Real-Time Inverse Kinematics for Humans

Mi, Z[1], Yang, J[1], and Abdel-Malek, K[1], and Mun, J[2], Nebel, K[3]

[1]The Digital Humans Laboratory
The University of Iowa
Iowa City, I A52242
Tel. (319) 335-5676
Fax. (319) 335-5669
karim-abdel-malek@uiowa.edu

[2]Department of Intelligent Mechanical Engineering
School of Science and Engineering
Tokyo Denki University
Tel. 81-49-296-2911 (Ext. 6370)
Fax. 81-49-296-6544
jmun@engineering.uiowa.edu

[3]U.S. Army TACOM
Automotive Research Center (ARC)
Mail Stop #233
6501 East 11 Mile Rd.
Warren, MI 48397-5000
nebelk@tardec.tacom.army.mil

Abstract

A general methodology and associated computational algorithm for predicting realistic postures of digital humans (mannequins) is presented. The basic plot for this effort is a task-based approach, where we believe that humans assume different postures for different tasks. The underlying problem is characterized by the calculation (or prediction) of the joint displacements of the human body in such a way to accomplish a specified task. In this work, we have not limited the number of degrees of freedom associated with the model. Each task has been defined by a number of human performance measures that are mathematically represented by cost functions that evaluate to a real number. Cost functions are then optimized, i.e., minimized or maximized subject to a number of constraints. The problem is formulated as a multi-objective optimization algorithm where one or more cost functions are considered as objective functions that drive the model to a solution. The formulation is then validated against existing posture prediction algorithms and confirmed with human experimental data.

Keywords: Posture prediction, inverse kinematics, human postures, task-based postures

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
15 MAR 2004	**Journal Article**	**12-02-2004 to 12-03-2004**

4. TITLE AND SUBTITLE
Real-Time Inverse Kinematics for Humans

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Z Mi; J Yang; K Abdel-Malek; J Mun; K Nebel

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Digital Humans Laboratory,The University of Iowa,28 W Jefferson St # W106,Iowa City,IA,52242

8. PERFORMING ORGANIZATION REPORT NUMBER
; #13981

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army TARDEC, 6501 East Eleven Mile Rd, Warren, Mi, 48397-5000

10. SPONSOR/MONITOR'S ACRONYM(S)
TARDEC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
#13981

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
A general methodology and associated computational algorithm for predicting realistic postures of digital humans (mannequins) is presented. The basic plot for this effort is a task-based approach, where we believe that humans assume different postures for different tasks. The underlying problem is characterized by the calculation (or prediction) of the joint displacements of the human body in such a way to accomplish a specified task. In this work, we have not limited the number of degrees of freedom associated with the model. Each task has been defined by a number of human performance measures that are mathematically represented by cost functions that evaluate to a real number. Cost functions are then optimized, i.e., minimized or maximized subject to a number of constraints. The problem is formulated as a multi-objective optimization algorithm where one or more cost functions are considered as objective functions that drive the model to a solution. The formulation is then validated against existing posture prediction algorithms and confirmed with human experimental data.

15. SUBJECT TERMS
Posture prediction, inverse kinematics, human postures, task-based postures

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Public Release**	**18**	

Introduction

Posture prediction from a human modeling and simulation point of view is a very important research area because of its eminent use in the ergonomic design process, to reduce occupational injuries, and to introduce rigor into the ergonomic design process. Over the past several years digital human modeling, simulation, and analysis has received increased attention from senior leadership within both the automotive and defense industry. This is primarily due to the fact that the traditional methods of human factor analysis in the evaluation of proposed system concepts and block design changes through the use of full-scale mock-ups, is being reduced, if not entirely eliminated. Full-scale mock-ups by themselves are extremely expensive, and require a considerable amount of time to build, thus limiting the number of design iterations that can be performed during development and modernization. It is then paramount to have digital solutions available that generate realistic and accurate digital human postures, providing the ability to address human interface issues in the virtual environment, early enough in the development cycle to have an impact on both cost and schedule. Furthermore, posture prediction is also of vital interest to biomechanics engineers with the aim to better understand the functionality of the upper and lower extremities and tasks associated with the muskoskeletal system.

There has been two schools of thought regarding posture prediction. The first, perhaps the more traditional, uses anthropometrical data, collected from performing thousands of experiments by human subjects, or simulation using three-dimensional computer-aided human-modeling software [see for instance; Porter et al. (1990) and Das and Singupta (1995)], which were statistically analyzed to form a predictive model of posture; e.g. regression models. This school of thought is referred to as *empirical-statistical modeling*. These models have been implemented in various simulation software systems with some variations as to the method for selecting the most probable posture. Among the empirical-statistical modelers were Beck and Chaffin (1992), Zhang and Chaffin (1996, 1997), Das and Behara (1998), and Faraway, et al. (1999).

The second school of thought often used biomechanics and kinematics as a predictive tool (often referred to as the *inverse kinematics solutions*), on a posture that has not been observed but has been estimated as a likely posture for a task (Tracy, 1990). This approach mathematically models the motion of a limb with the goal of formulating a set of equations that can be solved for the joint variables. Among the researchers who belong to this school of modeling are Kee et al. (1994), Jung et al. (1995), Jung and Kee (1996), Jung and Choe (1996), Kee and Kim (1997), and Wang (1999).

Researchers that belong to one school of modeling (in particular Beck and Chaffin 1992) cautioned that the inverse kinematics algorithm is not necessarily correct for prediction of posture because of its theoretical foundation, because of the difficulty with evaluating the Jacobian, determining a closed form equation for the posture, and in modeling large numbers of Degrees of Freedom (DOF). On the other hand, others (Faraway, et al. 1999; Abdel-Malek, et al. 2001) have stated that the use of only statistical models do not provide avenues for design. Furthermore, those that belong to the inverse kinematics school of modeling, state that most existing human models have not fully utilized anthropometric data due to the generalized formalism of data manipulation, which may result in serious problems when a system is upgraded or when a specific population of operators is considered (e.g., see the object-oriented anthropometric work by Jung and Kang 1995).

We believe that our approach using a task-based optimization formulation addresses most of these problems, does not attempt to determine a closed-form expression, predicts a realistic posture, and can handle a relatively large number of DOFs.

An approximate analytical reach prediction algorithm, was developed (Jung, et al. 1995), where the Denavait and Hartenberg (D-H) notation was used to represent human motion. They reportedly demonstrated that humans adopt postures of minimum discomfort among all feasible body configurations. Similar results were reported by Dysart and Woldstad (1996) who used three separate models and objective functions to predict the postures of humans performing static sagittal lifting tasks. The models used a common inverse kinematics characterization to represent mathematically feasible postures, but explore different criteria functions for selecting a final posture. Dysart and Woldstad (1996) results showed that the first objective function (minimum total torque) was more accurate.

Also using the concept of inverse kinematics, Kee and Kim (1997) proposed an approximate algorithm to generate the workspace including foot and trunk motion. More recent results have focused on a combination of methods such as both rule-based empirical and optimization to address the posture prediction problem (Wang 1999). The emergence of Artificial Neural Networks models to provide more accurate predictions over the standard statistical models (Eksioglu, et al. 1996; Jung and Park 1994; Hestenes 1994).

