

# Optimization-based layout design

doi:10.1533/abbi.2004.0046

K. Abdel-Malek<sup>1</sup>, Z. Mi<sup>1</sup>, J. Yang<sup>1</sup> and K. Nebel<sup>2</sup>

<sup>1</sup>*Virtual Soldier Research Program, Center for Computer-Aided Design, The University of Iowa, 111 Engineering Research Facility, Iowa City, IA 52242-1000, USA*

<sup>2</sup>*U.S. Army TACOM/RDECOM, AMSRD-TAR-NAC/157, 6501 East 11 Mile Rd., Warren, MI 48397-5000, USA*

**Abstract:** The layout problem is of importance to ergonomists, vehicle/cockpit packaging engineers, designers of manufacturing assembly lines, designers concerned with the placement of levers, knobs, controls, etc. in the reachable workspace of a human, and also to users of digital human modeling code, where digital prototyping has become a valuable tool. This paper proposes a hybrid optimization method (gradient-based optimization and simulated annealing) to obtain the layout design. We implemented the proposed algorithm for a project at Oral-B Laboratories, where a manufacturing cell involves an operator who handles three objects, some with the left hand, others with the right hand.

**Key words:** Layout design, global optimization, ergonomics, human modeling.

## INTRODUCTION

The layout design problem is defined as the method whereby positions of target points are specified in the work environment surrounding a human. Given a person's physical dimensions and ranges of motion, the person is required to locate a number of objects in the environment such that a specified cost function is optimized. The problem is of interest to ergonomists, automobile assembling engineers, and designers interested in locating targets (e.g., lever, buttons, control knobs, switches, etc.) within a person's reachable space. Since there are an infinite number of solutions to this problem, the field of optimization presents a viable venue for formulating the problem.

To a certain extent, some researchers have addressed implementing a systematic optimization scheme in ergonomics (Fisher 1993, Pham and Onder 1992). There are also many studies, most of them experimental, that delimit "comfort" and "convenient" reach zones in which objects may be placed to reduce operator effort and to minimize potential operator injuries (Lim and Hoffmann 1997, Das and Sengupta 1996).

The field of ergonomics received considerable attention after the appearance of digital human modeling software, where digital mannequins are manipulated to answer specific questions such as "Is this target point reachable? Is this load too heavy? Is this posture comfortable?" Whereas this type of software is a valuable tool for ergonomists and designers, it has not been able to provide a best design scenario.

In this paper, we present a method and accompanying code to address the layout problem from an optimization point of view. The general question is where to locate specific target points in the reachable space while optimizing one or more cost functions and while subject to a number of constraints. A global optimization method, simulated annealing, has been used to implement the algorithm and to yield global solutions. Because layout design must take into consideration a relatively large number of issues, some subjective, others more objective, we believe that our proposed optimization-based solution introduces a new method for making more educated design-related decisions. We have implemented this method at Oral-B Laboratories, where a manufacturing cell involves an operator who handles three objects, some with the left hand, others with the right.

## HUMAN KINEMATIC MODELING

To establish a systematic method for modeling human anatomy biomechanically, researchers have implemented conventions for representing segmental links and joints. Human anatomy can be represented as a sequence of rigid

---

Corresponding author:

Jingzhou Yang

Virtual Soldier Research Program

Center for Computer-Aided Design, The University of Iowa

111 Engineering Research Facility, Iowa City, IA 52242-1000, USA

Tel: +319 353 2249, Fax: +319 384 0542

Email: [jyang@engineering.uiowa.edu](mailto:jyang@engineering.uiowa.edu)

---

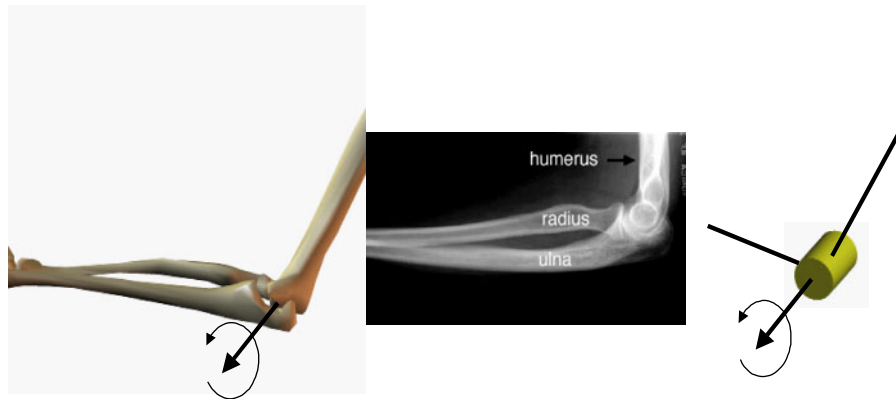


Figure 1 One DOF elbow.

bodies (links) connected by joints; this serial linkage could be an arm, a leg, a finger, a wrist, or other functional biomechanism. Joints in the human body vary in shape, function, and form, so the complexity offered by each joint must also be modeled, to the extent possible, to enable a correct simulation of the motion. The degree that a model replicates the actual physical model is called the level of fidelity.

Perhaps the most important element of a joint is its function, which may vary according to the joint's location and physiology. Physiology becomes important when we discuss the loading conditions of a joint. In terms of kinematics, the function is addressed in terms of the number of degrees of freedom (DOF) associated with the joint's overall movement. Muscle ligament, tendon, and actions at a joint are also important and contribute to the joint's function.

For example, consider the elbow joint, which is considered a hinge or one DOF rotational joint (e.g., the hinge of a door) because it permits flexion and extension in the sagittal plane (Figure 1), as the radius and ulna rotate about the humerus. We shall represent this joint by a cylinder that rotates about one axis and has no other motions (i.e., one DOF). Therefore, we can now say that the elbow is characterized by one DOF and is represented as a cylindrical rotational joint, as shown in Figure 1.

