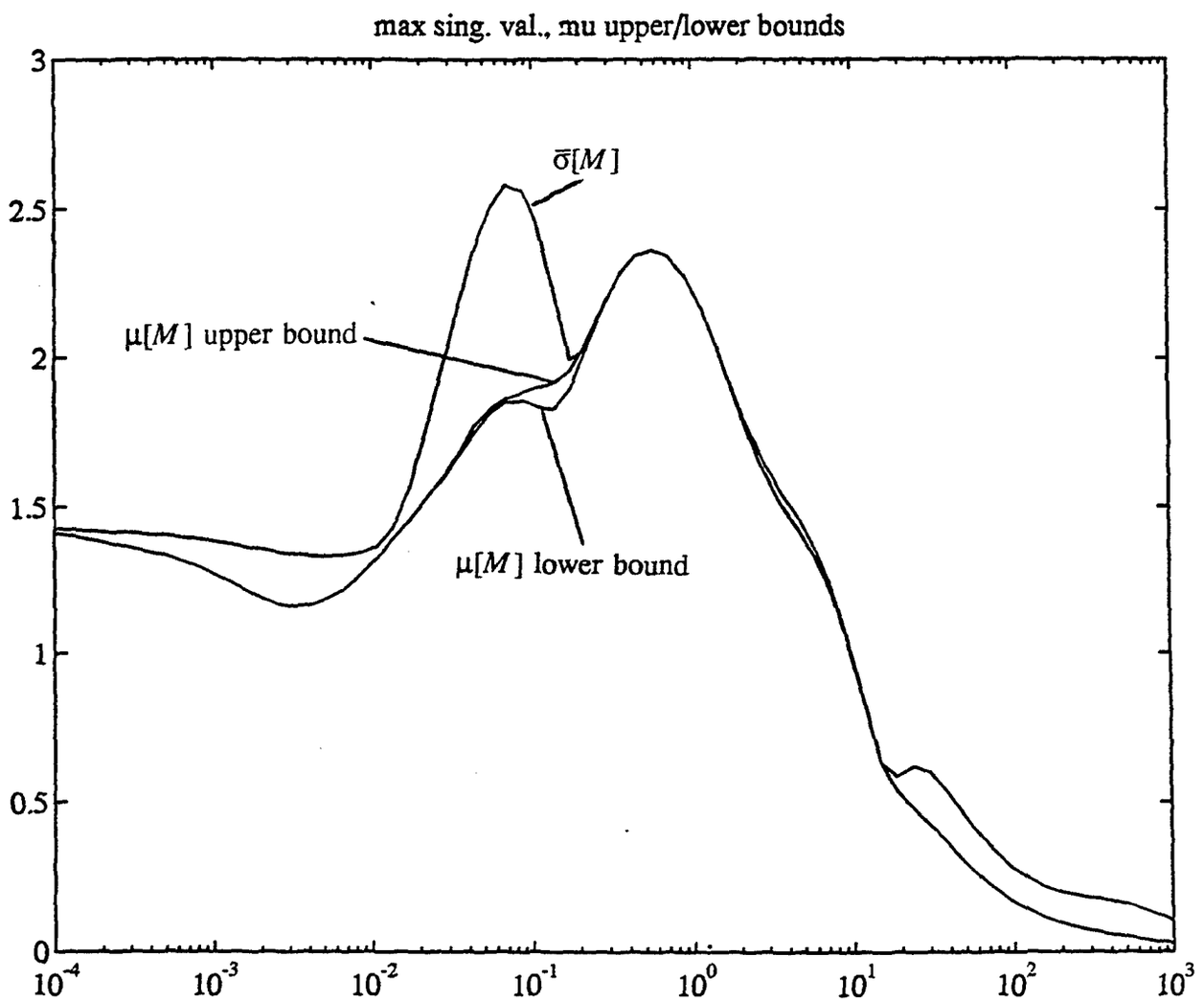


Figure 11.26: Maximum Singular Value and μ Bounds for Third Iteration.

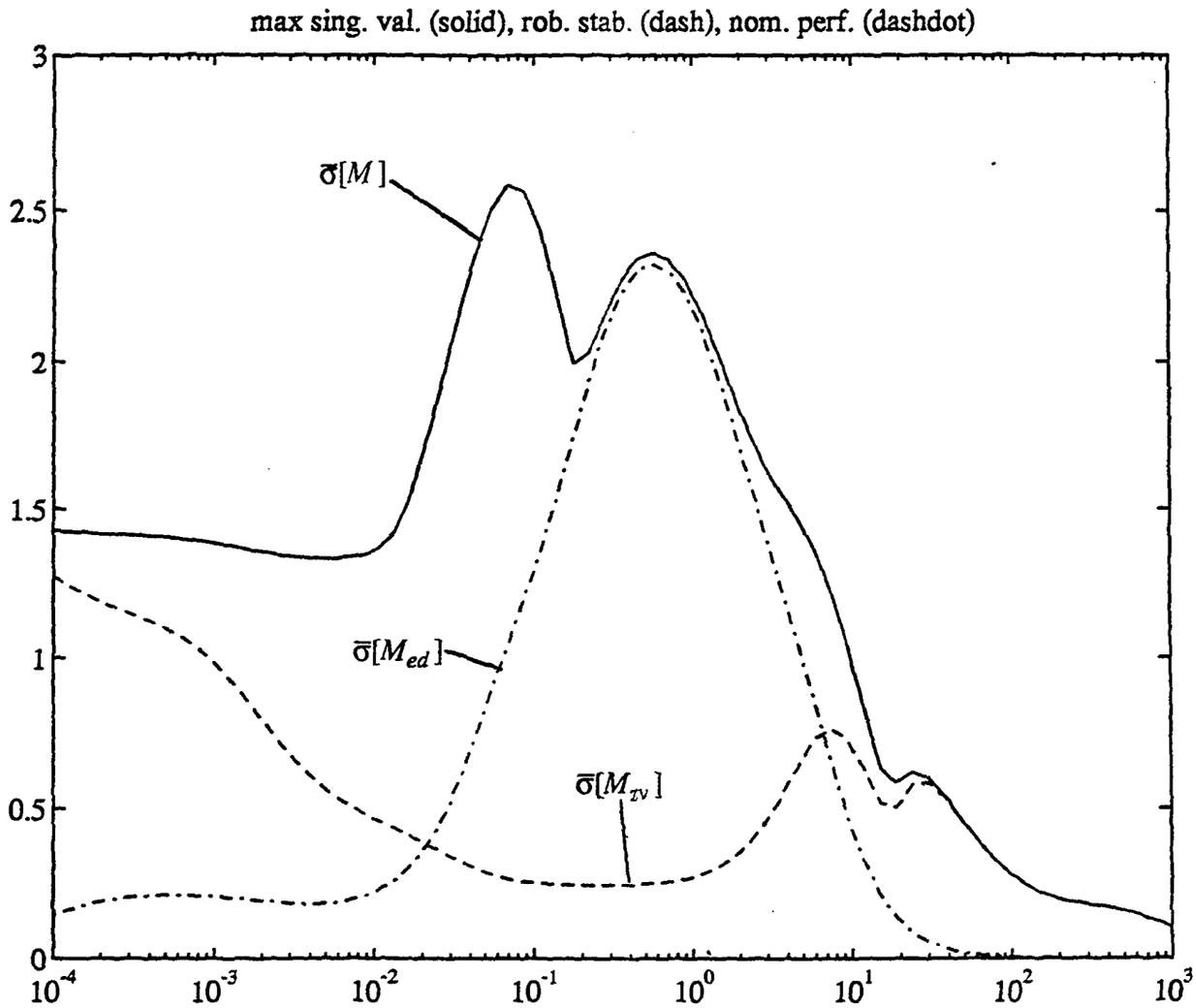


Figure 11.27: M-Analysis for Third Iteration.

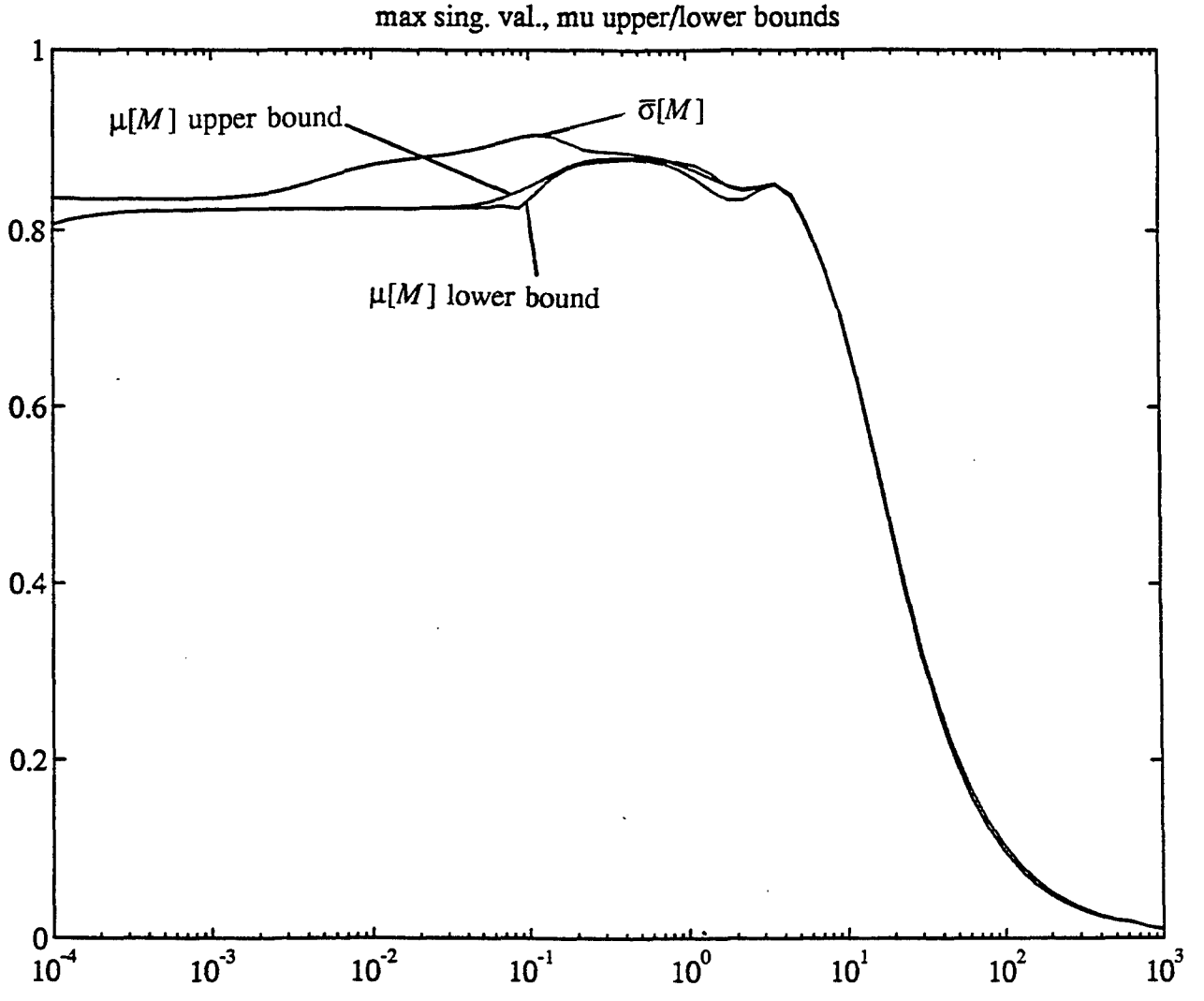


Figure 11.28: Maximum Singular Value and μ Bounds for Final Iteration.

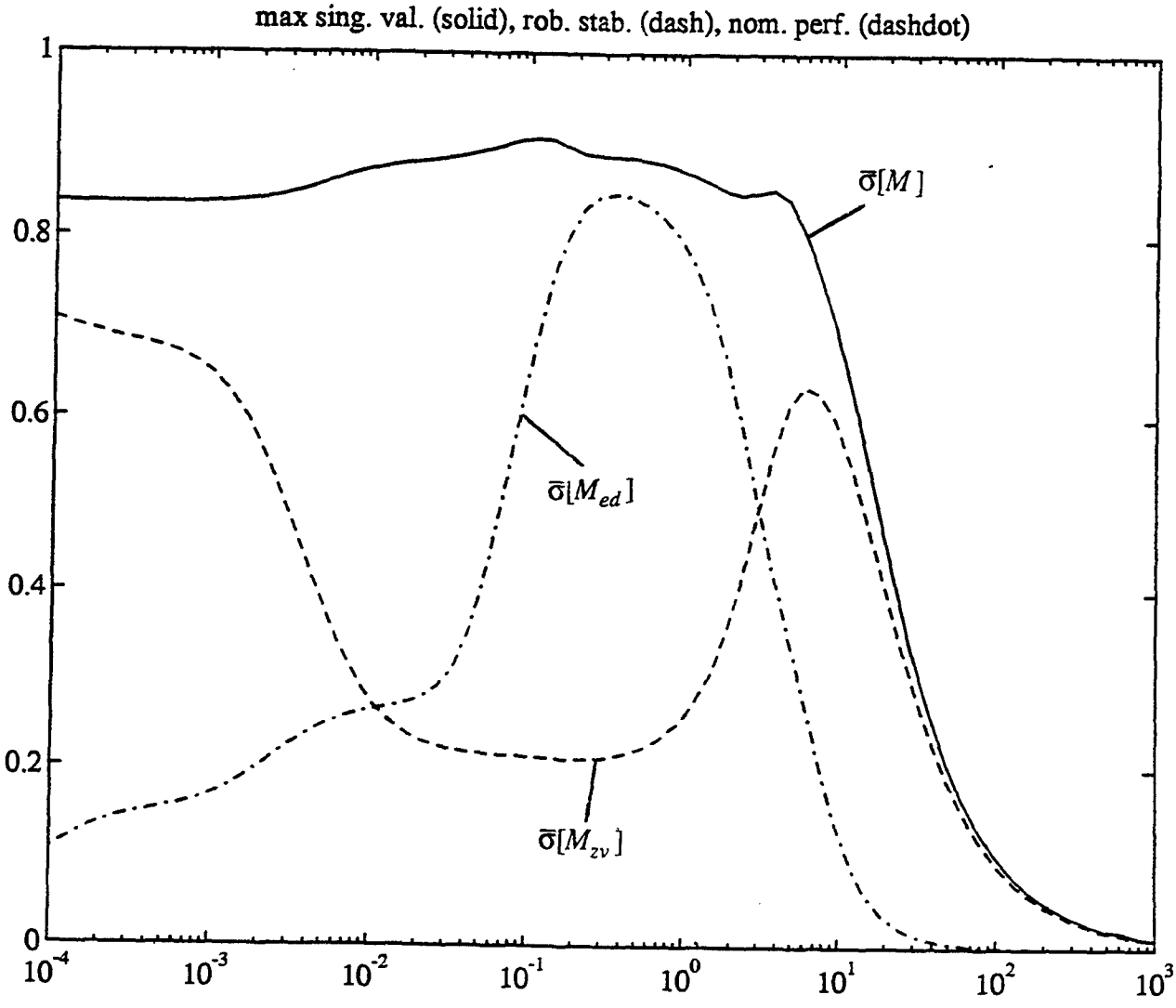


Figure 11.29: M-Analysis for Final Iteration.

for the full-order and residualized controller.

Next, the 30-state controller was transformed to a balanced realization, and we reduced state dimension to 6 by truncating the least significant states. We used no frequency-dependent weighting in computing this realization. This is the final reduced-order controller. Figure 11.31 compares frequency responses for the final 6-state controller with those for the original 38-state controller.

As an aside, we note that balancing the original 38-state model and then truncating did not yield a reasonable model. Presumably, the very high-frequency dynamics cause numerical difficulties. It is generally a good practice to residualize high-frequency modes in a working model before proceeding to the task at hand, be it synthesis, model reduction, or simulation.

11.8.1.7 Analysis

All closed-loop poles that result from using the 6-state controller with the 2-state short-period model were stable and well damped. Conventional single-loop-at-a-time analysis showed nearly 20 dB of gain margin and 70 deg of phase margin in all channels. Finally, we performed linear simulation with a longitudinal-stick step input. Figure 11.32 shows incremental changes from trim for a 1-g commanded step. Shown are q, α, nz at the cg and at the accelerometer and the slaved pitch-effector command. (Ultimately, this slaved effector would be split in some proportion among the three effectors.) All responses look reasonable, although an argument could be made that the response speed is a little slow (by a couple tenths of a second or so).

When the 6-state model was connected to the 4-state longitudinal model (i.e., short-period plus phugoid dynamics), the closed-loop exhibits an unstable root. The time to double for the unstable pole is 12.2 sec, which, according to the MIL-STD-1797, is Level 3 flying qualities. This motivated a full 4-state longitudinal design, which is described next.

11.8.2 Short-Period-Plus-Phugoid Design

In this design, we seek control in the short-period frequency range, as in Section 11.8.1, and closed-loop stability of both short-period and phugoid modes. We include the phugoid in the design phase with the knowledge that H_∞ produces stable closed-loop systems.

11.8.2.1 Interconnection Structure

The interconnection structure for this example appears in Figure 11.33. Note that only the pieces of the problem pertinent to demonstrating phugoid stabilization are included.

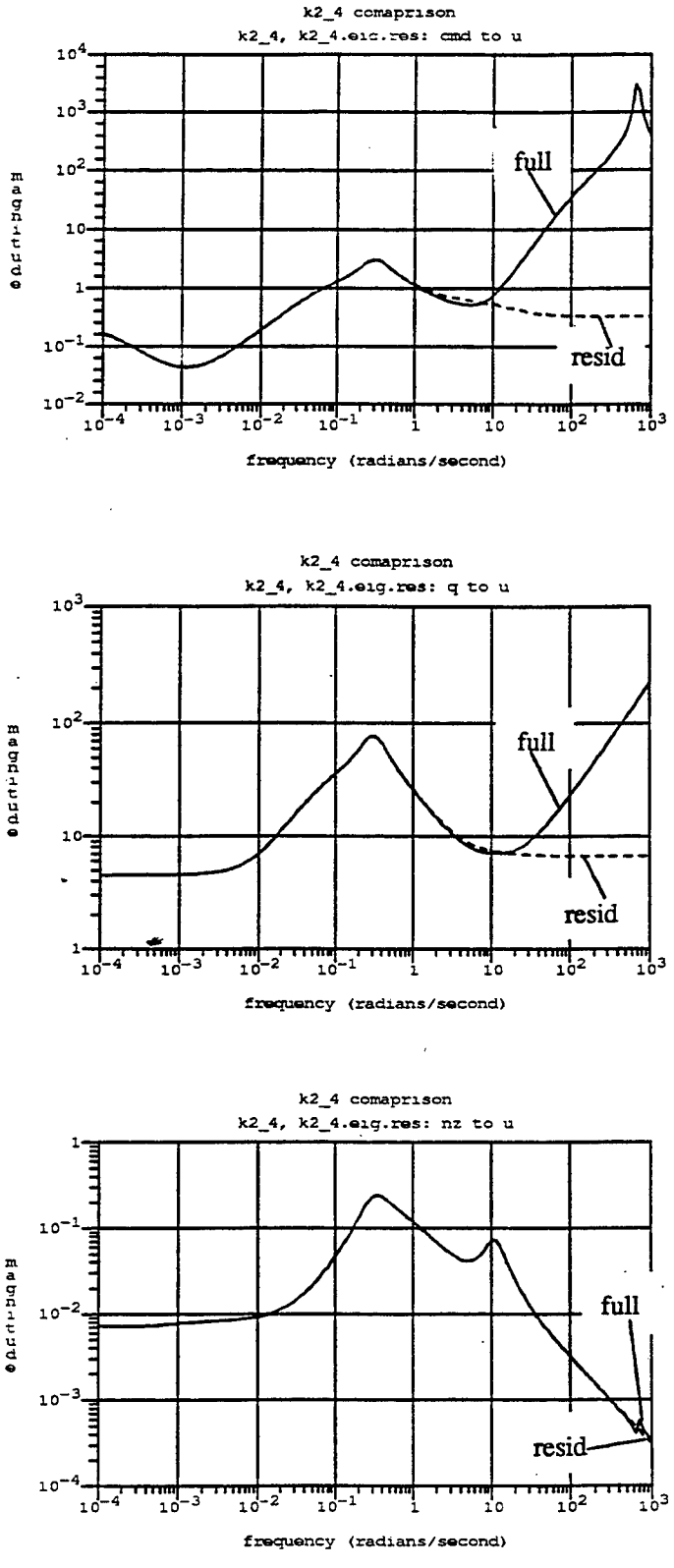


Figure 11.30: Comparison of Full-Order and Residualized Controllers.

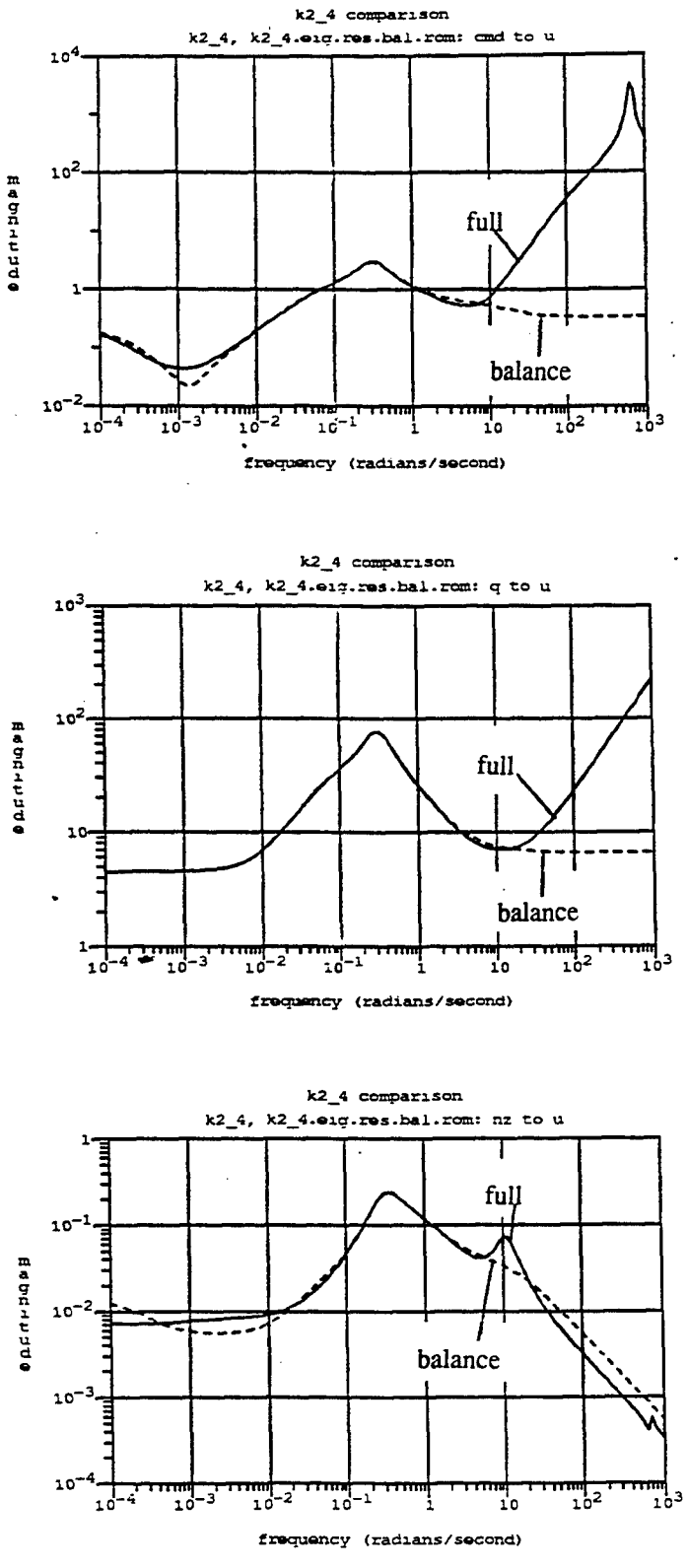


Figure 11.31: Comparison of Full- and Sixth-Order Controllers.