We will first present the background human modeling method necessary for the analysis. Our task-based approach will then be presented in view of a rigorous mathematicallly-based optimization formulation where cost functions characterizing human performance measures are used and evaluated to a real number. These cost functions are then implemented in the optimization formulation to predict postures.

Before proceeding, it is important to note that although our exposition has focused on the torso, shoulder and arm, it is applicable to any serial chain representing the human body.

Kinematic Modeling

In this section, we describe the general modeling method used in our development and adapted from the field of kinematics. The method is used to characterize joints of a mechanism in the study of motion, such that a position vector describing the location of a given point in terms of all joint displacements is determined. Indeed, the Denavit-Hartenberg (1955) representation method (also known as the DH method) has been demonstrated to yield an effective method for modeling humans (Jung, et al. 1995; Abdel-Malek, et al. 2001).

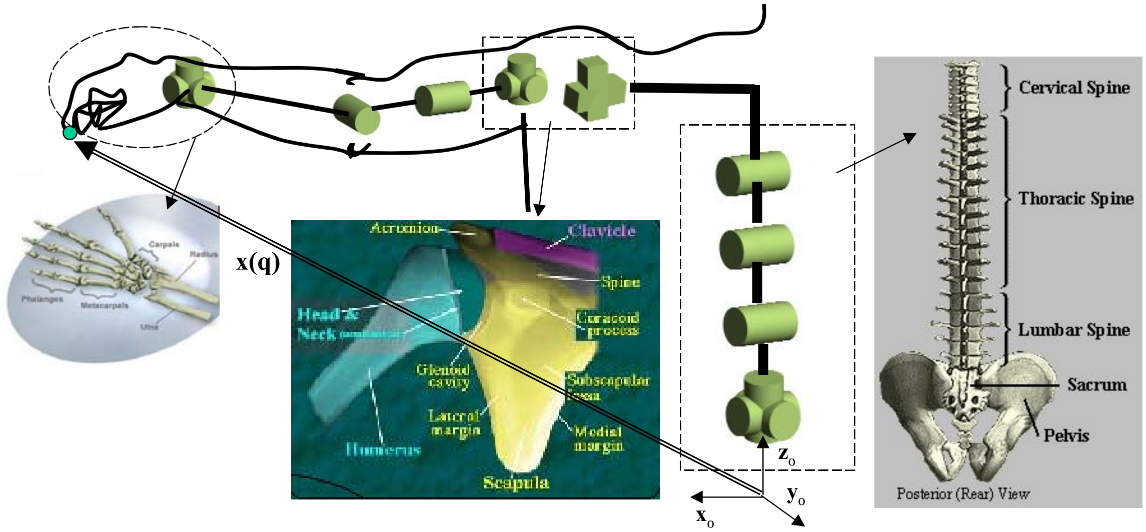


Fig. 1 Modeling of the torso-shoulder-arm movement

The position vector of a point of interest on the end-effector of a human articulated model (e.g., a point on the thumb with respect to the shoulder) can be written in terms of joint coordinates as

$$\mathbf{x} = \mathbf{x}(\mathbf{q}) \tag{1}$$

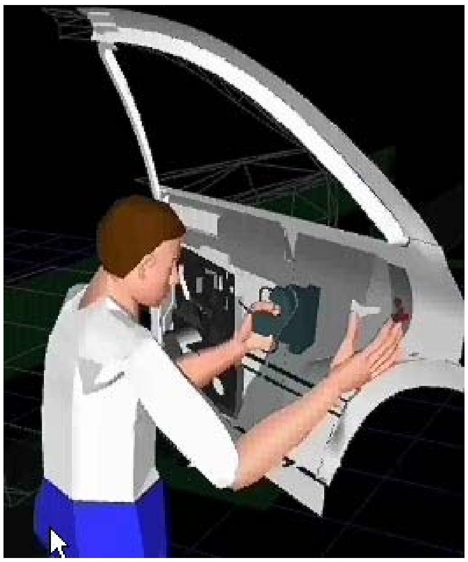
where $\mathbf{q} \in \mathbf{R}^n$ is the vector of *n*-generalized coordinates, and $\mathbf{x}(\mathbf{q})$ can be obtained from the multiplication of the homogeneous transformation matrices defined by the D-H representation method **(16, 17)** as

$$^{0}\mathbf{T}_n = {}^{0}\mathbf{T}_1\,{}^{1}\mathbf{T}_2 \ldots {}^{n-1}\mathbf{T}_n = \begin{bmatrix} {}^{0}\mathbf{R}_n(\mathbf{q}) & \mathbf{x}(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \tag{2}$$

where ${}^{i}\mathbf{R}_j$ is the rotation matrix relating coordinates frames *i* and *j*. The vector function $\mathbf{x}(\mathbf{q})$ characterizes the set of all points touched by the fingertip.

Cost Functions

In 1989, the Air Force Research Lab, formerly Armstrong Labs, Wright Patterson Air Force Base, conducted the first research and development of a task based behavioral solution for digital environments using digital avatars. Under the Design, Evaluation, for Personnel Training and Human Factors (D.E.P.T.H.) program, the Pennsylvania-Jack ergonomic simulation and analysis software, as it would come to be called, developed by Dr. Norman Badler, University of Pennsylvania, took advantage of the capability of the Penn-Jack software to capture, store, and reuse actual human motion capture data. This motion data was used to establish acceptable motion path parameters which were then used by software algorithms to control the digital avatar as the system modified the relative position of segments or limbs to reach manually positioned target points for the human interface of Computer Aided Design (CAD) models of components and systems in the digital world. The success of this program demonstrated both the need and feasibility for tasked based posture prediction algorithms, requiring that the solutions developed are fast, realistic, and robust, i.e. applicable to the whole digital human, not just the upper torso or limb, and scalable to different human populations and percentiles. The interest generated in the automotive industry by the success of developmental research and the trend to reduce if not completely eliminate the need to build full scale mock-ups is the reason why researchers have become increasingly interested in posture prediction algorithms. With the recent appearance of commercially available digital human code, for example, Jack, SafeWorks, and Robcad, enabling a user to model a human mannequin, to place the mannequin in a digital representation of a conceptual system, to provide the ability to interact with the proposed systems human interface, and to evaluate specific criteria like reach and occlusion.



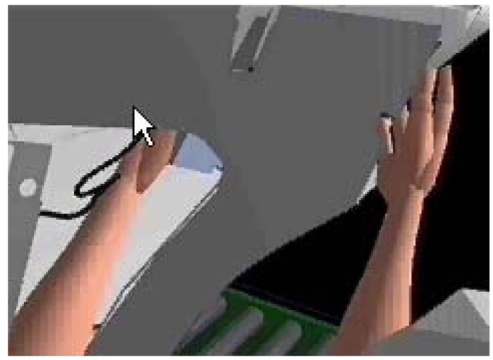


Fig. 4 As seen by the mannequin

Saving time, reducing cost, increasing the number of evaluations for proposed prototypes, and reducing the number of engineering change requests resulting from oversights and deficiencies for the systems human interface. Providing the system developer with the ability to optimize the analysis of processes, products, and vehicle systems, in a more efficient manner, prior to the first vehicle rolling off the assembly line.

In this section, we address the development of simple human performance measures that enable the mathematical evaluation of a cost function. The basic plot is based upon obtaining a real number that evaluates the task, where each task comprises several cost functions. Each cost function must evaluate to a number and must be mathematically defined. Once this is achieved, it is then possible to formulate an optimization algorithm that iteratively evaluates the task.

