Robotica (2009) volume 27, pp. 607–620. © 2008 Cambridge University Press
doi:10.1017/S0263574708004992 Printed in the United Kingdom

Optimization-based posture prediction for human upper body

Zan Mi†, Jingzhou (James) Yang†,‡,* and Karim Abdel-Malek†

†*Center for Computer-Aided Design, The University of Iowa, Iowa City, IA 52242-1000.*
‡*Department of Mechanical Engineering, Texas Tech University, Lubbock, TX 97409-1021.*

(Received in Final Form: July 2007. First published online: August 11, 2008)

SUMMARY

A general methodology and associated computational algorithm for predicting postures of the digital human upper body is presented. The basic plot for this effort is an optimization-based approach, where we believe that different human performance measures govern different tasks. The underlying problem is characterized by the calculation (or prediction) of the human performance measure in such a way as 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, including joint limits. The formulation is demonstrated and validated. We present this computational formulation as a broadly applicable algorithm for predicting postures using one or more human performance measures.

KEYWORDS: Posture prediction; Inverse kinematics; Human postures.

1. Introduction

Posture prediction from a human modeling and simulation point of view is an important research area because of its eminent use in the ergonomic design process; it introduces rigor into the process and reduces occupational injuries.

There have been two schools of thought regarding the posture prediction. The first, perhaps the more traditional, uses anthropometrical data, collected from performing thousands of experiments on human subjects, or simulation using three-dimensional computer-aided human-modeling software,[6,19] which were statistically analyzed to form a predictive model of posture, such as 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,[2] Zhang and Chaffin,[22] Das and Behara,[5] and Faraway *et al.*,[10] Cao *et al.*[3]

The second school of thought often used biomechanics and kinematics as a predictive tool (often referred to as *inverse kinematic solutions*), on a posture that has not been observed but has been estimated as a likely posture for a task.[20] 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.*,[17] Jung *et al.*,[13] Jung and Choe,[14] Kee and Kim,[18] and Wang.[21]

Researchers who belong to the first school of modeling (in particular, Beck and Chaffin[2]) cautioned that the inverse kinematics algorithm is not necessarily correct for prediction of posture because of its theoretical foundation and its difficulty in evaluating the Jacobian, in determining a closed-form equation for the posture, and modeling a large number of degrees of freedom (DOF). Others[1,10] meanwhile, have stated that the use of only statistical models does not provide avenues for rigorous ergonomic design and does not reflect a task-based approach. With this approach, an impracticable number of experiments involving human subjects must be conducted for each specific task, for every gender of every age, and anthropometric measure.

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

Jung *et al.*[13] developed an approximate analytical reach prediction algorithm where the Denavit and Hartenberg (DH) notation was used to represent human motion. They reportedly demonstrated that humans adopt postures of minimum joint displacement among all the feasible body configurations. Similar results were reported by Dysart and Woldstad,[8] 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 while exploring different criteria functions for selecting a final posture. The results from Dysart and Woldstad[8] showed that the first objective function (minimum total torque) was more accurate but that the models were limited to planar motion with a relatively small number of DOFs.

Also using the concept of inverse kinematics, Kee and Kim[18] proposed an approximate algorithm to generate the workspace including the foot and trunk motion. Some results have focused on a combination of methods such as rule-based empirical and optimization to address the posture prediction problem.[21] The emergence of Artificial Neural

* Corresponding author. E-mail: james.yang@ttu.edu

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

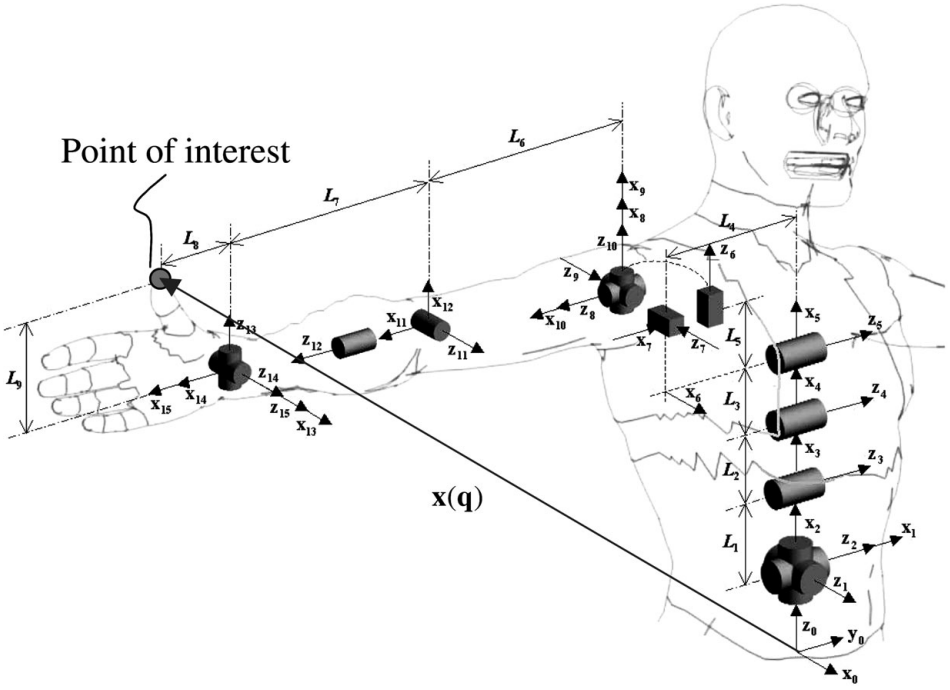


Fig. 1. Modeling of the upper body movement.

Networks models to provide more accurate predictions over the standard statistical models[9,12,16] is a promising development.

We will first present the human modeling method necessary for the analysis. Our task-based approach is presented in view of a rigorous mathematically-based optimization formulation, where cost functions characterizing human performance measures are implemented.

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

2. Kinematic Modeling

In this section, we describe the general modeling method used in our development, which has been adapted from the field of robotic kinematics. The method is used to characterize the 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[7] representation method (also known as the DH method) has been demonstrated to yield an effective method for modeling humans.[1,13]

The DH representation provides a systematic method for describing the relationship between adjacent links. The 4×4 transformation matrix describing a transformation from link $(i-1)$ to link i for joint i is

$$ {}^{i-1}\mathbf{T}_i = \begin{bmatrix} \cos q_i & -\cos\alpha_i \sin q_i & \sin\alpha_i \sin q_i & a_i \cos q_i \\ \sin q_i & \cos\alpha_i \cos q_i & -\sin\alpha_i \cos q_i & a_1 \sin q_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

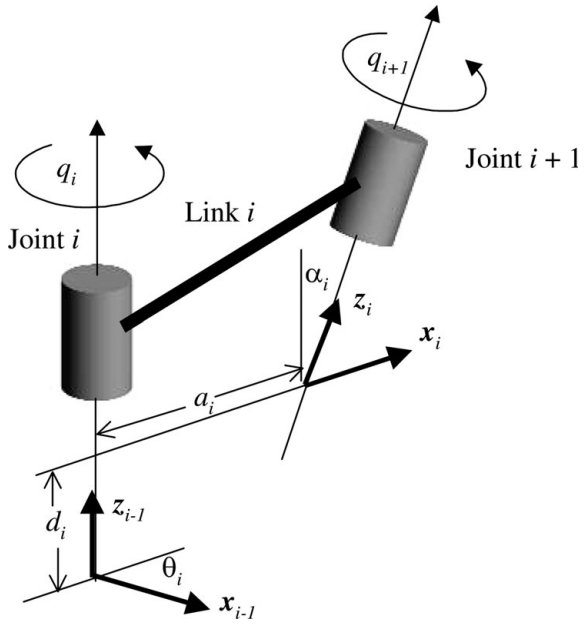


Fig. 2. Joint coordinate system convention and its parameters.

where θ_i, depicted in Fig. 2, is the joint angle from the x_{i-1} axis to the x_i axis, d_i is the distance from the origin of the $(i-1)$th coordinate frame to the intersection of the z_{i-1} axis with the x_i axis, a_i is the offset distance from the intersection of the z_{i-1} axis with the x_i axis, and α_i is the offset angle from the z_{i-1} axis to the z_i axis. Values for all the four variables are typically entered into a table called the DH table. This data set provides complete information about the model in terms of kinematic functionality as well as dimensions.