The torso, shoulder, and arm are modeled using a 15-DOF model (from waist up to the right hand) (Figure 2). The DH Table for the right hand is in Table 1(a). With a symmetric model to that shown in Figure 2 (with axes  $z_9, z_{10}, z_{11}, z_{13}, z_{14}$  having opposite directions to those shown in Figure 2), we have a DH Table for left arm as shown in Table 1(b), where joint variables  $q_{16}$  to  $q_{24}$  are symmetric to  $q_7$  to  $q_{15}$  of the right

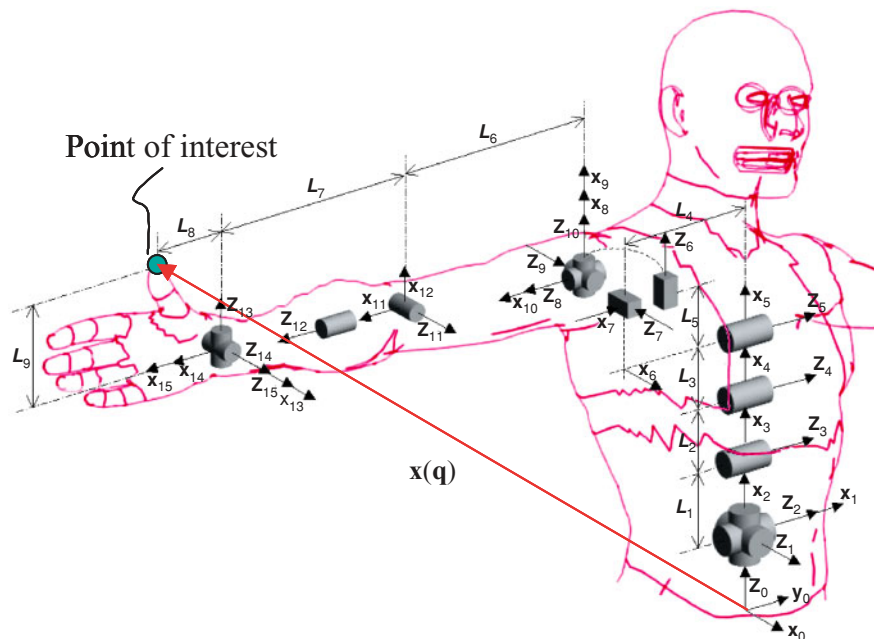


Figure 2 A 15-DOF model for upper body from waist up to right hand.

Table 1 DH Table for upper body

	(a)				(b)			
	$\theta_i$	$d_i$	$\alpha_i$	$a_i$	$\theta_i$	$d_i$	$\alpha_i$	$a_i$
1	$\pi/2 + q_1$	0	$\pi/2$	0	$\pi/2 + q_1$	0	$\pi/2$	0
2	$\pi/2 + q_2$	0	$\pi/2$	0	$\pi/2 + q_2$	0	$\pi/2$	0
3	$q_3$	0	0	$L_1$	$q_3$	0	$\pi/2$	$L_1$
4	$q_4$	0	0	$L_2$	$q_4$	0	0	$L_2$
5	$q_5$	0	0	$L_3$	$q_5$	0	0	$L_3$
6	$\pi/2 + q_6$	$-L_4$	$\pi/2$	0	$\pi/2 + q_6$	$L_4$	$\pi/2$	0
7	$\pi/2$	$L_5 + q_7$	$-\pi/2$	0	$\pi/2$	$L_5 + q_{16}$	$-\pi/2$	0
8	$-\pi/2$	$q_8$	$\pi/2$	0	$-\pi/2$	$q_{17}$	$\pi/2$	0
9	$q_9$	0	$\pi/2$	0	$\pi + q_{18}$	0	$\pi/2$	0
10	$\pi/2 + q_{10}$	0	$\pi/2$	0	$-\pi/2 + q_{19}$	0	$-\pi/2$	0
11	$q_{11}$	0	$-\pi/2$	$L_6$	$q_{20}$	0	$\pi/2$	$L_6$
12	$-\pi/2 + q_{12}$	0	$-\pi/2$	0	$-\pi/2 + q_{21}$	0	$\pi/2$	0
13	$-\pi/2 + q_{13}$	$L_7$	$-\pi/2$	0	$-\pi/2 + q_{22}$	$-L_7$	$\pi/2$	0
14	$-\pi/2 + q_{14}$	0	$-\pi/2$	0	$-\pi/2 + q_{23}$	0	$\pi/2$	0
15	$q_{15}$	0	0	0	$q_{24}$	0	0	0

arm. Therefore, we have a 24-DOF model for upper body, including both arms. The joint limits based on the experiments on three human subjects are  $-\pi/6 \leq q_1 \leq \pi/6$ ,  $-\pi/12 \leq q_2 \leq \pi/12$ ,  $-\pi/18 \leq q_3 \leq \pi/6$ ,  $-\pi/18 \leq q_4 \leq \pi/6$ ,  $-\pi/18 \leq q_5 \leq \pi/6$ ,  $-\pi/18 \leq q_6 \leq \pi/6$ ,  $-3.81 \leq q_7 \leq 3.81$ ,  $-3.81 \leq q_8 \leq 3.81$ ,  $-\pi/2 \leq q_9 \leq \pi/2$ ,  $-2\pi/3 \leq q_{10} \leq 11\pi/18$ ,  $-\pi/3 \leq q_{11} \leq 2\pi/3$ ,  $-5\pi/6 \leq q_{12} \leq 0$ ,  $-\pi \leq q_{13} \leq 0$ ,  $-\pi/3 \leq q_{14} \leq \pi/3$ ,  $-\pi/9 \leq q_{15} \leq \pi/9$ .