Discomfort

Consider a cost function that measures the level of discomfort from the most neutral position of a given joint. Let q_i^N be the neutral position of a joint measured from the starting home configuration (i.e., from the position and orientation specified in the DH Table). Then the displacement from the neutral position is given by $\left| q_i - q_i^N \right|$. Because the discomfort is usually felt higher in some joints, we also introduce a weight function w_i to stress the importance of one joint versus another. The total discomfort of all joints is then characterized by the function

$$f_{discomf}(\mathbf{q}) = \sum_{i=1}^{n} w_i \left| q_i - q_i^N \right| \qquad (1)$$

where w_i is a weight function assigned to each joint for the purpose of giving importance to joints that are typically more affected than others.

Effort

Effort is measured as the displacement of a joint from its original position. Effort will greatly depend on the initial configuration of the limb prior to moving to another location. For an initial set of joint variables $q_i^{initial}$ and for a final set of joint variables q_i, a simple measure of the effort is expressed by

$$f_{effort}(\mathbf{q}) = \sum_{i=1}^{n} w_i \left| q_i - q_i^{initial} \right| \qquad (2)$$

Note that f_{effort} depends on the initial configuration of each joint.

Potential Energy

For a second cost function, consider the potential energy exerted by a limb. Each link (e.g., the forearm) has a specified center of mass. The vector from the origin of the link's coordinate system to the center of mass is given by ${}^{i}\mathbf{r}_i$, where similar superscript and subscript indicate that the vector is resolved in the link's coordinate system as illustrated in Fig. 4.

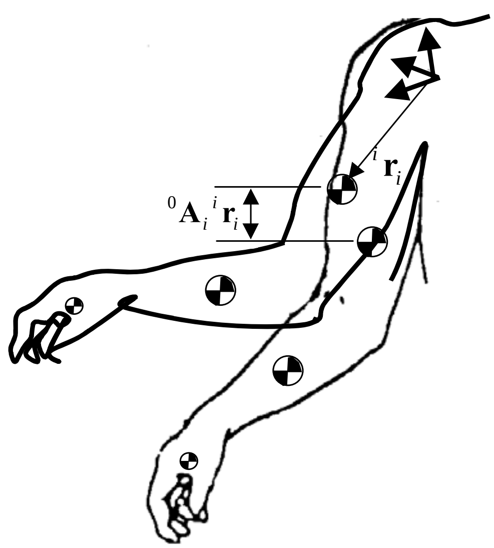


Fig. 6 Illustrating the potential energy of the forearm

The total potential energy $f_{potential}$ is the sum of all individual potential energies P_i. In order to determine the position and orientation of any one part of the arm, we shall use the transformation matrices ${}^{(i-1)}A_i$ that relates one part to another using the (4×4) transformation matrix. Let the vector ${}^{i}\overline{\mathbf{r}}_i$ denote the position of the center of mass of a body part from the origin of its own coordinate system and let $\mathbf{g}$ be the gravity vector (Fig. 4). Then for the first body part in the chain, the potential energy is $P_1 = m_1 \mathbf{g}\,{}^{0}\mathbf{A}_1\,{}^{1}\overline{\mathbf{r}}_1$.

However, for the second body part in the chain, we must compute the previous result in addition to the energy contribution by the second body part in the chain. We use a second transformation matrix in order to keep track of the second joint variable as $P_2 = m_2 \mathbf{g}\, {}^0\mathbf{A}_1\, {}^1\mathbf{A}_2\, {}^2\overline{\mathbf{r}}_2 + P_1$. For a complete chain (e.g., a 9 degree of freedom arm), the total potential energy is given by

For a the total potential energy of the arm is given by

$$f_{potential}(\mathbf{q}) = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} \left(-m_i \mathbf{g} ({}^0\mathbf{A}_i\, {}^i\mathbf{r}_i) \right) \tag{3}$$

where $\mathbf{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$ is the gravity vector.

Dexterity

It is believed that humans also configure their extremities around an object in such a way to have the maximum accessibility to that object. We define a cost function that is based on maximizing the dexterity at specified target points. Indeed, to mathematically formulate this problem, it is necessary to use a dexterity measure at specific target points. Such a measure must account for the ranges of motion for each joint. Because of the need for an analytical expression that can be used in the proposed optimization method, we define a new dexterity measure.

Since the extended Jacobian $\mathbf{H_q}$ inherently combines information about the position, orientation, and ranges of motion of the hand, it is a viable measure of dexterity. Furthermore, because of the simplicity in determining an analytical expression of $\mathbf{H_q}$, it is well suited as a cost function for an optimization problem. We define the dexterity measure as

$$f_{dexterity}(\mathbf{q}) = \sqrt{\left| \mathbf{H_q}(\mathbf{q}) \mathbf{H_q}^T(\mathbf{q}) \right|} \tag{4}$$

Note that the measure characterized by Eq. (22) takes into consideration all ranges of motion and singular orientations for a given kinematic chain. The proposed dexterity measure is more accurate in describing the manipulability of robot manipulators than that proposed by Yoshikawa (1995), because it considers all singularities (Jacobian and others) as well as joint limits.

Torque

Stress induced at a joint is a function of torque imposed at that joint due to the biomechanical interaction. A person will generate the torque at a given joint to overcome a load by exerting muscle forces but is also a function of the position and orientation of the joint during loading. In order to account for all of the elements that enter into calculating the torque at a given joint, we must employ a systematic formulation. To develop a mathematical expression for the torque, we first introduce a few preliminary concepts. The velocity of a point on the hand is obtained by differentiating the position vector as

$$\dot{\mathbf{x}} = \mathbf{J_x} \dot{\mathbf{q}} \tag{5}$$

where the position Jacobian $\mathbf{J}_{\mathbf{x}}(\mathbf{q}) = [\partial \mathbf{x} / \partial \mathbf{q}]$ is a $(3 \times n)$ matrix and $\dot{\mathbf{q}}$ is the vector of joint velocities. Note that the reach envelope can be determined from analytically stratifying the Jacoboian (Abdel-Malek, et al. 2001). Similarly, the angular velocity can be obtained as

$$\mathsf{w} = \mathbf{J}_{\mathsf{w}} \dot{\mathbf{q}} \tag{6}$$

where the orientation Jacobian $\mathbf{J}_{\omega}$ is a $(3 \times n)$ matrix. Combining equations (5 and 6) into one vector yields

$$\mathbf{v} = \begin{bmatrix} \dot{\mathbf{x}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \tag{7}$$

where $\mathbf{J}(\mathbf{q})$ is the Jacobian of the limb or kinematic structure defined by

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{x} \\ \mathbf{J}_{\omega} \end{bmatrix} \tag{8}$$

There are many methods for determining the Jacobian of a kinematic structure, we present a direct method in Appendix A.

The goal in this section is to determine the relationship between the generalized forces applied to the hand (e.g., carrying a load) and generalized forces applied to the joints. Let $\boldsymbol{\tau}$ denote the $(n \times 1)$ vector of joint torques and $\mathbf{F}$ the $(m \times 1)$ vector of hand forces applied at $\mathbf{p}$, where m is the dimension of the operational space of interest (typically six).