For an n-DOF model, 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 torso coordinate

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

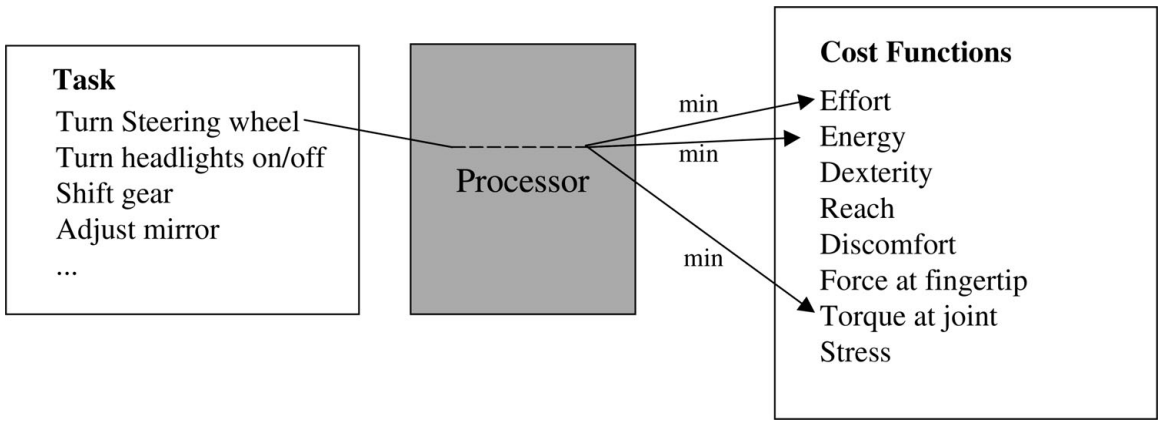


Fig. 3. The task-based approach to selecting cost functions.

system) can be written in terms of joint variables as

$$\mathbf{x} = \mathbf{x}(\mathbf{q}) \tag{2}$$

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 DH representation method as

$$\begin{bmatrix} \mathbf{x}(\mathbf{q}) \\ 1 \end{bmatrix} = {}^0\mathbf{T}_n \begin{bmatrix} \mathbf{v}_n \\ 1 \end{bmatrix} = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \ldots {}^{n-1}\mathbf{T}_n \begin{bmatrix} \mathbf{v}_n \\ 1 \end{bmatrix} \tag{3}$$

where $\mathbf{v}_n$ is the position vector with respect to coordinate frame n. The vector function $\mathbf{x}(\mathbf{q})$ characterizes the set of all points touched by the fingertip.

3. Task-Based Behavior

Researchers have become increasingly interested in posture prediction algorithms because of the digital human codes that enable a user to model a human mannequin, to manipulate the model, and to perform analysis on products and vehicles that would not otherwise be possible. Instead of building a complete prototype of a vehicle, for example, a virtual model is built from CAD data and the digital human is deployed into the vehicle to report various characteristics such as reach and field of view.

While there have been several posture prediction algorithms, a few have focused on calculating a posture for a given target point in space. Because of the apparent mathematical complexity of the problem and because the human body has many more than six DOFs, most of this research has focused on calculating as many inverse kinematic solutions (postures) as possible with a postprocessing technique to select those that are most realistic.

We propose in this paper a new framework and associated algorithm for predicting postures that is based on an individual task. Each task is comprised of one or more cost functions. In order to better understand the motivation behind the cost functions, consider the case of a driver in a vehicle who is about to reach a radio control button on the dashboard. It is believed that the driver will reach directly to the button while exerting minimum effort and perhaps expending minimum energy. However, the same driver, when negotiating a curve, will have to place his/her hand on the steering wheel in such a way as to be able to exert the force needed to turn the wheel. As a result, the driver will involuntarily select a posture that maximizes force at the hand, minimizes the torque at each joint, minimizes energy, and minimizes the effort needed to accomplish this task (as illustrated in Fig. 3). Therefore, our underlying plot is that each task is driven by the optimization of one or more cost functions. Note that simple logic has been implemented in the processor module to correlate between the task and the cost functions.

What we conclude from this simple example is that humans assume different postures for different tasks. Indeed, the objective leading to a given posture is one or more performance measures, which we denote by the cost functions. The designation comes from the field of optimization where the formulation contains three main ingredients:

(a) A cost function to be minimized or maximized. In our case, many cost functions, thus forming a multiobjective optimization problem.
(b) Design variables: the variables that will be calculated from the algorithm. In our case, the joint variables that define the position and orientation of each link connecting two joints.
(c) Constraints, which are mathematical expressions that bound the problem. In our case, joint-limit and distance constraints.

It is evident that the motion subtended by the human upper extremity to reach a specific target is directly dependent upon the arm's initial posture (i.e., initial conditions). A person will usually reach towards a target using the least motion of the joints possible. A person will also usually avoid exerting unnecessary energy against gravity (i.e., humans do not like to maintain their arms up in the air). In addition, we believe that the displacement of each joint from its most neutral position is a major factor in comfort and plays an important role in assuming a posture.

These functions are subject to kinematic constraints (developed in the following section). The most basic functions will be used to illustrate our formulation, although

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

other cost functions have been developed to characterize dexterity, stress, torque, and force. We believe that many more exist, such as strength, balance, shoulder mobility, and the orientation and weight of the objects being moved; this will be the subject of future investigations. This paper will focus on introducing a general formulation and demonstrating that realistic postures are predicted.

4. Cost Functions and Constraints

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.

4.1. Cost functions

Joint Displacement: Consider a cost function that measures the level of difference 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 $|q_i - q_i^N|$. Because joint displacement is felt more in some joints, we also introduce a weight function w_i to stress the importance of one joint versus another. The total joint displacement of all the joints is then characterized by the function

$$f_{discomf}(\mathbf{q}) = \sum_{i=1}^{n} w_i \left(q_i - q_i^N\right)^2 \tag{4}$$

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. Also we use a quadratic form to make sure the gradient is continuous.

Effort: Effort is measured as the displacement of a joint from its original position. It 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

$$f_{effort}(\mathbf{q}) = \sum_{i=1}^{n} \gamma_i \left(q_i - q_i^{initial}\right)^2 \tag{5}$$

where γ_i is a weight function. Note that f_{effort} depends on the initial configuration of each joint.

Delta Potential Energy: We represent the primary segments of the upper body with six lumped masses: three for the lower, middle, and upper torso; one for the upper arm; one for the forearm; and one for the hand. We then determine the potential energy for each mass. The heights of these masses, rather than the joint displacements for the generalized coordinates, provide the components of the human performance measure. Mathematically, the weight (force of gravity) of a segment of the upper body provides a multiplier for movement of that segment in the vertical direction. The height of each segment is a function of generalized coordinates, so, in a sense, the weights of the lumped masses replace the scalar multipliers, w_i, which are used in the joint displacement function. If the potential energy function were used directly, there would always be a tendency to bend over, thus reducing potential energy. Consequently, we actually minimize the *change* in potential energy. Each link in a segmented serial chain, as depicted in Fig. 4 (e.g., the forearm), has a specified center of mass.

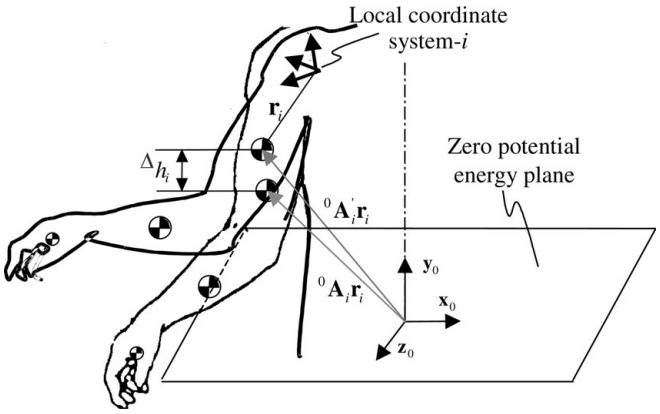


Fig. 4. Illustration of the potential energy of the upper body.