In order to obtain a systematic representation of any serial kinematic chain, we define  $\mathbf{q} = [q_1 \cdots q_n]^T \in \mathbf{R}^n$  as the vector of  $n$ -generalized coordinates defining the motion of a limbs with respect to each other, where  $q_i$  is the individual DOF variable. The position vector function (shown in Figure 2), generated by a point of interest written as a multiplication of rotation matrices and position vectors, is expressed by

$$\mathbf{x}(\mathbf{q}) = \begin{bmatrix} x(\mathbf{q}) \\ y(\mathbf{q}) \\ z(\mathbf{q}) \end{bmatrix} = \sum_{i=1}^{i=n} \left[ \prod_{j=1}^{j=i-1} {}^{j-1}\mathbf{R}_j \right]^{i-1} \mathbf{p}_i = \Phi(\mathbf{q}) \quad (1)$$

where both  ${}^i\mathbf{p}_j$  and  ${}^i\mathbf{R}_j$  are defined using the Denavit–Hartenberg (D–H) representation method (Denavit and Hartenberg 1955) such that

$${}^{i-1}\mathbf{R}_i = \begin{bmatrix} \cos q_i & -\cos \alpha_i \sin q_i & \sin \alpha_i \sin q_i \\ \sin q_i & \cos \alpha_i \cos q_i & -\sin \alpha_i \cos q_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix}$$

and  ${}^{(i-1)}\mathbf{p}_i = [a_i \cos q_i \quad a_i \sin q_i \quad d_i]^T \quad (2)$

where  $q_i$  is the joint angle from  $\mathbf{x}_{i-1}$  axis to the  $\mathbf{x}_i$  axis,  $d_i$  is the shortest distance between the  $\mathbf{x}_{i-1}$  and  $\mathbf{x}_i$  axes,  $a_i$  is the offset distance between the  $\mathbf{z}_i$  and  $\mathbf{z}_{i-1}$  axes, and  $\alpha_i$  is the offset angle from the  $\mathbf{z}_{i-1}$  and  $\mathbf{z}_i$  axes.

## FORMULATION

Because many parameters enter into the layout problem and also because it has no single best solution, the problem lends itself to optimization. We shall develop a formulation suitable for implementation into numerical optimization algorithms. To this end, an objective function, also called a cost function, must be optimized (maximized or minimized) subject to imposed constraints.

### Problem definition

Given the dimensions and joint ranges of motion of a human situated in a pre-specified position (e.g., seated in the workplace or in the cockpit of an aircraft), a requirement is to determine (or calculate) the coordinates of a number of target points that will be designed into the environment (Figure 3). These target points could be any combination of levers, buttons, control knobs, switches, etc., while optimizing an objective function.

### Cost functions and constraints

The cost functions could be discomfort, potential energy, effort, torque, dexterity, or any combination of these functions. The cost functions will first briefly be noted here. Then a multi-objective optimization problem is formed.

#### Discomfort

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

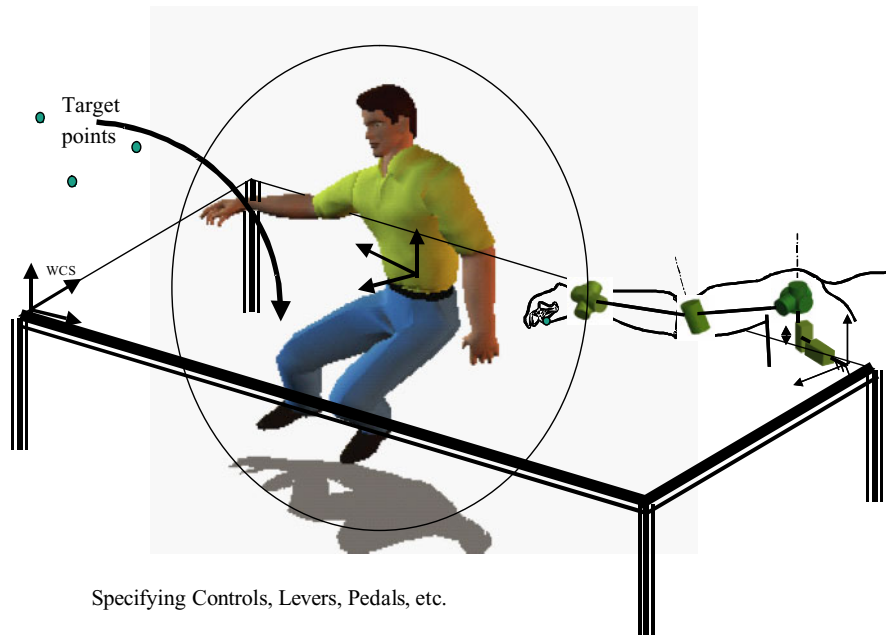


Figure 3 A layout problem.

The total discomfort of all joints is then characterized by the function

$$f_{\text{discomf}}(\mathbf{q}) = \sum_{i=1}^n w_i (q_i - q_i^N)^2 \quad (3)$$

where  $w_i$  is a weight function assigned to each joint for the purpose of giving more 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, i.e., prior to its moving to another position. For an initial set of joint variables  $q_i^{\text{initial}}$  and for a final set of joint variables  $q_i$ , a simple measure of the effort is expressed by

$$f_{\text{effort}}(\mathbf{q}) = \sum_{i=1}^n \gamma_i (q_i - q_i^{\text{initial}})^2 \quad (4)$$

where  $\gamma_i$  is a weight function assigned to each joint. Note that  $f_{\text{effort}}$  depends on the initial configuration of each joint.

