81-16

Techniques for Goal-Directed Motion

James U. Korein and Norman I. Badler
Dept. of Computer and Information Science
University of Pennsylvania

Abstract

When motions of linkages such as the human body must be specified in terms of joint angle changes, considerable effort is required to acheive a particular goal. We review some techniques useful for the automatic generation of joint angle adjustments from a goal specified in terms of a world coordinate system.

Introduction

As computer animation systems achieve greater graphics realism and highly interactive interfaces, it becomes a challenge to animate articulated figures such as the human body. Allowable motions of linkages such as an arm or leg are governed by well known rotation transformations. However, the design of a particular movement such as taking a step or reaching for an object may require considerable trial and error, if specified at the level of individual joint adjustments. For purposes of convenient animation, it is clear that, at a minimum, one must supply a movement primitive to achieve a goal position with a linked structure [4].

Parallel research in robotics can shed light on computational solutions to the problem of achieving a particular position and orientation in space. The problem of goal-directed motion will be cast in terms of robotics and linkage kinematics: the study of position (displacement) and its time derivatives (velocity and acceleration).

Considerations of force and mass (dynamics) [23,25,46], balance [36,37] and obstacle avoidance [34,35] are beyond the scope of this paper.

We proceed by establishing the necessary terminology, describing linked structures such as the human body, discussing constraints and outlining algebraic and numerical methods for goal acheivement.

Rigid Object Position and Orientation

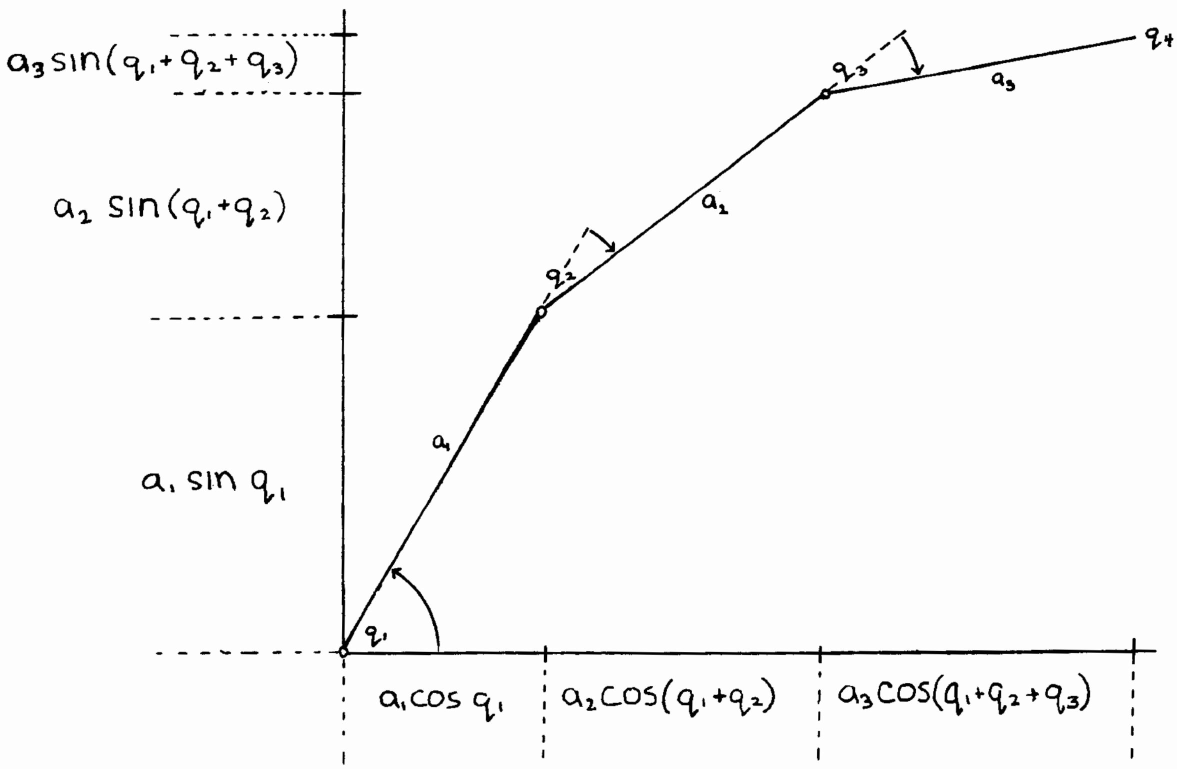
The relative position of a rigid 2-D object with respect to a given Cartesian reference frame is given by a translation (x, y) and a rotation (r). Hence, a rigid object in the plane is said to have three degrees of freedom (d.o.f.). In 3-D, six variables are necessary and sufficient to specify the position (x,y,z) and orientation (r1,r2,r3) of a rigid object. Therefore, a rigid object in space is said to have six d.o.f.