Using the principle of virtual work, we can determine a relationship of joint torques and forces at the hand. Since the upper extremity is a kinematic system with time-invariant, holonomic constraints, its configuration only depends on the joint variables $\mathbf{q}$ (not explicitly on time). Consider the virtual work performed by the two force systems. As for the joint torques, its associated virtual work is

$$dW_{\boldsymbol{\tau}} = \boldsymbol{\tau}^{T} d\mathbf{q} \tag{9}$$

For the hand forces $\mathbf{F} = \begin{bmatrix} \mathbf{f}^{T} & \mathbf{m}^{T} \end{bmatrix}^{T}$, comprised of a force vector $\mathbf{f}$ and moment vector $\mathbf{m}$, the virtual work performed is

$$dW_{\mathbf{F}} = \mathbf{f}^{T} d\mathbf{x} + \mathbf{m}^{T} \boldsymbol{\omega} dt \tag{10}$$

where $d\mathbf{x}$ is the linear displacement and $\boldsymbol{\omega} dt$ is the angular displacement. Substituting Eqs. (7 and 8) into Eq. (10) yields

$$dW_{\mathbf{F}} = \mathbf{f}^{T} \mathbf{J}_{\mathbf{x}} d\mathbf{q} + \mathbf{m}^{T} \mathbf{J}_{\omega} d\mathbf{q}$$

$$dW_{\mathbf{F}} = \mathbf{F}^{T} \mathbf{J} d\mathbf{q} \tag{11}$$

Since virtual and elementary displacements coincide, virtual works associated with the two systems are

$$\delta W_{\boldsymbol{\tau}} = \boldsymbol{\tau}^{T} \delta \mathbf{q} \tag{12a}$$

$$\delta W_{\mathbf{F}} = \mathbf{F}^{T} \mathbf{J} \delta \mathbf{q} \tag{12b}$$

where δ denotes a virtual quantity. The system is under static equilibrium if and only if

$$dW_{\mathbf{F}} = dW_{\boldsymbol{\tau}} \qquad \forall \delta \mathbf{q} \tag{13}$$

which means that the difference between the virtual work of the joint torques and the virtual work of the hand forces shall be null for all joint displacements. Substituting Eqs. (9 and 11) into (13) yields

$$\boldsymbol{\tau}^T \delta\mathbf{q} = \mathbf{F}^T \mathbf{J}(\mathbf{q})\delta\mathbf{q} \quad \forall\mathbf{q} \tag{14}$$

Therefore, the relationship between the joint torques and forces on the hand is given by

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} \tag{15}$$

where the torque vector is $t = \left[\tau_1, \tau_2, \ldots, \tau_n\right]^T$.

We now develop the cost function that is fundamental to our formulation. The objective function (the Torque Cost Function–TCF) is to be <u>minimized</u> and is comprised of the weighted summation of all joint torques

$$TCF = \sum_{i=1}^{n} w_i \left|\tau_i\right| \tag{16}$$

where w_i is a weight function used to distribute the importance of the cost function among all joints.

Constraints

For the point on the end-effector characterized by **x** as a function of all joint variables **x(q)** to reach a target point **p**, it is necessary that **x(q) – p = 0**. While many have attempted to implement this simple equation in an optimization formulation as a cost function, it is quickly realized that it is a difficult implementation. It is evident that criteria is indeed a constraint that must be driven by one or more cost functions addressed above. Therefore, we implement this equation as a constraint to be imposed within a specified tolerance ε, such that

$$\left\|\mathbf{x}(\mathbf{q}) - \mathbf{p}\right\| \le \varepsilon \tag{17}$$

Furthermore, each degree of freedom has unilateral constraints imposed in the form of

$$\begin{matrix} q_i^L \le q_i \le q_i^U \\ i = 1, \ldots, n \end{matrix} \quad q_i^L \le q_i \le q_i^U;\ i = 1, \ldots, n \tag{18}$$

where n is the number of DOF used in the model. Note that for a 15DOF model as implemented in our code, a total of 31 constraints must be imposed (two for each unilateral constraint and Eq. 17).

Real-Time Algorithm

Previously, GA-DOT method was used to calculate the posture for a given target point by using some task-driven cost functions. Genetics Algorithm (GA) is a global optimization method, but it takes a lot of computation time. Design Optimization Tools (DOT) is very fast but it only searches optimization point in a local area. GA-DOT combines them together; it uses GA with some cost function to search for a point in global area and gives its result during each searching step to DOT as initial point. Then DOT will refine the search locally to find the best point with minimized distance to target point. Through the combination, pretty good results can be obtained. Besides, the computation time is much improved and reduced to 15 minutes from several hours by using GA itself, however it is still too much for real time prediction. A much faster and still accurate method is needed. In order to utilize the fast property of DOT method, a workspace of our 15-DOF human model is pre-calculated and divided into 16 sections. A middle point is chosen within each section and is given to GA-DOT as target point for preprocessing. Results from GA-DOT are going to be used as starting points respectively for each section which the target points drop inside. On the basis of the above information, four faster methods were developed and tested. Each of the four methods makes justifications at the beginning of the algorithm; it will terminate right away if the target point is outside the workspace. The main difference of the four methods lies in that they use different optimization strategies. The flowcharts for the four methods are shown in figure 1, 2, 3 and 4 respectively.

DOT-DOT method uses two layers of optimization. Inner DOT works as distance constraint, it is used to find the optimal point with minimal distance to target point and sends the result to outer DOT, which is to minimize some cost function, like discomfort here. The real discomfort is calculated and regarded as the value of cost function only if the distance satisfies some tolerance, otherwise, a big penalty is given as the cost function value. This way, the search is driven to the point which has low cost and guarantees the end-effector reaches the given target point. DIS-CONS method uses traditional constrained optimization method which minimizes discomfort with the distance to the target point as constraint. MOO method is a non-constrained optimization method which actually is doing multi-objective optimization. The cost function for MOO combines discomfort and distance together with some weights so as to realize finding the point with minimal discomfort and satisfying distance. The motivation for proposing CONS-DOT method comes from the limitation of normal gradient-based optimization method used in DOT with our special problem here. Normally, for each target point, the starting point hardly satisfies the distance constraint, i.e., the end-effector will not be on the target point at the beginning of the optimization process. So there will always be violated constraint at the first iteration inside the optimization procedure. Since the problem here is highly nonlinear with 15 variables, the search of feasible design becomes extremely difficult. DOT will terminate its optimization process if 20 iterations pass without overcoming the constraint violations. Thus providing a starting point with no violated constraint will have a strong influence on the efficiency and reliability of the result. CONS-DOT method calls DOT first to look for a new starting point with any given target and starting point, by only minimizing distance so that the new starting point satisfies the distance constraint when second DOT is called. Then DOT optimizes the cost with distance constraint but with new starting point.