**Torque**

Stress induced at a joint is a function of torque imposed at that joint due to biomechanical interaction. To overcome a load, a person will generate the torque at a given joint by exerting muscle forces, but the torque is also a function of the position and orientation of the joint during loading. In order to account for all 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

$$\dot{\mathbf{x}} = \mathbf{J}_x \dot{\mathbf{q}} \quad (5)$$

where the position Jacobian  $\mathbf{J}_x(\mathbf{q}) = [\partial \mathbf{x} / \partial \mathbf{q}]$  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 Jacobian (Abdel-Malek et al 2001). Similarly, the angular velocity can be obtained as

$$\boldsymbol{\omega} = \mathbf{J}_\omega \dot{\mathbf{q}} \quad (6)$$

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

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

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

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_x \\ \mathbf{J}_\omega \end{bmatrix}. \quad (8)$$

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 between joint torques and generalized forces acting on the hand. Since the upper extremity is a kinematic system with time-invariant, holonomic constraints, its configuration depends only on the joint variables  $\mathbf{q}$  (not explicitly on time). Consider the virtual work performed by the two force systems (joint torques and generalized hand

forces). As for the joint torques, their associated virtual work is

$$dW_{\tau} = \boldsymbol{\tau}^T d\mathbf{q}. \quad (9)$$

For the hand forces  $\mathbf{F} = [\mathbf{f}^T \ \mathbf{m}^T]^T$ , comprised a force vector  $\mathbf{f}$  and moment vector  $\mathbf{m}$ , the virtual work performed is

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

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

$$dW_{\mathbf{F}} = \mathbf{f}^T \mathbf{J}_x d\mathbf{q} + \mathbf{m}^T \mathbf{J}_{\omega} d\mathbf{q} \quad dW_{\mathbf{F}} = \mathbf{F}^T \mathbf{J} d\mathbf{q}. \quad (11)$$

Since virtual and elementary displacements coincide, virtual work associated with the two systems is

$$\delta W_{\tau} = \boldsymbol{\tau}^T \delta \mathbf{q} \quad (12a)$$

$$\delta W_{\mathbf{F}} = \mathbf{F}^T \mathbf{J} \delta \mathbf{q} \quad (12b)$$

where  $\delta$  denotes a virtual quantity. The system is under static equilibrium if and only if

$$dW_{\mathbf{F}} = dW_{\tau} \quad \forall \delta \mathbf{q} \quad (13)$$

which means that the difference between virtual work of the joint torques and virtual work of the hand forces shall be null for all joint displacements. Substituting equations (11) and (12) into equation (13) yields

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

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

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} \quad (15)$$

where the torque vector is  $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_n]^T$ .

Torque cost function comprises the weighted summation of all joint torques

$$f_{\text{torque}} = \sum_{i=1}^n \lambda_i |\tau_i| \quad (16)$$

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

### Potential energy

The third objective function represents the potential energy of an arm. Often with upper body motion, the change in height for the torso's center of gravity, and thus the change in potential energy, is relatively small. In addition, minimizing the torso's potential energy is not as crucial in dictating natural postures as minimizing the arm's potential energy. Consequently, in initial studies, only the potential energy of the arm has been considered.

Each link (e.g., the forearm) has a specified center of mass. The vector from the origin of a link's local coordinate system to the center of mass is given by  $\mathbf{r}_i$ , where the

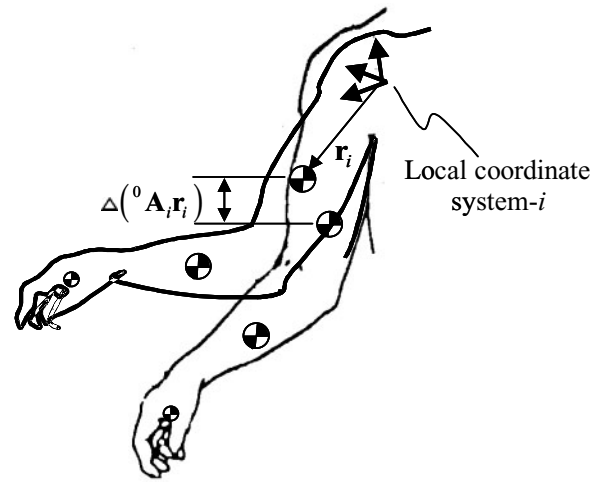


Figure 4 Illustration of the potential energy of the forearm.

subscript indicates the relevant local coordinate system, as illustrated in Figure 4.

In order to determine the position and orientation of any part of the arm, we use the transformation matrices  $({}^{i-1}A_i)$ , which are  $4 \times 4$  matrices that relate one part to another part. Consequently,  $\mathbf{r}_i$  is actually an augmented  $4 \times 1$  vector, rather than a  $3 \times 1$  vector typically used with Cartesian space.  $\mathbf{g}$  is the augmented gravity vector. Then, for the first body part in the chain, the potential energy is  $P_1 = m_1 \mathbf{g}^T {}^0 A_1 \mathbf{r}_1$ . The potential energy for the second body part is  $P_2 = m_2 \mathbf{g}^T {}^0 A_1 {}^1 A_2 \mathbf{r}_2$ . The total potential energy is the total potential energy for all of the links of both body parts under consideration. Thus, for the complete chain, the total potential energy is given by

$$P_{\text{total}} = \sum_{i=1}^k (m_i \mathbf{g}^T ({}^0 A_i \mathbf{r}_i)) \quad (17)$$

where  $k$  is the number of lumped masses,  $\mathbf{g} = [0 \ 0 \ -g \ 0]^T$  is the gravity vector and  $P_i$  is the potential energy for link- $i$ . Because the actual value of potential energy is negative, the objective function, which is minimized, is defined as follows:

$$f_{\text{potential}}(\mathbf{q}) = -P_{\text{total}} \quad (18)$$

In a layout problem, it is usual for the positions of some target points to be constrained to some subregion, e.g., a line, a surface, or some specified region. We will, therefore, have target-region constraints to constrain each target point to some specified space. Some target points are also required to be near each other, and are formulated as distance constraints. In the process of optimizing the positions of several target points, we also need to constrain them not to overlap. Moreover, each target point needs to be reachable with each joint within its joint limits. Typically, we have the following types of constraints:

1. Target-region constraints

$$\mathbf{p}_i \in \mathbf{R}_i \quad (19)$$

where  $\mathbf{R}_i$  is a subregion within the Cartesian space,  $\mathbf{p}_i$ ,  $i = 1, 2, 3, \dots, m$ , are  $m$  target points.