Interpolation

Consider any degree of freedom in position or orientation of an object (x, for example). It may be expressed as a function of time which will be denoted x(t). The x components of velocity and acceleration are the first and second time derivatives. Given a temporal sequence of values for x, a function x(t) may be obtained by any of a number of interpolation methods. Since physical objects cannot achieve infinite acceleration, realistic simulation requires that the acceleration be everywhere finite, hence that x(t) be everywhere twice differentiable. This requirement prohibits use of linear interpolation schemes common in computer animation systems. Quadratic methods have been discussed by Paul [44] and Herbison-Evans [22]. Finkel [17] discusses the use of cubic spline interpolation for computing robot arm trajectories. Several interpolation schemes are compared by Mujtaba [38].

Joints and Chains

Joints which connect rigid links may be divided into those with a single d.o.f., such as revolute and sliding (prismatic) joints, and those with more, such as spherical (ball and socket) joints. A spherical joint has three degrees of freedom. Suppose that the position and orientation of one of the links connected by a spherical joint is fixed. Two variables suffice to give the direction in which the axis of the free link is pointing, and a third to give its rotational position about that axis. For



$a_3 \sin(q_1+q_2+q_3)$
$a_2 \sin(q_1+q_2)$
$a_1 \sin q_1$
q_1
a_1
q_2
a_2
q_3
a_3
q_4
$a_1 \cos q_1$
$a_2 \cos(q_1+q_2)$
$a_3 \cos(q_1+q_2+q_3)$

Figure 1

purposes of analysis, multiple d.o.f. joints may be decomposed into "kinematically equivilent" sequences of one d.o.f. joints. For example, a spherical joint may be decomposed into a sequence of three revolute joints separated by zero length links, all of whose axes intersect at a point. Further discussion of joints and kinematic equivalence may be found in texts by Suh and Radcliffe [56], Dijksman [14], Hunt [26] and Paul [41].

A linear sequence of links connected pairwise by joints is called a (kinematic) chain. We will consider chains whose joints have been decomposed into one d.o.f. joints. A chain has a free (distal) end and a fixed (proximal) end. It is useful to think of the proximal end as being attached by a joint to a reference link, possibly imbedded in the world coordinate system. A concise notation for the description of such chains was developed by Denavit and Hartenberg [12], and is widely used in robotics applications, [45,46,58]. In the Denavit-Hartenberg notation, a chain with n joints and n links, and its current configuration is described by n four-element vectors. Each vector contains a joint variable value and three "link geometry" values relating the position and orientation of two consecutive joints. The system applies to both revolute and prismatic joints. This information is sufficient to define a transformation between a coordinate system embedded in one link and one embedded in a consecutive link. Specification of joint limits requires two additional values per joint.

We specify the the configuration of a chain with n one d.o.f. joints as a vector (q[1], ... , q[n]), or simply q^. Given a temporal sequence of configurations for a chain we can, by interpolating each variable of the configuration vector, obtain the configuration of the system as a vector function of time, $\hat{q}(t)$.

Application to Human Body Animation

The human body may be described as a tree structure, where "approximately rigid" segments of the body are taken as nodes, connected by joints represented as arcs, [2,3,4,5,6]. The number of arcs impinging on each node n is typically two, as in the case of the upper arm segment, or three, as in the case of the pelvis. If the position of some segment is constrained, and a second segment is to be moved with respect to the first, then the joints and links relevant to the motion form a simple path through the tree. A path through the tree may be abstracted as a chain.