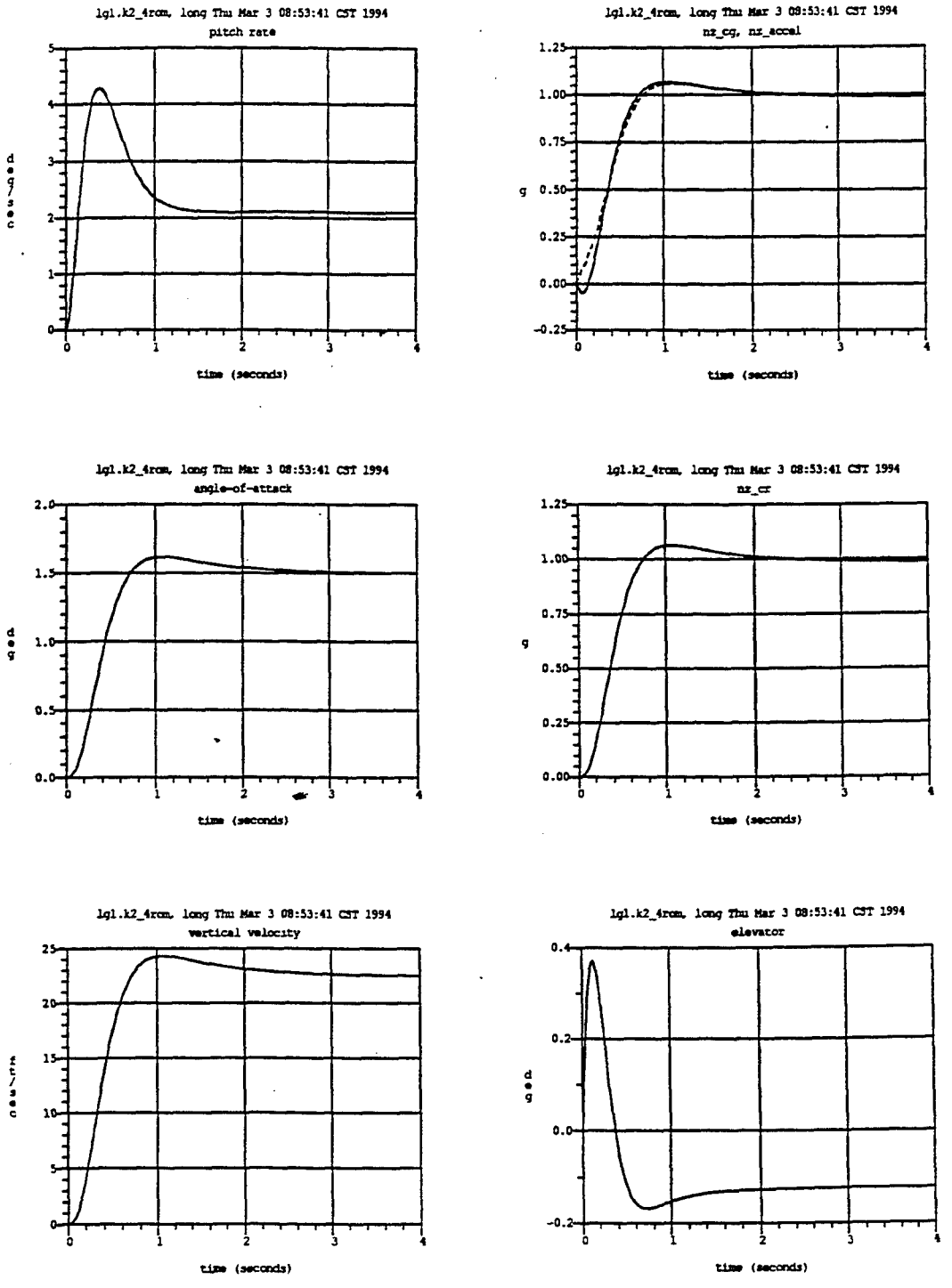


Figure 11.32: Linear Simulation of Closed-Loop Short-Period Design.

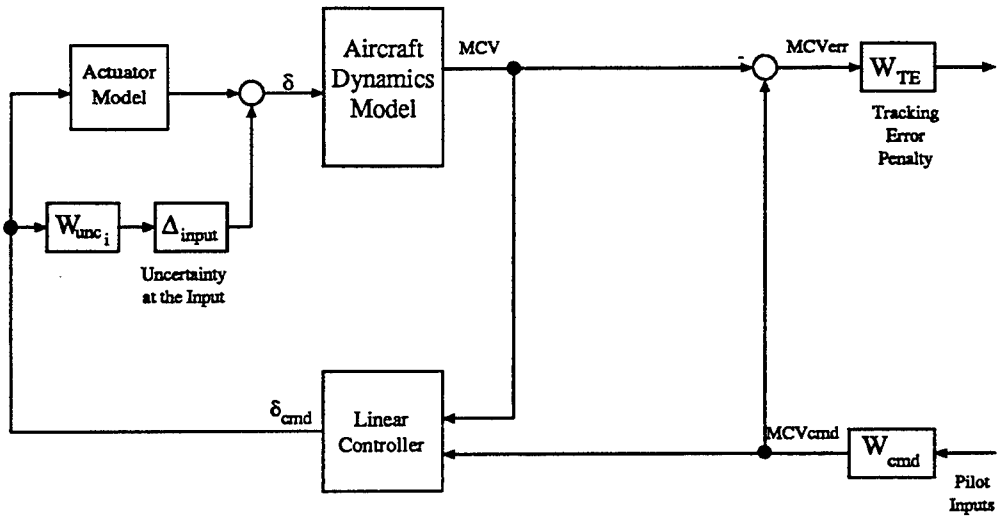


Figure 11.33: Interconnection Structure for Short-Period-Plus-Phugoid Design.

11.8.2.2 Aircraft Model

The aircraft model short-period dynamics are unchanged, but the phugoid ($\omega_{ph} = 0.16$, $\zeta_{ph} = 0.13$) is added. In addition, we define an additional output for this design: a longitudinal controlled variable, MCV. The new design using MCV will now be consistent with the other designs in these guidelines in that an MCV is defined prior to application of the design methodology. For this example,

$$MCV = \frac{V_{co}}{g}q + \frac{3}{s+3}nz$$

where $V_{co} = 300$ ft/sec, and nz is measured in g's. Note that this controlled variable is similar to the pitch controlled variable defined in Section 5.3. Obvious differences are the use of nz rather than α and the explicit filtering of nz. Recall from the previous example that we used output uncertainty weights to affect a rolled-off nz.

11.8.2.3 Weights

W_{unc_i}: This weight is defined as in the previous short-period design presented in Section 11.8.1.4.

W_{TE}: This weight penalizes errors between pilot stick command and the MCV response and is given by

$$W_{TE} = 71.9\frac{[s^2 + 1.4(0.05)s + 0.05^2](s+5)}{[s^2 + 1.4(0.2)s + 0.2^2](s+0.2)(s+100)}$$

Above 0.2 rad/sec (i.e., above the phugoid frequency) errors are penalized in the usual fashion: that is, high gain at 0.2 rad/sec that rolls off between 0.2 rad/sec and the desired crossover of 5 rad/sec. The pole at 100 rad/sec decreases the penalty for very high frequencies. Below 0.2 rad/sec the weighting function also rolls off to deemphasize the impact of phugoid dynamics on the MCV response, while still allowing H_∞ to stabilize the closed loop.

11.8.2.4 *D-K* Iteration

Just a couple *D-K* iterations were necessary to lower μ to an acceptable level. Figure 11.34 shows the μ plot along with the nominal performance (M_{ed}) and robust stability (M_{zv}) plots. Note again the typical behavior in that nominal performance dominates μ below crossover and robust stability dominates above crossover. Guaranteed robust performance is not achieved here ($\mu > 1$), but adequate robustness is achieved ($M_{zv} < 1$).

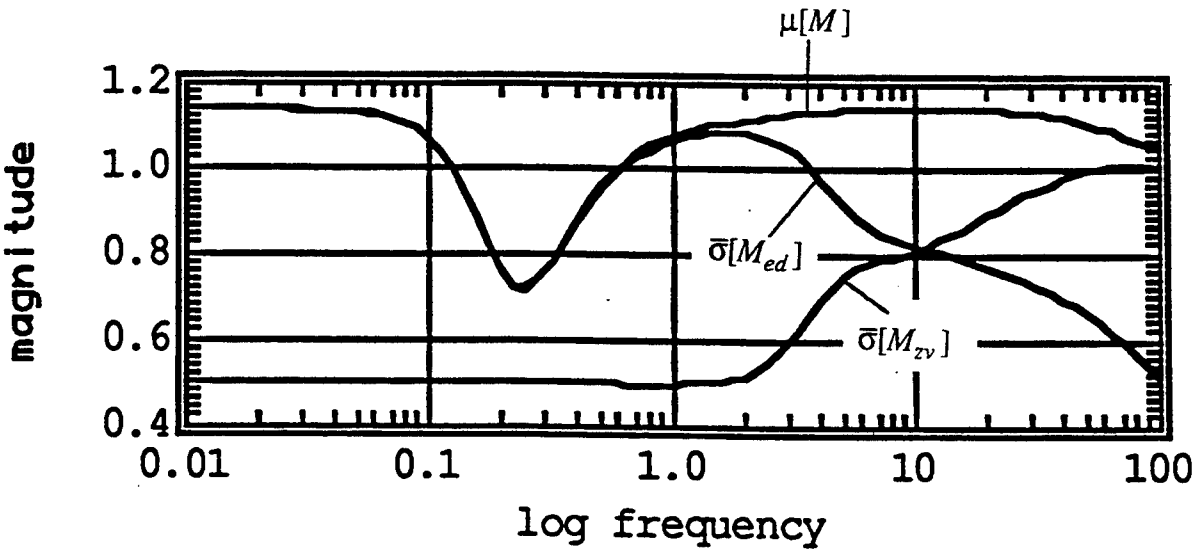


Figure 11.34: M-Analysis for Final Iteration.

11.8.2.5 Analysis

We were primarily concerned in this example with stabilizing the phugoid, and a check of the closed-loop poles reveals this was accomplished.

11.8.3 Short-Period Design with Real-Parameter Variation

This design example repeats the short-period design with the inclusion of real-parameter uncertainties in C_{m_α} and C_{m_δ}.

11.8.3.1 Interconnection Structure

Figure 11.35 shows the interconnection structure for this case. It is identical to Figure 11.23, except for the addition of a block for real-parameter uncertainty.

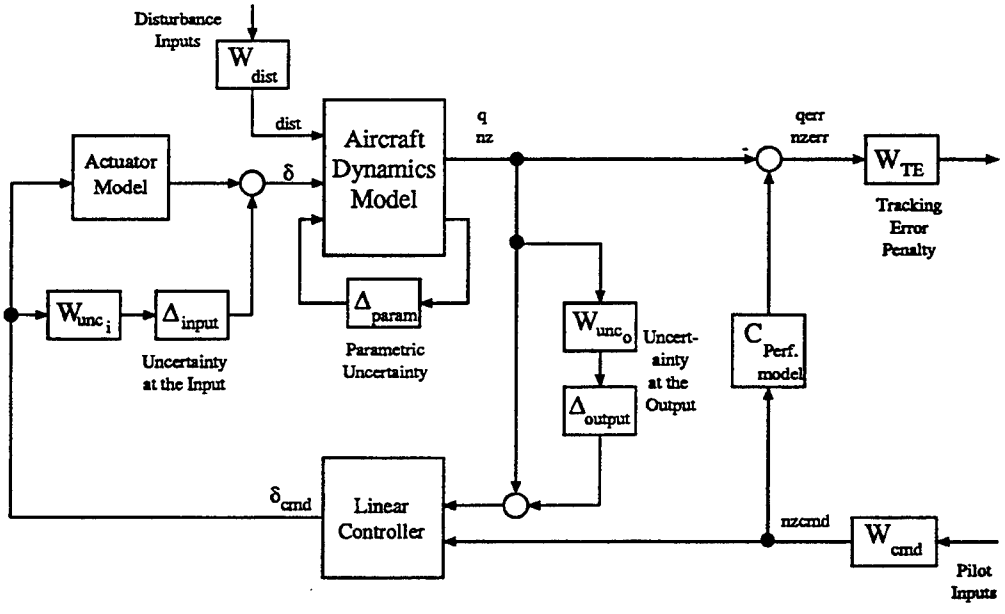


Figure 11.35: Interconnection Structure for Short-Period Design With Real Parameter Variations.

The short-period aircraft model is unchanged from the previous short-period design, except for addition of inputs and outputs associated with modeling uncertainty in C_{m_α} and C_{m_δ}. See Section 11.2.3 for details of how to add these types of inputs and outputs.

All other weights and models in Figure 11.35 are as for the previous short-period design. Note that the real parameter uncertainty block has no associated weights because: (1) the normalization scaling is handled with the method for setting up the additional inputs and outputs described in Section 11.2.3, and (2) any frequency dependence associated with these uncertainties is related to unsteady-aerodynamics effects, which generally have very high bandwidth (proportional to velocity) and can be ignored.

11.8.3.2 Design and Analysis

The design proceeds much like the previous short-period design presented in Section 11.8.1. The same design trade regarding gust disturbance is made, and the D-K iteration is stopped when $\bar{\sigma}[M]$ and $\mu[M]$ bounds reach those shown in Figure 11.36. Figure 11.37 shows the M-analysis: The solid line is a repeat of $\bar{\sigma}[M]$, the dashed line is robust stability due to the complex uncertainties, the dotted line is robust stability due to the two real-parameter uncertainties, and the dash-dot line is nominal performance. The resulting compensator has 52 states, but is reduced to eight in the same manner described for the previous short-period design. The resulting analysis is also similar.

11.9 Lateral-Directional Example

In this section, we describe a μ-synthesis design for lateral-directional control of a linear model of the F-117 stealth fighter. Additional examples of μ-synthesis designs for lateral-directional flight control are given in the references [Enns 1990a, Jackson 1990].

11.9.1 Problem Description

The F-117 stealth fighter lateral-directional dynamics differ slightly from more conventional aircraft in that the stability derivative for side force due to sideslip angle (C_{y_β}) is small. For this reason, a lateral accelerometer is not as useful as a control sensor, and a sideslip-angle measurement is assumed for this example.

The linear model has three inputs: effective aileron (δ_a), effective rudder (δ_r), and sideslip-angle gust (β_g). There are four measured variables: stability-axis roll and yaw rates (p_s, r_s), sideslip angle (β) and bank angle about the velocity vector (μ).

The control objectives are to track lateral stick and rudder pedal commands with little coupling, while rejecting side wind gusts, and perform well under model uncertainty. The lateral-tracking objectives are to provide quick response from lateral stick to stability-axis roll rate for commands of 4 rad/sec. The directional-tracking objectives are to provide quick response from rudder pedals to NCV response for commands of 0.5 g. NCV is a blend of stability-axis yaw rate and sideslip angle with an additional term to deal with trim. (See Section 5.4 for a detailed description of NCV.)

The two types of model uncertainty considered for this example are sensor uncertainty and unstructured (coupled) input uncertainty. We assume that the four measurements are accurate to some percentage of their nominal values (3% for β and μ sensors, and 1.5% for rate gyros). We assume aileron and rudder effectiveness are uncertain and allowed to couple together to 30% of their nominal effectiveness.

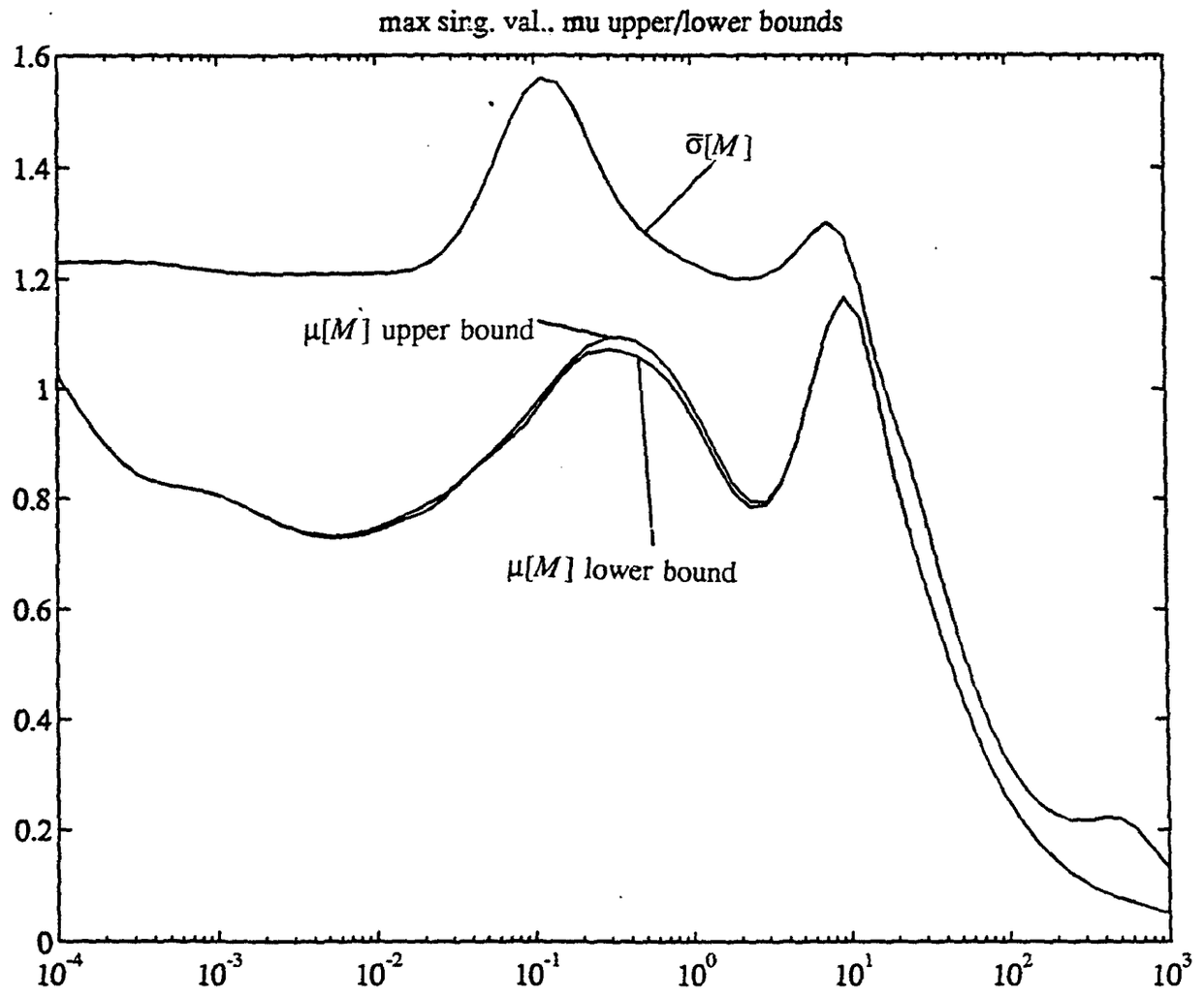


Figure 11.36: Maximum Singular Value and μ Bounds for Final Iteration.

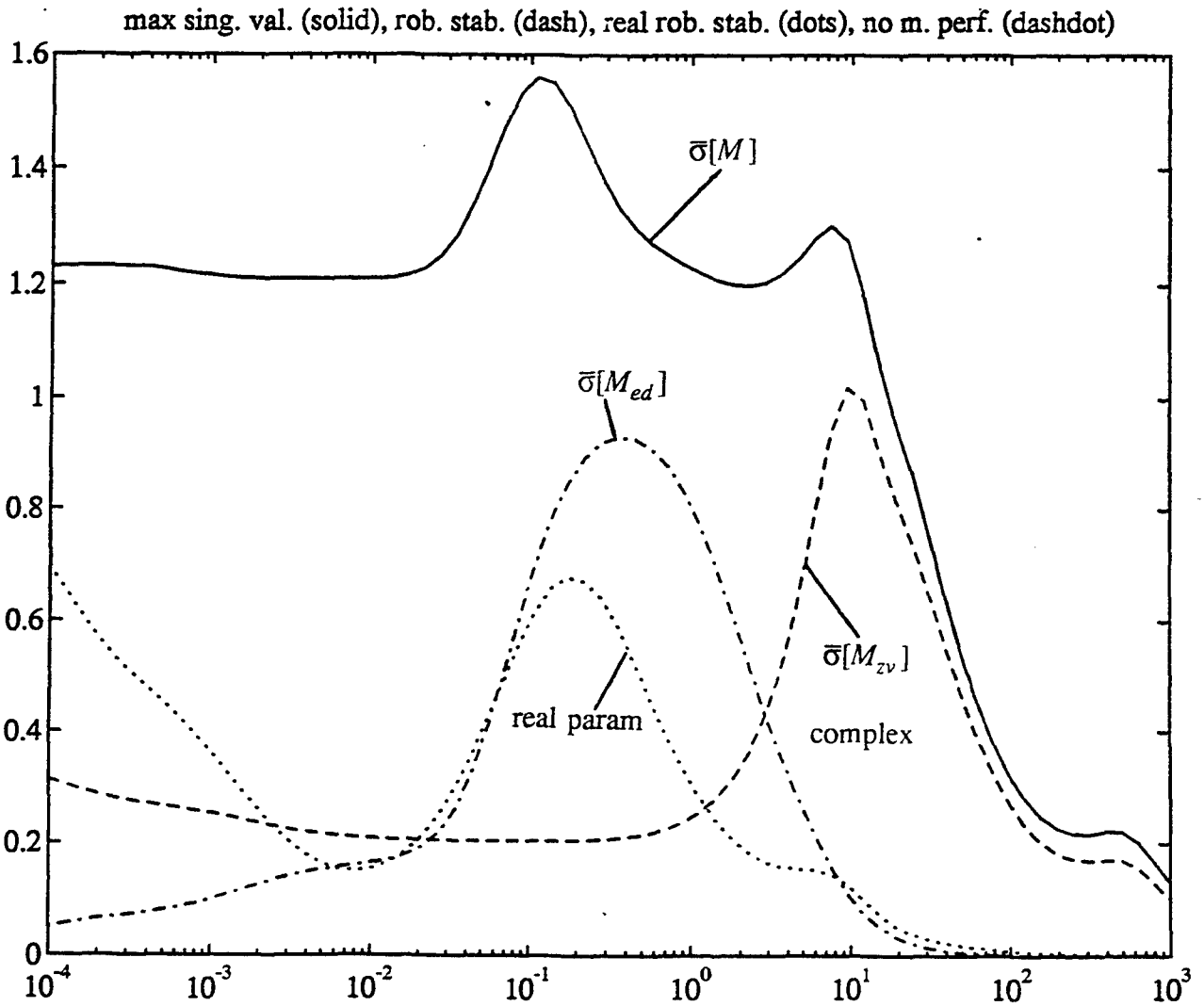


Figure 11.37: M-Analysis for Final Iteration.

11.9.2 Interconnection Structure

Figure 11.38 shows the interconnection structure, which is similar to Figure 11.20, but with no performance model and no parametric uncertainty. The interconnection P is built from the block diagram. The lower loop of P is closed with controller K to form closed-loop system M.

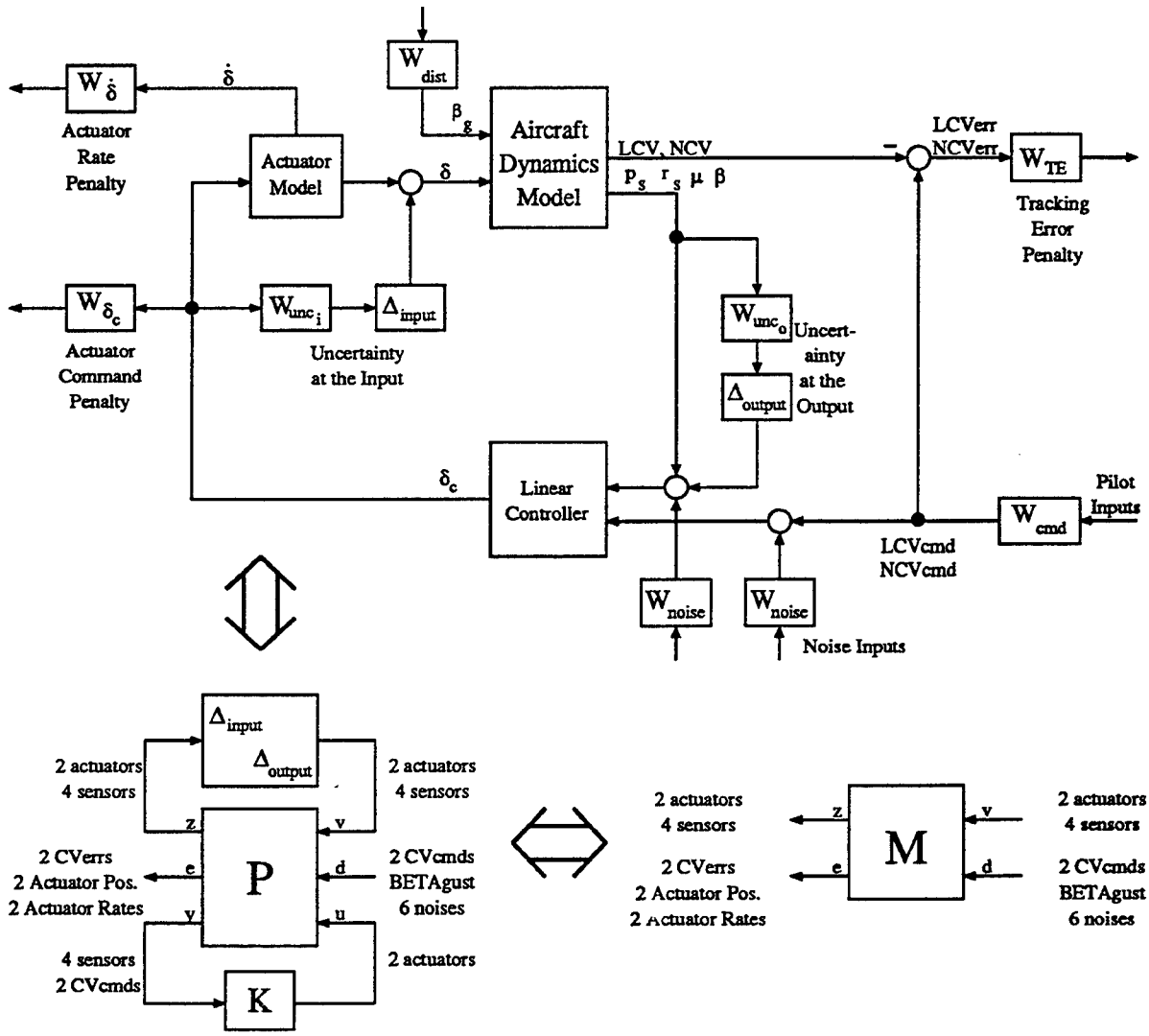


Figure 11.38: Lateral-Directional Interconnection Structure Example.

11.9.3 Performance Weights

Note that the primary performance inputs and outputs are pilot inputs and tracking errors, which gives a 2-input/2-output system. Figure 11.39 provides the context for

discussion of command and tracking error weights, which can be interpreted in several ways: One is described in Section 11.2.4, where each weight is considered separately. Another is how these weights act in combinations.