The vector from the origin of a link's local coordinate system to its center of mass is given by ${}^i\mathbf{r}_i$, where the subscript indicates the relevant local coordinate system. In order to determine the position and orientation of any part of the body, we use the transformation matrices ${}^{(i-1)}A_i$, which are 4×4 matrices that relate the local coordinate system-i to the local coordinate system-$i-1$. Consequently, $\mathbf{r}_i$ is actually an augmented 4×1 vector with respect to local coordinate system i, rather than a 3×1 vector typically used with Cartesian space. $\mathbf{g} = [0 \ -g \ 0 \ 0]^T$ is the augmented gravity vector. When the human upper body moves from one configuration to another, there are two potential energies: P_i', which is associated with the initial configuration, and P_i, which is associated with the current configuration. Therefore, for the first body part in the chain (the lower torso), the potential energies are $P_1' = m_1\mathbf{g}^T{}^0\mathbf{A}_1'\mathbf{r}_1$ and $P_1 = m_1\mathbf{g}^T{}^0\mathbf{A}_1\mathbf{r}_1$. The potential energies for the second body part (the upper torso) are $P_2' = m_2\mathbf{g}^T{}^0\mathbf{A}_1'{}^1\mathbf{A}_2'\mathbf{r}_2$ and $P_2 = m_2\mathbf{g}^T{}^0\mathbf{A}_1{}^1\mathbf{A}_2\mathbf{r}_2$. The potential energies for the i^{th} body part are $P_i' = m_i\mathbf{g}^T{}^0\mathbf{A}_1'\mathrm{L}^{\,i-1}\mathbf{A}_i'\mathbf{r}_i$ and $P_i = m_i\mathbf{g}^T{}^0\mathbf{A}_1\mathrm{L}^{\,i-1}\mathbf{A}_i\mathbf{r}_i$. In Fig. 4, Δh_i is the y-component of the vector ${}^0\mathbf{A}_1' \cdots {}^{i-1}\mathbf{A}_i'\mathbf{r}_i - {}^0\mathbf{A}_1 \cdots {}^{i-1}\mathbf{A}_i\mathbf{r}_i$. The final objective function, which is minimized, is defined as follows:

$$f_{Delta-potential-energy}(\mathbf{q}) = \sum_{i=1}^{\kappa} (P_i - P_i')^2 \tag{6}$$

Note that Eq. (6) can be written in the form of a weighted sum as follows:

$$f_{Delta-potential-energy}(\mathbf{q}) = \sum_{i=1}^{\kappa} (m_i g)^2 (\Delta h_i)^2 \tag{7}$$

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

where $(m_i g)^2$ represents the weights, $(\Delta h_i)^2$ acts as the individual objective functions, and $\kappa = 6$ is the number of lumped masses. In this case, the initial position is the neutral position described in relation to joint displacement. With this performance measure, the digital human again gravitates toward the neutral position. However, horizontal motion of the lumped masses has no effect on the objective function.

4.2. Constraints

For the point on the end-effector characterized by $\mathbf{x} = [\,x\ y\ z\,]^T$ as a function of all joint variables $\mathbf{x(q)}$ to reach a target point $\mathbf{p}$, it is necessary that $\mathbf{x(q) - p = 0}$. 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 computational problem. This criterion is indeed better suited for a constraint that must be driven by one or more cost functions, i.e., we impose a driving function to move the human linkage while enforcing the zero-distance constraint. Therefore, we implement this equation as a constraint to be imposed within a specified tolerance ε (e.g., 0.001), such that

$$\|\mathbf{x}(\mathbf{q}) - \mathbf{p}\| \leq \varepsilon \tag{8}$$

Furthermore, each DOF has unilateral constraints imposed in the form of

$$q_i^L \leq q_i \leq q_i^U;\ i = 1, \ldots, n \tag{9}$$

where n is the number of DOFs used in the model. Note that for a 15-DOF model as implemented in our code, a total of 31 constraints must be imposed (two for each unilateral constraint and Eq. (9)).

5. Optimization Formulation

Genetic algorithms (GA) are based on an evolution of random tries by "individuals," not on logic as regular algorithms are.[11] The driver in a GA is the *cost function*, which determines the fitness of each individual. Each cycle in the process of evaluation, selection of survivors, breeding, evaluation, etc. is called a generation. The GA continues the iteration of generations until the goal is found or some other end criterion occurs. Mutation helps prevent a degenerate population, which tends to settle prematurely at a nonoptimum solution. Optimization can improve both the GA's speed and its capability to solve problems. Genetic algorithms have two important characteristics that make them applicable to calculating postures:

(1) Genetic algorithms tend to a global minimum versus only a local solution.
(2) Genetic algorithms are usually used when the search space is very large.

An adaptation of a GA algorithm to the problem described herein is shown in Fig. 5.

First, lower and upper limits for each joint variable are input along with other information, such as position of target point, population size, crossover probability, mutation probability, and number of possibilities per parameter. In our implementation, the latter is specified as 1024. Then, the total number of chromosomes (bits) required is counted. For example, there are $2^{10} = 1024$ possibilities for each parameter, which means that each will need 10 bits of binary to characterize its description, i.e., the number of chromosomes needed is 10. Subsequently, for a total of 15 joint variables, 150 chromosomes are required. Subsequently, the main generational processing loop begins: each individual of the population is evaluated, i.e., binary values are converted to a decimal and a function evaluator is executed to calculate the fitness. At this time, a commercial gradient-based optimization algorithm called the Design Optimization Toolkit (DOT) is called inside the function evaluator to perform a minimization of the distance between a specified point on the end-effector and the target point (i.e., to enforce the distance constraint).

The overall procedure is as follows: the GA passes the parameters to the function evaluator while the DOT algorithm inside the function uses these parameters as initial values to minimize the distance. If the distance is less than a small tolerance, the weighted cost function (e.g., joint displacement) for the 15 joints is calculated and its negative is passed as the function value; otherwise, a very large negative value is passed. Since the GA only processes maximization, a negative value of the cost functions is passed. This value is assigned as fitness in GA, where the individual with the best fitness is obtained. Subsequently, a tournament selection is implemented in the GA. Before each selection cycle for a new generation, a shuffle is first performed on the parent population. Then, each tournament selects the parent with better fitness from two possible parents for mating. After all the parent pairs for the creation of the child population have been selected, a uniform crossover is performed between each randomly selected pair, where each bit in the chromosome of one parent is exchanged with the bit from the other parent under some random probability.

Following the mating process, two kinds of mutations are performed. One is called jump mutation, in which some randomly selected bits jump from 1 to 0 or from 0 to 1. Jump mutation is performed on binary code directly while the other type of mutation, creep mutation, is performed on the decimal number, where each randomly selected parameter is moved one discrete position away within the imposed limits. After all the children have been generated, a child array is written back into the parent array for a new generation. Finally, a check is performed to verify whether the best parent was replicated. If not, the GA reproduces the best parent into a random slot. After the main loop has been executed for a specified number of generations, the optimized parameters are obtained.

For the purpose of our computer code, the *Input File* to the system (initial configuration) contains the following information.

(a) *The DH table*: The DH table provides all the necessary information to model the human body, including joints and dimensions (see Appendix).