Some approximations are inherent in this representation. Complex joints such as wrist and shoulder are first approximated as spherical joints, which are then decomposed into a kinematically equivalent sequence of one d.o.f. joints. (In fact, the actual motion of the shoulder involves a center of rotation which depends on the position of the upper arm). It is further assumed that each joint may move independently of any other, and that the position of one joint does not effect the range of motion of any other. A refinement to this model is the direct representation of spherical joints. This avoids inaccuracies inherent in modelling a spherical joint by a a sequence of lower pairs with independent joint limits, at the cost of introducing a joint limit function. A simple approach is to restrict the position of the distal link of a spherical joint to lie within some angle of a specified center position, thereby allowing it to move within a cone. The rotation of the distal joint about its axis is modelled independently [2].

The number of degrees of freedom to be controlled when moving segments relative to one another is just the number of joints after each spherical joint has been decomposed. Once all joint variables are specified, the body is effectively made rigid. Thus, there are six additional degrees of freedom required to specify the position and orientation of the body with respect to the world coordinate system. Note that each body segment is defined in terms of its own coordinate system. One of these, called the root, must be chosen to specify the relation of the rigid body to the world coordinate system. Once the root is specified, a position vector for a human body consists of joint angles for each joint, and six variables relating the root and world coordinate systems. If each of these variables is given as a function of time, as might be generated by interpolating between key position vectors, the motion of the body is entirely specified [65].

Goals

For many purposes, we wish to specify the actions to be performed by a body or linkage, without the burden of having to specify motion of all degrees of freedom explicitly [4]. The first step towards this end is to provide a facility by which segments of the body may be positioned with respect to the world coordinate system, and other objects within it. Most of the work done on this type of problem has been in the context of industrial robotics [18,46].

We define a goal as a set of constraints on the position or orientation of a body segment. For example, we might require a certain point on the tip of the right forefinger to be at a certain position $p^{\wedge}$ =(kx,ky,kz) in the world coordinate system. Alternatively, we might require that the coordinate system imbedded in the left forearm be positioned at a particular position and orientation in the world coordinate system. The goal $p^{\wedge}$ for the forearm would be specified as a constant position-orientation vector (kx,ky,kz,kr1,kr2,kr3). To require that a foot segment be flat on the support plane, we would specify required values for height (ky) and two rotations, (say kr1 and kr2), leaving the others unspecified.

These sets of constraints may be written as equations. For the first example, we have:

$$
\begin{aligned}
x &= kx \\
y &= ky \qquad (1) \\
z &= kz
\end{aligned}
$$

Clearly, restrictions of the type discussed above may produce up to six such equations.

More complex equations may be used to describe more complex constraints. For example, we might restrict the fingertip to lie on a circle in the x-y plane by the following constraints:

(2) x**2 + y**2 = kr**2

z = kz

where kr and kz are constants. In the examples above, we have considered only equality constraints. We might also wish to constrain one or more degrees of freedom to lie is a specified range of values. For example, we might constrain the fingertip to lie within a rectangular box, by specifying maximum and minimum value for x, y and z. Six inequality constraints result:

kxmin <= x <= kxmax

(3) kymin <= y <= kymax

kzmin <= z <= kzmax

Additional constraints may arise from conditions imposed on the body segments. For example, the feet may be pinned to the floor, the waist restrained by a seatbelt or the upper torso by a harness [53]. Constraints on the position and orientation of the body may be treated just as goals are. The only distinction is that body constraints usually refer to conditions which held in a previous configuration, as opposed to goals, which must be acheived. The term "constrained segment" will be used to encompass both segments for which a goal is specified and those which are to maintain their previous position with respect to the world coordinate system.

Positional Constraints and Joint Variables

Consider two constrained segments belonging to a common chain. In considering their relative positions we may take the coordinate system embedded in one (proximal) segment as the world coordinate system. The constraints constitute a (possibly partial) specification of the position and orientation of the distal link in the coordinate system of the proximal link.