2. Distance constraints

$$\|\mathbf{p}_i - \mathbf{p}_j\| \leq d_k, \quad i, j, k = 1, 2, 3, \dots, m, \quad i \neq j \quad (20)$$

where  $d_k$  is some positive number,  $\mathbf{p}_j$  are  $m$  target points.

3. Overlapping constraints

$$\|\mathbf{p}_i - \mathbf{p}_j\| \geq \mu_k, \quad i, j, k = 1, 2, 3, \dots, m, \quad i \neq j \quad (21)$$

where  $\mu_k$  is some positive number.

4. Reachability constraints

$$\|\mathbf{p}_i - \mathbf{x}(\mathbf{q})\| \leq \varepsilon_i \quad (22)$$

where  $\varepsilon_i$  is some small positive number.

5. Joint ranges of motion

$$q_i^L \leq q_i \leq q_i^U, \quad i = 1, \dots, n \quad (23)$$

where the superscripts L and U denote the lower and upper limits, respectively, where  $n$  = number of DOF.

The above constraints are enforced on different design variables during the optimization. For example, constraints shown in equations (19)–(21) are constraints on target-point positions, while equations (22) can be constraints on the target-point positions and constraints on joint variables. Equation (23) are constraints only on joint variables. The multi-objective optimization problem can be defined as

Find  $\mathbf{q} \in R^{\text{DOF}}$ .

To minimize:  $f(\mathbf{q}) = w_1 f_{\text{discomfort}}(q) + w_2 f_{\text{effort}}(q) + w_3 f_{\text{potential}}(q) + w_4 f_{\text{torque}}(q).$  (24)

Subject to:  $\mathbf{p}_i \in \mathbf{R}_i$

$$\mu_k \leq \|\mathbf{p}_i - \mathbf{p}_j\| \leq d_k, \quad i, j, k = 1, 2, 3, \dots, m, \quad i \neq j$$

$$\|\mathbf{p}_i - \mathbf{x}(\mathbf{q})\| \leq \varepsilon_i$$

$$q_i^L \leq q_i \leq q_i^U, \quad i = 1, 2, \dots, n.$$

**Optimization algorithm**

We use a combination of a gradient-based optimization method, Broydon–Fletcher–Goldfarb–Shanno (BFGS) method, global optimization method, and simulated annealing, to obtain an optimal layout.

Simulated annealing (SA) is a popular global optimization method. Similar to the genetic algorithm that we used in Mi et al (2002), it is an iterative random search procedure with adaptive moves along coordinate directions (Corana et al 1987). It permits uphill moves under the control of a probabilistic criterion, thus tending to avoid the local minima encountered. The SA optimization algorithm can be considered analogous to the physical process by which a

material changes state while minimizing its energy. A slow, careful cooling brings the material to a highly ordered, crystalline state of lowest energy, whereas a rapid cooling yields defects and glass-like intrusions inside the material. As shown in Figure 5 (main loop is simulated annealing), this algorithm proceeds iteratively: starting from a given point  $\mathbf{q}_0$ , it generates a succession of points:  $\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_i, \dots$  tending to the global minimum of the cost function. Applying random moves along each coordinate direction. In turn, new candidate points are generated around the current point  $\mathbf{q}_i$ . The new coordinate values are uniformly distributed in intervals centered on the corresponding coordinates of  $\mathbf{q}_i$ . Half the size of these intervals along each coordinate is recorded in the step vector  $\mathbf{v}$ . If the point falls outside the defined domain of the cost function  $f(\mathbf{q})$ , a new point is randomly generated until a point belonging to the defined domain is found. A candidate point  $\mathbf{q}'$  is accepted or rejected according to the Metropolis criterion:

If  $\Delta f \leq 0$ , then accept the new point:  $\mathbf{q}_{i+1} = \mathbf{q}'$   
 else accept the new point with probability:

$$p(\Delta f) = \exp(-\Delta f / T)$$

where  $\Delta f = f(\mathbf{q}') - f(\mathbf{q}_i)$  and  $T$  is a parameter called temperature.

The SA algorithm starts at some “high” temperature  $T_0$  given by the user. A sequence of points is then generated until an “equilibrium” is approached; that is, a sequence of points  $\mathbf{q}_i$  whose average value of  $f$  reaches a stable value as  $i$  increases. During this phase, the step vector  $\mathbf{v}_m$  is periodically adjusted (every  $N_s$  cycles) to better follow the function behavior. The best point reached is recorded as  $\mathbf{q}_{\text{opt}}$ . After thermal equilibration, the temperature  $T$  is reduced (every  $N_T$  steps) and a new sequence move is made starting from  $\mathbf{q}_{\text{opt}}$ , until thermal equilibrium is reached again, and so on. The process is stopped at a temperature low enough that no more useful improvement can be expected, according to a stopping criterion.

The combination of SA and BFGS is as follows: in SA (main loop), design variables are the positions of target points; for each iteration in SA, a set of positions of target points are generated within the target region (ranges of design variables). If some of them do not satisfy the overlapping constraints, a large penalty value is given to the cost function. If they do not overlap, the target points are given as the input of the posture prediction algorithm (BFGS optimization method for fast prediction as developed in Mi (2003), while combining discomfort and distance as a single cost function, joint limits as constraints with the reach envelope divided into 16 sections and an initial point within each section pre-calculated using a GA). If the target is reachable, a set of joint variables is found and corresponding discomfort, potential energy can be calculated; a weighted summation of each measurement along with distance to the target point is evaluated as the cost function in SA. If the posture prediction algorithm finds the target is not reachable, another large penalty value is given to the cost function. SA is thus used to find an optimized layout