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

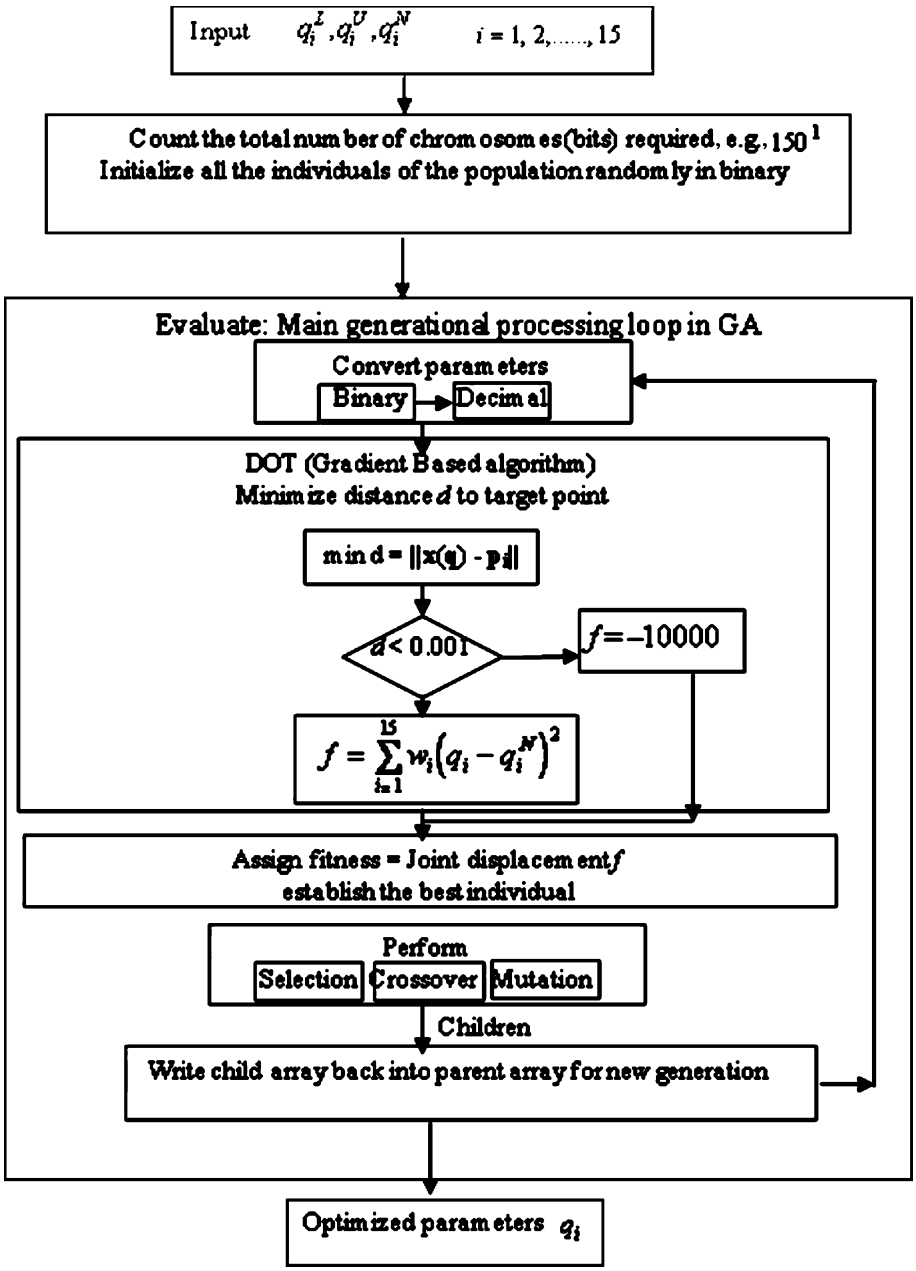


Fig. 5. Algorithm for predicting a posture using genetic algorithms.

(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:

(c) $q_i^N = 0; \quad i = 1, \ldots, 9, 11, 12, 14, 15, \quad q_{10}^N = \pi/2, \quad$ and $q_{13}^N = -\pi/2$.

(d) *Initial positions*: This is the last known configuration of the model prior to executing the command. The initial positions are used to evaluate the effort function.

(e) *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 $q_i - q_i^N$
$q_3 \ldots q_6$	10	When $q_i - q_i^N > 0$
	100	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 $q_i - q_i^N > 0$

6. Examples

To demonstrate our formulation and computer implementation, the 15-DOF model is developed in Fig. 1. In order

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

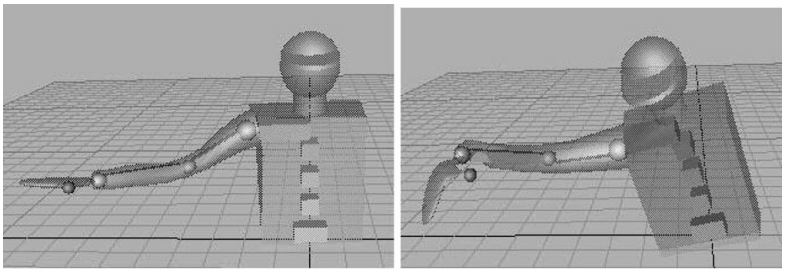


Fig. 6. Target Point 1 (41.2, −57.0, 31.5), Joint displacement = 2.2022.

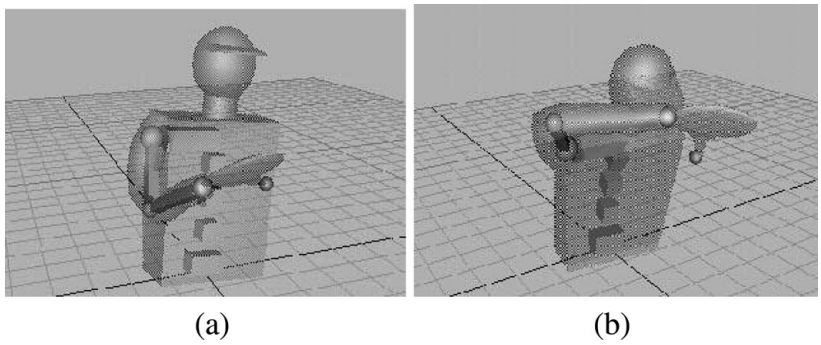


Fig. 7. Target Point 2 (40.0, 0.0, 36.0), Joint displacement = 7.38.

to obtain a better understanding of our results, we have compared the calculated postures with those produced by 3-D Studio Max's inverse kinematics module. For a given target position defined below, we have optimized a posture for a maximum joint displacement function. For each set of figures, Figs. 6–15(a) is a result of the optimization algorithm, and Figs. 6–15(b) is obtained from 3-D Studio Max's inverse kinematic algorithm.

Note that the calculation of the cost function for each posture is an important element of the prediction. For the first posture in Fig. 6, the calculated cost function is 2.2022, a relatively low joint displacement. 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. 12 and 14, where the calculated joint displacement values are also relatively small, 0.72 and 0.496, respectively. Difficult postures (the most uncomfortable) are observed in Fig. 8 (joint displacement = 12.825) and Fig. 7 (joint displacement = 7.38). Note that these postures contain several joint angles that are far from neutral, therefore adding to the cost function. The computer to run these examples has a 1.8 GHz processor with 512 MB memory. The CPU time to run each example is less than 10 s based on the 2000 generation in GAs. The CPU time using 3-D Studio Max is around 20 s.

Because we have opted to use genetic algorithms for optimization, the stopping criterion is not well defined (an

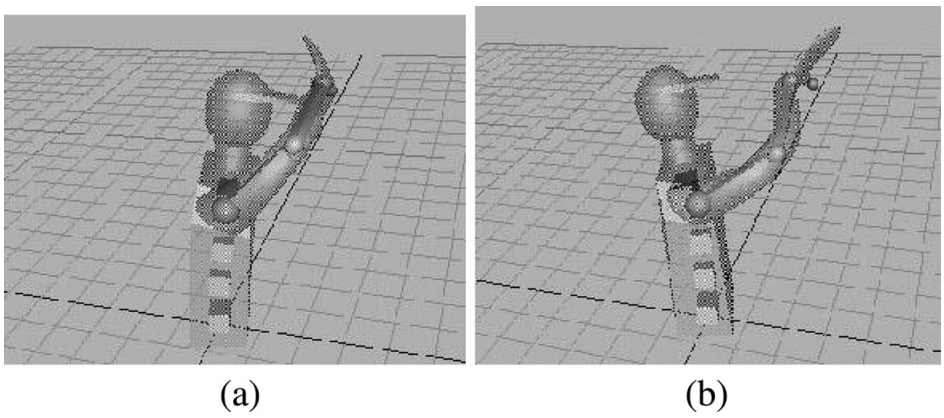


Fig. 8. Target Point 3 (20.0, 35.0, 50.0), Joint displacement = 12.8254.