Now, the coordinate systems embedded in each consecutive pair of links in the chain are related by a rigid transformation, including translation and rotation. Since the joint between the links has only one d.o.f., the transformation depends on a single joint variable. The composition of these transformations from link to link gives a vector expression for the position and orientation of the distal link in the proximal link coordinate system. This expression, in conjunction with the specified goals and body constraints, describes implicitly the set of chain configurations that will satisfy those constraints.

Let us consider a simple example [15,25]. Consider the three-link planar chain shown in figure 1. The proximal end is constrained to the origin. The link lengths are a[1], a[2] and a[3]. The joint variables (all angular) are q[1], q[2] and q[3]. For convenience, we will give each joint the same name as its joint variable, and call the chain's distal terminal q[4]. Similarly, the ith link from the proximal end is labelled a[i].

The position of any joint α[i] may be obtained by examining the projections of more proximal links on the x and y axes, marked off in the figure. Letting x[i] and y[i] be the x and y components of the position of joint α[i], we see from the figure that:

```
(4)   x[2]  =  a[1]*cos(α[1])

(5)   y[2]  =  a[1]*sin(α[1])

(6)   x[3]  =  a[1]*cos(α[1]) + a[2]*cos(α[1]+α[2])

(7)   y[3]  =  a[1]*sin(α[1]) + a[2]*sin(α[1]+α[2])

(8)   x[4]  =  a[1]*cos(α[1]) + a[2]*cos(α[1]+α[2])
               + a[3]*cos(α[1]+α[2]+α[3])

(9)   y[4]  =  a[1]*sin(α[1]) + a[2]*sin(α[1]+α[2])
               + a[3]*sin(α[1]+α[2]+α[3])
```

The orientation, r[i], of link a[i] is just the accumulation of more proximal joint angles:

```
(10)  r[1]  =  α[1]

(11)  r[2]  =  α[1] + α[2]
```

(12) r[3] = q[1] + q[2] + q[3]

Equations for position and orientation for links of a planar chain of arbitrary length may be computed in exactly the same way [15,25].

Let us now consider several different classes of goals for the depicted chain, and their implications for the configuration of the chain. It will be useful to define the workspace of a chain as the set of points which may be reached by its distal end. In particular, the workspace of the chain with distal end q[i] will be denoted W[i].

Let the goal be that joint q[3] (not the tip q[4]) move to a point (kx, ky). This imposes two constraints:

(13) x[3] = kx

(14) y[3] = ky

Combining these constraints with the equations for position of q[3] (eq. 6 and 7), we get:

(15) kx = a[1]*cos(q[1]) + a[2]*cos(q[1]+q[2])

(16) ky = a[1]*sin(q[1]) + a[2]*sin(q[1]+q[2])

We have two equations in two unknowns, namely q[1] and q[2], so the two link subchain ending at q[3] is perfectly constrained. These particular equations may be solved algebraically for q[1] and q[2] [25]. The workspace of this subchain is either a disk-shaped or annular region, depending on relative link lengths [25], and that there are a finite number of solution configurations (one or two) at each point in the workspace. This is characteristic of a perfectly constrained system. The workspaces of systems which are overconstrained have a lower dimensionality than the space in which they lie. For example, the one link subchain ending at q[2] can only reach points lying on the perimeter of a circle. Underconstrained systems are of central importance in animation applications, and will be discussed in a subsequent section.

Returning to the example, we may add to the goal a constraint on the orientation of the last link:

$$(17) \quad r[3] = kr$$

Then combining with eq. 12, we have:

$$(18) \quad kr = q[1] + q[2] + q[3]$$

Since q[1] and q[2] are already constrained, this gives the solution for orientation, so that the entire chain is perfectly constrained.

The concept of a workspace for purely positional goals may be extended to that of a goalspace for goals of other types. For example the goalspace for position-orientation goals (kx, ky, kr) is a set in the 3-space with axes x, y, r.

If joints are limited, the goalspace will be a subset of that of the same chain with unlimited joints. In the perfectly constrained case, if an algebraic solution is available, all solution configurations may be found. If none of them satisfy joint limits, it is immediately clear that the goal is not in the goalspace of the restricted chain.