The best combination of the parameters inside DOT was obtained by trying lots of different combinations and methods in DOT. Results got by using the best combination and the four methods are listed in table 2. CPU times needed by the four methods are listed in table 3. Results from the previous global optimization method GA-DOT are listed in table 1 for comparison. From table 2 and table 3 we can see DOT-DOT gives us the most accurate results, but is the slowest just as what was expected. It uses inner DOT guaranteeing the end-effector is right on the target point so that outer DOT is able to search for the result well within all the points with end-effector on the target. However, the two-layer search brings too much cost on the computation time. DIS-CONS gives us the worst results. This is mainly due to the fact that it always fails to overcome the constraint violations during 20 iterations and terminates the optimization process. CONS-DOT much improves the reliability of the algorithm and verifies the importance of providing an initial feasible design to the optimization process. It found good results except for point 8. Moreover, it is fast. Although it takes more average time than DIS-CONS, we can see that generally it takes much less time. The reason for this is that since a new starting point satisfying the constraint is provided to the DOT, it will save much time used for searching direction back toward a feasible region in the optimization process. However as we can see, this method is not robust enough and will give bad result for certain target point. MOO gives us results with acceptable accuracy at every point and it is the fastest due to that it only searches for the minimum cost function without any constraint, and avoids the cost of the iterations related to the constraint. In practice, since MOO runs very fast and gives accurate enough results, it was selected and implemented into a plug in of posture prediction to 3D Studio MAX.

	GA-DOT	
Point	Distance	Discomfort
1	0.0000	2.2022
2	0.0002	7.3800
3	0.0003	12.8254
4	0.0002	2.0873
5	0.0005	1.5824
6	0.0001	1.0783
7	0.0003	0.7253
8	0.0007	3.8352
9	0.0005	0.4966
10	0.0008	3.3709

Table 1. Distance and Discomfort obtained from GA-DOT

	Method 1(DOT-DOT)		Method 2(DIS-CONS)		Method 3(MOO)		Method 4(CONS-DOT)	
Point	Distance	Discomfort	Distance	Discomfort	Distance	Discomfort	Distance	Discomfort
1	0.0000	3.2911	146.2477	1.1662	0.0001	3.6126	0.0001	3.6507
2	0.0003	8.2012	0.0096	4.6063	0.0026	8.6291	0.0002	8.5723
3	0.0010	15.1879	54.8682	7.9918	0.3558	15.3929	0.0026	8.0009
4	0.0014	3.2147	0.0234	1.8227	0.0001	4.2680	0.0009	4.3472
5	0.0006	1.6103	0.0041	1.5178	0.0008	1.6217	0.0004	1.6103
6	0.0005	3.2093	54.0299	0.7429	0.0009	2.9490	0.0001	3.2093
7	0.0000	0.6639	0.0117	0.6097	0.0030	0.6954	0.0031	0.7002
8	0.0000	4.9564	176.6827	2.2751	0.0000	4.5921	27.9220	2.6904
9	0.0002	0.5771	7.3989	0.3014	0.0000	0.9554	0.0000	0.9576
10	0.0006	8.3686	221.7487	0.8704	0.0003	7.0304	0.0290	8.4179

Table 2. Distance and Discomfort obtained from the four faster methods

	CPU Time(*Seconds*)			
Point	Method 1(DOT-DOT)	Method 2(DIS-CONS)	Method 3(MOO)	Method 4(CONS-DOT)
1	0.74	3.05E-07	2.21E-11	2.21E-11
2	0.74	4.03E-05	2.05E-13	1.24E-12
3	0.82	2.08E-05	3.05E-07	2.73E-03
4	102.4	6.71E-06	3.52E-14	3.52E-14
5	8.96E-07	1.75E-06	3.57E-12	3.57E-12
6	8.96E-07	2.21E-06	2.05E-13	2.05E-13
7	3.38E-02	3.43E-06	3.52E-14	2.05E-13
8	3.62E-02	5.24E-11	3.57E-12	3.22E-08
9	0.21	1.75E-06	3.52E-14	2.05E-13
10	2.10E-04	1.96E-07	2.05E-13	3.57E-12
Mean	**10.49802118**	**7.75E-06**	**3.05E-08**	**2.73E-04**

Table 3. CPU time of computations on a HP-UX workstation

Input File to the System (initial configuration)

a. The DH Table (shown in Fig. 1): The DH Table provides all necessary information to model the human body including joints and dimensions.

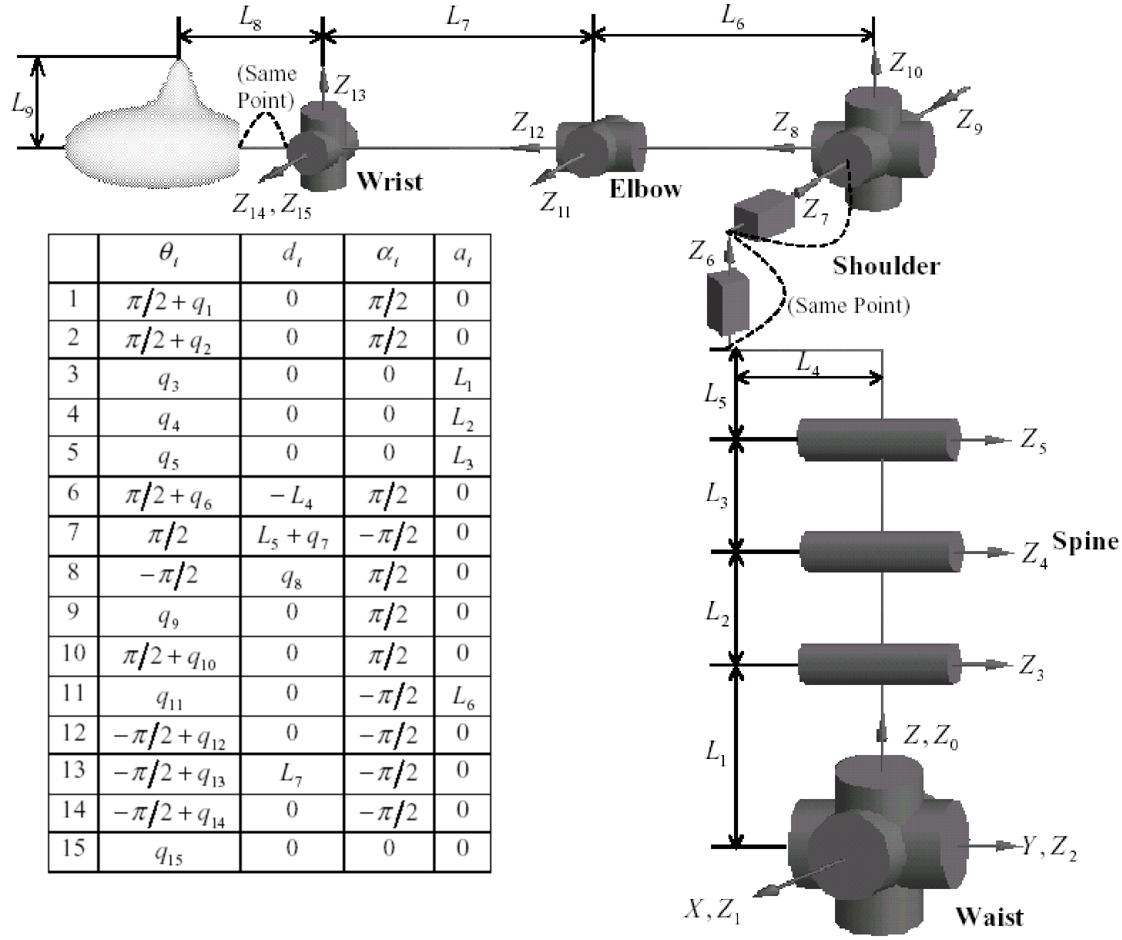
b. Neutral Positions: These are set constants (constant joint angle) that characterize the most comfortable position for each joint. The variable is measured from home configuration (i.e., wherefrom the posture has started). For this human model, the neutral positions are as follows:
$q_i^N = 0;\ i = 1,...,9,11,12,14,15$, $q_{10}^N = \pi/2$, and $q_{13}^N = -\pi/2$.

c. Joint Weights: It is natural that some human joints tend to be activated more than others (passive versus active). To reflect that in the model, we have assigned a scalar number to each joint therefore setting more importance on certain joints.