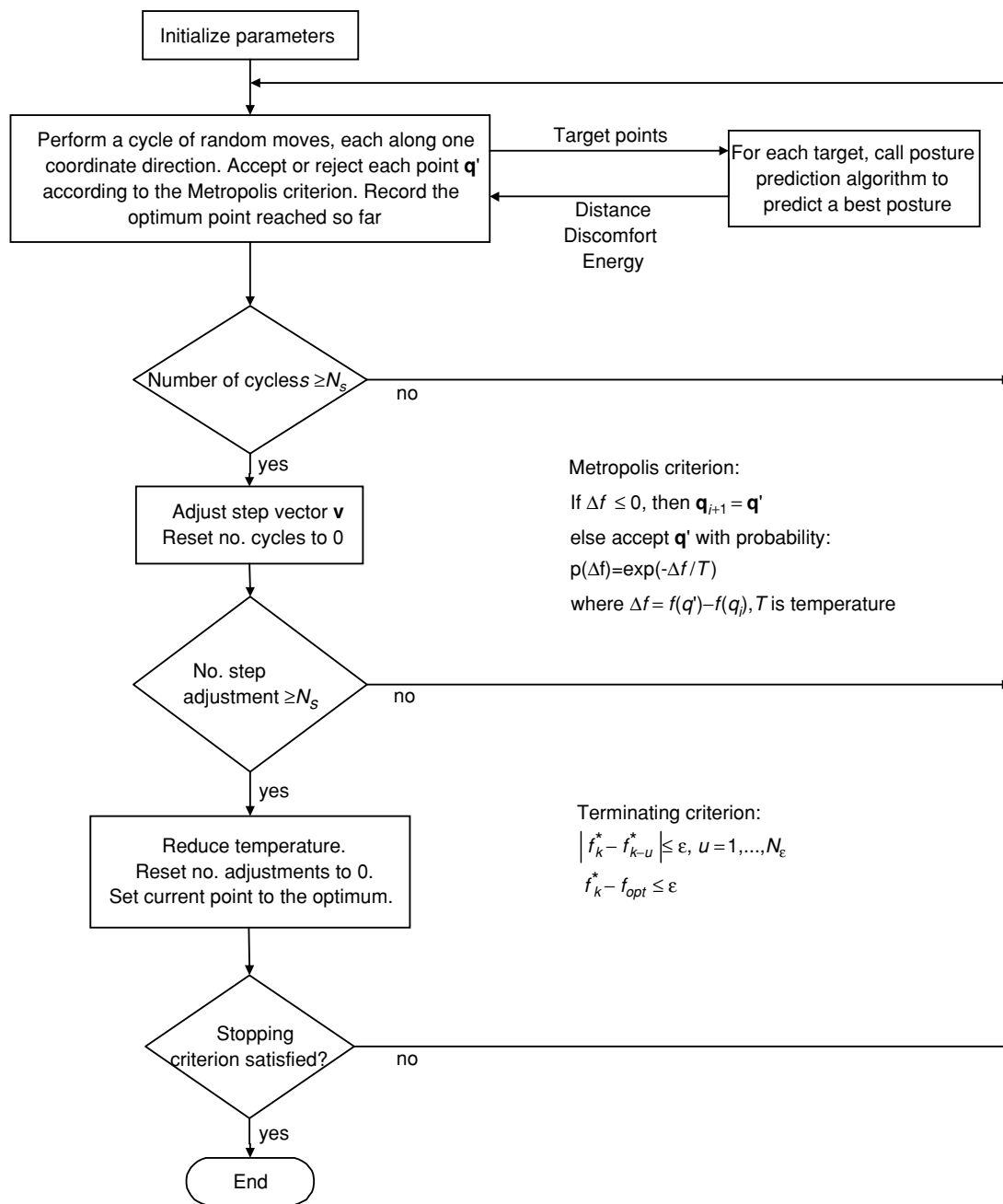


Figure 5 Optimization scheme.

of target points globally, whereas BFGS (posture prediction algorithm) is used to find a natural posture associated with reaching specific target. Although BFGS is itself a local optimization method, since we already picked an initial posture globally, it can provide a fast and near global solution.

#### Comparison of GA and SA algorithms

Since genetic algorithms (GA) and simulated annealing (SA) have both been used to solve optimization problems (Mi et al 2002), this section will briefly compare these two popular global optimization methods.

Theoretically, SAs and GA are quite close relatives, and much of their difference is superficial. The two approaches are usually formulated in ways that look very different and using very different terminology. With SA, one usually talks about solutions and their costs, neighbors, and moves; while with a GA, one talks about individuals (or chromosomes) and their fitness, selection, crossover, and mutation. In a GA, a chromosome is thought of as a “genotype” that only indirectly expresses a solution that has traditionally been specific to a GA, although there is no reason why a similar approach could not be applied in SA as well.

Basically, SA can be thought of as a GA, but with a population size of only one. The current solution is the only one individual in the population; since there is only one individual, there is no crossover, only mutation. This characteristic without crossover is, in fact, the key difference between SA and a GA. Whereas SA creates a new solution by modifying only one solution with a local move, a GA creates solutions by combining two different solutions. Whether this different method to obtain the new solution actually makes the algorithm better or worse, depends on the problem and the representation.

It should be noted that SAs and GAs both share the fundamental assumption that good solutions are most likely found “near” already known good solutions than by randomly selecting them from the whole solution space. If this were not the case with a particular problem or representation, then GAs and SAs would perform no better than random sampling. What a GA does differently here is treat combinations of two existing solutions as being “near”, making the assumption that such combinations (children) meaningfully share the properties of their parents, so that a child of two good solutions is probably better than a child of a random solution. Again, if, for a particular problem or representation this is not the case, then a GA will not provide an advantage over SA.

It should also be noted that the relative weight given to mutation and recombination is a crucial parameter affecting what a GA actually does. If mutation is the dominant way of creating new solutions, then a GA in fact acts as a paralleled version of SA, where several solutions are being independently improved. From a practical viewpoint, it should be noted that for some problems, evaluating solutions that are near an existing solution may be very efficient, which may give a big performance advantage to SA, when compared to a GA, if evaluating recombined solutions is not so efficient. A more practical, empirical comparison should be performed when solving the same problem given the same execution time, which is not conducted here.