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

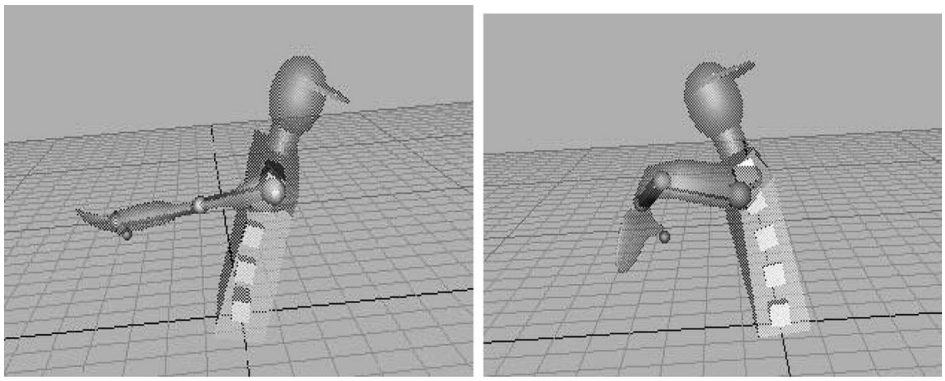


Fig. 9. Target Point 4 (−30.0, 10.0, 20.0), Joint displacement = 2.0873.

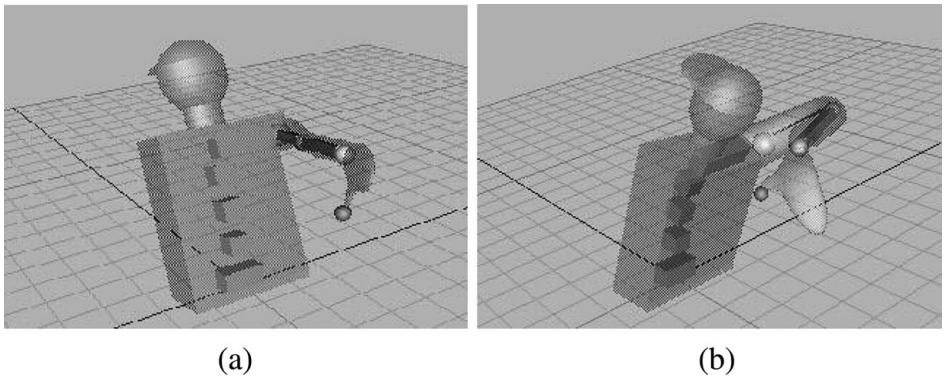


Fig. 10. Target Point 5 (−40.0, 0.0, 36.0), Joint displacement = 1.5824.

inherent characteristic of GAs). 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 the finger and the target point, the program must continue to minimize the joint displacement cost function until an acceptable low value is achieved.

Consider now a task that requires the evaluation of several cost functions simultaneously as a multiobjective optimization problem. For a target point of (41.2, −57, 31.5), the joint displacement function is optimized to $f_{Joint-Disp} = 3.4703$, the effort function to $f_{effort} = 1.8405$, and the potential function to $f_{potential} = 6.6597$. The posture is predicted and shown in Fig. 16. Note that, in comparison with the previous posture using one cost function (Fig. 6), the joints have slightly changed.

Similarly, compare the posture predicted and shown in Fig. 7 with that in Fig. 17, for which the target point is (40.0, 0.0, 36.0). Joint variables are calculated to a similar posture, but one that has slightly different joint values. Note that all of the effort, potential energy, and joint displacement values evaluate to nonzero numbers, indicating that both cost functions play an important role in predicting this posture. Joint displacement = 11.0847, Effort = 5.8041, and Potential Energy = 7.6123.

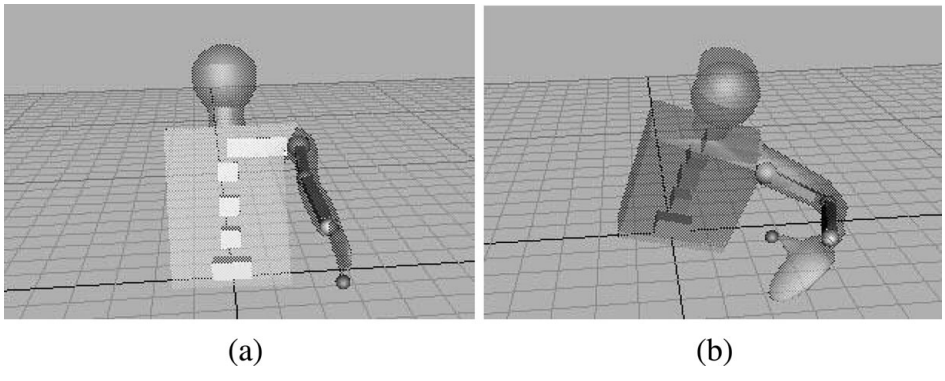
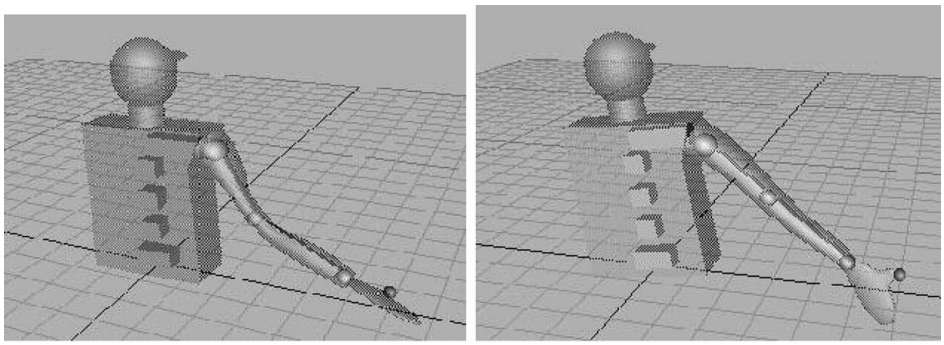


Fig. 11. Target Point 6 (−50.0, −20.0, 20.0), Joint displacement = 1.0783.

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press



(a) (b)

Fig. 12. Target Point 7 (0.0, −60.0, 5.0), Joint displacement = 0.7253.

To demonstrate the applicability of various cost functions, consider two different initial configurations, i.e., initial postures from which the arm and torso will move to reach a desired target point specified by (41.2, −57, 31.5). We shall use the *Effort* function, which depends upon the initial configuration (more effort is needed if each joint travels longer distances).

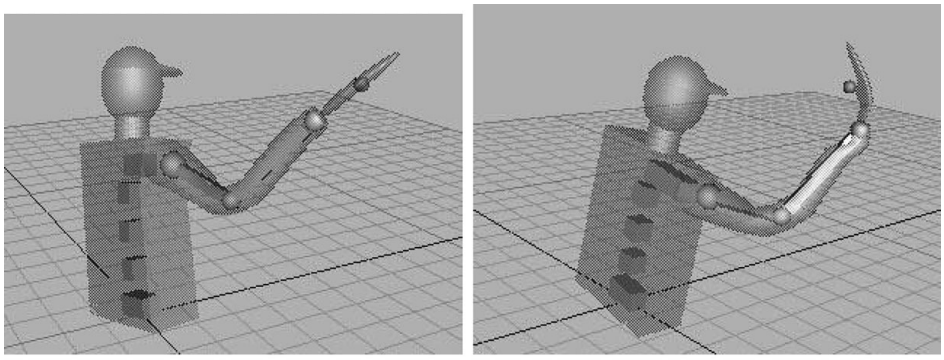
Case 1 will start from an initial posture that has all the joints in their defined neutral configuration (i.e., the most relaxed position) and given by $q_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, \pi/2, 0, 0, -\pi/2, 0, 0]$. The calculated final posture is shown in Fig. 18(a), where the effort is calculated as $f_{\text{Effort}} = 1.2075$.