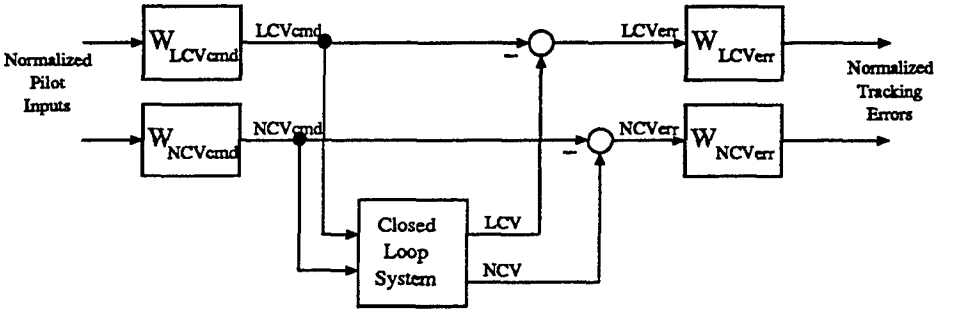


Figure 11.39: Command and Tracking Performance Weights.

For example, we expect response from NCV_{cmd} to NCV_{err} to look like a classical sensitivity function plot: low at low frequency, peaking above 1 near crossover, and equal to 1 at high frequency. The H_∞ design method attempts to make the response from normalized inputs to outputs flat across frequency and as small as possible. Thus, we chose weights so that the product of weights on NCV input and output approximate the sensitivity function we desire for NCV response. Although there appears to be some freedom in how we allocate weights between input and output, this choice does impact the response from LCV_{cmd} to NCV_{err}, as well as actuation, sensing, and disturbances responses.

LCV response dynamics are limited by a zero at the origin in the linear system, so we cannot expect good tracking at low frequency. Thus, the weighting on LCV response is chosen to allow low-frequency errors. However, we do want small LCV tracking errors throughout the frequency range of about 0.1 to 2 rad/sec. The following two weights are candidates for achieving these goals:

$$W_{LCV_{cmd}} = 0.08\left(\frac{s}{s+0.06}\right)\left(\frac{s+3}{s+0.06}\right)\left(\frac{60}{s+60}\right)$$

$$W_{LCV_{err}} = 6.25$$

The inverse of the product of these two weights ($1/W_{LCV_{err}}W_{LCV_{cmd}}$) defines the bound for the actual sensitivity response (LCV_{err}/LCV_{cmd}), as illustrated in the plot labeled *LCVerr/LCVcmd* in Figure 11.40. The solid lines in this figure are the weights; the dotted lines are the achieved transfer functions. If the H_∞ norm from normalized pilot input to normalized error is less than 1 across all frequencies, the actual response will be less than the bound, as it is for this example. These weights define a reasonable upper bound for sensitivity: They allow errors to be as large as 200% below 0.001 rad/sec, under 10% near 0.1 rad/sec, cross 1 near 2 rad/sec, and flatten out at 200% error until 60 rad/sec, where they ramp back up.

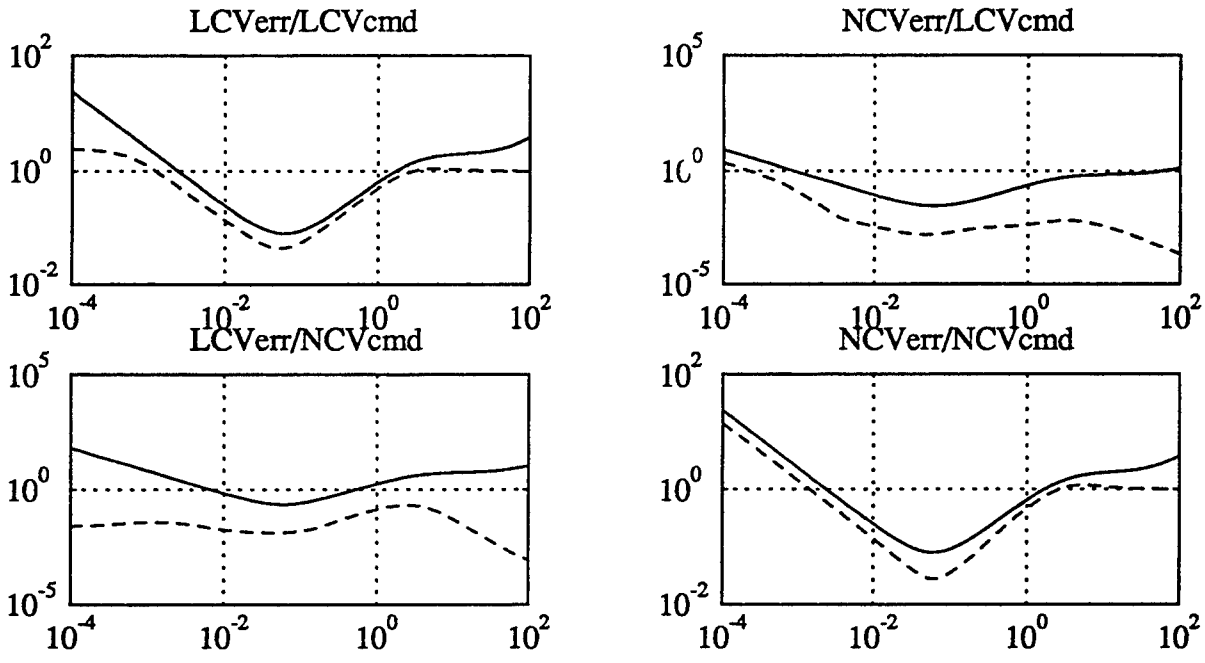


Figure 11.40: Performance Weights.

We chose NCV weights in a manner similar to the choice of LCV weights. Unlike the LCV response, NCV response is not limited by a zero at the origin. Nevertheless, it is still difficult to control NCV response at low frequency with sensor and actuator uncertainty. Thus, errors are allowed to be large at low frequency:

$$W_{NCV_{cmd}} = 0.028\left(\frac{s}{s+0.06}\right)\left(\frac{s+3}{s+0.06}\right)\left(\frac{60}{s+60}\right)$$

$$W_{NCV_{err}} = 17.7$$

We scaled the weights — the first upward and the second downward — to change relative weights of the cross terms. This scaling does not change the $NCVerr/NCVcmd$ weight, shown in the lower right plot of Figure 11.40, but does change gains of the cross terms. Without this scaling, the coupling from a LCV command to a NCV error is higher than desired.

The cross-coupling weights shown in Figure 11.40 are $(1/W_{LCV_{err}}W_{NCV_{cmd}})$ in the lower left and $(1/W_{NCV_{err}}W_{LCV_{cmd}})$ in the upper right.

We chose the weight on disturbance to reflect the expected size and frequency content of side wind gusts:

$$W_{\beta_g} = 0.21\frac{0.1}{s+0.1}$$

The maximum expected wind gust is 0.21 radians at relatively low frequency.

We did not use noise inputs as a driving requirement for this design, but we did include them to meet the rank requirements for H_∞ design when model uncertainty is absent for the first iteration. We chose a constant weight for all of the noise inputs with a value of 0.0005, which is small enough to not drive the design, but large enough to avoid numerical problems.

We also require a penalty on actuator command to meet the rank requirements for H_∞ design. The weight chosen was a constant value of 0.01 for each actuator command:

$$W_{\delta_c} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

11.9.4 Limiting Bandwidth

Two primary methods are available for limiting bandwidth of the closed-loop system: (1) adding model uncertainty that increases with frequency, or (2) adding a penalty on actuator rates. The first method is physically motivated, but requires the addition of frequency-dependant weights. The second method is just as effective, but it is not obvious how the weight should be chosen to achieve the desired bandwidth. We used the second method for this example. We chose weight on actuator rate to be very small initially and iteratively increased it until crossover occurred at the desired frequency (about 3–5 rad/sec).

$$W_{\dot{\delta}} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

11.9.5 Uncertainty Model

We modeled sensor uncertainty with a separate multiplicative uncertainty for each sensor, weighted by the percentage error that the sensor is rated for. In the context of Figure 11.38, we chose uncertainty weight as follows:

$$W_{unc_o} = \begin{bmatrix} 0.015 & 0 & 0 & 0 \\ 0 & 0.015 & 0 & 0 \\ 0 & 0 & 0.03 & 0 \\ 0 & 0 & 0 & 0.03 \end{bmatrix}$$

where the sensors are in the order: p_s, r_s, β, μ. We define uncertainty Δ_{output} as a 4-by-4 matrix consisting of four scalar complex uncertainty blocks.

We defined the actuator uncertainty model to allow 20% uncertainty in aileron and rudder effectiveness and to allow coupling between the two:

$$W_{unc_i} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}$$

The uncertainty Δ_{input} is defined as a 2-by-2 complex uncertainty block. Note that actuator uncertainty is wrapped around the actuator model, which allows uncertainty to ramp up from the given 20% error at the actuator break frequency, even though the weight is frequency independent. This trick was described in Section 11.2.5.

11.9.6 Computation of Controller (D-K Iteration)

We removed the uncertainty model inputs and outputs of the interconnection structure P for the first iteration of the H_∞ design. It is usually impractical to include the uncertainty model until an initial controller is designed and analyzed using μ to get a reasonable set of D-scales. An alternative to removing uncertainty connections is to set uncertainty weights to zero, but this may lead to numerical difficulties. We first perform an H_∞ design to generate a controller for the nominal system. The value of γ, the achieved H_∞ norm of the closed-loop weighted system M, should be small (less than 1). If γ is not small, performance weights are unreasonable. It is often desirable to do some iteration on performance weights until a satisfactory nominal performance controller is achieved, without the complication of uncertainty weights and μ-analysis. For this example, the value of γ for the first iteration was 0.75.

We then use the full interconnection structure, including uncertainty, to analyze robust performance of the controller with μ-analysis. Robust performance is usually very poor because the controller was designed with no knowledge of the uncertainty. However, we can fit the D-scales computed from μ-analysis with rational transfer functions and incorporate them into the interconnection structure to scale uncertainty inputs and outputs for the H_∞ design.

We next perform D-K iteration until the μ plot is relatively flat and the μ-optimal controller is nearly reached. It is usually not desirable to proceed until *full* convergence is reached because at least one closed-loop pole tends toward infinity and numerical problems result. Figure 11.41 shows a plot of the maximum singular value of the weighted closed-loop system ($\bar{\sigma}[DMD^{-1}]$) for each iteration, and Figure 11.42 shows the same plot for the final iteration along with the $\mu[M]$ bounds. Here, $\bar{\sigma}$ is nearly equal to the bounds.

For this example, we fit D-scales after each iteration with first- to fourth-order transfer functions. Because there are five uncertainty blocks, there are five D-scales, and the full D-scale state-space matrix was usually between about 15-th and 25-th order. Figure 11.43 shows a plot of the D-scales for the final iteration. The solid lines are the actual D-scales and the dotted lines are the curve fits from μ analysis.

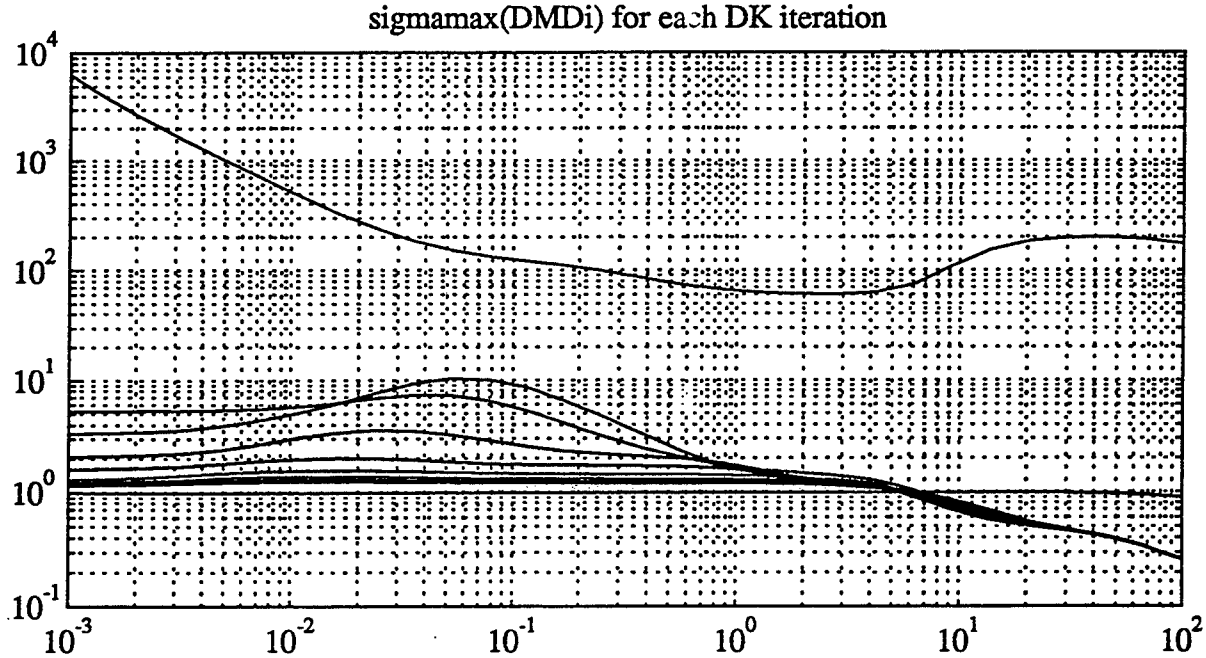


Figure 11.41: Change in $\bar{\sigma}[DMD^{-1}]$ During D-K Iteration.

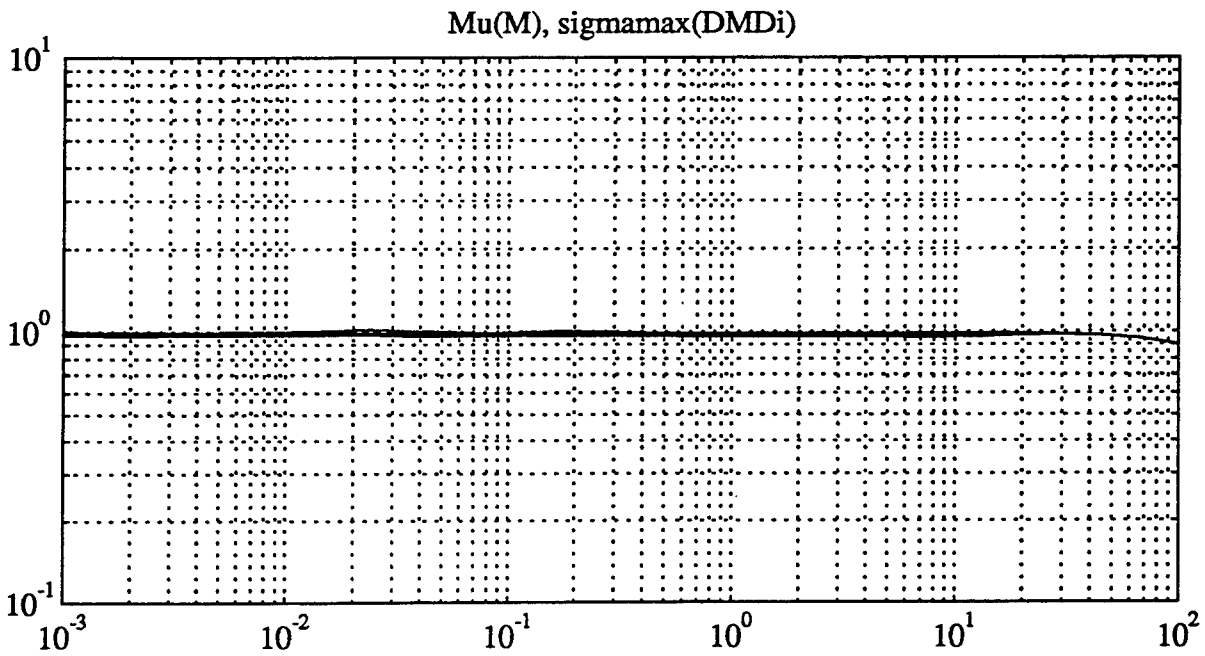


Figure 11.42: $\bar{\sigma}.[DMD^{-1}\cdot]$ and $\mu[M]$ Upper and Lower Bounds for Final Iteration.

11.9.7 M-Analysis

The μ-optimal system should achieve $\mu[M] < 1$. When we are unable to achieve this, we often must determine which inputs and outputs of M are the primary contributors to the value of $\mu[M]$. This process, called M-analysis, consists of plotting the transfer function M for various sets of inputs and outputs. However, it is usually impractical to look at all inputs and outputs on an individual basis. For this example, there are 17 inputs and 13 outputs. These inputs and outputs can be divided into groups and the matrix M similarly partitioned. We can then plot the maximum singular values of elements of M corresponding to these groups of inputs and outputs. Those elements whose maximum singular values are close to μ define the particular groups of inputs and outputs that are causing difficulties in meeting objectives. We can then "zoom in" on these troublesome cases and look at individual elements of these groups, or at least make the groups smaller.

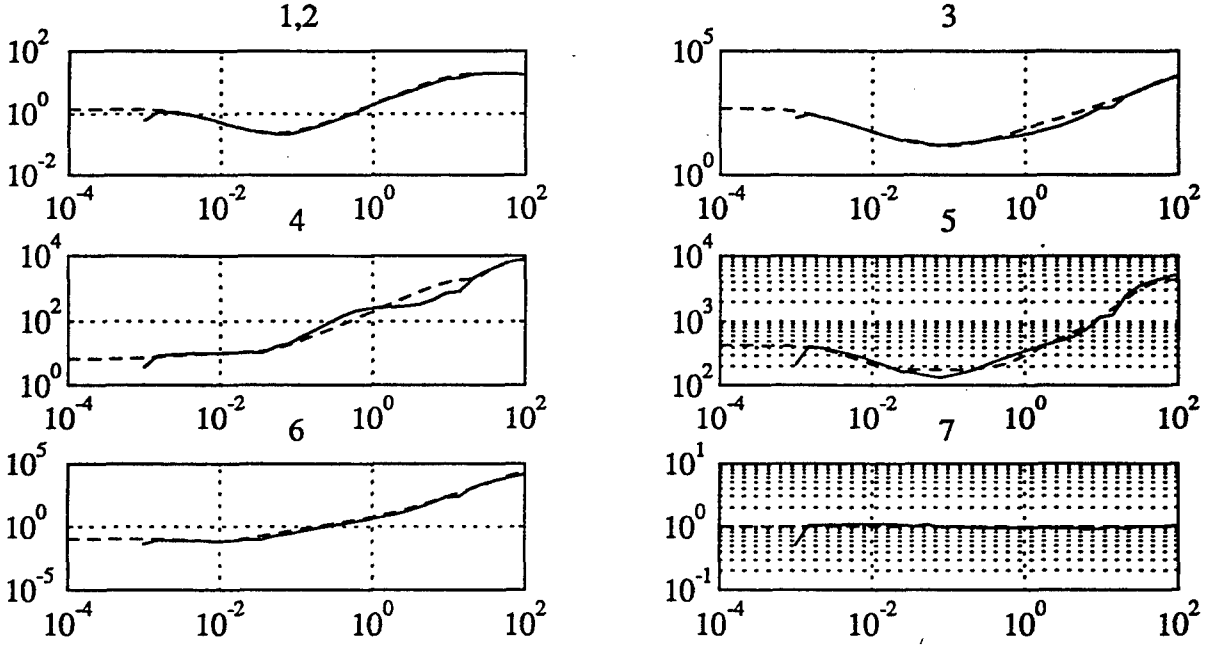


Figure 11.43: D-Scales (Solid), D-Scale Rational Fits (Dashed).

Figure 11.38 earlier showed the inputs and outputs of the closed-loop system M. The first step in M-analysis is usually to look at maximum singular value plots of the gains of the four main transfer functions: (1) v to z, (2) v to e, (3) d to z, and (4) d to e. These four plots appear in Figure 11.44. The upper curve in each plot is $\bar{\sigma}[DMD^{-1}]$, which is approximately equal to 1 for all frequencies, and the lower curve is the maximum singular value of the subset of DMD^{-1} in question. (Note that $\bar{\sigma}$ is approximately 1 in each of the four plots.) It is important to note the separation between curves, as this determines how significant the contribution this subset of M makes to the maximum singular value

of M. We offer the following observations from looking at these plots:

- Interaction between model uncertainty and performance (d to z and v to e) drives the problem at low frequency.
- Nominal performance (d to e) drives the problem at high frequency.
- Stability robustness (v to z) does not seem to be a problem.

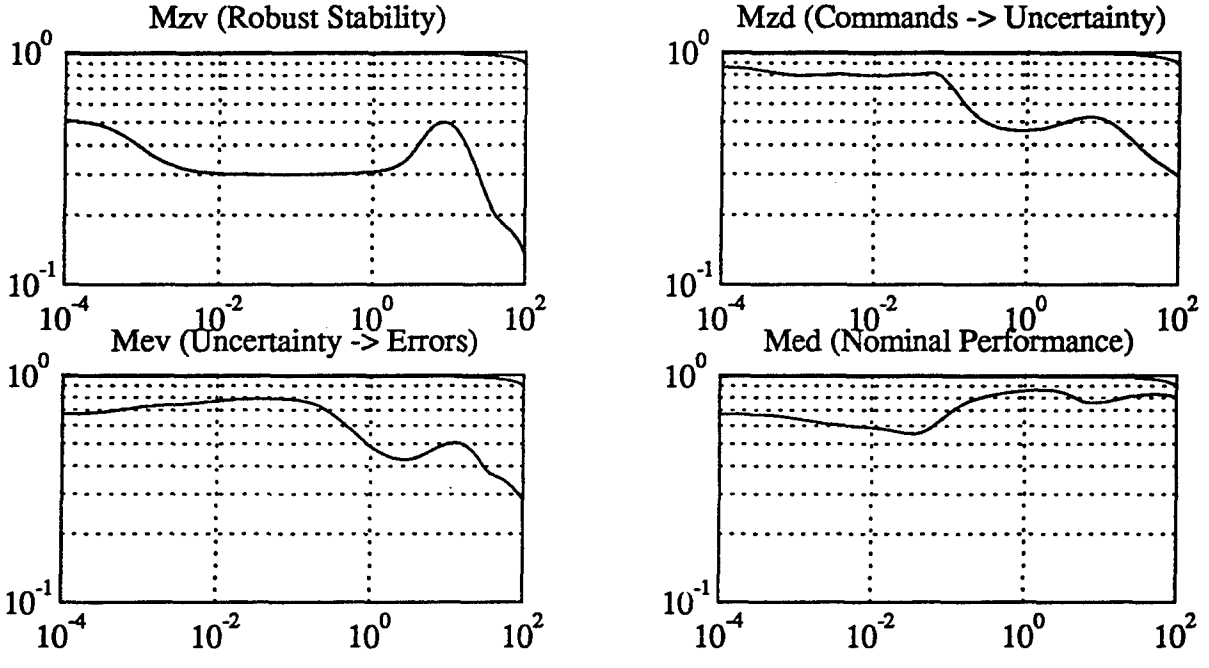


Figure 11.44: M-Analysis — First Step.

The next step taken in the M-analysis is to look at all 25 transfer functions from the five groups of inputs to the five groups of outputs, as defined in the lower right of Figure 11.38. Figure 11.45 shows four of the 25 plots that have the most significant contribution to M, and we offer the following observations:

- Interaction between model uncertainty and performance noted above is due to sensor uncertainty and controlled variable commands and errors.
- Tracking performance (CV commands to CV errors) is the dominant path in the frequency range from about 0.5 to 5 rad/sec.
- Actuator rate penalty is dominant above about 5 rad/sec.

It is evident from the dominance of actuator rate outputs at high frequency that this penalty is forcing the system to roll off.

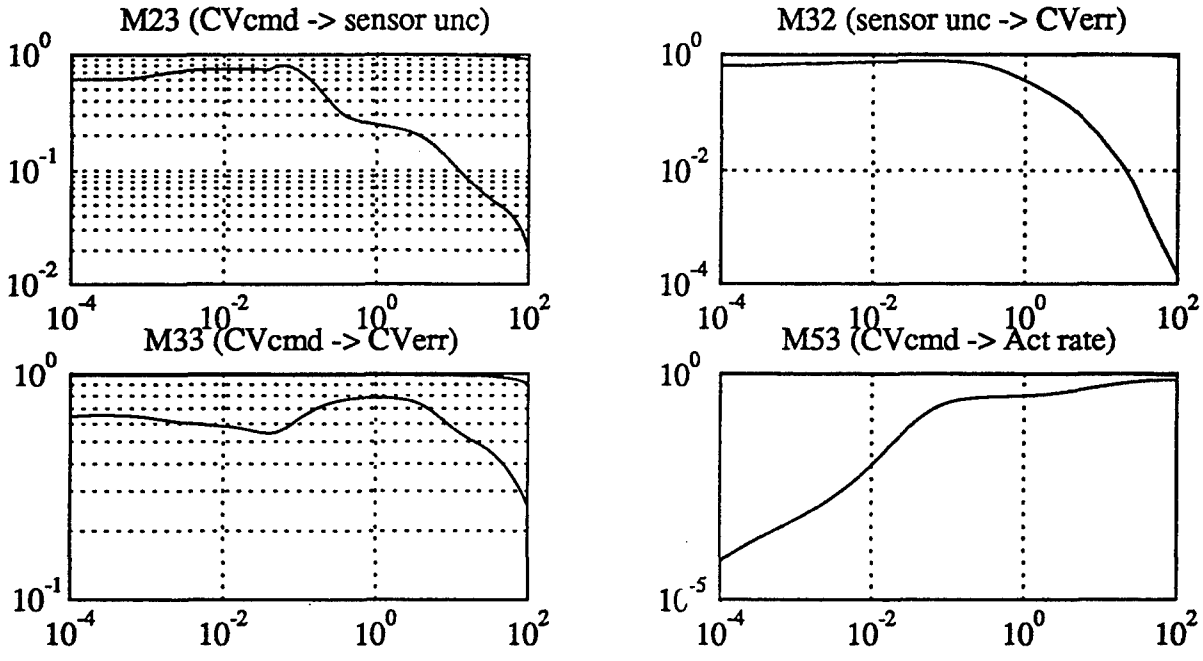


Figure 11.45: M-Analysis — Zooming In.