### ILLUSTRATIVE EXAMPLE

We now consider the layout of target points in the reachable workspace of a human. These target points could be the placement of control knobs in an aircraft, levers in a tank, or knobs in a vehicle. In this example, we have a manufacturing cell involving an operator who handles three objects: tuff bin, button, and pliers. As shown in Figure 6, the operator will be sitting in the chair and handle the products passing by on a conveyer belt on the table. During the operation, the operator needs to press the button, to pick up the parts from a tuff bin, and to get pliers to assembly parts on the unfinished products as they pass by. The tuff bin is supposed to be reached by the left hand, while the button and pliers are to be reached by the right hand. We shall calculate the spatial positions of the tuff bin, button, and pliers (shown in the table of Figure 7) in

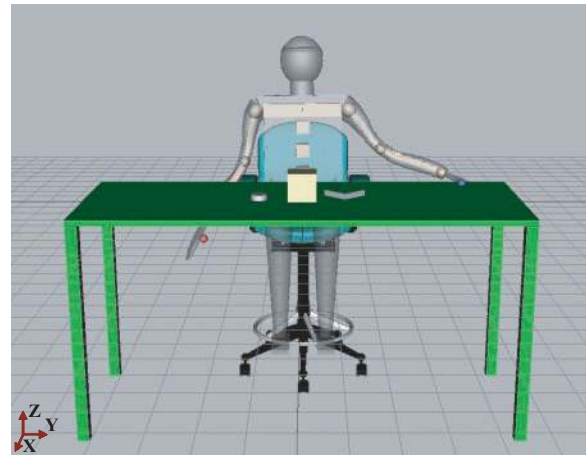


Figure 6 A manufacturing cell.



Figure 7 Designed layout.

the reachable workspace of a human, while minimizing a cost function comprised discomfort, potential energy, and reachability.

The table is 152 cm long, 62 cm wide, and the height is 107 cm from the floor and 11 cm from the waist of the operator. The operator (waist) is 15 cm away (horizontally) from the table’s edge. The origin of the coordinate system is located at the center of the table. There are several target region constraints: since there is a conveyer belt on the table, the tuff bin has to be in the space above the table. And also because the tuff bin is to be reached by the left hand, it is constrained to the left half of the space. The button and pliers are required to be on the top of the table in front of the operator and reachable by the right hand, so they are constrained to the right half of the table top. The distance between button and pliers must be no less than 10 cm (an overlapping constraint). We have a total of five design variables for SA, which are the  $x$ ,  $y$ ,  $z$  coordinates of the bin,  $y$  coordinate of the button, and pliers with the



following constraints:

$$\begin{aligned}
 0 &\leq \text{Bin}_x \leq 31 \\
 0 &\leq \text{Bin}_y \leq 76 \\
 28 &\leq \text{Bin}_z \leq 48 \\
 -76 &\leq \text{Button}_y \leq 0 \\
 -76 &\leq \text{Pliers}_y \leq 0 \\
 \|\mathbf{P}_{\text{Button}} - \mathbf{P}_{\text{Pliers}}\| &\geq 10
 \end{aligned} \tag{25}$$

where  $\mathbf{P}_{\text{Bin}} = (\text{Bin}_x, \text{Bin}_y, \text{Bin}_z)^T$ ,  $\mathbf{P}_{\text{Button}} = (-31, \text{Button}_y, 0)^T$ , and  $\mathbf{P}_{\text{Pliers}} = (-31, \text{Pliers}_y, 0)^T$  are the positions of bin, button, and pliers, respectively. In each iteration, when the new design variables attempted satisfy the constraints of equation (25), the positions of these three objects are sent to the posture prediction algorithm as target points, where design variables are the 15 joint variables, either with right arm or left arm. Postures with minimized distance to the target and minimized discomfort are predicted, and the values of associated distance to the target, discomfort, and potential energy are returned to SA. Now SA is able to evaluate its cost function, a combination of distance, discomfort, and potential energy, etc. (values returned by the posture prediction algorithm), where the reachability constraints have been enforced by converting them to distance measurements and including them in a cost function with the following format:

$$\begin{aligned}
 f &= \sum_{i=1}^3 f_{\text{discomf}}^i + \sum_{i=1}^3 f_{\text{potential}}^i + \sum_{i=1}^3 f_{\text{effort}}^i \\
 &+ \sum_{i=1}^3 f_{\text{torque}}^i + 1000 \times \sum_{i=1}^3 \text{distance}_i
 \end{aligned} \tag{26}$$

where  $f_{\text{discomf}}^i$ ,  $f_{\text{potential}}^i$ ,  $f_{\text{effort}}^i$ ,  $f_{\text{torque}}^i$  and  $\text{distance}_i$  are the discomfort, potential energy, effort, torque, and distance to the specific target of each posture reaching bin, button, and pliers respectively.