Case 2 will start from an initial posture that has all joints at zero (i.e., home posture where we start our simulations from), $q_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. The calculated effort is $f_{\text{Effort}} = 0.3541$, and the calculated posture is shown in Fig. 18(b).

The important result from the above two postures is that they are different, which is a behavior observed in humans.

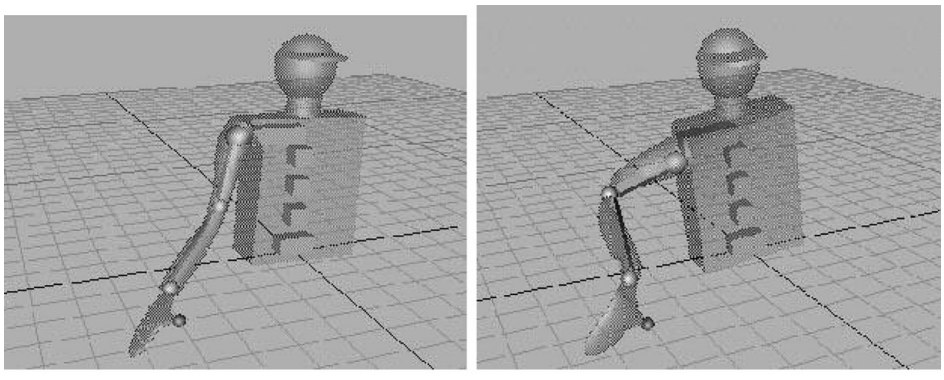
7. Validation

To validate our approach, we use the experimental database[4] through motion captured by HUMOSIM (Human Motion SIMulation at the Center for Ergonomics, University of



(a) (b)

Fig. 13. Target Point 8 (30.0, −40.0, 60.0), Joint displacement = 3.8352.



(a) (b)

Fig. 14. Target Point 9 (30.0, −40.0, 0.0), Joint displacement = 0.4966.

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

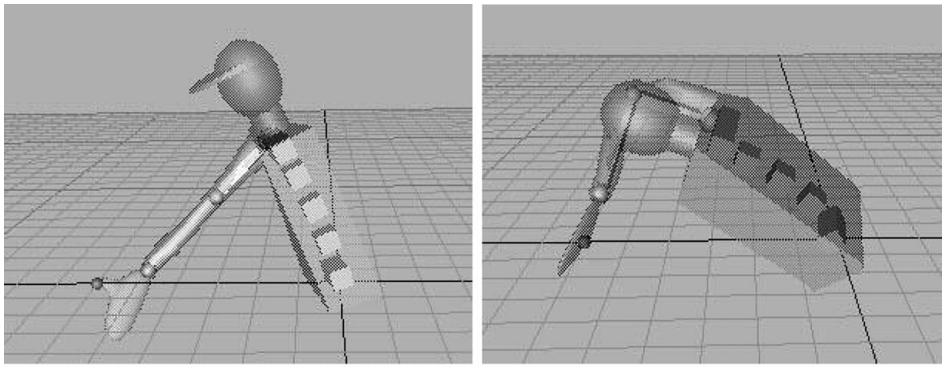


Fig. 15. Target Point 10 (60.0, 0.0, 0.0), Joint displacement = 3.3709.

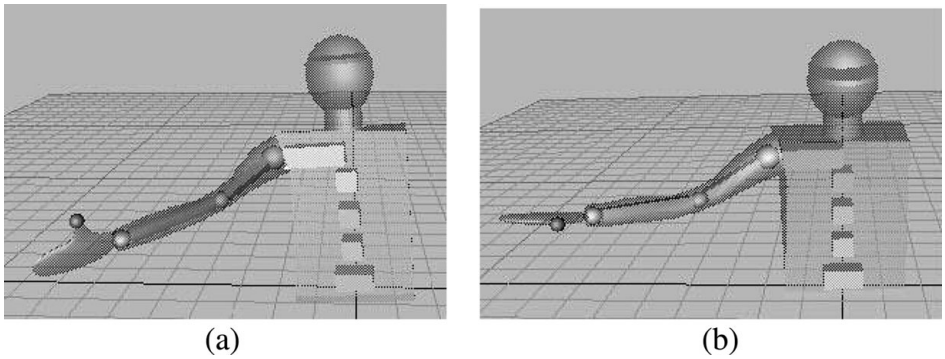


Fig. 16. Posture prediction for (41.2, −57, 31.5) (a) Multiobjective optimization (b) Single cost function.

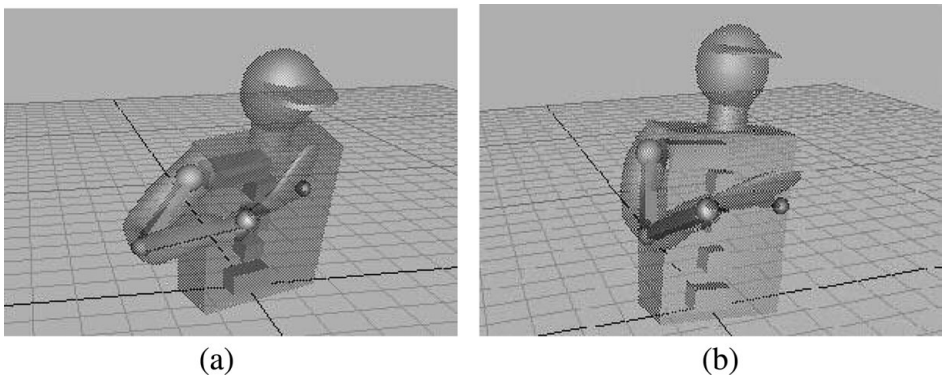


Fig. 17. Posture prediction for (40.0, 0.0, 36.0) (a) Multiobjective optimization (b) Single cost function.

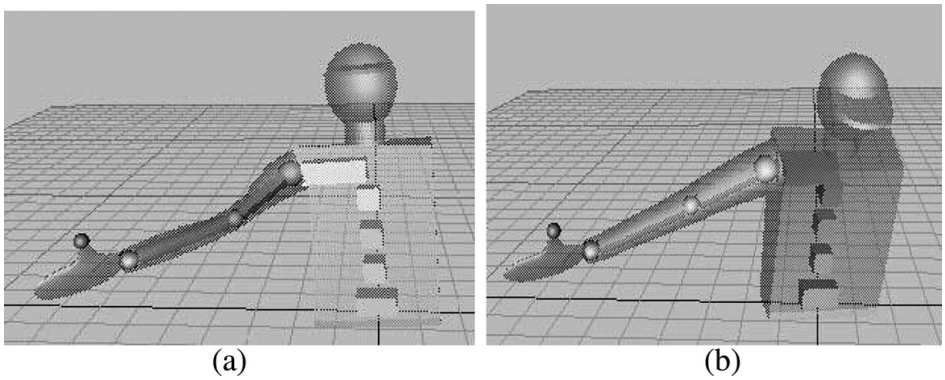


Fig. 18. Prediction postures based on two different initial postures (a) Initial configuration is neutral position (b) Initial configuration has joints at zero.

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

Michigan) to compare the simulation and experimental results. HUMOSIM database files are of the following five types: event records (.evt), global angles (.gan), local angles (.lan), location of joint centers and landmarks (.loc), and unit vector for rotation matrix (.uvt). Event records indicate the status of each data entry in the other tracking data files. There are two types of reach, Go To and Return. For our purpose of posture validation, we need to extract the last row of Go To type and the first row of Return type from the data where the subject was in stable contact with the target. To find the position and orientation of the target, the data from files with .loc and .uvt extensions were extracted. We chose the location of the electromagnetic marker on the back of the hand as our end-effector position.