Figure 11.46 shows the transfer functions from the two controlled variable inputs to the two controlled variable errors. We offer these observations:

- Both LCV and NCV tracking performance drive the problem near the crossover frequency range.
- Tracking cross terms are not significant.

11.9.8 Controller Model Reduction

We reduced the controller order using the frequency-weighted balanced-realization method described in Section B.1.5.

11.9.9 Time Response

Figures 11.47 and 11.48 show time responses for step input commands in LCV and NCV, respectively.

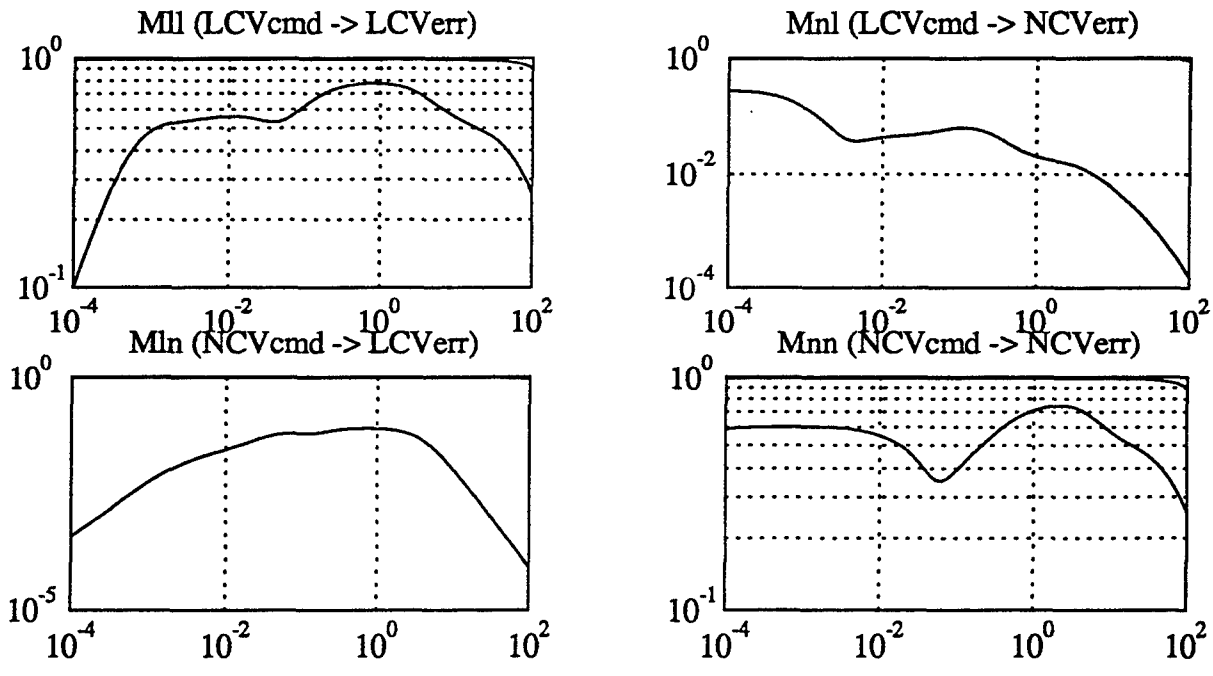


Figure 11.46: M-Analysis — Performance Inputs/Outputs.

11.9.10 Lateral-Directional Example Summary

This example showed how to build the interconnection structure and weight performance and uncertainty of the lateral-directional dynamics for a linear model of the F-117. The weights used provide a good starting point for designing a controller for another aircraft. The process of M-analysis was discussed in detail to illustrate how one assesses which inputs and outputs are driving the problem.

11.10 General Observations

In this subsection, we offer some general observations on the μ synthesis design methodology in light of the designs performed at the Lockheed facility.

11.10.1 Design Problem Formulation Issues

Time-domain simulation, for specified maneuvers, is an important part of assessing the quality of a given design. The H_∞/μ synthesis approach involves using frequency dependent weighting functions to specify performance. It is recommended that a nominal design (without the inclusion of perturbations in the interconnection structure) be run, and simulated, to determine whether or not the selected weighting functions result in a

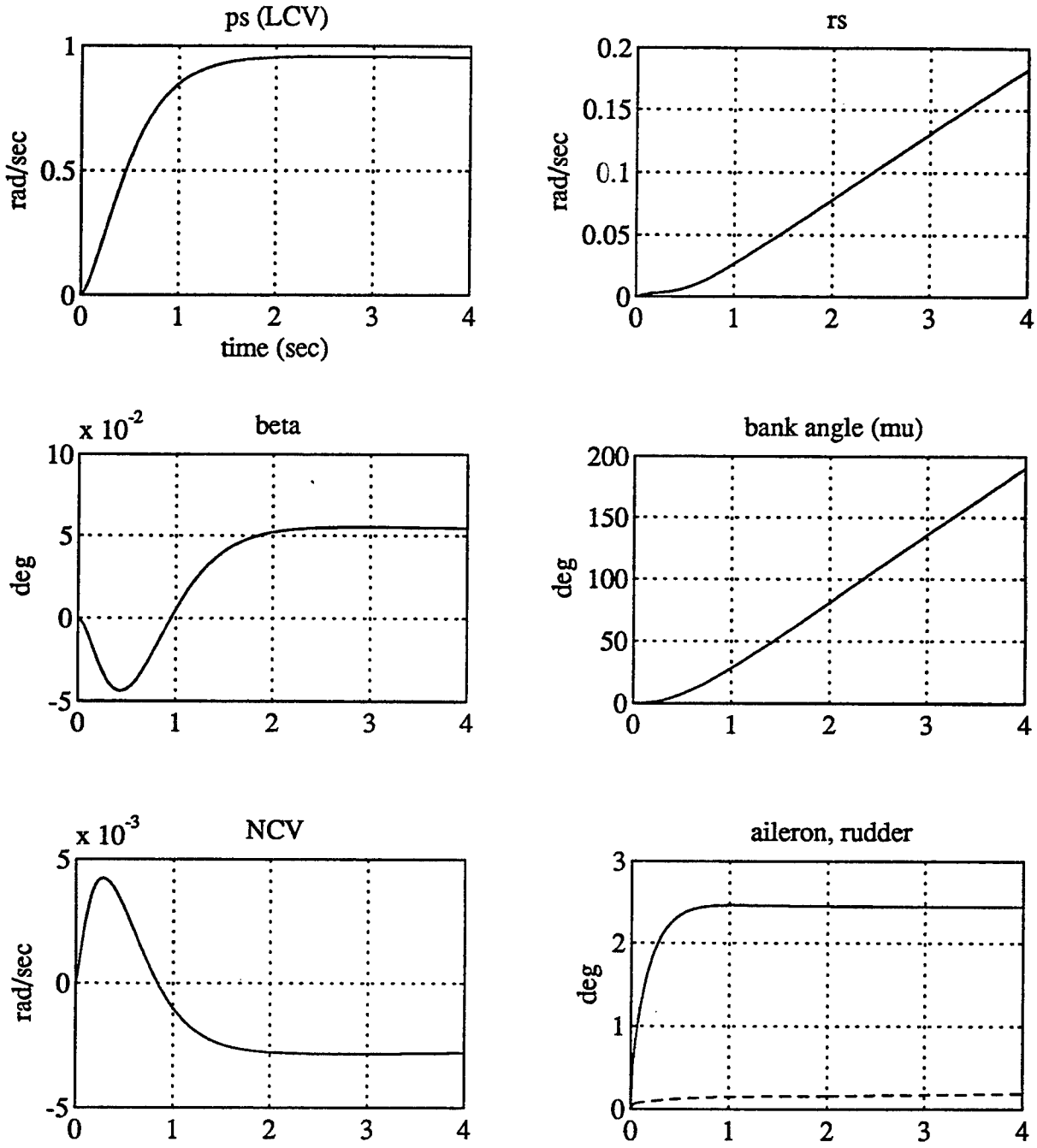


Figure 11.47: Time Response to *LCV* Step Input.

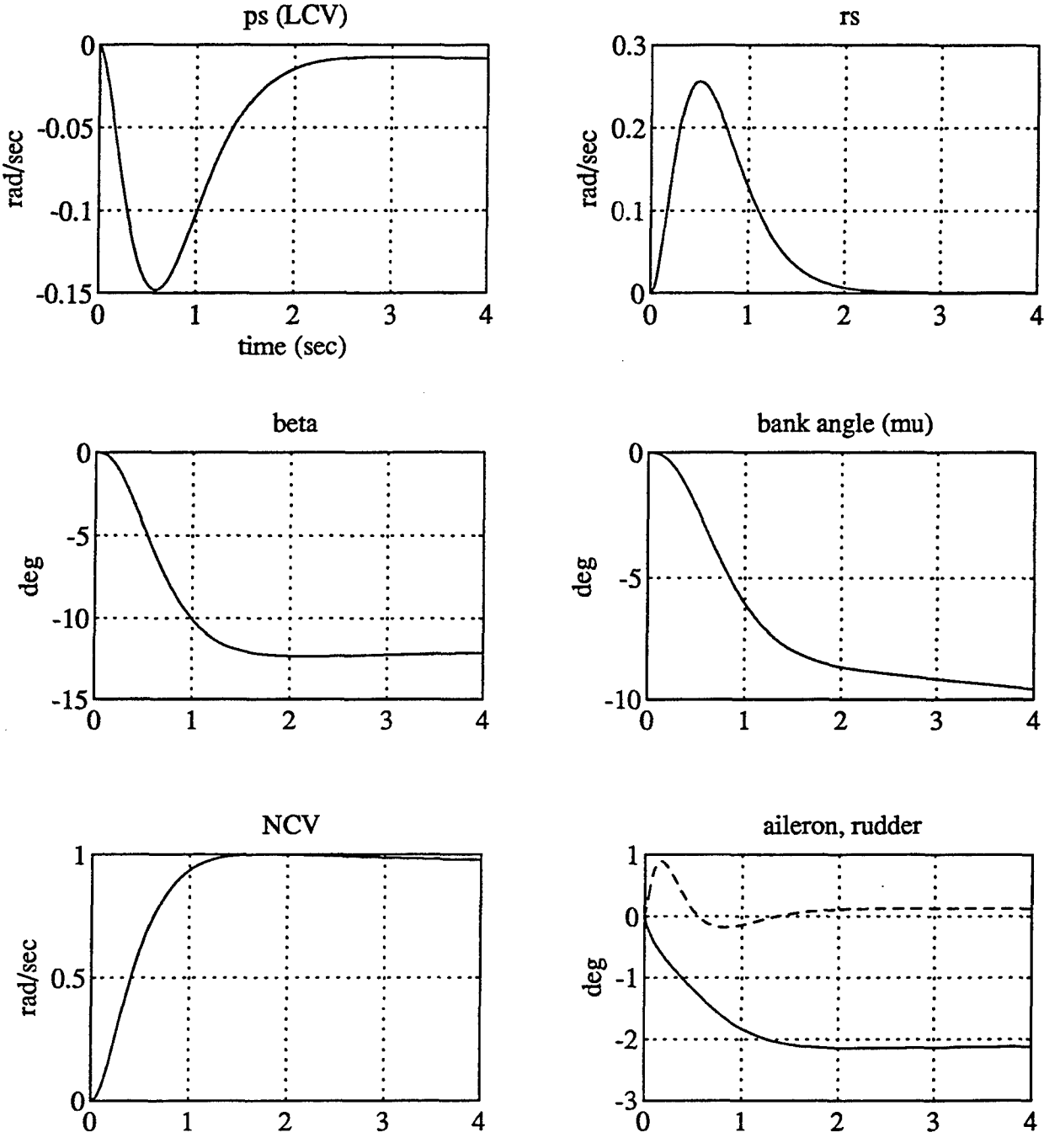


Figure 11.48: Time Response to NCV Step Input.

suitable design.

The weighting functions should be modified such that the nominal design results in satisfactory time-domain performance (as assessed via simulation), and the γ value in the H_∞ optimization is less than one. The resulting nominal controller may be a good initial controller for the D-K iteration.

In general, as the D-K iteration proceeds, nominal performance is traded off to give additional robustness. The value of γ achieved for the nominal problem is a lower bound for the value of γ (and for μ) that can be achieved for the full robust-performance problem.

The interconnection structure used influences the D-K iteration, and, as the iteration is not convex, can influence the final result. For example, weighting functions can commute with the perturbations. From a robust performance/stability point of view, this has no effect. However the minimum γ in the first iteration can differ by orders of magnitude, and the problems can have different iteration characteristics. In the problems studied here, this was not an issue.

It is possible that for a specified interconnection structure, the final value of μ is significantly greater than one. In such cases it may not be possible to achieve the requested robust performance, and the designer may have to select different weighting functions to trade between performance and robustness requirements. Examining nominal performance, robust stability, and robust performance as a function of frequency, can give some idea of what is the limiting factor. Note that robust stability and robust performance are, in general, both μ tests. The M-analysis discussed in the guidelines provides greater detail on how individual perturbations or performance inputs/outputs can be studied to determine the limiting issues. After changing the weights, the designer should redesign and simulate a new purely nominal performance controller to reassess suitability of the specified nominal performance.

11.10.2 D-K Iteration Issues

The objective with μ-synthesis is to improve closed-loop robust performance (i.e., reduce $\mu(F_l(P,K))$, where $F_l(P,K)$ denotes the interconnection structure, P, with the lower loop closed around the controller, K).

The D-scales, calculated via $\mu(F_l(P,K))$, are such that $\bar{\sigma}(D\,F_l(P,K)\,D^{-1})$ is a close upper bound to $\mu(F_l(P,K))$. Here $\bar{\sigma}$ denotes the maximum singular value, and this is strongly affected by the inclusion of D-scales. However, μ is invariant under the D-scaling operation, and the iteration proceeds by using rational approximations to the D-scales as scalings for the next H_∞ design step.

At each step, $\mu(F_l(P,K))$ is guaranteed to be less than the γ value obtained in the H_∞ calculation of K. This does not mean that γ is necessarily reduced at each step. Do not use the achieved γ levels as a measure of the D-K-iteration performance at each step;

the calculated value of μ is the correct measure. In most problems, γ is actually reduced at each step.

The D-K iteration is not always convex, and (irrespective of the fitting method, or its accuracy) is not guaranteed to decrease μ on the next step, although it often does.

The D-scales are fit over a finite frequency range, and the H_∞ design can be viewed as being applied over all frequencies. If the problem exhibits dynamics outside the specified frequency range, these will not be captured by the D-K iteration. Prior experience has shown this to be a potential consequence of selecting high-order D-scale fits. High-order fits often result in near pole/zero cancellations, and these may occur outside the frequency range considered. This issue did not arise in any of the problems studied here. To ensure that this is not a problem, the designer should select a frequency range that captures all the dynamics of the weighted interconnection structure.

Several choices of weighting functions for D-scale fitting are possible. The default choice is the sensitivity of μ with respect to the D-scale value. The heuristic here is that this choice will produce D-scale fits that give closer approximations to the μ bound. An alternative choice is the value of μ itself. The heuristic in this case is that this choice will emphasize frequencies where μ is largest, and this may help reduce μ on the next iteration. In the problems studied here, we used D-scale sensitivity and found it to give good results.

When μ is approximately constant over the desired controller bandwidth, subsequent D-K iterations often affect only the higher frequency behavior, which is a useful stopping criteria for the D-K iteration.

11.10.3 Numerical Considerations

Prior experience with flexible-structure vibration-suppression problems has indicated that lightly damped modes may cause numerical difficulties in the H_∞ design phase and may limit the number of D-K iterations that can be performed. This was not an issue in the problems studied here. As many as eight iterations were performed on some problems with no indication of excessive numerical sensitivity. Numerical sensitivity is often due in part to the rational D-scale fits having dynamics outside the frequency range over which μ is calculated. Selecting a wider frequency range usually alleviates some of these problems.

Several machine specific considerations are noted. The VAX/VMS version of Xmath does not support IEEE standard arithmetic. Divide by zero operations cause an arithmetic trap and termination of the Xmath/Xmu function. This can cause a problem in the existing Xmu D-scale fitting function, `fitsys.msf`, for fit orders greater than three or four. An algorithmic modification has been made that alleviates this problem. Initial indications suggest that this has resolved the problem, although additional testing on VMS versions of Xmath is recommended.

A similar problem was noted in the function that displays system pole and zero locations, `rifd.msf`, and has been fixed.

11.10.4 Potential Enhancements

The following should be considered for future efforts.

The μ software can calculate the worst-case dynamic perturbation (or a random perturbation) for a given robust performance/stability μ analysis. Time-domain simulations should include, in addition to the nominal response, the effects of several perturbations. The perturbations can be chosen to reflect expected flight condition variations and/or worst case effects. The μ-synthesis procedure trades off nominal performance against robustness to such perturbations, and it is often instructive to use simulation to assess the value of this tradeoff. Note that for comparison purposes, a purely nominal performance design should be simulated on a perturbed plant.

THIS PAGE LEFT INTENTIONALLY BLANK

Appendix A
Mathematical Details

A.1 Singular Value Decomposition

It is a fact [Stewart 1973] that any complex $n \times m$ matrix can be decomposed via

$$A = U\Sigma V^* \tag{A.1}$$

where

$$\Sigma = \begin{cases} \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix} & \text{if } n < m \\ \Sigma_1 & \text{if } n = m \\ \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} & \text{if } n > m \end{cases} \tag{A.2}$$

with

$$\Sigma_1 = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_r\} \tag{A.3}$$

$$r = \min(n, m) \tag{A.4}$$

$$\bar{\sigma}[A] = \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r = \underline{\sigma}[A] \geq 0 \tag{A.5}$$

and U and V are unitary matrices. That is

$$\begin{aligned} UU^* &= U^*U = I \\ VV^* &= V^*V = I \end{aligned} \tag{A.6}$$

The columns of U and V are right eigenvectors of AA^* and A^*A, respectively, and are known as the right and left singular vectors of the matrix A. They describe the directions of the gains of A. The eigenvalues of AA^* and A^*A, denoted by $\lambda[AA^*]$ and $\lambda[A^*A]$, respectively are nonnegative, and

$$\sigma_i[A] = \lambda_i^{1/2}[AA^*] = \lambda_i^{1/2}[A^*A] \quad \text{for } i = 1,\ 2,\ \ldots, r \tag{A.7}$$

Several standard equalities, inequalities, and theorems involving singular values are stated in Table A.1.

Table A.1: Some Singular Value Equalities, Inequalities, and Theorems.

Equalities	
Inverse Relationship	$\underline{\sigma}[A] = \frac{1}{\bar{\sigma}[A^{-1}]}$
	$\bar{\sigma}[\alpha A] = \lvert\alpha\rvert\bar{\sigma}[A]$ for any complex scalar α
Inequalities	
Triangle Inequality	$\bar{\sigma}[A+B] \leq \bar{\sigma}[A] + \bar{\sigma}[B]$
Cauchy-Schwarz Inequality	$\bar{\sigma}[AB] \leq \bar{\sigma}[A]\bar{\sigma}[B]$
	$\underline{\sigma}[A] - 1 \leq \underline{\sigma}[I+A] \leq \underline{\sigma}[A] + 1$
	$\underline{\sigma}[A] \leq \lvert\lambda[A]\rvert \leq \bar{\sigma}[A]$
Theorems	
	$\underline{\sigma}[A],\ \underline{\sigma}[B] \neq 0 \Rightarrow \underline{\sigma}[AB] \geq \underline{\sigma}[A]\underline{\sigma}[B]$
	$\bar{\sigma}[E] < \underline{\sigma}[A] \Rightarrow \underline{\sigma}[A+E] > 0$
	$\bar{\sigma}[A] < 1 \Rightarrow \underline{\sigma}[I+A] \geq 1 - \bar{\sigma}[A]$

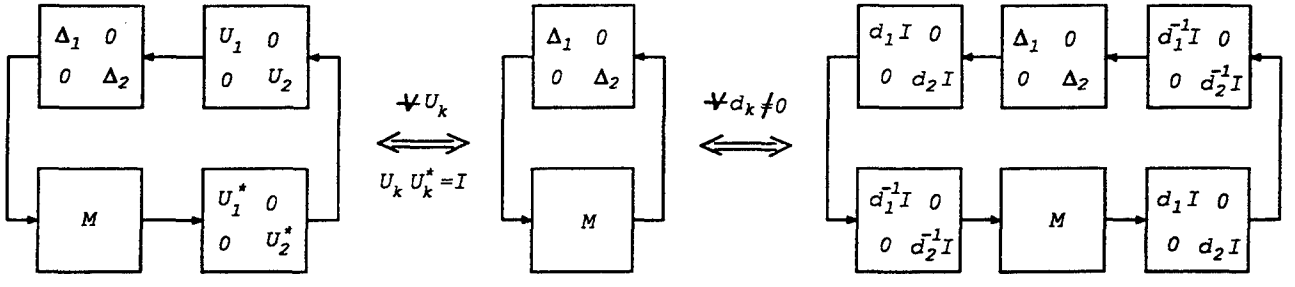


Figure A.1: Manipulations of Perturbations That Preserve Size.

A.2 Bounds for μ

For general structures, it is not always possible to numerically quantify the value of μ. But it is always possible to bound it between the values for the two extreme cases discussed in Sections 2.6.3.1 and 2.6.3.2. Furthermore, we can perform other manipulations to tighten the bounds. Often the bounds can be tightened to the point where upper and lower bounds are equal, in which case μ is known exactly. In other cases, the upper and lower bounds may be sufficiently close to obtain an adequate approximation of μ.

Consider the diagram shown in Figure A.1. These three diagrams are all equivalent, but the stability robustness analyses conducted (after the manipulations introduced on the left and right of Figure A.1) for the two extreme cases discussed in Sections 2.6.3.1 and 2.6.3.2 lead to lower and upper bounds for μ.

The manipulations on the left involve unitary matrices, U where

$$U^*U = UU^* = I \tag{A.8}$$

In addition, these matrices have structure (with blocks indexed by k) that is consistent

with the Δ structure. The size of the uncertainty set is unchanged by these manipulations, which can be shown as follows:

$$\bar{\sigma}[\Delta_k U_k] = \lambda^{\frac{1}{2}}[\Delta_k U_k U_k^* \Delta_k^*] = \lambda^{\frac{1}{2}}[\Delta\Delta^*] = \bar{\sigma}[\Delta_k] \tag{A.9}$$

Because the size of the uncertainty set is unchanged, $\mu[M] < 1$ for the problem in the center of Figure A.1 is equivalent to $\mu[M\bar{U}] < 1$ for the problem on the left of Figure A.1. But $\rho[M\bar{U}] < \mu[M\bar{U}]$ for any $\bar{U} = \text{diag}\{U_k\}$, so steps can be taken to tighten the bound. In summary, for the lower bound

$$\sup_{U_k} \rho[\bar{U}M] \leq \mu[M] \tag{A.10}$$

and further details can be found in the literature [Doyle 1982a, Doyle 1982b].

The manipulations on the right involve scaling matrices, D, that have structure (with blocks indexed by k) that is consistent with the Δ structure. Figure A.1 assumes full Δ blocks. Thus, $d_k I$ can be replaced with a full D_k if the blocks are repeated scalars. The size of the uncertainty set is unchanged by these manipulations which can be shown as follows

$$\bar{\sigma}[d_k \Delta_k d_k^{-1}] = \bar{\sigma}[\Delta_k] \tag{A.11}$$

Because the size of the uncertainty set is unchanged, $\mu[M] < 1$ for the problem in the center of Figure A.1 is equivalent to $\mu[DMD^{-1}] < 1$ for the problem on the right of Figure A.1. Because $\mu[DMD^{-1}] < \bar{\sigma}[DMD^{-1}]$ for any $D = \text{diag}\{d_k\}$, steps can be taken to tighten the bound. In summary, for the upper bound,

$$\mu[M] \leq \inf_{d_k} \bar{\sigma}[DMD^{-1}] \tag{A.12}$$

Further details can be found elsewhere [Doyle 1982a, Doyle 1982b].

In conclusion, both lower and upper bounds can be computed for $\mu[M]$. These bounds depend on the structure of the uncertainty set for Δ. There are a variety of possible structures that can occur in real applications and software exists to give fast and accurate bounds for most cases of interest.

A.2.1 Real vs. Complex Structure

Clearly if the perturbations Δ are real, the corresponding μ is different than if the perturbations are complex. Consider the complex plane. Then, the set of all complex perturbations is a disk that extends over the real and imaginary portions of the complex plane. Contrast this with the set of real numbers subject to the same bound. Here, the set of bounded real perturbations is just a line segment that is the intersection of the disk and the real line, which is a much different set.

Currently software is available for either case. Only bounds are computed in general. Improved bounds, both in the sense of tighter bounds and faster algorithms, are the subject of current research (e.g., [Young 1992b]).

A.2.2 Constant But Unknown vs. Arbitrary Time-Varying Structure

Clearly there is a difference between modeling perturbations Δ that are constant but unknown and perturbations that are varying in some arbitrary manner. In the case of constant but unknown perturbations, the perturbed closed-loop system remains time invariant. But when Δ varies with time, the system becomes time varying. It follows that stability analyses for these two cases will be different.

Recall that D-scales in Section A.2 were arranged such that the size of the set of possible $D^{-1}\Delta D$ was the same as the size of the set of possible Δ. In the case of constant Δ, a frequency dependent D-scale commutes with Δ, and thereby cancels the D^{-1} because the composition of time invariant operators corresponds to multiplication of transfer functions and the D-scales are just scalars for each individual structural block of Δ. But if the Δ is a time-varying operator, $D^{-1}\Delta D$ implies the composition of these three operators, so that the D scales only cancel in this case for a constant operator D (i.e., non frequency dependent). To see this, consider a signal passing through a filter D and then being altered in some time varying (but bounded) manner by the operator Δ. It is not possible to undo the frequency response of D by filtering the output of ΔD with D^{-1} unless D is constant in frequency.