Joint variable	**Joint Weight**	**Comments**
q_1, q_2	10	A weight multiplying the Cost Function (*f*), for both negative and positive values of $q_i - q_i^N$
$q_3 ... q_6$	10 100	When $q_i - q_i^N > 0$ When $q_i - q_i^N < 0$
q_7	50	For both negative and positive values of $q_i - q_i^N$
q_9	50	When $q_i - q_i^N > 0$

Simulation and Validation

To demonstrate our formulation and computer implementation, the 15DOF model and associated DH Table shown in Fig. 6 has been used. Note that the model shown is in home configuration.



	θ_i	d_i	α_i	a_i
1	$\pi/2+q_1$	0	$\pi/2$	0
2	$\pi/2+q_2$	0	$\pi/2$	0
3	q_3	0	0	L_1
4	q_4	0	0	L_2
5	q_5	0	0	L_3
6	$\pi/2+q_6$	$-L_4$	$\pi/2$	0
7	$\pi/2$	L_5+q_7	$-\pi/2$	0
8	$-\pi/2$	q_8	$\pi/2$	0
9	q_9	0	$\pi/2$	0
10	$\pi/2+q_{10}$	0	$\pi/2$	0
11	q_{11}	0	$-\pi/2$	L_6
12	$-\pi/2+q_{12}$	0	$-\pi/2$	0
13	$-\pi/2+q_{13}$	L_7	$-\pi/2$	0
14	$-\pi/2+q_{14}$	0	$-\pi/2$	0
15	q_{15}	0	0	0

Fig. 8 Modeling of the torso, shoulder, and arm as a 15 DOF system

In order to obtain a better understanding of our results, we have compared the calculated postures with those produced by 3D Studio Max's inverse kinematics module. For a given target position defined below, we have optimized a posture for a maximum Discomfort function and have asked a person to reach (with thumb touching the target point). We have also verified that the calculated postures are close (but not identical) to those assumed by a human subject. Note that the calculation of the cost function for each posture is an important element of the prediction. For the first posture in Fig. 1, the calculated cost function is 2.2022, a relatively low discomfort. This is due to the fact that the calculated posture has only a very small deviation from the neutral position (recall that the neutral position is perceived to give the most comfortable posture). Therefore, the arm fully extended provides for most of the angles being close to zero. Indeed, a similar situation occurs for postures assumed in Figs. 7 and 9, where the calculated discomfort values are also relatively small, 0.72 and 0.496, respectively. Difficult postures (most uncomfortable) are observed in Fig. 3 (discomfort = 12.825), and Fig. 2 (Discomfort 7.38). Note that these postures contain several joint angles that are far from neutral, therefore adding to the cost function.

Because we have opted to use genetics algorithms for optimization, the stopping criteria is not well defined (an inherent characteristic of genetic algorithms). The selection, mutation, and crossover processes continue to run indefinitely over the population. Therefore, time (in terms of computational complexity) must be defined as a criterion for stopping the calculations coupled with an error estimate of the distance of the tip of the finger from the target point. Of course, any posture that has zero distance error and that satisfies the constraints is indeed a solution, but is not the optimum solution. Therefore, although the distance is zero between the tip of finger and target point, the program must continue to minimize the Discomfort cost function until an acceptable low value is achieved.

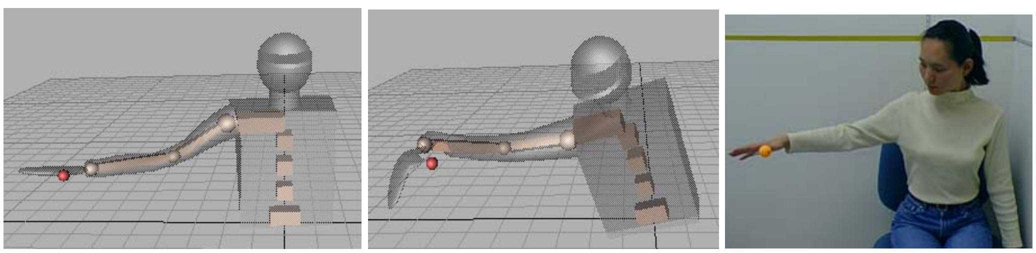


Fig. 1: Target Point 1 (41.2, -57, 31.5), Discomfort = 2.2022

$$\mathbf{q} = [.0847, -.0007, .0407, .0091, .0567, .0820, -.0019, .0075, .0110, .3465, .5328, -.4244, -1.4772, -.1081, .1557]^T$$

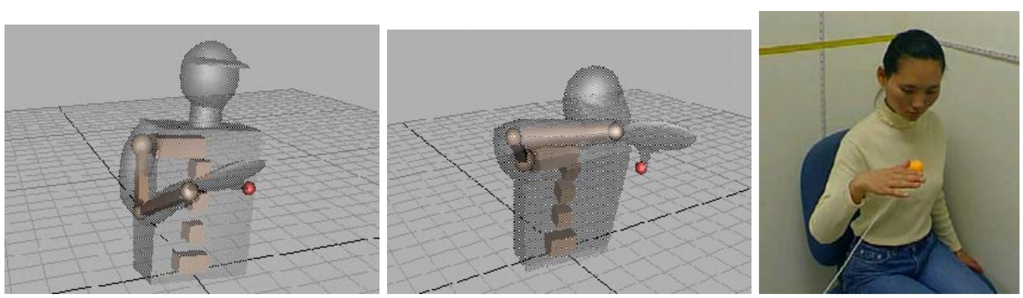


Fig. 2: Target Point 2 (40, 0, 36), Discomfort = 7.38

$$\mathbf{q} = [.1022, -.1310, -.0235, .0198, .0014, .0072, -.0112, .0444, -.7829, -.1346, 1.3475, -1.2451, -1.4099, -.1625, -.3101]$$

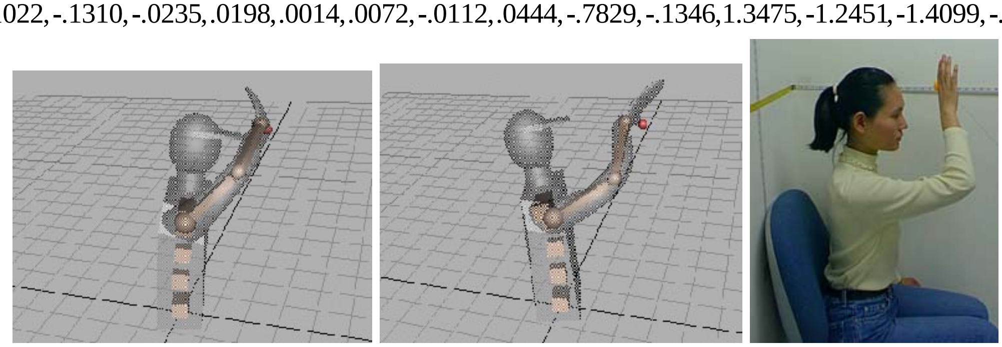


Fig. 3: Target Point 3 (20, 35, 50), Discomfort = 12.8254

$\mathbf{q} = [.3087, -.2618, .0510, .0843, .0416, .0020, .0022, .2022, -.2543, -1.4352, .3640, -1.1986, -1.2240, -.3469, .3487]$