All the tracking data are given for Subject 19, and there are 480 different targets obtained by counting all the combinations of the following: Left 45 Degrees, Forward, Right 45 Degrees, or Right 90 Degrees; Push or Pull; Top Pod, Middle Pod, or Lowest Pod; Center Surface, Left Surface, Top Surface, Right Surface, or Bottom Surface of the tower block; Thumb on Top, Thumb on Right, Thumb on Bottom, or Thumb on Left of each surface. We took twelve files for validation (reach type is Go To, Return type files will give the same target information):

O19g003h: Left 45 Degrees, Push, Top Pod, Center, Thumb on Bottom
O19g004h: Left 45 Degrees, Push, Top Pod, Center, Thumb on Left
O19g007h: Left 45 Degrees, Push, Top Pod, Left, Thumb on Bottom
O19g097l: Left 45 Degrees, Pull, Middle Pod, Bottom, Thumb on Top
O19g122h: Forward, Push, Top Pod, Center, Thumb on Right
O19g170h: Forward, Push, Lowest Pod, Top, Thumb on Right
O19g218l: Forward, Pull, Middle Pod, Bottom, Thumb on Right
O19g244h: Right 45 Degrees, Push, Top Pod, Center, Thumb on Left
O19g340l: Right 45 Degrees, Pull, Middle Pod, Bottom, Thumb on Left
O19g349l: Right 45 Degrees, Pull, Lowest Pod, Top, Thumb on Top
O19g363h: Right 90 Degrees, Push, Top Pod, Center, Thumb on Bottom
O19g458l: Right 90 Degrees, Pull, Middle Pod, Bottom, Thumb on Right

The extracted location (from .loc) files have 119 columns, where columns 73–75 are for the right hand, i.e., the location of the electromagnetic marker on the back of the hand. The data of the unit vector were actually derived from the tracked location data. Among the 155 columns of the extracted unit vector data file, columns 46–54 are the right hand orientation. The position and orientation of the right hand were obtained by re-extracting the above columns from the extracted location and unit vector files.

The coordinates used in the HUMOSIM database are different from those in our model. For validation and comparison, transformations were made within our program to allow for a direct comparison of *two equivalent models* (Fig. 19). Human measurements, hand position, and orientation from UM were used as our optimization algorithm input. Joint variables were obtained for the above 12 targets with position and orientation while using joint displacement as a cost function.

Results are shown in Figs. 20–31. These figures illustrate posture prediction using our digital human modeling environment; postures for our 15-DOF model are shown along with the wireframe model representing the Michigan data. The figure captions correspond to the Michigan data file names.

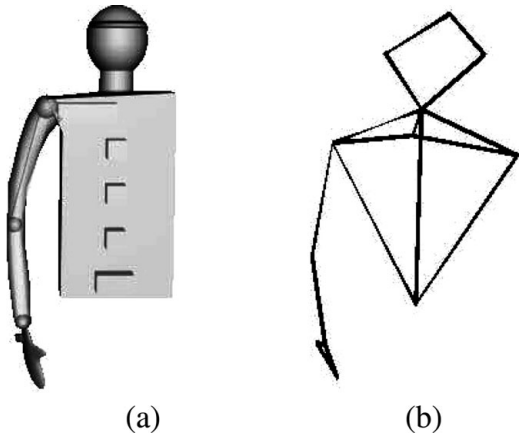


Fig. 19. (a) Our 15-DOF model (b) Michigan model.

As the examples illustrate, our optimization algorithm generally provides good results by using a simple joint displacement cost function (a more elaborate cost function has been developed since this study). The postures obtained from our program match those assumed by a real human subject as measured by HUMOSIM experiments in general.

In some cases, our results (Figs. 21, 23, and 26) predicted arms stretched out, while the experimental data did not. This is related to the use of our simple joint displacement cost function, which measures only the derivation of the arm from the neutral joint angles (e.g., straight elbow). This also suggests that potential energy should be included in the

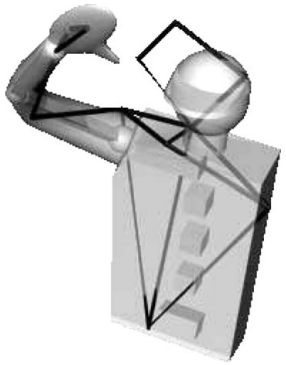


Fig. 20. Posture (g003h).

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

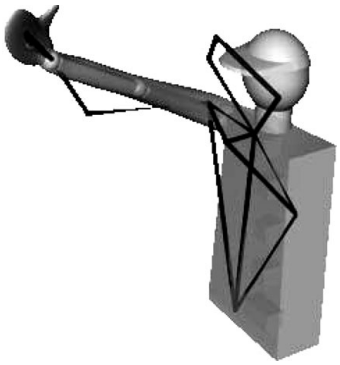


Fig. 21. Posture (g004h).

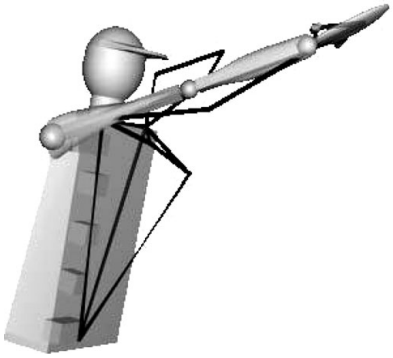


Fig. 22. Posture (g007h).

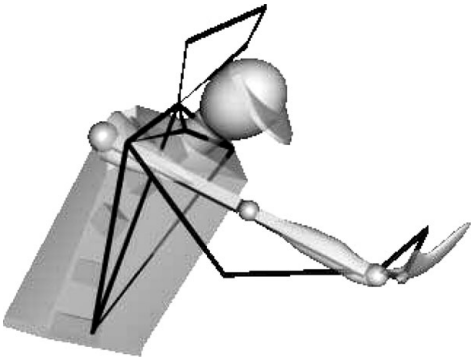


Fig. 23. Posture (g097l).

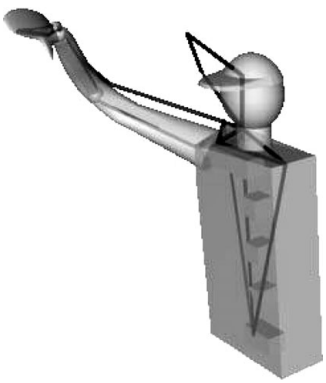


Fig. 24. Posture (g122h).

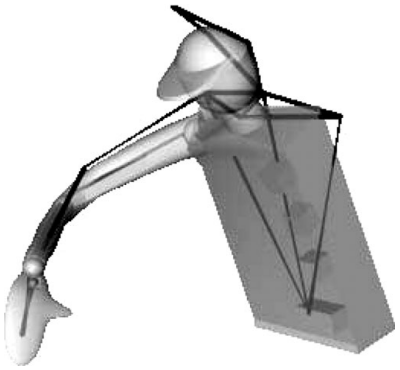


Fig. 25. Posture (g170h).

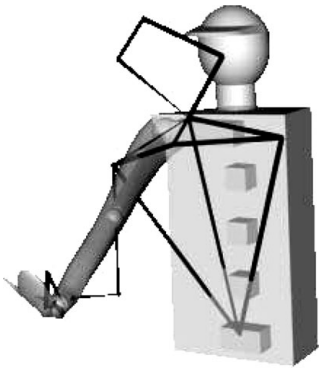


Fig. 26. Posture (g218l).

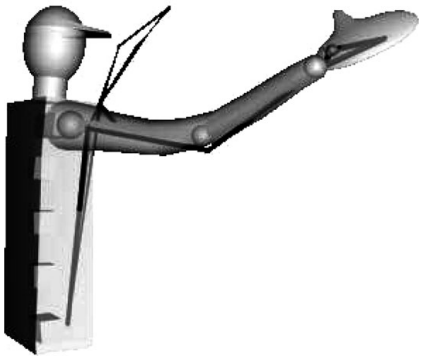


Fig. 27. Posture (g244h).

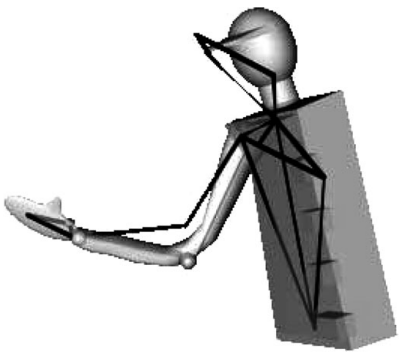


Fig. 28. Posture (g340l).