As a consequence of this, μ will be larger for time varying Δ because there are not as many degrees-of-freedom available for minimization of the upper bound for μ given by $\bar{\sigma}DMD^{-1}$. The D-scales can be frequency dependent for constant Δ but must be independent of frequency for the case of time varying perturbations.

Technical details are discussed further in the references [Doyle 1987, Krause 1988].

A.3 Desensitization

Recall that in Section 2.4.1 we showed that the closed-loop response was given by equation (2.25). Now assuming a divisive representation for plant uncertainty given by equation (2.41), we show below that the sensitivity, S, provides a useful measure of feedback control robustness. Substituting G_{actual} from (2.41) for G in equation (2.25), with $d = 0$ and $n = 0$, gives

$$\begin{aligned} y &= [I + (I+\Delta)^{-1}GK]^{-1}(I+\Delta)^{-1}GKy_c \\ &= [(I+\Delta)^{-1}(I+\Delta+GK)]^{-1}(I+\Delta)^{-1}GKy_c \\ &= (I+\Delta+GK)^{-1}GKy_c \\ &= \{(I+GK)[I+(I+GK)^{-1}\Delta]\}^{-1}GKy_c \\ &= [I+(I+GK)^{-1}\Delta]^{-1}(I+GK)^{-1}GKy_c \\ &= (I+\Delta_{cl})^{-1}(I+GK)^{-1}GKy_c \end{aligned} \tag{A.13}$$

where

$$\Delta_{cl} = (I + GK)^{-1}\Delta \tag{A.14}$$

This says that for the output response to commands to be insensitive to the model error, Δ_{cl} must be small. And for Δ_{cl} to be small, $(I + GK)^{-1}$ must be small.

THIS PAGE LEFT INTENTIONALLY BLANK

Appendix B
Practical Considerations

B.1 Model Simplification

In some cases, it is desirable to find a mathematical model that is a simplified version of a higher fidelity model. Such model reduction and can be performed on both linear and nonlinear models. In the early stages of control law development, it is typically used for model-order reduction of aircraft dynamics, whereas in the latter stages, it is often used for reducing the order of the resulting control laws.

The early part of this subsection deals with simplification of aircraft models, while the latter addresses model simplification in general.

B.1.1 Nonlinear Model Reduction

Nonlinear model reduction in aircraft applications is commonly used for two purposes: to separate longitudinal and lateral-directional motions into two sets and to approximate aircraft motion as motion of a point mass instead of a rigid body.

B.1.1.1 Separation of Longitudinal and Lateral-Directional Motions

Most aircraft have a plane of symmetry that separates the left and right halves of the aircraft. Thus, aircraft motion can be represented using two independent sets of equations: one for longitudinal motion and another for lateral-directional motion. Longitudinal motion occurs within the plane of symmetry of the aircraft, whereas lateral-directional motion occurs outside of this plane of symmetry.

Longitudinal motion involves the state variables u, w, q, θ for body components or V, α, q, γ for flight path components. The longitudinal forces and moments are L, D, m_a^c, m_p^c.

Lateral-directional motion involves the state variables v, p, r, ϕ for body components or β, p, r, μ for flight path components. The lateral-directional forces and moments are Y, ℓ_a^c, ℓ_p^c, n_a^c, n_p^c.

It is a consequence of the aircraft's plane of symmetry that longitudinal motions do not create lateral-directional forces or moments. Thus pure longitudinal motion is possible. In this case, the equations of motion are equations 4.36, 4.38, 4.40, 4.42, 4.44, and 4.46

for body components, and equations 4.50, 4.52, 4.54, 4.56, 4.58, and 4.60 for flight path components.

To a good approximation, lateral-directional motions do not create longitudinal forces or moments (e.g., neglect the drag produced by sideslip, aileron, or rudder). Thus pure lateral-directional motion is often assumed. In this case, the equations of motion are equations 4.35, 4.37, 4.39, 4.41, 4.43, and 4.45 for body components and equations 4.49, 4.51, 4.55, 4.57, 4.53, and 4.59 for flight path components.

B.1.1.2 Point-Mass Approximation

A common nonlinear model reduction approach in aircraft applications is to use a point-mass model in place of a rigid-body model, with the degrees of freedom associated with the body rotations in equilibrium. Moment equilibrium is accomplished by control surface or thrust vector trim settings.

In flight path components, a reduced-order model results from using equations 4.49, 4.50, and 4.51 to compute the trim settings of the controls, neglecting equations 4.55, 4.56, and 4.57 but retaining equations 4.52, 4.53, 4.54, 4.58, 4.59, and 4.60 and regarding T, μ, α, β as the inputs. This approximation is useful in trajectory control applications.

B.1.2 Linear Model Reduction

Many techniques exist for reducing the order of linear models. Assuming the model to be reduced is in first-order state-space form, as in equation 4.3, *model order* refers to the number of differential equations, the number of state variables, or the dimension of the square A matrix, all of which are identical.

The model reduction techniques to be discussed here start with some original state-space description given by

$$\begin{aligned} \dot{x}_o &= A_o x_o + B_o u \\ y &= C_o x_o + D u \end{aligned} \tag{B.1}$$

A similarity transformation, T, is introduced to define a new state variable x such that

$$x_o = Tx \tag{B.2}$$

Thus, the new state-space description in transformed coordinates becomes

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{B.3}$$

where the state-space matrices are related by

$$A = T^{-1} A_o T \tag{B.4}$$

$$B = T^{-1}B_o \tag{B.5}$$

$$C = C_oT \tag{B.6}$$

Note that the direct feedthrough term, D, remains unchanged under a similarity transformation.

Many choices are possible for the similarity transformation, two of which will be discussed in Subsections B.1.3 and B.1.5. Most model reduction methods involve partitioning the state space as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \tag{B.7}$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du \tag{B.8}$$

where x_1 are the states to be retained in the model and x_2 are the states to be discarded through either *truncation* or *residualization.*

B.1.2.1 Truncation

The basic assumptions employed for model truncation are that either the x_2 states have negligible contribution to the original model response or the x_2 states are much *slower* in responding to commands and disturbances than the x_1 states. In the latter case, the x_2 states contribute only an offset (i.e., trim) to the x_1 states and responses that we neglect in the development below.

In any case, the reduced-order model that results from truncation is

$$\begin{aligned} \dot{x}_1 &= A_{11}x_1 + B_1u \\ y &= C_1x_1 + Du \end{aligned} \tag{B.9}$$

where x_2 has been truncated.

In aircraft applications, this approximation can be used to find short-period, roll-subsidence, and dutch-roll approximations by truncating the V, γ, χ, μ ξ, η, h states and retaining the p, q, r, α, β states for flight path components.

B.1.2.2 Residualization

The fundamental assumptions employed for model residualization are that the x_2 states are much *faster* in reaching their equilibrium than the x_1 states and these faster states can be approximated by their steady-state contribution. This implies that the eigenvalues of A_{22} are large compared to those for A_{11} and are in the left half plane (i.e., stable).

Thus, the reduced-order model that results from residualization is

$$\begin{aligned} \dot{x}_1 &= \left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)x_1 + \left(B_1 - A_{12}A_{22}^{-1}B_2\right)u \\ y &= \left(C_1 - C_2A_{22}^{-1}A_{21}\right)x_1 + (D - C_2A_{22}^{-1}B_2)u \end{aligned} \tag{B.10}$$

where x_2 has been residualized.

Residualization produces a reduced-order model that has the same steady-state gain for y/u as the original model, but the high-frequency gain is usually quite different.

In aircraft applications, residualization is often used to eliminate or reduce the number of the elastic modes. With rigid-body models and flight path components, this approximation can be used to find phugoid- and spiral-mode approximations by residualizing the p, q, r, α, β states after first truncating the χ, ξ, η, h states and retaining the V, γ, μ states.

Residualization is also often used for eliminating high-frequency dynamics of control compensators arising from H_∞ or μ-Synthesis designs described in Section 11.

B.1.3 Eigenvalue Realizations

From linear algebra theory, we know that we can quasi-diagonalize the original A matrix using a similarity transformation. This results in the so called Jordan form discussed in many textbooks [Stewart 1973, Strang 1976, Kailath 1980]). In practice, obtaining this particular similarity transformation can be difficult due to numerical roundoff problems.

Fortunately, in most model reduction applications, it is not necessary to find the similarity transformation that reduces A to Jordan form. Rather, it suffices to find a similarity transformation that block diagonalizes the A matrix.

B.1.4 Block-Diagonal Realizations

Sometimes it is desirable to separate the state space into slow and fast modes. This can be accomplished using a Schur decomposition, which is well conditioned numerically. The Schur decomposition produces an upper triangular A matrix with different options for state ordering. Next, a Sylvester equation is solved to find another similarity transformation to zero out the off-diagonal block, leaving a block-diagonal matrix.

The objective here is to decompose a given transfer-function matrix

$$G(s) = C(sI - A)^{-1}B + D = G_1(s) + G_2(s) \tag{B.11}$$

where

$$G_1(s) = C_1(sI - A_1)^{-1}B_1 + D_1 \tag{B.12}$$

$$G_2(s) = C_2(sI - A_2)^{-1}B_2 + D_2 \tag{B.13}$$

$$\lambda[A_1] \in S = \text{Subset of the complex plane} \tag{B.14}$$

$$\lambda[A_2] \in \bar{S} = \text{Complement of S} \tag{B.15}$$

An approach to performing this decomposition is to first obtain a Schur vector matrix for A with the Schur vector ordering consistent with the desired set S. This ordering can be accomplished for either S = left half plane or S = inside the unit disk. These subsets are shown in Figure B.1. The Schur decomposition of A is given by

$$U^{-1}AU = \Lambda \tag{B.16}$$

where U is the (unitary) Schur vector matrix and Λ is an upper triangular matrix.

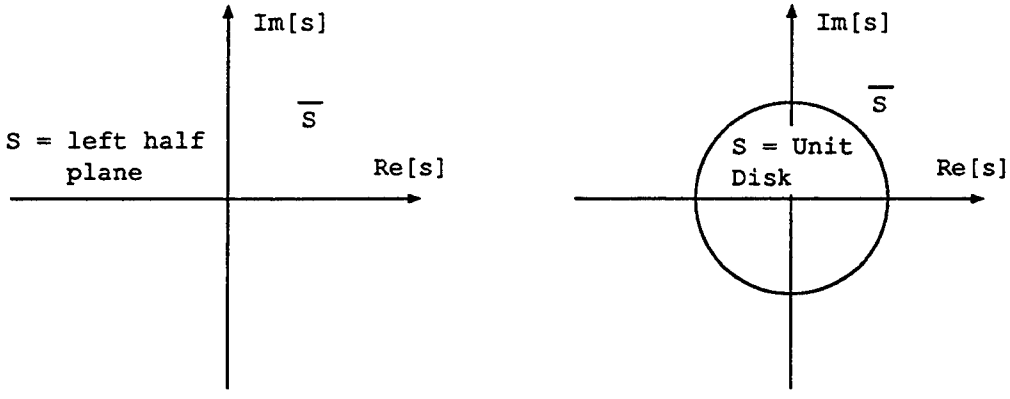


Figure B.1: Sets for Schur Vector Ordering.

More general decompositions of the complex plane can be obtained by making use of two facts from linear algebra:

$$U^{-1}(\alpha I + A)U = \alpha I + \Lambda \tag{B.17}$$

$$U^{-1}(\frac{1}{\beta}A)U = \frac{1}{\beta}\Lambda \tag{B.18}$$

where α and β are scalars.

By exploiting these equalities, the decompositions of the complex plane shown in Figure B.2 can be accomplished. The appropriate Schur vector ordering (for the decompositions shown in Figure B.2) is obtained by finding the Schur vectors of either $\alpha I + A$ for the half-plane set S or $\frac{1}{\beta}A$ for the disk set S and noting that the Schur vectors obtained are also Schur vectors of A (but with the desired ordering).

We next perform the similarity transformation,

$$\left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] = \left[\begin{array}{c|c} U^{-1}AU & U^{-1}B \\ \hline CU & D \end{array}\right] = \left[\begin{array}{cc|c} A_{11} & A_{12} & \tilde{B}_1 \\ 0 & A_{22} & \tilde{B}_2 \\ \hline \tilde{C}_1 & \tilde{C}_2 & D \end{array}\right] \tag{B.19}$$

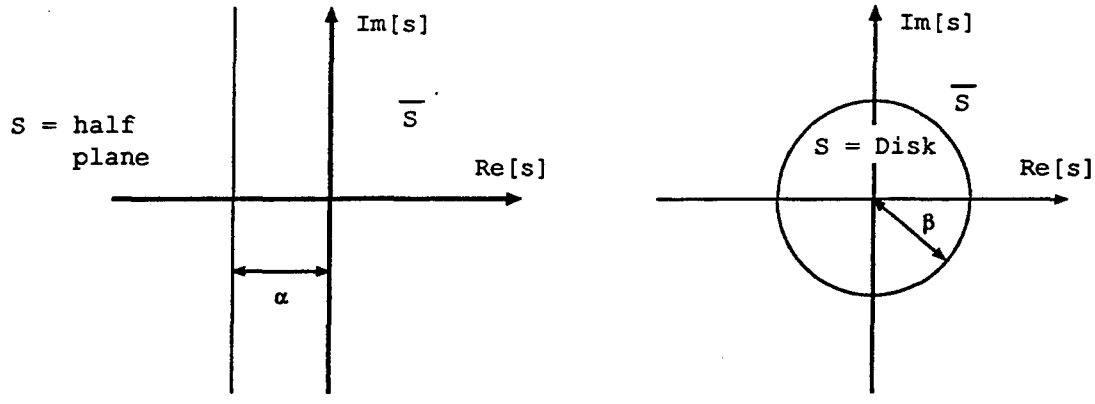


Figure B.2: Modified Sets for Schur Vector Ordering.

where the new $\tilde{A}$ matrix is upper triangular since U is a Schur vector matrix. Here the partitions are assumed consistent with the set S.

The next step is to eliminate the off-diagonal block, A_{12}. To do so, we now consider a second similarity transformation matrix

$$T = \begin{bmatrix} I & T_{12} \\ 0 & I \end{bmatrix} \tag{B.20}$$

where the submatrix T_{12} will be derived next. The second similarity transformation results in

$$\left[\begin{array}{c|c} T^{-1}\tilde{A}T & T^{-1}\tilde{B} \\ \hline \tilde{C}T & D \end{array}\right] = \left[\begin{array}{cc|c} A_{11} & A_{11}T_{12} - T_{12}A_{22} + A_{12} & \tilde{B}_1 - T_{12}\tilde{B}_2 \\ 0 & A_{22} & \tilde{B}_2 \\ \hline \tilde{C}_1 & \tilde{C}_1T_{12} + \tilde{C}_2 & D \end{array}\right] . \tag{B.21}$$

The upper-right block of the new A matrix can be zeroed by solving the Sylvester equation

$$A_{11}T_{12} - T_{12}A_{22} + A_{12} = 0 \tag{B.22}$$

which is a linear equation for T_{12} that always has a solution whenever the eigenvalues of A_{11} and A_{22} are distinct.

In conclusion, transforming the original state-space matrices with the product of the two similarity transformations, UT, gives the desired decomposition:

$$\left[\begin{array}{c|c} (UT)^{-1}A(UT) & (UT)^{-1}B \\ \hline C(UT) & D \end{array}\right] = \left[\begin{array}{cc|c} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ \hline C_1 & C_2 & D \end{array}\right] \tag{B.23}$$

Note that the advantage of this approach is that U is a unitary matrix and hence poses fewer numerical problems than an eigenvector matrix, which in general is not unitary.

B.1.5 Balanced Realizations

Balanced realizations [Moore 1981] provide a powerful method for reducing the order of a model. A balancing transformation results from solving two Lyapunov equations and then finding the eigenvector matrix for the product of the two solutions. The method also provides a bound for the error obtained by truncation of the balanced realization [Enns 1984, Glover 1984]. A refinement to the method exists that allows manipulation of this error bound with respect to frequency, which is often useful in control applications [Enns 1984].

It has often been said that one drawback of the balanced realization method is that the "balanced states" no longer have any physical meaning, which is in contrast to truncation or residualization of *physical states* discussed in Subsections B.1.2.1 and B.1.2.2. This drawback can actually be removed as follows.

After finding the reduced-order model via "balancing," all of the original states can be approximated in terms of the truncated "balanced" states; that is, the states of the original model, x_o (those with physical meaning), are related to the balanced states, x_1, through a similarity transformation. Since the reduced-order model truncates some of the balanced states, the original states are given approximately by

$$x_o = Tx = \begin{bmatrix} T_1 & T_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cong T_1 x_1 \tag{B.24}$$

Furthermore, if r is the order of the reduced-order model, then some set of the r *original* states can be in the reduced-order model. This is accomplished by first defining another similarity transformation, T_{1_r}, that selects the r rows of T_1 corresponding to the r original states of interest (i.e., $x_{r_o} = T_{1_r} x_1$). Provided these states are selected such that T_{1_r} is nonsingular, the resulting reduced-order model can be expressed in terms of the r original states of interest, x_{r_o}, by

$$\dot{x}_{r_o} = T_{1_r}^{-1} A_{11} T_{1_r} x_{r_o} + T_{1_r}^{-1} B_1 u \tag{B.25}$$

$$y = C_1 T_{1_r} x_{r_o} + Du \tag{B.26}$$

where

$$\left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array}\right] \tag{B.27}$$

is a balanced realization of the original model. It should be noted that such transformations from balanced coordinates back to original coordinates may degrade numerical conditioning of the well-conditioned balanced model.

B.2 Gain Scheduling

In many design methodologies, control laws are designed for each flight condition; that is, there is a distinct set of "gains" (compensators or state-space matrices) that define control laws for each flight condition. In such cases, these control laws must be scheduled for operation throughout the flight envelope.

It is common to schedule these gains with variables that define the flight condition. For example, if flight conditions are defined in terms of altitude and velocity, then it is natural to think of the gains as functions of altitude and velocity. However, gains also depend on requirements; that is, a higher bandwidth control system (with improved tracking and disturbance rejection) implies larger gains, independent of flight condition.

When requirements are constant across flight condition, it is possible to draw some conclusions about how proportional-plus-integral gains vary with flight condition for special cases of aerodynamic control surfaces and thrust vectoring.

To facilitate the discussion that follows, we refer to the generic control block diagram shown in Figure B.3. This control law assumes both an aerodynamic surface (u_1) and thrust vectoring (u_2), and includes proportional (k_1, k_2) and integral (k_3) gains, a lead-lag compensator (k_4, k_5), and a low-pass filter (k_6, k_7). There are two sensor measurements (y_1, y_2) and one command (y_c).

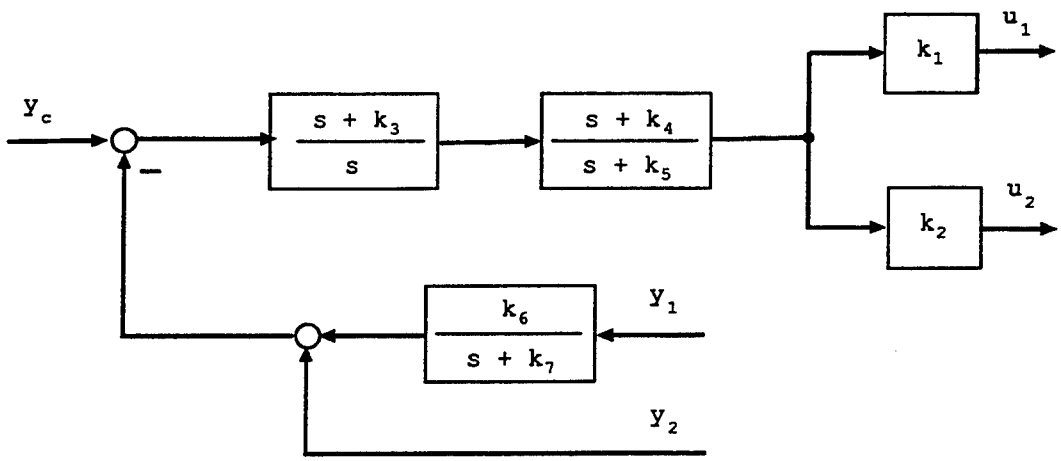


Figure B.3: Generic Control Law Gains.

If the control effector is an aerodynamic control surface (e.g., u_1), it is reasonable (based on aerodynamics) to assume that proportional gain (k_1) will vary inversely with dynamic pressure. Because aerodynamics vary with Mach number (other than for low subsonic conditions), it is also reasonable to expect that gains will vary with Mach number. On the other hand, if the control effector is thrust vectoring (e.g., u_2), it is reasonable to expect that proportional gain (k_2) will vary inversely with thrust or power setting for small deflection angles. The integral gain can often be mechanized or implemented in

such a way that it is independent of flight condition (i.e., k_3 = constant). Interpretations are less clear, however, when gains correspond to compensator poles and zeros (e.g., k_4, k_5, k_6, k_7).

In any case, gain scheduling has historically been posed as a correlation problem; that is, find the correlation between gains (or dependent variables) in the control law and measured (independent) variables that correspond to the flight condition. The correlation problem can be approached with either a formal regression/curve-fitting approach or an informal "seat of the pants" approximation that fits the data. A formal correlation approach to gain scheduling was employed for the F-8 Oblique Wing Research Aircraft [Enns 1987c]. Here 22 gains were scheduled, of which 15 could be correlated with seven independent gains that were fit to straight lines (either linear or logarithmic) as a function of impact pressure and wing sweep angle.

The worst fear for the gain scheduler is that plots of gains versus independent scheduling variables have the look of a "shotgun blast." This may result if the underlying aircraft dynamics vary significantly with flight condition in ways uncorrelated with the scheduling variables. But the design technique together with the requirements used can also lead to scheduling difficulties. In particular, if requirements are changed as part of the design process, discontinuously from one flight condition to the next, there is no reason to expect gain scheduling to be continuous.

The gain scheduling process usually works quite well in practice, despite its ad hoc flavor. Some of the reasons it works so well include:

- the feedback aspects of control make it more tolerant of compensator inaccuracies,
- the independent variables used for correlation with control system gains vary slowly relative to the primary flight control variables,
- the independent variables and the form of the gain schedules used in practice are really just an approximation to a more formal dynamic inversion control design approach.

B.2.1 Gain Scheduling Examples

In addition to the correlation example discussed above, three other examples are presented below.

B.2.1.1 Example 1: Averaged State-Space Models

One approach to gain scheduling that has been applied successfully with with many techniques, including μ-Synthesis, is to utilize an averaged or mean flight condition with provisions for the dependence of aerodynamic effectiveness on dynamic pressure. In

particular, this approach has been employed in a study of localizer capture and track using lateral directional models of the B-737 aircraft.

Here it was assumed that flight conditions are all low subsonic, with aerodynamic control surfaces, and that only dynamic pressure variations are to be considered as the independent variable for gain scheduling; that is, the dependence of gains on weight, cg position, flap setting, and anything else are neglected.

Aircraft state-space matrices A_i, B_i, C_i, D_i for each flight condition (i) were averaged over all $n = 14$ flight conditions after first scaling the B_i and D_i matrices by inverse dynamic pressure ($\bar{q}_{ref}/\bar{q}_i$). This averaged model

$$\begin{bmatrix} A_{avg} & B_{avg} \\ C_{avg} & D_{avg} \end{bmatrix} = \frac{1}{n} \sum_{i=1}^{n} \begin{bmatrix} A_i & B_i \frac{\bar{q}_{ref}}{\bar{q}_i} \\ C_i & D_i \frac{\bar{q}_{ref}}{\bar{q}_i} \end{bmatrix} \tag{B.28}$$

was used for a single-point μ-Synthesis control design. The resulting control law (K_{avg}) was then scheduled for each flight condition by reintroducing dynamic pressure via

$$K_i = K_{avg} \frac{\bar{q}_{ref}}{\bar{q}_i} \tag{B.29}$$

This approach worked quite well for dynamic pressures between 44 and 153 lbs/ft^2, even with different flap and gear settings. The scheduled design was later evaluated for all 14 flight conditions. Both time and frequency responses showed compliance with requirements, and in many cases the curves all fell within a small band of one another. Figure B.4 shows roll-rate error response to a step lateral stick input. Note how the gain schedule produces a similar transient for all 14 flight conditions.

B.2.1.2 Example 2: Averaged Short-Period Model

Another approach to autoland flare control for the MD-11 aircraft used average short-period stability derivatives across 18 flight conditions on a glide slope 50 ft above the ground, with different weights, cg, and flap settings. In this case, constant gains were utilized and bandwidth was allowed to vary with flight condition.

B.2.1.3 Example 3: Explicit Gain Derivation

Yet another approach to gain scheduling is to derive control gains directly from aircraft data and control system requirements. This is appropriate if a simple design rule can accommodate the requirements.

For the special case of short-period stabilization and normal acceleration and/or pitch-rate control (actually MCV; see Section 5.3), such a design rule has been used effectively for a homing missile, a hypersonic vehicle during ascent to orbit [Enns 1990b], and a supersonic high-dynamic-pressure condition for the F-15 aircraft.