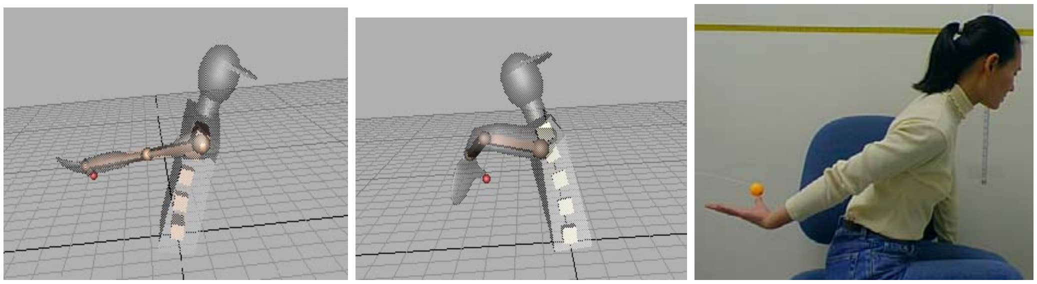


Fig. 4: Target Point 4 (-30, 10, 20), Discomfort = 2.0873

$\mathbf{q} = [-.0713, .0075, .1673, .0884, .1153, .0497, .0097, .0760, -.6950, 1.9193, -.2518, .0000, -2.3357, .4159, -.2897]$

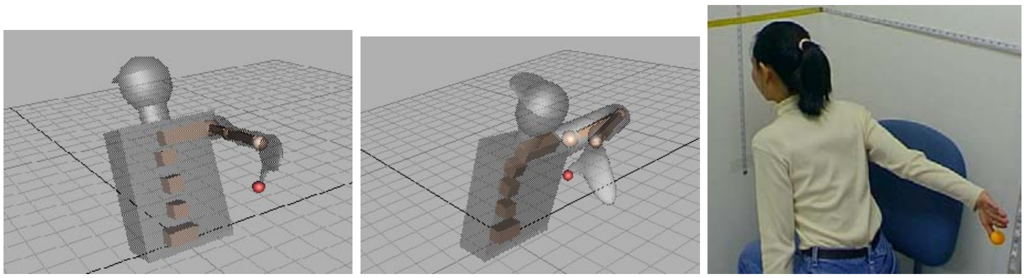


Fig. 5: Target Point 5 (-40, 0, 36), Discomfort = 1.5824

$\mathbf{q} = [.0135, .0206, .1265, .0720, .1204, .0805, .0021, -.0005, -.8226, 1.9184, -.5517, -.0003, -1.7336, .1164, -.1126]$

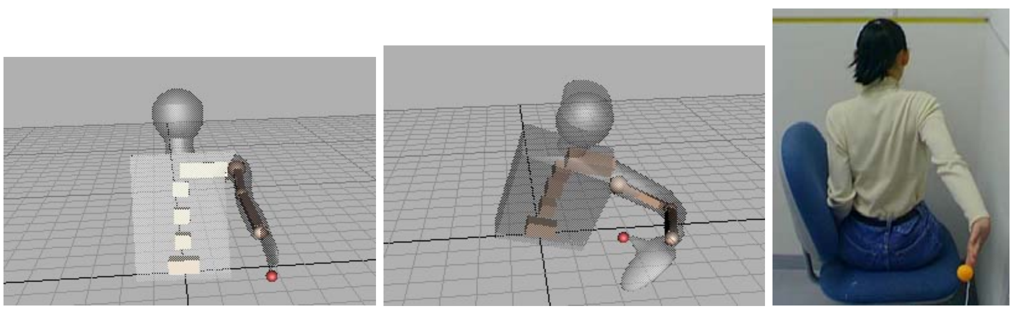


Fig. 6: Target Point 6 (-50, -20, 20), Discomfort = 1.0783

$\mathbf{q} = [-.0871, -.0053, .0546, .0661, .0340, .0676, .0074, .0200, -.7906, 1.5271, -.4086, -.1063, -1.5175, .0095, .2451]$

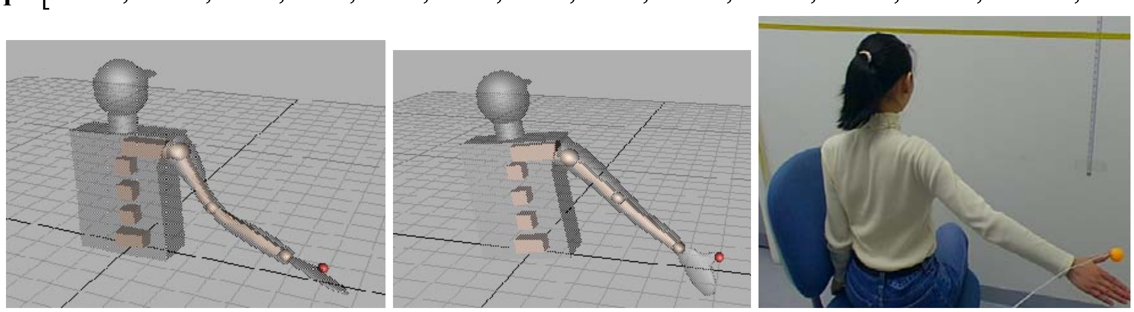


Fig. 7: Target Point 7 (0, -60, 5), Discomfort = 0.7253

$\mathbf{q} = [-.0120, -.0348, .0052, .0096, .0344, .0022, .0016, .0265, -.0643, .9742, -.1020, -.5219, -1.7381, -.0602, .1526]$

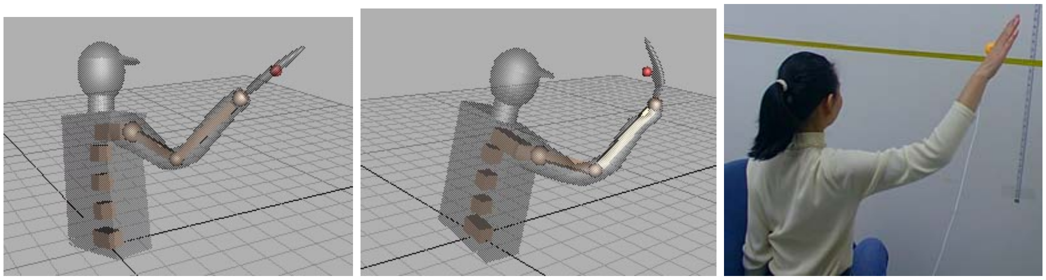


Fig. 8: Target Point 8 (30, -40, 60), Discomfort = 3.8352

$\mathbf{q} = [.1634, -.2485, .0292, .0468, .0703, .0418, -.0048, .0400, -.1602, .3555, .1484, -1.0230, -1.0510, .0806, -.0517]$