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

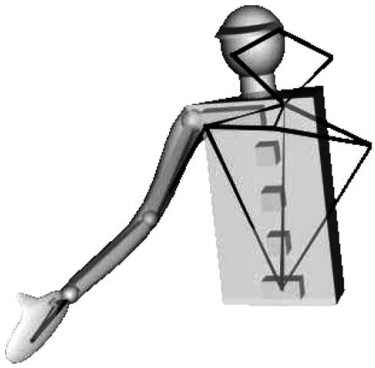


Fig. 29. Posture (g349l).

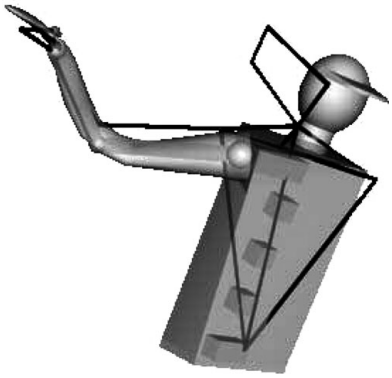


Fig. 30. Posture (g363h).

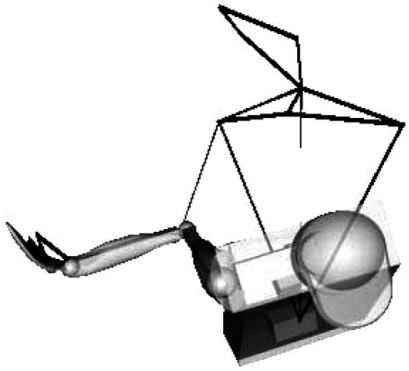


Fig. 31. Posture (g458l).

cost function. Another influential factor causing some joints to be actuated before others are the weighted scale factors corresponding to different joints. From all the figures, we can also observe that vision plays a very important role in the final posture and should be considered in our cost function (Fig. 31). A more accurate and complex cost function can be developed by considering all the factors mentioned above and by careful evaluation of the weight factors. Other ways to improve the accuracy of the prediction are to provide more accurate joint limits given as constraints and to use human models with more DOFs.

8. Conclusions

A general optimization-based formulation for predicting posture has been proposed, demonstrated, and validated. Each task comprises one or more human performance measures. These measures are rigorously defined and are used within an optimization algorithm to iteratively calculate the joint variables that would be assumed by a human 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 genetic algorithms can be used to calculate a global solution. The modeling method was not restricted to any number of degrees of freedom.

Perhaps the most important aspect of this method is that it does not employ the Jacobian in the numerical evaluation of a posture. This fact has enabled us to surpass the traditional limitation of the 6-DOF. Indeed, we have demonstrated the prediction of a 15-DOF human model using several cost functions.

Benefits of this method are also evident in its ability to represent tasks in terms of one or more cost functions. As demonstrated, more realistic posture prediction of human models is possible; it depends on the initial configuration, the range of motion, and the exact dimensions.

Validation of the method against a well-known commercial inverse kinematics algorithm and confirmation against 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.

Appendix: DH Table

The following is the DH table for the 15-DOF human model.

	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	$\pi/2$	$L_1 = 9$
4	q_4	0	0	$L_2 = 9$
5	q_5	0	0	$L_3 = 9$
6	$\pi/2 + q_6$	$-L_4 = -18$	$\pi/2$	0
7	$-\pi/2$	$L_5 + q_7$ ($L_5 = 9$)	$\pi/2$	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	$-\pi/2$	$L_6 = 20$
12	$-\pi/2 + q_{12}$	0	$-\pi/2$	0
13	$-\pi/2 + q_{13}$	$L_7 = 25$	$-\pi/2$	0
14	$-\pi/2 + q_{14}$	0	$-\pi/2$	0
15	q_{15}	0	0	0

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press

Acknowledgements

This research was partly supported by the U.S. Army TACOM project-Digital Humans and Virtual Reality for Future Combat Systems (FCS) (Contract No.: DAAE07-03-Q-BAA1).

References

1. K. Abdel-Malek, W. Yu, E. Tanbour and M. Jaber, "Posture Prediction Versus Inverse Kinematics," *Proceedings of the ASME Design Automation Conference*, Pittsburgh, PA (2001).
2. D. J. Beck and D. B. Chaffin, "Evaluation of Inverse Kinematic Models for Posture Prediction," *Proceedings of the International Conference on Computer Aided Ergonomics and Safety – CAES* '92, Rome, Italy, May 18–20, 1992 pp. 329.
3. L. Y. Cao, B. G. Kim, J. Kurths and S. Kim, "Detecting determinism in human posture control data," *Int. J. Bifur. Chaos* **8**(1), 179–188 (1998).
4. D. B. Chaffin, Motion Database (2003) http://www.engin.umich.edu/dept/ioe/HUMOSIM/database.html.
5. B. Das and D. N. Behara, "Three-dimensional workspace for industrial workstations," *Human Factors* **40**(4), 633–646 (1998).
6. B. Das and A. K. Sengupta, "Computer-aided human modeling programs for workstation design," *Ergonomics* **38**, 1958–1972 (1995).
7. J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. Appl. Mech.* **77**, 215–221 (1955).
8. M. J. Dysart and J. C. Woldstad, "Posture prediction for static sagittal-plane lifting," *J. Biomech.* **29**(10), 1393–1397 (1996).
9. M. Eksioglu, J. E. Fernandez and J. M. Twomey, "Predicting peak pinch strength: Artificial neural networks vs. regression," *Int. J. Ind. Ergonomics* **18**(5–6), 431–441 (1996).
10. J. J. Faraway, X. Zhang and D. B. Chaffin, "Rectifying postures reconstructed from joint angles to meet constraints," *J. Biomech.* **32**(7), 733–736 (1999).
11. D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, New York, 1989).
12. D. Hestenes, "Invariant body kinematics; reaching and neurogeometry," *Neural Networks* **7**(1), 79–88 (1994).
13. E. S. Jung, D. Kee and M. K. Chung, "Upper body reach posture prediction for ergonomics evaluation models," *Int. J. Ind. Ergonomics* **16**(2), 95 (1995).
14. E. S. Jung and J. Choe, "Human reach posture prediction based on psychophysical joint displacement," *Int. J. Ind. Ergonomics* **18**(2–3), 173–179 (1996).
15. E. S. Jung, J. Choe and S. H. Kim, "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), , Vol 1, Nashville, TN (Oct. 24–28, 1994) pp. 636–640.
16. E. S. Jung and S. Park, "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, Vol 27 n 1–4, Ashikaga, Japan (Sep. 1994) pp. 369–372 .
17. D. Kee, E. S. Jung and S. Chang, "A man-machine interface model for ergonomic design," *Comput. Ind. Eng.* **27**(1–4), 365–368 (1994).
18. D. Kee and S. H. Kim, "Analytic generation of workspace using the robot kinematics," *Comput. Ind. Eng.* **33**(3/ 4), 525–528 (1997).
19. J. M. Porter, K. Case and M. C. Bonney, "Computer workspace modelling,". **In**: *Evaluation of Human Work* (J. R. Wilson and E. N. Corlett, eds.) , (Taylor & Francis, London, 1990) pp. 472–499.
20. M. F. Tracy, "Biomechanical methods in posture analysis,". **In**: *Evaluation of Human Work* (J. R. Wilson and E. N. Corlett, eds.) (Taylor & Francis, London, 1990) pp. 571–604.
21. X. Wang, "Behavior-based inverse kinematics algorithm to predict arm prehension postures for computer-aided ergonomic evaluation," *J. Biomech.* **32**(5), 453–460 (1999).
22. X. Zhang and D. B. Chaffin, "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), Vol 1, Philadelphia, PA (Sep. 2–6 1996) pp. 594–598.

https://doi.org/10.1017/S0263574708004992 Published online by Cambridge University Press