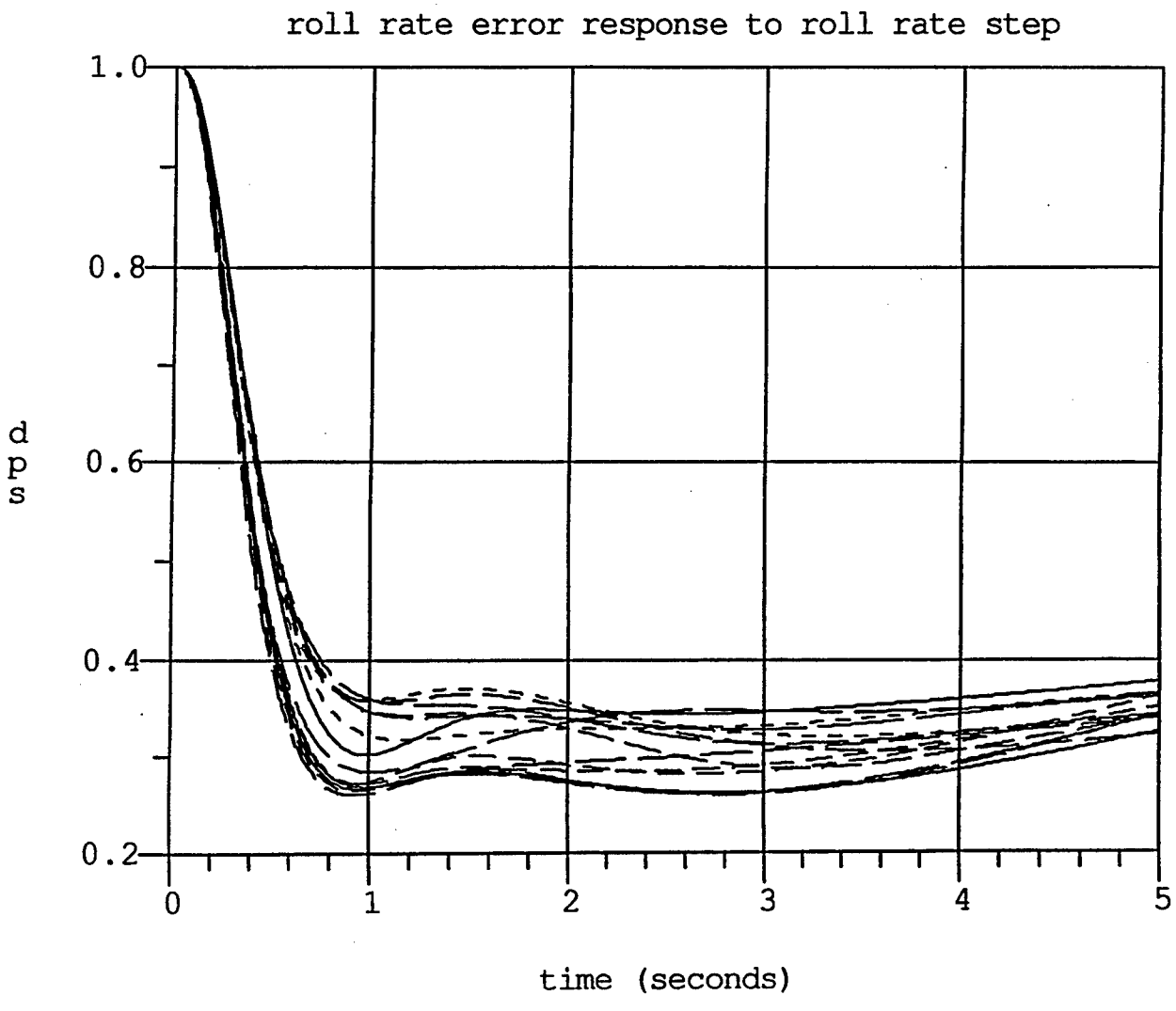


Figure B.4: Roll-Rate-Error Responses for 14 Different Flight Conditions.

In this case, it is a SISO algebraic design rule that relates control system requirements and short-period stability derivatives. The requirements are Bode loop crossover frequency, phase margin, and crossover velocity in the definition of MCV of Section 5.3. Crossover frequency and phase margin are robust stability requirements, whereas crossover velocity is a handling qualities requirement related to speed of normal acceleration response versus pitch rate overshoot.

In another study [Nesline 1984], a homing missile was successfully stabilized using a short-period design rule that scheduled short-period stability derivatives and in turn control gains with estimated cg position. In this study, the airframe open-loop stability varied from stable (short-period frequency of 30 rad/sec) to unstable (unstable pole at +20 rad/sec) and pitch control power varied by a factor of 2 over the 10-second boost portion of the trajectory.

As part of the National Aerospace Plane (NASP) program, 16 ascent and 14 reentry flight conditions were examined with this design rule. The parameters for requirements were held constant, and raw short-period stability derivatives, rather than gains, were scheduled as a function of flight condition. This approach produced satisfactory stability and handling quality results for an extensive range of Mach numbers and dynamic pressure.

B.2.2 Dynamic Inversion and Gain Schedules

Most of the need for gain schedules is eliminated with the dynamic inversion design technique. However, like the design rule approach discussed above, there is still a need to schedule or store raw aircraft modeling data. Although the raw data may well consume as much flight control computer memory as conventional gain schedules, the data stored is *directly* related to aircraft open-loop dynamics, as opposed to a combination of aircraft dynamics, control design technique, and control system requirements. The dynamic inversion approach can be viewed as a systematic approach to gain scheduling.

B.2.3 Alternate Approaches

Other systematic approaches to gain scheduling are discussed in the literature.

For example, one approach is based on structured singular value theory with repeated blocks [Packard 1994, Becker 1994]. In this approach, one block is associated with the aircraft dynamics and another is associated with a gain schedule in the control law. The schedule compensates for the change associated with the aircraft dynamics, so that the two blocks are equal but time varying.

A formal approach for gain scheduling is to develop a nonlinear control law such that the closed-loop linearization has specified properties at equilibrium points [Rugh 1991a, Rugh 1991b]. A necessary and sufficient condition for the existence of a nonlinear con-

troller with required operating point family and linearization family is that there exists a solution to a certain linear partial differential equation [Rugh 1991b]. This approach has been applied to short-period control of a missile where the aerodynamic coefficients are nonlinear functions of angle of attack and Mach number.

Another approach is based on the structure of the H_∞ optimal controller [Reichert 1992]. This structure is defined in terms of state-space matrices A, B, C, D for the aircraft linearized model at an operating point, two Riccati equation solutions (X_∞, Y_∞), and the optimal H_∞ norm (γ). The basic idea is to use this structure to define the control law at any design point. To simplify the real-time implementation, the design synthesis steps are performed off line and X_∞, Y_∞, γ are recorded for a few design points, which are then interpolated for the implementation. This approach has been applied to a missile short-period control law design with nonlinearities in angle of attack and Mach number.

B.3 Nonlinearities and Limits

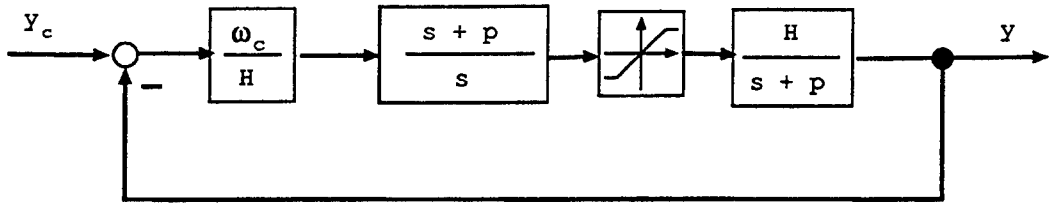
All real-world systems are nonlinear. Aircraft equations of motion include nonlinearities both in their kinematics (i.e., rotating coordinate systems) and in their dynamics (i.e., aerodynamics). Aerodynamic nonlinearities can usually be linearized about a particular flight condition, whereas those due to rotating coordinate systems are negligible for straight-and-level flight. Even for highly dynamic maneuvers, these kinematic nonlinearities can often still be linearized about a "nominal" maneuver trajectory. But other nonlinearities are associated with aircraft controls (e.g., aircraft control surface limits) that cannot be linearized in the usual static sense.

B.3.1 SISO Examples Motivate MIMO Discussion

Recall that Section 3 presented a simple short-period example that demonstrated the potential pitfalls associated with control effector limits. As that example showed, a simple saturation or clipping of each control is often a convenient design approach. When this does not work well, the problem becomes more difficult. This is especially true in the multivariable case, where there can be nonnegligible coupling between controls and performance objectives. It can even be nontrivial in the SISO case, as the next few examples will illustrate.

B.3.1.1 Example 1

Consider the design for a low-pass model of the aircraft dynamics that might correspond to a roll command augmentation system. The objective is to implement a proportional-plus-integral compensation with antiwindup prevention as the lagged, limited, *positive* feedback shown in Figure B.5. This is a proven practical implementation.



(Equivalent Diagrams When Not Limited)

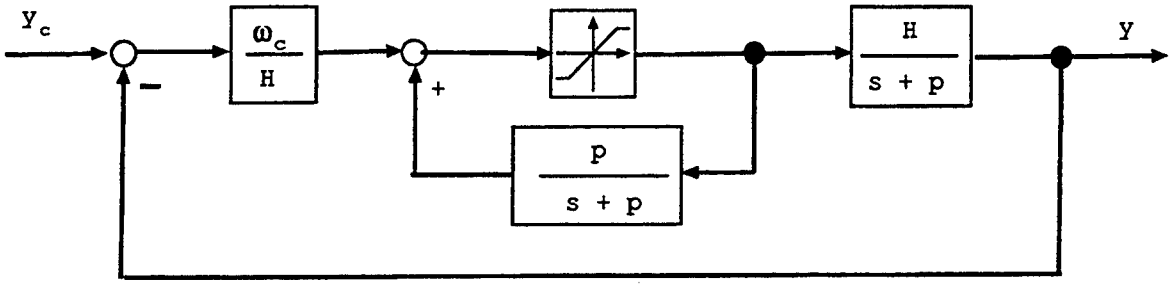


Figure B.5: Lagged, Limited, Positive Feedback Implementation of P+I.

It is of interest to observe that this block diagram is equivalent to the diagram in Figure B.6, which is from a theoretical formulation called Internal Model Control (IMC) [Morari 1989]. In general, G is the plant operator and Q the control law operator. In particular, if $G = \frac{H}{s+p}$ and $Q(s) = \frac{\omega_c}{H}\frac{(s+p)}{s+\omega_c}$, then the IMC diagram of Figure B.6 is equivalent to the proportional-plus-integral diagram of Figure B.5. Note that the control law utilizes a model of the plant. With this formulation, it can be shown that the control system is stable in the presence of limits for any stable Q if the open-loop system, G, is stable. This theoretical formulation is of particular interest because it is applicable to the multivariable case as well as SISO.

A multivariable implementation of the proportional-plus-integral compensation shown in Figure B.5 was also developed for a helicopter flight-test experiment [Enns 1987a]. The helicopter was the AH-64A, and there were four loops — pitch, roll, yaw and vertical translation motion.

We now consider the case where there is a dead zone in the actuation of the control surface, as Figure B.7 illustrates. Two approaches to implementation of the proportional-plus-integral compensation are shown. Both are equivalent for the case of no limiting or dead zone ($h = 0$), The first implementation without feedback of the actuation achieves zero steady-state error, even in the presence of the dead zone. However, the lagged, limited, positive feedback implementation does not achieve zero steady-state error due to the dead zone.

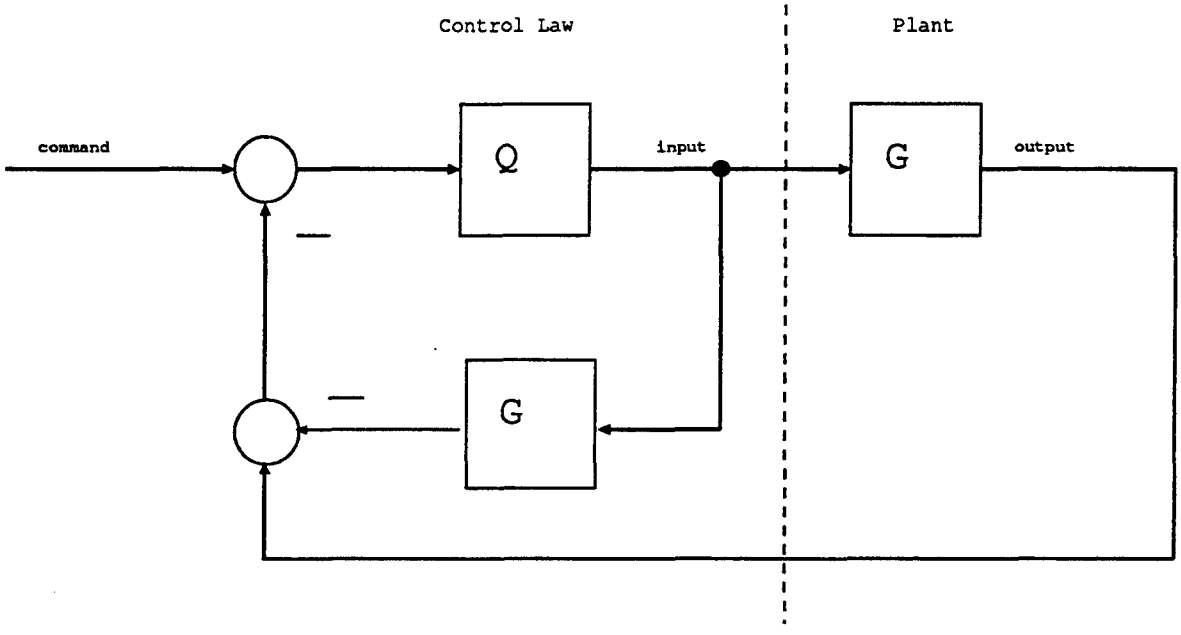


Figure B.6: Internal Model Control (IMC).

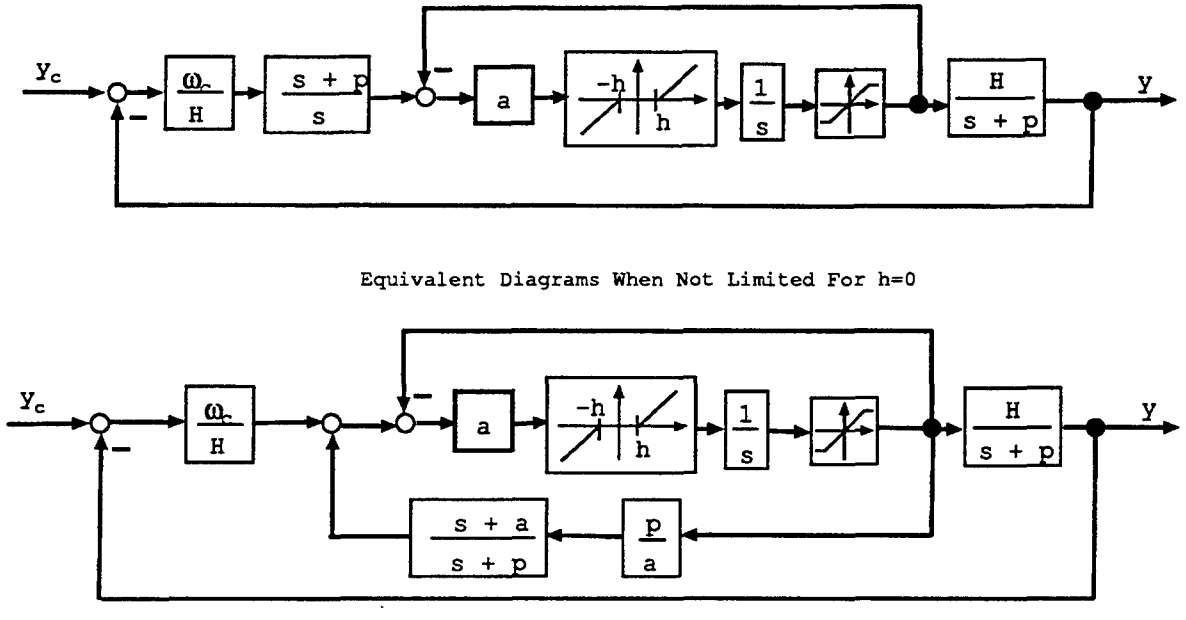


Figure B.7: Implementations of P+I With Actuator Deadband.

B.3.1.2 Example 2

We next consider a double-integrator plant with limited input, as shown in Figure B.8, where feedback of position and velocity are employed. In this case, the timing of control reversals is significant for large amplitude behavior. In this case, simple limiting of a standard second-order feedback law produces many overshoots for large step changes (see solid lines in Figure B.9).

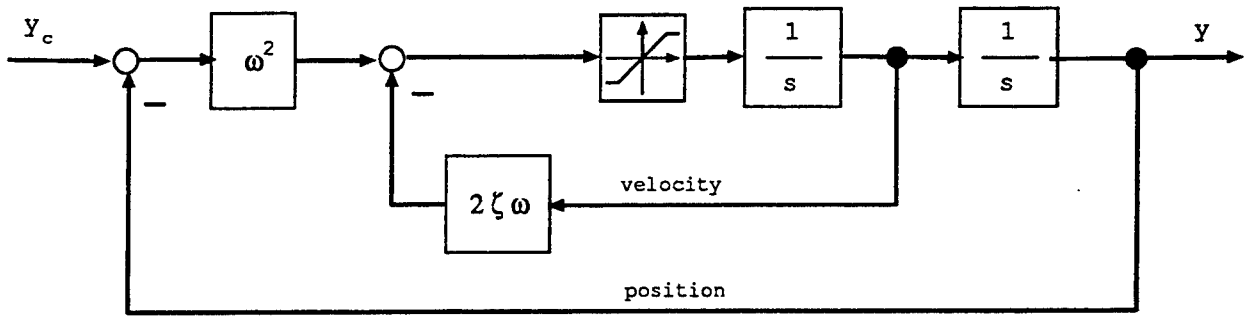


Figure B.8: Standard Linear Control Law for Double-Integrator Plant.

An alternative double-integrator design shown in Figure B.10 employs a bang-bang, minimum-time solution whenever tracking errors in position and velocity lie outside of an elliptical region centered about the current position command and zero velocity. The minimum time solution is developed in textbooks [Athans 1966, Ogata 1970]) and involves a parabolic switch curve in the phase plane. Linear control is used within the elliptical region. Results of a time history using this control strategy are shown with the dashed lines in Figure B.9.

Smooth blending between time-optimal control and linear control has been applied in a directed energy weapon [Harvey 1992] and other applications.

B.3.1.3 Example 3

Phase-plane-based time-optimal control has been applied to aircraft short-period response [Buffington 1996]. The theory for an unstable case with two real roots (documented in the text [Athans 1966]) was combined with the switch from time-optimal control to linear control upon entering a time-invariant set [Buffington 1996]. The time-invariant set just described is analogous to the ellipse in Figure B.10, where any initial condition inside the set remains within the set as the error decays to zero for a fixed command or set point. Introduction of a ramp command change can be accommodated with a similar control structure by defining tracking-error rate in terms of commanded (ramp) rate and actual rate. A constant acceleration can also be included, but the formal assumptions do not hold for arbitrary time-varying commands. This approach was applied to full longitudinal motion with scheduling based on short-period stability derivatives. It worked quite well,

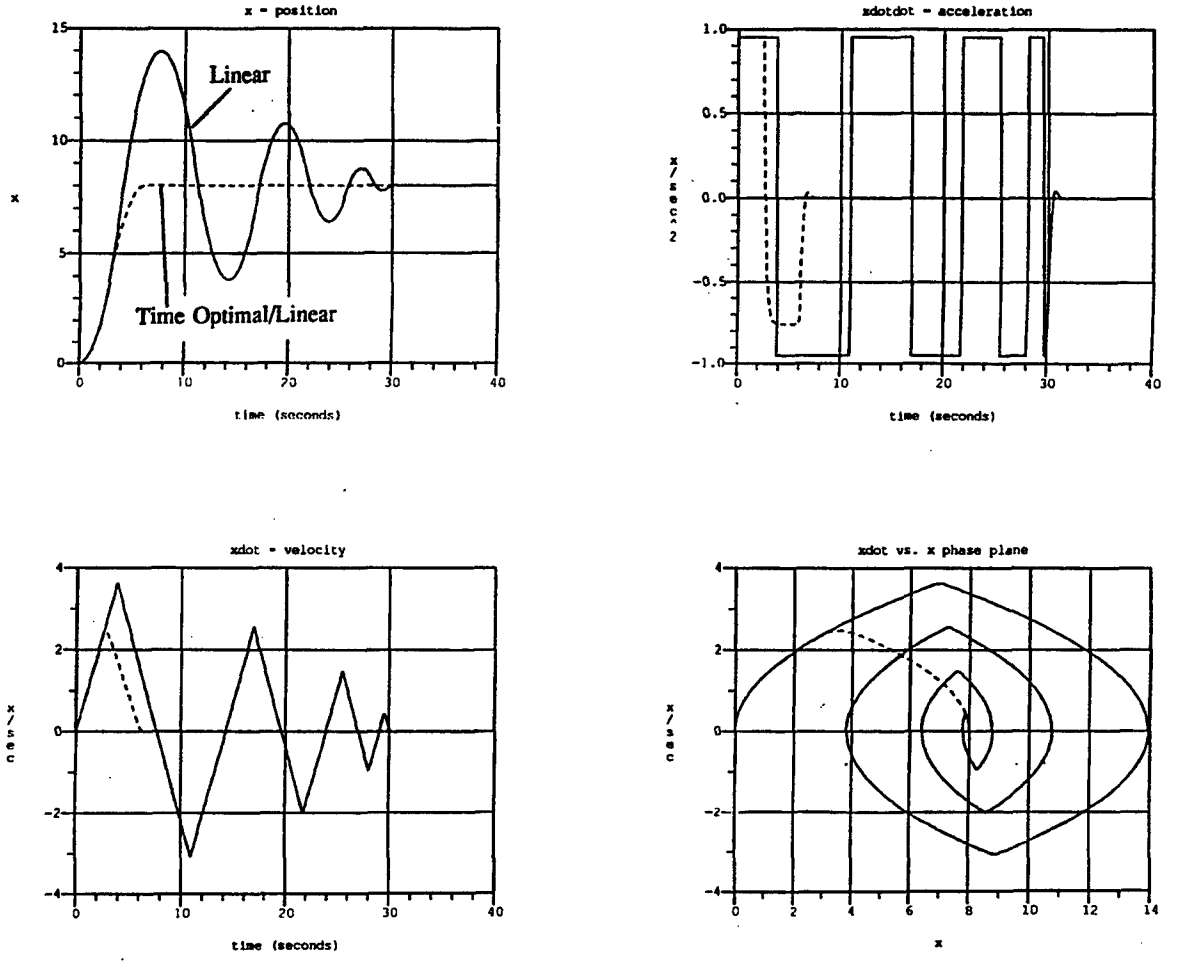
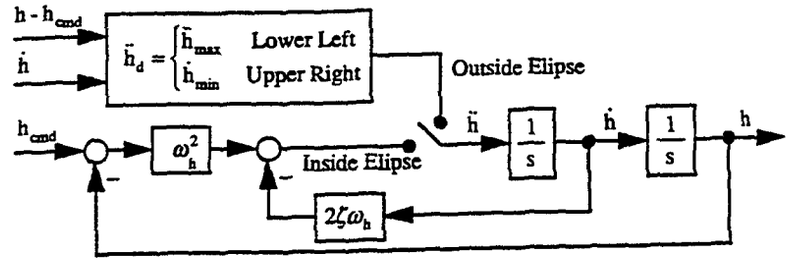


Figure B.9: Time Histories From a Double-Integrator Simulation With Saturation.



$$\ddot{h} = \frac{L\cos\gamma}{m} - g + \frac{V^2\cos^2\gamma}{r} - \frac{D\sin\gamma}{m} + 2\omega_E V\sin\chi\cos\gamma\cos(\text{lat})$$

$$L = \bar{q}SC_L(M, \alpha)$$

$$D = \bar{q}SC_D(M, \alpha)$$

V, γ, h, m known $\Rightarrow$ $\bar{q}$, M, r

and solve for α such that $\ddot{h} = \ddot{h}_{desired}$

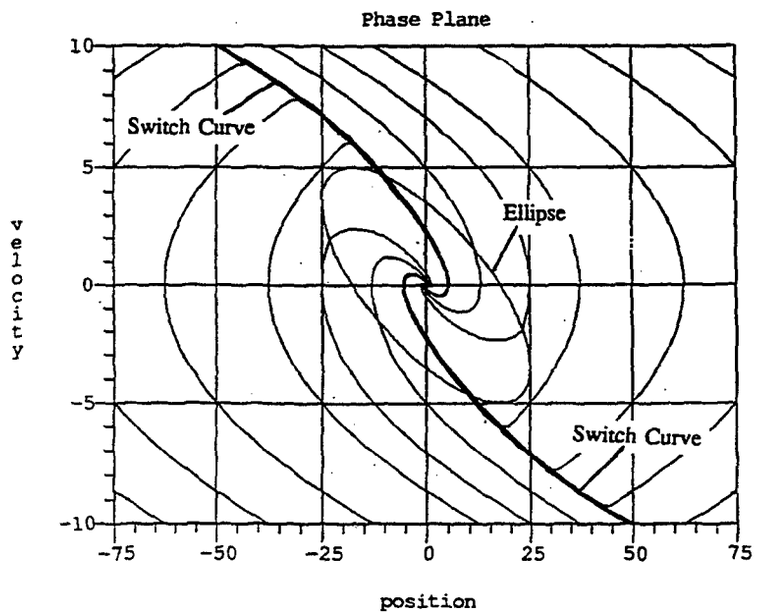


Figure B.10: Time-Optimal/Linear Control Law for Double-Integrator Plant.

despite violating the assumptions of linear equations with constant stability derivatives subject to deflection limits on the single longitudinal control.

B.3.2 MIMO Discussion

B.3.2.1 Phase-Plane-Based Switch Between Time-Optimal/Linear Control

The phase-plane-based approach has been applied to aircraft trajectory control of translational acceleration for both vertical and horizontal (normal to velocity) motion. The equations of motion for these two accelerations were inverted algebraically according to a phase-plane-based time-optimal (linear) control switch for outside (inside) a prescribed, and possibly invariant, set. There were two position errors, two rate errors, and two acceleration commands — one each for vertical and horizontal degrees of freedom. Some interesting issues arise since acceleration limits are not independent but can be traded off one for another. This is illustrated in Figure B.11, where the controls are angle of attack and bank angle. The boundary also depends on airspeed and flight path and is not constant. The boundary shown is for a particular flight condition with angle of attack and bank angle limits. This shows that it is possible to sacrifice horizontal acceleration for vertical acceleration, depending on relative priority.