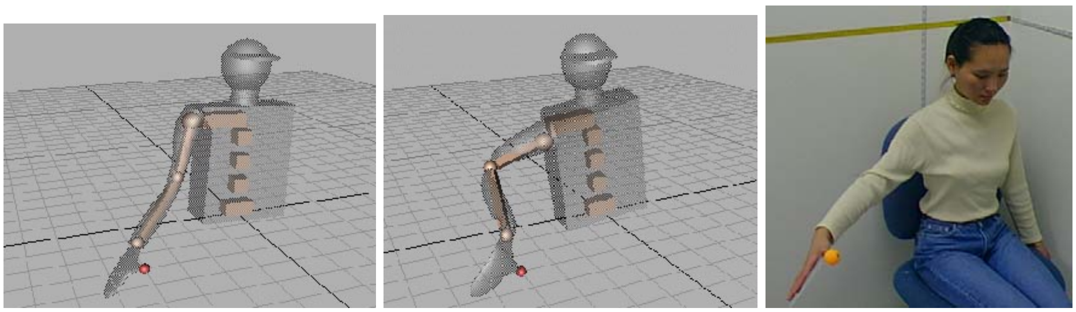


Fig. 9: Target Point 9 (30, -40, 0), Discomfort = 0.4966

$\mathbf{q} = [.0568, -.0618, .0135, .0030, -.0030, -.0056, -.0103, .0795, -.0314, 1.2111, .3822, -.3199, -1.7188, -.0346, -.0813]$

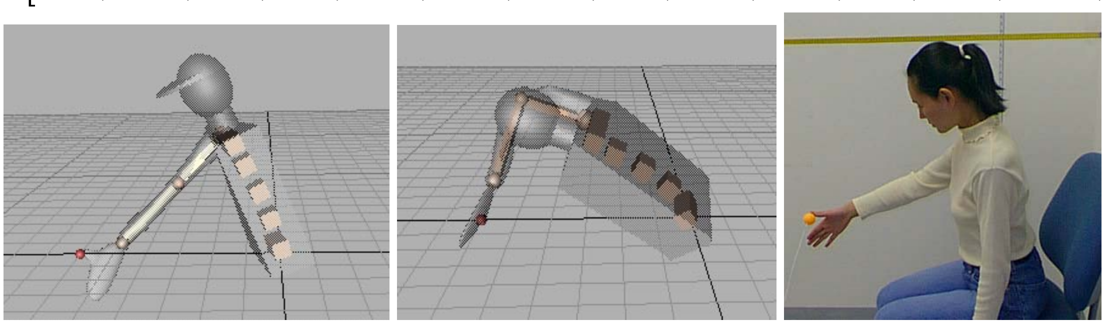


Fig. 10: Target Point 10 (60, 0, 0), Discomfort = 3.3709

$\mathbf{q} = [.2311, .0104, .2571, .0849, .1081, -.0150, .0203, -.3032, -.0402, 1.9117, 1.2721, -.0113, -1.6577, .0791, .3176]$

Conclusions

A general task-based real-time formulation for predicting posture has been proposed and demonstrated. Each task is proposed to comprise of one or more human performance measures and will be used within an optimization algorithm to iteratively calculate the joint variables that would be assumed in forming a posture. It was also proposed that each human performance measure be mathematically characterized as a function that evaluates to a real number and that can be used in a rigorous computational optimization algorithm. It was shown that the cost functions are minimized or maximized while converging on a set of joint variables that identify a posture. It was also shown that genetics algorithms are used to calculate a global solution. The modeling method was not restricted to any number of degrees of freedom.

Validation of the method against a well-known commercial inverse kinematics algorithm and confirmation with human subjects was presented. It is evident that the proposed method yields postures that minimize the specified cost function. However, it is also evident that many more cost functions are needed and more elaborate mathematical descriptions of human performance measures are required for various tasks. On the other hand, it is evident that this method provides a robust approach to realistic posture prediction that can handle a biomechanically accurate model.

References

Beck, D.J.; Chaffin, D.B.; "Evaluation of Inverse Kinematic Models for Posture Prediction", *Proceedings of the International Conference on Computer Aided Ergonomics and Safety - CAES* '92 May 18-20, pp. 329.

Cao, L.Y., Kim, B.G., Kurths, J., and Kim, S., "Detecting Determinism in Human Posture Control Data", *International Journal of Bifurcation and Chaos*, Vol. 8, No. 1, pp. 179-188.

Dysart, Marc J.; Woldstad, Jeffrey C., 1996, "Posture prediction for static sagittal-plane lifting"**,** *Journal of Biomechanics*, v29 n10 Oct 1996, pp. 1393-1397.

Faraway, Julian J.; Zhang, Xudong; Chaffin, Don B., "Rectifying postures reconstructed from joint angles to meet constraints", *Journal of Biomechanics*, v 32 n 7 1999, pp. 733-736.

Goldberg, D., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.

Jung, E.S.; Kee, D.; Chung, M.K., "Upper Body Reach Posture Prediction for Ergonomics Evaluation Models", *International Journal of Industrial Ergonomics*, v16 n2 1995, p 95.

Jung, Eui S.; Choe, Jaeho, 1996, "Human reach posture prediction based on psychophysical discomfort**",** *International Journal of Industrial Ergonomics*, v18 n2-3 Sep 1996, pp. 173-179.

Jung, Eui S.; Choe, Jaeho, 1996, "Human reach posture prediction based on psychophysical discomfort**",** *International Journal of Industrial Ergonomics*, v18 n2-3 Sep 1996, pp. 173-179.

Jung, Eui S.; Choe, Jaeho; Kim, Sung H., 1994, "Psychophysical cost function of joint movement for arm reach posture prediction**"** *Proceedings of the 38th Annual Meeting of the Human Factors and Ergonomics Society,* Part 1 (of 2) Oct 24-28 1994 v1, Nashville, TN, pp. 636-640.

Jung, Eui S.; Kee, Dohyung; Chung, Min K., "Reach posture prediction of upper limb for ergonomic workspace evaluation**",** *Proceedings of the 36th Annual Meeting of the Human Factors Society*, Part 1 (of 2) Oct 12-16 1992 v 1, Atlanta, GA, pp. 702-706.

Jung, Eui S.; Park, Sungjoon, "Prediction of human reach posture using a neural network for ergonomic man models", *Proceedings of the 16th Annual Conference on Computers and Industrial Engineering*, Mar 7-9 1994 v27 n1-4 Sep 1994 Ashikaga, Japan, pp. 369-372.

Wang, Xuguang, **“**Behavior-based inverse kinematics algorithm to predict arm prehension postures for computer-aided ergonomic evaluation**,”** *Journal of Biomechanics*, v 32 n 5 1999, pp. 453-460.

Zhang, Xudong; Chaffin, Don B., 1996, “Task effects on three-dimensional dynamic postures during seated reaching movements: an analysis method and illustration”, *Proceedings of the 1996 40th Annual Meeting of the Human Factors and Ergonomics Society*. Part 1 (of 2) Sep 2-6 1996 v1, Philadelphia, PA, pp. 594-598.

Zhang, Xudong; Chaffin, Don B., 1996, “Task effects on three-dimensional dynamic postures during seated reaching movements: an analysis method and illustration”, *Proceedings of the 1996 40th Annual Meeting of the Human Factors and Ergonomics Society*. Part 1 (of 2) Sep 2-6 1996 v1, Philadelphia, PA, pp. 594-598.