Results of this layout design are shown in Figure 8 where  $w_1 = w_3 = w_2 = w_4 = 1$ , and

$$\mathbf{P}_{\text{Bin}} = [1.3940, 40.1072, 28.8529]^T$$

$$f_{\text{discomf}}^{\text{Bin}} = 4.3448$$

$$f_{\text{potential}}^{\text{Bin}} = 10.7160$$

$$f_{\text{effort}}^{\text{Bin}} = 35.2473$$

$$f_{\text{torque}}^{\text{Bin}} = 54.5460$$

$$\text{distance}_{\text{Bin}} = 0.0000$$

$$\mathbf{P}_{\text{Button}} = [-31.0, -58.4235, 0.0]^T$$

$$f_{\text{discomf}}^{\text{Button}} = 1.1521$$

$$f_{\text{potential}}^{\text{Button}} = 8.8921$$

$$f_{\text{effort}}^{\text{Button}} = 2.3401$$

$$f_{\text{torque}}^{\text{Button}} = 36.2674$$

$$\text{distance}_{\text{Button}} = 0.0006$$

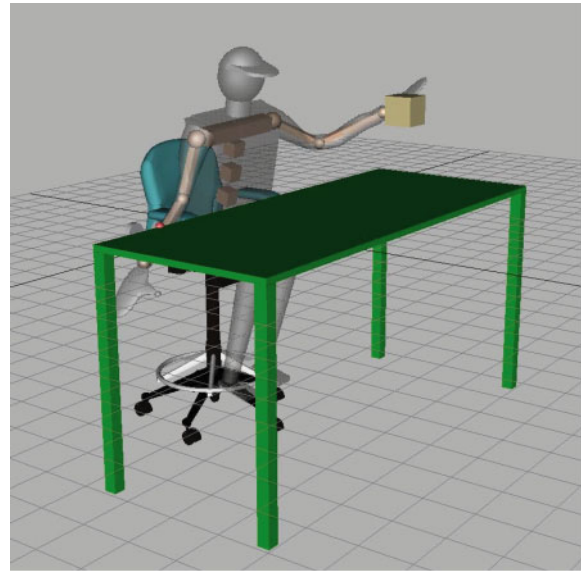


Figure 8 Posture reaching tuff bin.

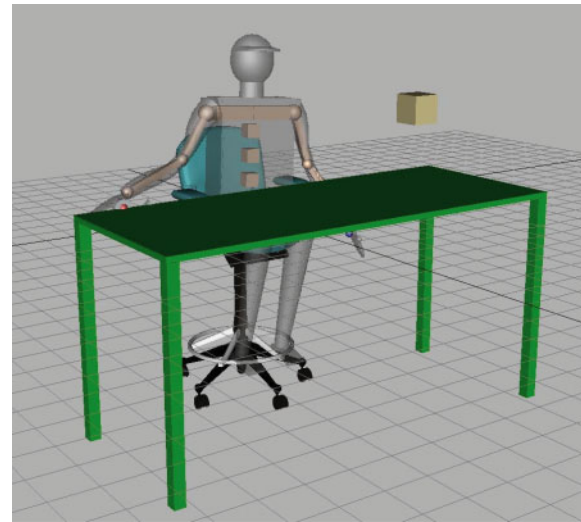


Figure 9 Posture pressing button.

$$\mathbf{P}_{\text{Pliers}} = [-31.0, -48.4231, 0.0]^T$$

$$f_{\text{discomf}}^{\text{Pliers}} = 2.2602$$

$$f_{\text{potential}}^{\text{Pliers}} = 9.4202$$

$$f_{\text{effort}}^{\text{Pliers}} = 0.3425$$

$$f_{\text{torque}}^{\text{Pliers}} = 20.1894$$

$$\text{distance}_{\text{Pliers}} = 0.0003.$$

The postures reaching the tuff bin and button are shown in Figure 8 and Figure 9, respectively. It is observed that each object can be touched easily and comfortably, and a numerical optimization algorithm has created a successful layout design.


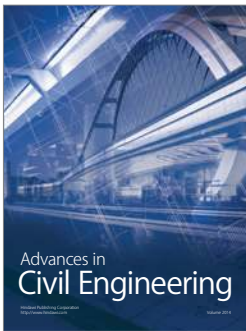
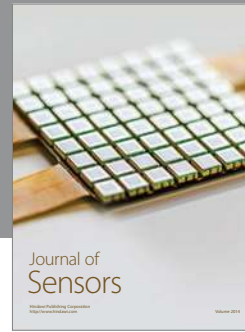
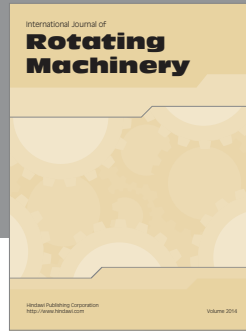
## CONCLUSIONS

A rigorous formulation for optimization-based layout has been presented. The layout problem occurs in the field of ergonomic design and encompasses many solution sets, one of which is to be selected. We have presented a method for calculating a best solution based on minimizing or maximizing a cost function or a combination of cost functions.

This formulation was shown to be implementable in computer code. Indeed, we believe this type of formulation augments capabilities digital human modeling software offers, where it facilitates digital prototyping, shortens lead times, and saves costs.

## REFERENCES

- Abdel-Malek K, Yang J, Brand R, Tanbour E. 2001. Towards understanding the workspace of the upper extremities. *SAE Trans J Passenger Cars: Mech Sys*, 110 (Section 6): 2198–206.
- Corana A, Marchesi M, Martini C, Idella S. 1987. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Trans Math Software*, 13(3):262–80.
- Das B, Sengupta A K. 1996. Industrial workstation design: a systematic ergonomics approach. *Appl Ergon*, 27(3):157–63.
- Denavit J, Hartenberg R S. 1955. A kinematic notation for lower-pair mechanisms based on matrices. *J Appl Mech*, 77:215–21.
- Fisher D L. 1993. Optimal performance engineering, good, better, best. *Hum Factors*, 35(1):115–39.
- Lim J, Hoffmann E. 1997. Appreciation of the zone of convenient reach by naive operators performing an assembly task. *Int J Ind Ergon*, 19(3):187–99.
- Mi Z, Yang J, Abdel-Malek K, Mun J H, Nebel K. 2002. Real-time inverse kinematics for humans In Proceedings of ASME Design Engineering Technical Conferences, Montreal, Canada.
- Mi Z. 2003. Task-based prediction of upper body motion, PhD Dissertation, The University of Iowa, 2003.
- Pham D T, Onder H H. 1992. Knowledge-based system for optimizing workplace layouts using a genetic algorithm. *Ergonomics*, 35(12):1479–87.



Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