B.3.2.2 Axis Prioritization

One important observation that can be drawn from the previous illustration is that whenever multiple control effectors are available but limited, they can be better used to accomplish a single objective than two or more control objectives. Whenever a control system is intended to operate at its limits, this issue must be considered in the synthesis process. In the case of angular-rate flight control, this principle is termed *axis prioritization.* There are multiple approaches, depending on the strategy. One approach is to minimize a quadratic function of the unsatisfied angular acceleration whenever the controls are position or rate limited [Bugajski 1992a]. The relative weighting between roll rate and yaw rate (i.e., turn coordination) is adjusted with a design parameter and proved quite effective.

For general three-axis motion of an airplane, all three axes are coupled together, and the strategy requires computing lateral-directional differential tail limits as a result of the higher-priority pitch commands. One approach to axis prioritization for three axes is to give pitch highest priority, followed by turn coordination, and finally roll [Bugajski 1992a]. The control laws are implemented separately for longitudinal and lateral-directional axes. In the digital implementation, the longitudinal calculations are performed first. Then, the remaining differential tail available for lateral-directional control is evaluated and lateral-directional calculations are performed.

The coupling between lateral-directional control effector limits is illustrated graphically

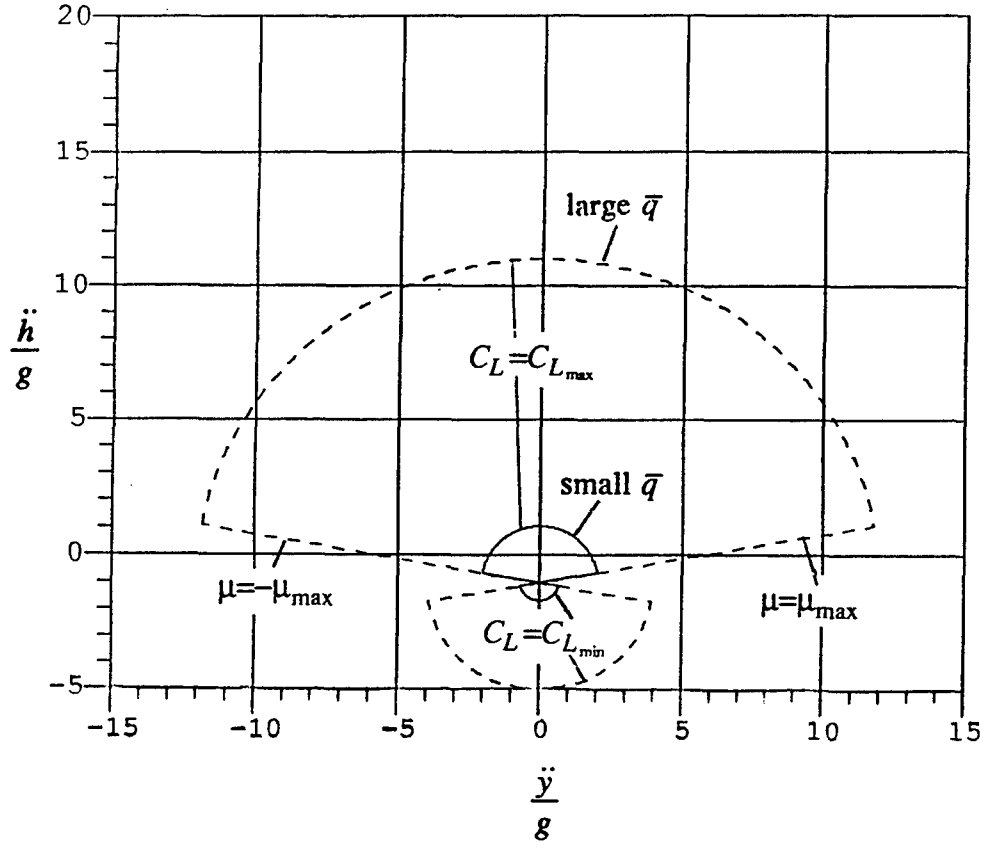


Figure B.11: Horizontal and Vertical Acceleration Limits.

in Figures B.12 and B.13. Here we represent roll acceleration, y_1, and yaw acceleration, y_2, as mappings of the limited controls — aileron, u_1, and rudder, u_2, for the former, with the addition of differential tail, u_3, as a third control for the latter. Angular accelerations are achievable if they lie inside the boundary, and are not if they lie outside the boundary.

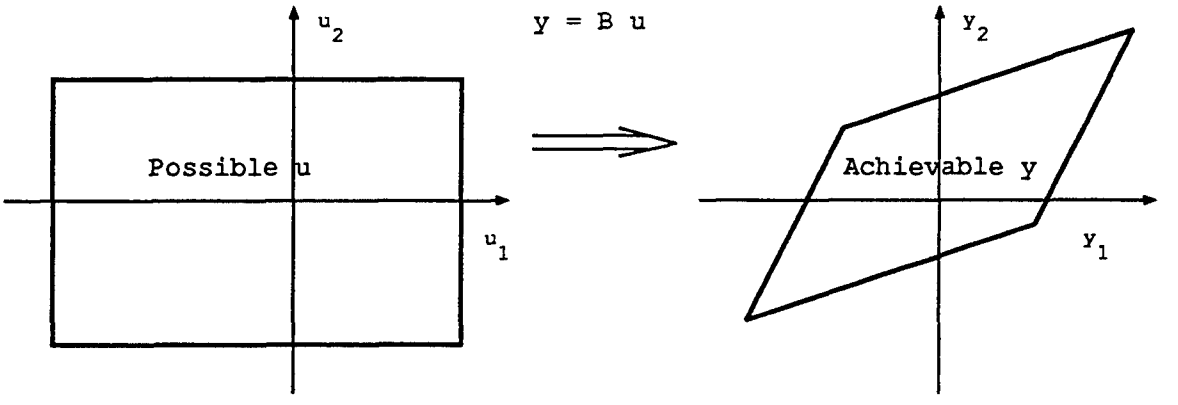


Figure B.12: Mapping by B of Two Inputs to Two Outputs.

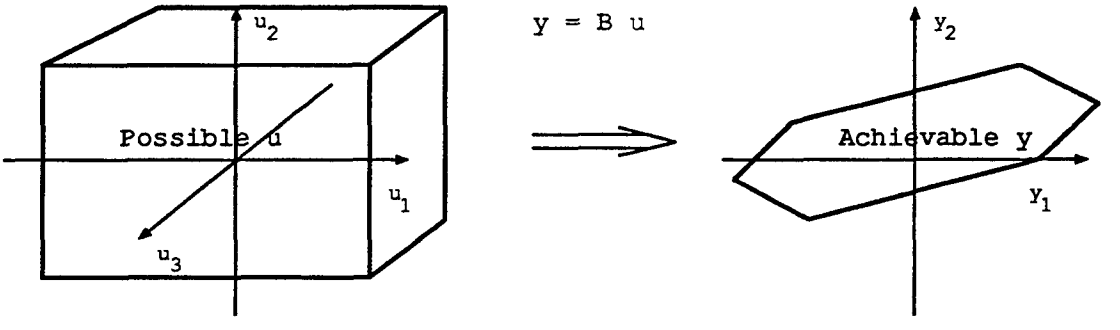


Figure B.13: Mapping by B of Three Inputs to Two Outputs.

Similar pictures can be developed for limits of quantities other than angular acceleration; for example, turn coordination requires inclusion of sideslip angular rate. For other applications, translational accelerations may be limited, such as in short-takeoff-and-vertical-landing aircraft configurations. In general, these accelerations are referred to as desired accelerations. Many of these ideas can be generalized to multiple dimensions, but it becomes much more difficult to illustrate them graphically. Theory and approaches for computing the boundaries, as well as three-dimensional illustrations, have been developed [Durham 1994].

B.3.2.3 Projections and Constrained Optimization

These types of illustrations motivate the use of projections. The simplest approach for equal priority among axes is illustrated as Projection 1 in Figure B.14. Here we scale the unachievable control back to the origin until we reach the boundary and then

issue that desired acceleration, rather than the unachievable acceleration (i.e., desired y). Algorithms for performing this projection to the boundary are available [Durham 1994].

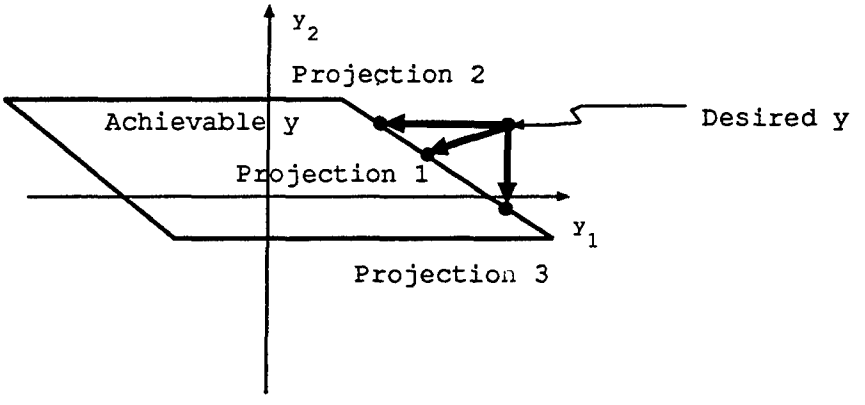


Figure B.14: Three Examples of Projections From Desired to Achievable.

Other projections are also possible, such as Projections 2 and 3 in Figure B.14 that satisfy one control objective exactly at the expense of the other. The choice of possible projections are many, and axis prioritization requirements can also be addressed with projections derived from constrained optimization algorithms. One approach is to minimize the sum of weighted absolute values of unsatisfied desired accelerations. Linear programming can be utilized to minimize this sum subject to specified control effector bounds.

A weighted sum can also be formed by using the square of the unsatisfied desired accelerations in place of the absolute value employed in Subsection 10.3.2. Additional quadratic terms can also be included to incorporate strategies that utilize redundancy in controls when limiting is not the issue.

B.3.2.4 Other Issues

Additional issues arise with respect to computational complexity. Timing estimates (in terms of number of floating-point operations) are available for linear programming and quadratic optimization based on the number of desired accelerations and control effectors. Actual computation times depend on the particular processor used, the details of the software implementation, and the programming language. This is an area of active research, and there is insufficient evidence to date to favor one algorithm over another.

Some complications associated with multivariable extensions of these ideas for general aircraft applications are that assumptions are often violated and requirements are difficult to precisely state. An example of the former concerns the assumption of linear equations with constant bounded inputs. In actuality, these equations are often nonlinear and the bounds are not constant but vary with state. Moreover, often commands are not constant, but time varying, and do not satisfy constant rate or acceleration requirements. This

violates assumptions for the phase-plane-based switch between time-optimal and linear control. An example of the latter concerns axis prioritization, where it is often not clear whether the requirement should be posed in terms of the one- or two-norm of some signal, or something entirely different.

B.4 Describing Function Analysis

One of the characteristics of nonlinear system behavior that cannot be explained by linear theory is the potential for a self-excited oscillation called a *limit cycle*. The primary mathematical theory used for assessing the potential for such behavior is *describing function analysis* [Gelb 1968, Graham 1961].

The describing function method assumes that in a nonlinear SISO feedback loop, we can isolate the nonlinear element from the linear element, as shown in Figure B.15. The method is a frequency domain technique that approximates the transfer function of the nonlinear element by its gain and phase with respect to a sinusoidal input, as a function of the amplitude (and sometimes frequency) of the input. Classical application of the method is restricted to nonlinearities that are amplitude dependent but insensitive to frequency. In such cases, we can represent the nonlinearity via an amplitude-dependent function, $N(a)$. Assuming a sinusoidal input to the nonlinearity, $u = a\sin\omega t$, with amplitude a and frequency ω, we can express the output of the nonlinearity as a Fourier series

$$\begin{aligned} v &= f(u) = f(a\sin\theta), \quad \theta = \omega t \\ &= c + \sum_{n=1}^{\infty}(a_n \sin\theta + b_n \cos\theta) \end{aligned} \tag{B.30}$$

where the Fourier coefficients for the fundamental component are given by

$$a_1 = \frac{1}{2\pi}\int_0^{2\pi} f(a\sin\theta)\sin\theta d\theta \tag{B.31}$$

$$b_1 = \frac{1}{2\pi}\int_0^{2\pi} f(a\sin\theta)\cos\theta d\theta \tag{B.32}$$

and the dc component is given by

$$c = \frac{1}{2\pi}\int_0^{2\pi} f(a\sin\theta)d\theta \tag{B.33}$$

The describing function of the nonlinear element then is defined as the ratio of the fundamental response to the input

$$N(a) = \frac{a_1 + jb_1}{a} \tag{B.34}$$

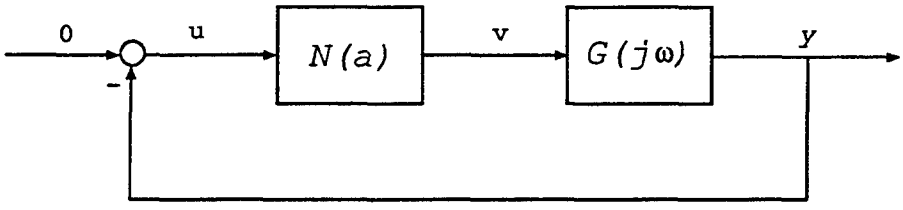


Figure B.15: Nonlinear Feedback Control System With Isolated Nonlinearity.

where $N(a)$ is a complex number corresponding to the equivalent gain (real and imaginary) of the nonlinearity. Typically $N(a)$ is expressed as an equivalent magnitude and phase

$$|N(a)| = \frac{\sqrt{a_1^2 + b_1^2}}{a} \tag{B.35}$$

$$\angle N(a) = \arctan(b_1/a_1) \tag{B.36}$$

Stability of the nonlinear feedback loop in Figure B.15 can be assessed (approximately) with a Nyquist like criterion that counts encirclements of a critical "point."[1] Here the usual critical point $1 + j0$ must, however, be replaced by the line segment $-1/N(a)$, $0 < a < \infty$; that is, we look for intersections where $G(j\omega) = -1/N(a)$ for some frequency ω and amplitude a. Such crossings indicate only the potential for a limit cycle. Depending on the direction of the crossing, the potential limit cycle may be either stable, in which case it persists, or unstable, in which case it dies out. If stable, the theory predicts that a limit cycle will occur at the fundamental frequency ω with an amplitude (at the input to the nonlinearity) of a. The rule of the thumb for limit cycle stability is illustrated for a memoryless nonlinearity, which has amplitude but no phase, in Figure B.16. Basically, the *limit cycle is stable if in traveling along the curve of increasing $G(j\omega)$ on a Nyquist plot, the curve for $-1/N(a)$ crosses from right to left with increasing amplitude a.*

Describing function analysis is, of course, only approximate. Its validity is critically dependent upon the amount of attenuation in the magnitude of the transfer function $G(j\omega)$ for frequencies around any potential limit cycle frequencies (i.e., frequencies where $G(j\omega)$ and $-1/N(a)$ cross). This attenuation, coupled with the fact that most nonlinearities also attenuate higher order harmonics, is necessary to ensure that higher order harmonics fed back around to the input of the nonlinearity are significantly attenuated. If not, these higher order harmonics will in turn generate new fundamental frequencies at the output of the nonlinearity, with their own higher order harmonics ad infinitum. This invalidates the assumption of a dominant fundamental frequency and brings validity of the limit cycle analysis into question. A common cause for inadequate attenuation in $G(j\omega)$ is structural flexibility.

This issue of flexibility was particularly troublesome in limit cycle analyses conducted

[1] A more rigorous assessment of stability is possible using the circle criterion [Safonov 1978], but this method is very conservative.

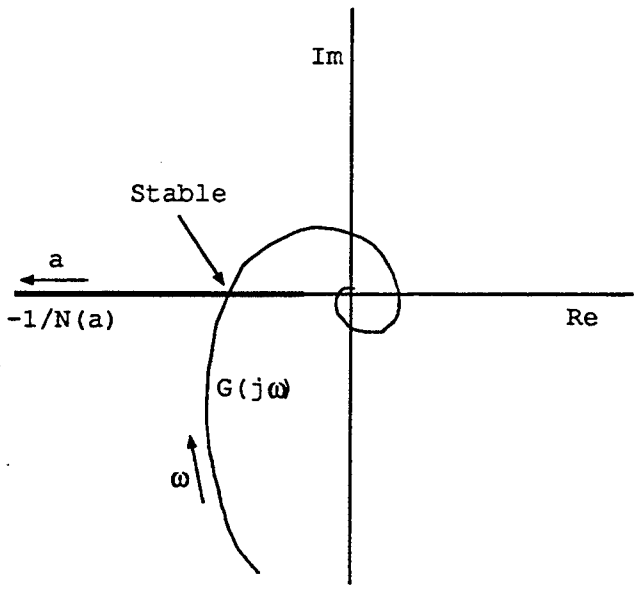


Figure B.16: Limit Cycle Stability With Describing Function Analysis.

for the Space Shuttle's reaction control system during entry [Barrett 1982]. That paper also deals with a minimum on-time constraint for the reaction jets, which produces a frequency-dependent nonlinearity $N(a,\omega)$, for which it is no longer possible to separate the frequency dependence of $G(j\omega)$ from the amplitude dependence of $-1/N(a)$. Here we must deal with the product $N(a,\omega)G(j\omega)$ as an overall frequency response, parameterized by limit cycle amplitude a. Alternatively, we may continue to treat the linear and nonlinear elements as separate entities by plotting $G(j\omega)$ as a function of ω versus $-1/N(a,\omega)$, plotted as a function of a but parameterized by ω, as was done in the paper. In either case, limit cycle analysis is very tedious and would be totally impractical without a computer. The paper also examines a more sophisticated method for limit cycle analysis in the absence of higher-order harmonic attenuation. It is known as *Tsypkin's method* [Gelb 1968], and we will address it next.

B.4.1 Tsypkin's Method

This method is an exact stability analysis procedure that is applicable only to very simple nonlinearities such as relays. It requires no assumptions regarding linear loop attenuation because it carries along all significant terms in the Fourier series expansions of all signals. It begins by assuming an explicit periodic waveform for a potential limit cycle that might be generated by the nonlinearity in question, as illustrated in Figure B.17 for the case of a relay with deadband d and hysteresis h. This waveform is then represented as the infinite Fourier series expansion

$$v(t;\omega,\rho) = \frac{4D}{\pi} \sum_{n\ odd} \frac{\sin(n\rho\pi/2)}{n} \mathrm{Re}\{e^{jn(\omega t-\pi/2)}\} \qquad \text{(B.37)}$$

where D is the output level of the relay output, $\omega \equiv 2\pi/T$ is the frequency, and ρ for $0 \leq \rho \leq 1$ is the duty cycle of the periodic waveform. Because the waveform is π-symmetric, the expansion contains only odd terms and no dc term. The relay output, when passed through the linear system $G(j\omega)$, in turn generates the input to relay with the (exact) Fourier expansion

$$u(t;\omega,\rho) = -\frac{4D}{\pi}\sum_{nodd}\frac{\sin(n\rho\pi/2)}{n}\mathrm{Re}\{G(jn\omega)e^{jn(\omega t-\pi/2)}\} \tag{B.38}$$

For a limit cycle to exist, it is apparent from the figure that the following conditions must hold at and between switching times[2]:

$$\begin{array}{lll} (1) & u(t_{on}) = d, & \dot{u}(t_{on}) > 0 \\ (2) & u(t_{off}) = d-h, & \dot{u}(t_{off}) > 0 \\ (3) & u(t) > d-h, & t\varepsilon(t_{on}, t_{off}) \\ (3) & u(t) < d, & t\varepsilon[0, t_{on}) \cup (t_{off}, T/2] \end{array} \tag{B.39}$$

for some $\omega \geq 0$ and $0 \leq \rho \leq 1$.

The equality parts of conditions (1) and (2) are verified by evaluating the Fourier expansion for $u(t;\omega,\rho)$ for $t = t_{on}$ and $t = t_{off}$ and comparing them against the trip levels d and $d-h$, respectively. This is easily done by plotting $y(t_{on})$ versus $y(t_{off})$ as a function of ρ and parameterized by discrete values of ω. The critical point $(d, d-h)$ is also displayed on the same plot to facilitate locating intersections. Exact intersections will likely not occur for discrete values of ω, so a finer sweep in frequency ω may be necessary for frequencies where potential intersections occur. This graphical solution only identifies potential limit cycles. To determine if a potential limit cycle is real, we must verify that the inequalities in conditions (1) and (2) also hold at the on and off times, respectively. In addition, we must do a sweep through time for one limit cycle to ensure that $u(t)$ does not violate the trip levels of conditions (3) and (4). If all of these conditions are satisfied, a limit cycle is unavoidable. The advantage of Tsypkin's method over describing function analysis is that it is exact, subject, of course, to the reliability of all models. The disadvantage is that it lacks the intuitive feel associated with describing function analysis.

[2]For simplicity, we have omitted the minimum on-time constraint.

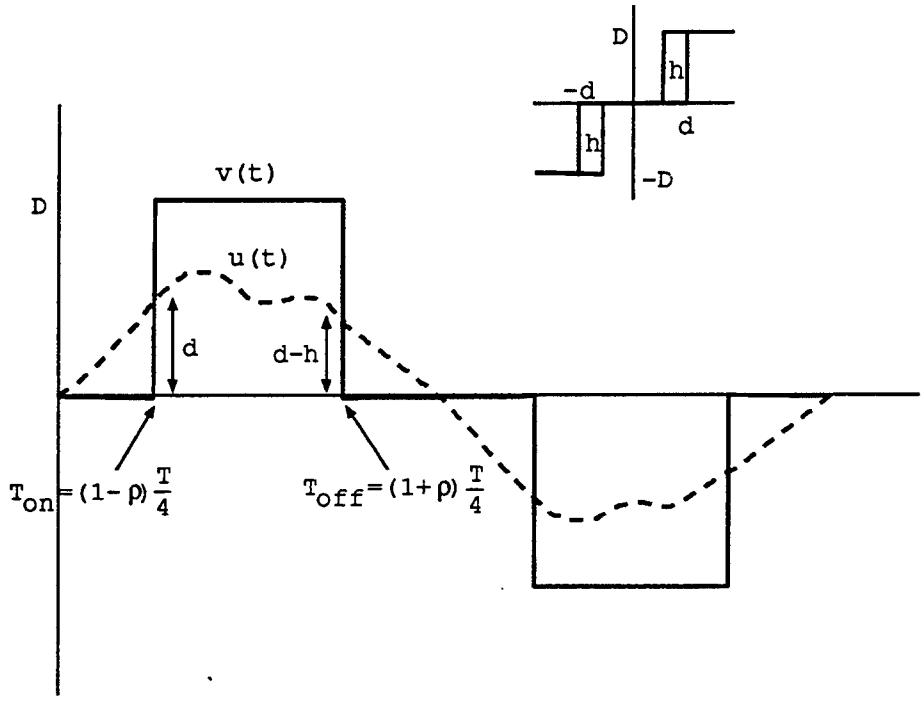


Figure B.17: Limit Cycle Waveform With Tsypkin's Method.

THIS PAGE LEFT INTENTIONALLY BLANK

Bibliography

[Ackerman 1985] Ackerman, J.E., *Sampled-Data Control Systems,* Springer Verlag, Berlin, 1985.

[Ackerman 1992] Ackerman, J.E., "Does it Suffice to Check a Subset of Multilinear Parameters in Robustness Analysis?" IEEE TAC-37(4), April 1992.

[Anderson 1986] Anderson, B.D.O., "Weighted Hankel-Norm Approximation: Calculation of Bounds, Systems and Control Letters, Vol. 7, 1986.

[Andry 1983] Andry, A.N., E.Y. Shapiro and J.C. Chung, "On Eigenstructure Assignment for Linear Systems," IEEE TAES-19, September 1983.

[Asseo 1970] Asseo, S.J., "Application of Optimal Control to Perfect Model Following," J.Aircraft, Vol 7(4), July-August 1970.

[Astrom 1990] Astrom, K., and B. Wittenmark *Computer-Controlled Systems,* 2nd edition, Prentice Hall, Engle Cliffs, NJ. 1990.

[Athans 1966] Athans, M., and Peter L. Falb, *Optimal Control,* McGraw-Hill Book Company, 1966.

[Athans 1971] Athans, M., Special Issue on Linear Quadratic Gaussian Problems, IEEE TAC-16(6), December 1971.

[Axsater 1966] Axsater, S., "Suboptimal Time-Variable Feedback Control of Linear Dynamical Systems with Random Inputs," Int. J. Control, vol. 4(6), 1966.

[Banda 1991] Banda, S.S., H.H. Yeh and S.A. Heise, "A Surrogate System Approach to Robust Control Design," Int. J. System Science, Vol. 22(1), 1991.

[Barmish 1985] Barmish, B.R., "Necessary and Sufficient Conditions for Quadratic Stabilizability of an Uncertain Linear System," J. Opt. Theory and Appl., Vol. 46, 1985.

[Barrett 1982] Barrett, M. F., "Nonlinear Methods for Stability Analysis of the Space Shuttle Lateral-Axis Control System," 35th Guidance and Control Panel Symposium on Advances in Guidance and Control Systems, Lisbon, Portugal, Oct. 12-14, 1982.

[Becker 1994] Becker, G. and A. Packard, "Robust Performance of Linear Parameter Varying Systems Using Parametrically Dependent Linear Feedback," Systems and Control Letters, Volume 23, pp. 205-215, 1994.

[Bernstein 1990] Bernstein, D.S. and D.C. Hyland, "The Optimal Projection Approach to Robust, Fixed -Structure Control Design," Mechanics and Control of Space Structures, J.L. Junkins, editor, AIAA, 1990.

[Bernstein 1989] Bernstein, D.S. and W.M. Haddad, "LQG Control with H_∞ Performance Bound: A Riccati Equation Approach," IEEE TAC-34(3), 1989.

[Bisplinghoff 1955] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., *Aeroelasticity,* Addison-Wesley, 1955.

[Bisplinghoff 1962] Bisplinghoff, R. L. and Ashley, H., *Principles Of Aeroelasticity,* Dover, 1962.

[Blakelock 1965] Blakelock, J.H., *Automatic Control of Aircraft and Missiles,* John Wiley & Sons, New York, 1965.

[Blakelock 1991] Blakelock, J.H., *Automatic Control of Aircraft and Missiles,* John Wiley & Sons, New York, 1991.

[Bode 1945] Bode, H. W., *Network Analysis and Feedback Amplifier Design,* D. Van Nostrand, 1945.

[Boppe 1980] Boppe, C. W., "Transonic Flow Field Analysis For Wing-Fuselage Configurations," Grumman, NASA CR 3243, 1980.

[Boyd 1991] Boyd, S.P. and C.H. Barratt, *Linear Controller Design, Limits of Performance,* Prentice Hall, 1991.

[Boyd 1989] Boyd, S.P. and Q. Yang, "Structured and Simultaneous Lyapunov Functions for System Stability Problems," Int. J. Control, Vol. 49, 1989.

[Boyd 1992] Boyd, S.P., V. Balakrishnan and L. El Ghaoui, "Computing Bounds for the Structured Singular Value Via Interior Point Algorithms," American Control Conference, Chicago 1992.

[Broussard 1978] Broussard, John R., O'Brien, Mike J., "Feedforward Control to Track the Output of a Forced Model," 18th Conference on Decision and Control, 1978, pp1149-1155.

[Buffington 1996] Buffington, James M., "Control Design and Analysis for Systems with Redundant Limited Controls," Ph. D. dissertation, Control Science and Dynamical Systems, University of Minnesota, March 1996.

[Bugajski 1992a] Bugajski, D.J., D.F. Enns, R. C. Hendrick, "Nonlinear Control Law Design for High Angle-of-Attack," NASA High-Angle-of-Attack Projects and Technology Conference, NASA Conference Publication Cp-3137, Volume 4, April, 1992.

[Bugajski 1992b] Bugajski, D.J., D.F. Enns, "Nonlinear Control Laws with Applications to High Angle-of-Attack Flight," J. Guidance Control and Dynamics, Vol 15(3), May-June, 1992.

[Calvo-Ramon 1986] Calvo-Ramon, J.R., "Eigenstructure Assignment by Output Feedback and Residue Analysis," IEEE TAC-31(3), 1986.

[Chang 1991] Chang, B.C., O. Ekdal, H.H. Yeh and S.S. Banda, "Computation of the Real Structured Singular Value via Polytopic Polynomials," J. Guidance, Control and Dynamics, Vol. 14(1), January-February 1991.

[Dahleh 1987] Dahleh, M.A. and J.B. Pearson, "ℓ_1-Optimal Feedback Controllers for MIMO Discrete-Time Systems," IEEE TAC-32(4), April 1987.

[deGaston 1988] deGaston, R.R.E. and M.G. Safonov, "Exact Calculation of the Multiloop Stability Margin," IEEE TAC-33, February 1988.

[DeMarco 1990] DeMarco, C.L., V. Balakrishnan and S. Boyd, "Branch and Bound Methodology for Matrix Poly-

tope Stability Problems," IEEE CDC, Honolulu, 1990.

[Desoer 1975] Desoer, C. A., Vidyasagar, M., *Feedback Systems: Input-Output Properties* Academic Press, 1975.

[Doyle 1978] Doyle, J.C. "Guaranteed Margins for LQG Regulators," IEEE TAC-23(4), August 1978.

[Doyle 1981] Doyle, J.C. and G. Stein, "Multivariable Feedback Design: Concepts for a Classical / Modern Synthesis," IEEE TAC-26(1), February 1981.

[Doyle 1982a] Doyle, J.C., "Analysis of Feedback Systems with Unstructured Uncertainties," IEE Proceedings, 129-D(6), November 1982.

[Doyle 1982b] Doyle, J.C., J.E. Wall, G. Stein, "Performance and Robustness Analysis for Structured Uncertainty," Proceedings of the IEEE 20th Conference on Decision and Control, December, 1982.

[Doyle 1983] Doyle, J.C., "Synthesis of Robust Controllers and Filters," IEEE Conference on Decision and Control, San Antonio, TX, 1983.

[Doyle 1985] Doyle, J.C., Cheng-Chih Chu, "Matrix Interpolation and H_∞ Performance Bounds," 1985 American Control Conference Proceedings, Boston, 19-21 June 1985.

[Doyle 1987] Doyle, J.C., Packard, A., "Uncertain Multivariable Systems from a State Space Perspective," 1987 American Control Conference Proceedings, Minneapolis, MN, 10-12 June 1987.

[Doyle 1989] Doyle, J.C., K. Glover, P.P. Khargonekar and B.A. Francis, "State Space Solutions to Standard H_2 and H_∞ Control Problems," IEEE TAC-34(8), August 1989.

[Doyle 1992] Doyle, J. C., Francis, B., A., Tannenbaum, A. R., *Feedback Control Theory,* Macmillan, 1992.

[Durham 1994] Durham, W. C., "Constrained Control Allocation: Three-Moment Problem," Journal of Guidance, Control, and Dynamics, Volume 17, Number 2, March-April, 1994.

[Elgersma 1991] Elgersma, M., G. Stein, M. Jackson and J. Yeichner, "Robust Controllers for Space Station Momentum Management," IEEE CDC, Brighton 1991.

[Englehart 1991] Englehart, M., D.F. Enns, "A Design Comparison Between Dynamic Inversion and μ-Synthesis," Honeywell Internal Memo, August 1991.

[Enns 1982] Enns, D. F., "Analysis Of The Flight Control Laws Of The Forward Swept Wing Aircraft," Honeywell ADG Report F0858-SR7, October 1982.

[Enns 1984] Enns, D. F., "Model Reduction For Control Design," Ph.D. dissertation, Dept. Of Aeronautics And Astronautics Engineering, Stanford University, June 1984.

[Enns 1987a] Enns, D. F., "Multivariable Flight Control for an Attack Helicopter," IEEE Control Systems Magazine, Volume 7, Number 2, April, 1987.

[Enns 1987b] Enns, D. F., "Analysis Of Control Laws For An Unstable Flexible Aircraft," 1987 AIAA Guidance, Navigation and Control Conference, August 1987.

[Enns 1987c] Enns, D. F., and D. J. Bugajski, "Development of Control Law Methodology for the Oblique Wing Research Aircraft," Honeywell ADG Report No. F0275-SR1, December, 1987.

[Enns 1990a] Enns, D.F., "Robustness Of Dynamic Inversion v.s. μ-Synthesis: Lateral-Directional Flight Control Example," Proceedings of the 1990 AIAA Guidance, Navigation, and Control Conference, August 1990.

[Enns 1990b] Enns, D.F., R. Hendrick, J. Caton and R. Roatcap, "A Longitudinal Flight Control Law Design Method with Application to a NASP Configuration," Paper No. 169, Ninth National Aero-Space Plane Symposium, Orlando, November 1990.

[Enns 1991] Enns, D.F., "Rocket Stabilization as a Structured Singular Value Synthesis Design Example," IEEE Control Systems Magazine, Vol. 11, No. 4, June 1991.

[Enns 1994] Enns, D.F., D.J. Bugajski, R.C. Hendrick, G. Stein, "Dynamic Inversion: An Evolving Methodology for Flight Control Design," International Journal of Control, January, 1994.

[Etkin 1959] Etkin, B., *Dynamics of Flight,* John Wiley & Sons, New York, 1959.

[Etkin 1972] Etkin, B., *Dynamics of Atmospheric Flight,* John Wiley & Sons, New York, 1972.

[Etkin 1982] Etkin, B., *Dynamics of Flight - Stability & Control,* John Wiley & Sons, New York, 1982.

[Franklin 1990] Franklin, G.F., J.D. Powell, and M.L. Workman, *Digital Control of Dynamic Systems,* 2nd edition, Addison Wesely, Reading, MA, 1990.

[Freudenberg 1985] Freudenberg, J. S. and D. P. Looze, "Right Half Plane Poles and Zeros and Design Tradeoffs in Feedback Systems," IEEE TAC-30(6), June 1985.

[Garg 1989] Garg, S. "Stability Robustness Improvement of Direct Eigenspace Assignment Based Feedback Systems using Singular Value Sensitivities," American Control Conference, Pittsburg, 1989.

[Garrard 1989] Garrard, William L., Low, Eicher, Prouty, Scott, "Design of Attitude and Rate Command Systems for Helicopters Using Eigenstructure Assignment," AIAA Journal of Guidance, Control and Dynamics, Vol 12, No. 6. Nov/Dec 1989, pp. 783-791.

[Gelb 1968] Gelb, A. and W. E. Vander Velde, *Multiple-Input Describing Functions and Nonlinear System Design,* McGraw-Hill, 1968.

[Giesing 1976] Giesing, J. P., Kalman, T. P., and Rodden, W. P., "Correction Factor Techniques for Improving Aerodynamic Prediction Methods," McDonnell Douglas Corporation, Douglas Aircraft Company, NASA CR-144967, May 1976.

[Glover 1984] Glover, K., "All Optimal Hankel-Norm Approximations of Linear Multivariable Systems, and Their L_∞ Error Bounds," International Journal of Control, Vol. 39, 1984.

[Graham 1961] Graham, D. and D. McRuer, *Analysis of Nonlinear Control Systems,* John Wiley & Sons, New York, 1961.

[Gupta 1980] Gupta, N.K., "Frequency-Shaped Cost Functionals: Extension of Linear-Quadratic-Gaussian Design Methods," J. Guidance and Control, Vol 3(6), November-December, 1980.

[Harvey 1978] Harvey, C.A. and G. Stein, "Quadratic Weights for Asymptotic Regulator Properties," IEEE TAC-23(3), June 1978.

[Harvey 1992] Harvey, C.A. and D. J. Bugajski, "Smooth Blend of Time-Optimal and Linear Control," Journal of Guidance, Control, and Dynamics, Volume 15, Number 2, March-April 1992.

[Herbst 1980] Herbst, W. B., "Future Fighter Technologies," Journal of Aircraft, Volume 17, No. 8, pp. 561-566, August 1980.

[Herbst 1983] Herbst, W. B., "Dynamics of Air Combat," Journal of Aircraft, Volume 20, No. 7, pp. 594-598, July, 1983.

[Horowitz 1963] Horowitz, I. M., *Synthesis of Feedback Systems,* Academic, 1963.

[Horowitz 1979] Horowitz, I.M., "Quantitative Synthesis of Uncertain Multiple-Input Multiple-Output Feedback Systems," Int. J. Control, Vol. 30(1), 1979.

[Hung 1982] Hung, Y.S. and A.G.J. MacFarlane, *Multivariable Feedback: A Quasi-Classical Approach,* Springer Verlag, 1982.

[Hyland 1982] Hyland, D.C., "Maximum Entropy Stochastic Approach to Controller Design for Uncertain Structural Systems," American Control Conference, Arlington, 1982.

[Jackson 1990] Jackson, M.R., and D.F. Enns, "Lateral-Directional Control Of An Aircraft Using μ Synthesis," 1990 AIAA Guidance, Navigation, and Control Conference, Portland, OR, 1990.

[James 1947] James, H. M., Nichols, N. B., Phillips, R. S., *Theory of Servomechanisms,* McGraw-Hill, 1947.

[Jordan 1953] Jordan, P. F., "Aerodynamic Flutter Coefficients For Subsonic, Sonic, And Supersonic Flow (Linear Two Dimensional Theory)," R. & M. No. 2932, British Aeronautical Research Council, 1957. See also: Royal Aircraft Establishment Report Structures 141, 1953.

[Kailath 1980] Kailath, T., *Linear Systems,* Prentice Hall, New Jersey, 1980.

[Karpel 1980] Karpel, M., "Design For Active And Passive Flutter Suppression And Gust Alleviation," Ph.D. Dissertation, Dept. of Aeronautics And Astronautics Engineering, Stanford University, August 1980.

[Khou 1993] Khou, K., "Frequency Weighted L_∞ Norm and Optimal Hankel Norm Model Reduction," To appear, 1993.

[Krause 1988] Krause, J., B. Morton, D.F. Enns, G. Stein, J.C. Doyle, A. Packard, "A General Statement of Structured Singular Value Concepts," 1988 American Control Conference, Atlanta, GA, 1988.

[Lallman 1985] Lallman, Frederick J., "Relative Control Effectiveness With Application to Airplane Control," NASA TP-2416, April 1985.

[Latham 1986] Latham, G.A., B.D.O. Anderson, "Frequency Weighted Hankel-Norm Approximation of Stable Transfer Function," Systems and Control Letters, Vol. 5, 1986.

[Levine 1970] Levine, W.S. and M. Athans, "On the Determination of the Optimal Constant Output Feedback Gains for Linear Multivariable Systems," IEEE TAC-15(1), February 1970.

[Lockheed Martin 1994] Lockheed Martin, "Application of Multivariable Control Theory to Aircraft Control Laws Interim Report 1: Definition of Control Problem and Requirements," Air Force Contract No. F33615-92-C-3607, July 1994.

[Mayne 1973] Mayne, D.Q. "The Design of Linear Multivariable Systems," Automatica, Vol. 9, 1973.

[McFarlane 1992] McFarlane, D. and K. Glover, "A Loop Shaping Design Procedure Using H_∞ Synthesis," IEEE TAC-37(6), June 1992.

[McRuer 1973] McRuer, D., I. Ashkenas, D. Graham, *Aircraft Dynamics and Automatic Control,* Princeton University Press, Princeton, New Jersey, 1973.

[MIL-STD-1797 1987] Military Standard, "Flying Qualities of Piloted Vehicles." MIL-STD-1797, 31 March 1987.

[Moore 1981] Moore, B.C., "Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction," IEEE Transactions on Automatic Control, Vol. AC-26, February 1981.

[Moorhouse 1982] Moorhouse, D. J., Woodcock, R. J. "Present Status of Flying Qualities Criteria For Conventional Aircraft," AGARD Conference Proceedings No. 333 on "Criteria for Handling Qualities of Military Aircraft," April 1982.

[Morari 1989] Morari, M. and E. Zafirou, *Robust Process Control,* Prentice Hall, 1989.

[Morton 1985] Morton, B., McAfoos, R, "A Mu-Test for Real-Parameter Variations," Proceedings of the American Control Conference, Boston, June, 1985.

[Morton 1993] Morton, B., D. Enns and E. Zafirou, Honeywell Internal Memo, 1993.

[Morton 1994] Morton, B., D. Enns and Bing-Yu Zhang, Institute for Math and Its Applications, University of Minnesota, IMA Preprint Series # 1245, July, 1994.

[Moynes 1992] Moynes, J. and G. Stein, "The Approach for B-2 Flight Control Algorithms," American Control Conference, Chicago 1992.

[Nesline 1984] Nesline, F.W., and M.L. Nesline "How Autopilot Requirements Constrain the Aerodynamic Design of Homing Missiles," American Control Conference, 1984.

[Ogata 1970] Ogata, K., *Modern Control Engineering,* Prentice-Hall, Inc., 1970.

[Packard 1988] Packard, A., M.K.H. Fan and J. Doyle, "A Power Method for the Structured Singular Value," IEEE CDC, Austin, 1988.

[Packard 1993] Packard, A., J. Doyle and G. Balas, "Linear Multivariable Robust Control with a μ perspective," ASME Journal of Dynamics, Measurements and Control, June, 1993.

[Packard 1994] Packard, A., "Gain Scheduling via Linear Fractional Transformations," Systems and Control Letters, Volume 22, Number 2, pp. 79-82, February 1994.

[Peczkowski 1978] Peczkowski, J.L. and M.K. Sain, "Linear Multivariable Synthesis with Transfer Functions," in *Alternatives for Linear Multivariable Control,* Sain, Peczkowski and Melsa, editors, National Engineering Consortium, 1978.

[Peterson 1986] Peterson, I.R., and C.V. Hollot, "A Riccati Equation Approach to the Stabilization of Uncertain Linear Systems," Automatica, July 1986.

[Postlethwaite 1979] Postlethwaite, I., and A.G.J. MacFarlane, *A Complex Variable Approach to the Analysis of Linear Multivariable Feedback Systems,* Springer Verlag, 1979.

[Reichert 1992] Reichert, R. T., "Dynamic Scheduling of Modern-Robust-Control Autopilot Designs for Missiles," IEEE Control Systems Magazine, October, 1992.

[Ridgely 1992] Ridgely, D.B., L.S. Valavani, M.A. Dahleh and G. Stein, "Solution to the General Mixed H_2/H_∞ Problem - Necessary Conditions for Optimality," American Control Conference, Chicago, 1992.

[Rosenbrock 1974] Rosenbrock, H.H., *Computer Aided Control System Design,* Academic Press, 1974.

[Roskam 1972] Roskam, J., *Flight Dynamics of Rigid and Elastic Airplanes, Parts One and Two,* Roskam Aviation and Engineering Corporation, 519 Boulder, Lawrence, Kansas, 66044, 1972.

[Rotea 1991] Rotea, M.A. and P.P. Khargonekar, "Mixed H_2/H_∞ Control: A Convex Optimization Approach," IEEE TAC-36(7), July 1991.

[Rugh 1991a] Rugh, Wilson .J., "Analytical Framework for Gain Scheduling," IEEE Control Systems Magazine, Vol. 11, No. 1, pp. 79-84, January 1991.

[Rugh 1991b] Rugh, Wilson .J., "Gain Scheduling and High Performance Autopilot Design," AFOSR Workshop On the Theory and Application of Nonlinear Control, Washington University, St. Louis, Missouri, August 1991.

[Safonov 1978] Safonov, M. G. and M. Athans, "A Multiloop Generalization of the Circle Criterion," Twelfth Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, CA, November 1978.

[Serrin 1959] Serrin, James, "Mathematical Principles Of Classical Fluid Mechanics," Handbook Der Physik, Verlag, 1959.

[Shaw 1985] Shaw, Robert L., *Fighter Combat, Tactics and Maneuvering,* Naval Institute Press, Annapolis, Maryland, 1985.

[Skelton 1989] Skelton, R. and M. Ikeda, "Covariance Controllers for Linear Continuous-time Systems," Int. J. Control, Vol. 49(5), 1989.

[Sobel 1985a] Sobel, Kenneth M., Shapiro, Elizer Y., "Eigenstructure Assignment: A Tutorial," Proceedings of the 1985 American Conference on Control, pp. 456-467.

[Sobel 1985b] Sobel, Kenneth M., Shapiro, Elizer Y., "A Design Methodology for Pitch Pointing Flight Control Systems," Journal of Guidance, Control and Dynamics, Vol 8, No.2, Mar/Apr 1985, pp. 181-187.

[Sobel 1987] Sobel, Kenneth M., Shapiro, Elizer Y., "Application of Eigenstructure Assignment to Flight Control Design: Some Extensions."

[Sobel/Lallman 1988] Sobel, Kenneth M., Lallman, Frederik J., "Eigenstructure Assignment for the Control of Highly Augmented Aircraft," Proceedings of the 1988 American Control Conference, pp. 1266-1276.

[Sobel 1990] Sobel, K.M., W. Yu and F.J. Lallman, "Eigenstructure Assignment with Gain Suppression Using Eigenvalue and Eigenvector Derivatives," J. Guidance, Control and Dynamics, Vol. 13(6), November-December 1990.

[Smith 1991] Smith, P.R., "Application of Eigenstructure Assignment to the Control of Powered Lift Combat Aircraft," Royal Aerospace Establishment report number RAE-TM-FS-1009, 1991.

[Sparks 1990] Sparks, A., S.S. Banda and H.H. Yeh, "A Mixed H_2 and H_∞ Approach to the Boeing 737 Autopilot Design Problem," AIAA Guidance and Control Conference, Portland, August 1990.

[Srinathkumar 1979] Srinathkumar, S., "Eigenvalue/Eigenvector Assignment Using Output Feedback," IEEE Transactions on Automatic Control, Vol. AC-23, No. 1, February, 1979.

[Stein 1971] Stein, G. and Henke, A. H., "A Design Procedure And Handling-Quality Criteria For Lateral-Directional Flight Control Systems," AFFDL-TR-70-152, February 1971.

[Stein 1987] Stein, G. and M. Athans, "The LQG/LTR Procedure for Multivariable Feedback Design," IEEE TAC-32, February 1987.

[Stein 1991] Stein, G. and J.C. Doyle, "Beyond Singular Values and Loop Shapes," J. Guidance, Control and Dynamics, January-February 1991.

[Stewart 1973] Stewart, G. W., *Introduction To Matrix Computations*, Academic Press, 1973.

[Stone 1984] Stone, C. R., "Modelling For Control," Proceedings Of The Aeroservoelastic Specialists Meeting, AFWAL-TR-84-3105, Vol. II, Oct. 1984.

[Strang 1976] Strang, G. *Linear Algebra and Its Applications,* Academic Press, New York, 1976.

[Theodorsen 1935] Theodorsen, T., "General Theory Of Aerodynaimc Instability And The Mechanism Of Flutter," NACA Report 496, 1935.

[Tyler 1966] Tyler, Jr., J.S. and F.B. Tuteur, "The Use of a Quadratic Perfomance Index to Design Multivariable Control Systems," IEEE TAC-11(1), January 1966.

[Utkin 1977] Utkin, V.I., "Variable Structure Systems with Sliding Modes," IEEE TAC-22(2), 1977.

[VanDierendonck 1972] VanDierendonck, A.J., "Design Methods for Fully Augmented Systems for Variable Flight Conditions," AFFDL-TR-71-152, January 1972.

[VanDierendonck 1973] VanDierendonck, A.J. and G.L. Hartmann, "Quadratic Methodology: A Short Course on Applications of Optimal Control Theory to Design of Practical Control Systems," Honeywell Report F016-FR, for CCV Branch at AFFDL, October 1973.

[Vepa 1975] Vepa, R. "Finite State Modeling Of Aeroelastic Systems," Ph.D. Dissertation, Dept. Of Applied Mechanics, Stanford University, June 1975. See also: NASA CR-2779.

[Well 1982] Well, K. H., B. Fabre, and E. Berger, "Optimization of Tactical Aircraft Maneuvers Utilizing High Angles-of-Attack," Journal of Guidance, Control and Dynamics, Vol. 5, pp. 131-137, March-April 1982.

[Wilson 1989] Wilson, D.J. and Riley, D.R., "Cooper-Harper Rating Variability," AIAA-89-3358, August 1989.

[Wilson 1990] Wilson, D.J. and Riley, D.R., "More on Cooper-Harper Rating Variability," AIAA-90-2822, August 1990.

[Wilson 1990] Wilson, R.F. and J.R. Cloutier, "Optimal Eigenstructure Achievement with Robustness Guarantees," American Control Conference, San Diego, 1990.

[Wilson 1993] Wilson, David J., David Riley, and Kevin D. Citurs, "Aircraft Maneuvers for the Evaluation of Flying Qualities and Agility, Volume 1: Maneuver Development Process and Initial Maneuver Set, Volume 2:

Maneuver Descriptions and Selection Guide," WL-TR-93-3082, U. S. Air Force Wright Laboratory, August 1993.

[Wilson 1993a] Wilson, D.J., "Aircraft Maneuvers for the Evaluation of Flying Qualities and Agility - Maneuver Descriptions and Selection Guide," WL-TR-93-3082, August 1993.

[Wilson 1993b] Wilson, D.J., "Aircraft Maneuvers for the Evaluation of Flying Qualities and Agility - Maneuver Development Process and Initial Maneuver Set," WL-TR-93-3081, August 1993.

[Wolovitch 1974] Wolovitch, W.A., Linear Multivariable Systems, Springer Verlag, 1974.

[Wonham 1979] Wonham, W.M., Linear Multivariable Control: A Geometric Approach, Springer Verlag, 1979.

[Yates 1966] Yates, E. C., Jr., "Modified-Strip-Analysis Method For Predicting Wing Flutter At Subsonic To Hypersonic Speeds," Journal Of Aircraft, Vol. 3, No. 1, Jan-Feb. 1966.

[Yeh 1992] Yeh, H.H., S.S. Banda and B.C. Chang, "Necessary and Sufficent Conditions for Mixed H_2/H_∞ Optimal Control," IEEE TAC-37(3), March 1992.

[Yeh 1991] Yeh, H.H., S.S. Banda and A.G. Sparks, "Loop Shaping in Mixed H_2 and H_∞ Optimal Control," American Control Conference, Boston 1991.

[Yeh 1990] Yeh, H.H., S.S. Banda, S.A. Heise and A.C. Bartlett, "Robust Control Design with Real Parameter Uncertainty and Unmodelled Dynamics," J. Guidance, Control and Dynamics, Vol. 13(6), November-December 1990.

[Young 1992a] Young, P., M.P. Newlin and J.C. Doyle, "Practical Computation of the Mixed μ Problem," American Control Conference, Chicago 1992.

[Young 1992b] Young, P., M.P. Newlin and J.C. Doyle, "Let's Get Real," Proceeedings of the Institute of Math and its Applications (of the University of Minnesota) on Robust Control Theory , Minneapolis, 1992.

[Zames 1983] Zames, G. and B.A. Francis, "Feedback, Minimax Sensitivity and Optimal Robustness," IEEE TAC-28(5), May 1983.

[Zhou 1989] Zhou, K., J.C. Doyle and B. Bodenheimer, "Optimal Control with Mixed H_2 and H_∞ Performance Objectives," American Control Conference, Pittsburg, 1989.