Three Dimensions

The equations for the position of joints of a 3-D chain may be obtained just as in the planar case, by considering the projections of those points on x, y and z axes [15]. Obtaining orientation equations directly requires that we obtain the contribution to each of the three rotation angles by each link and joint of the chain [15]. Rather than proceeding directly along these lines, the usual approach is to use homogenous transformation matrices.

The relationship between the ith and i+1st links of a chain may be written as a homogeneous transformation matrix A[i], which is a function of the ith set of parameters for the chain. For a given chain, A[i] is a function of joint variable q[i] only. These matrices may be multiplied to obtain new matrices relating any two non-consecutive links. The first and last links are related by the product of all the matrices A[1] through A[n]. But the position and orientation of the distal link in the world coordinate system may also be expressed as a homogeneous transform matrix T defined in terms of the goal. Thus we obtain the matric equation:

$$(19) \quad T = A[1]*A[2]*...*A[n]$$

for a chain with n d.o.f. Since each of the matrices in eq. 19 is 4 by 4, it represents 16 scalar equations. The number of these which are linearly independent correspond to the number of goal constraints embodied in the matrix T. These equations give a (possibly partial) specification of position and orientation of the distal link in terms of the joint angles q[1] through q[n] [45,46,47].

Algebraic Solution

When the system of equations arising from the specification of a goal for a chain is perfectly constrained, it is sometimes the case that an algebraic solution may be found. The situation that has received the most attention in robotics is that where the goal is a complete specification of position and orientation for the distal link of a spatial chain with six degrees of freedom.

It was first shown by Pieper [47] that if the axes of three consecutive revolute joints intersect at a point, a six d.o.f. system may be decomposed and solved for the joint variables q[1],...,q[6]. Such chains are called "kinematically simple" [45].

An algebraic solution to the MIT Vicarm robot is described by Horn [24] in an intuitive, geometric way. Paul solved the Stanford Schienman robot arm [42], and has recently published descriptions of general algebraic techniques for the solution of kinematically simple robot arms [44,46]. Other accounts of algebraic solutions may be found in [32,43,52].

Algebraic solutions have two primary advantages over numerical techniques. First, they can be performed more quickly. This speed is important for real-time robotic applications. Second, all solution configurations are found, providing greater flexibility than an iterative techniques, which converge to a single solution.

Numerical Solutions

Pieper [47] evaluated two methods of numerical solution for six d.o.f. systems: the first method uses Newton-Raphson iteration; the second method is based on the use of velocity screws. Pieper obtained solutions faster with the latter method. The use of these numerical methods is not restricted to kinematically simple chains. Whitney [60,61] proposed a numerical method called "resolved rate control" using velocity, rather than position, since the relationship between the components of velocity of the distal link and that of the joint angles is linear.

The fact that numerical methods converge on only one solution presents a practical problem: it is necessary to insure that the solution reached is not prohibited by joint limits. Which solution, if any, an iterative method will converge to depends on the initial estimate of the solution. The use of the initial position of the chain has been proposed for this purpose [60]. If this choice is not sufficiently close to the final configuration to provide convergence, an intermediate goal partway between initial and final positions may be chosen, and the process repeated recursively. This procedure still does not guarantee convergence to a configuration satisfying joint limit restrictions (discussed subsequently). A more general discussion of numerical methods for the solution of non-linear system of equations may be found in Ortega [40].

Underconstrained Systems

If a system has more degrees of freedom than the number of constraints imposed by the goal parameters, it is underconstrained or redundant. The difference between the degrees of freedom and the goal-imposed constraints is the degree of redundancy.

We return, for example, to the three link planar chain in figure 1. Consider the goal of positioning the distal terminal q[4] at a specified position (kx,ky). From equations 8 and 9 we get:

(20) kx = a[1]*cos(q[1]) + a[2]*cos(q[1]+q[2])
 + a[3]*cos(q[1]+q[2]+q[3])

(21) ky = a[1]*sin(q[1]) + a[2]*sin(q[1]+q[2])
 + a[3]*sin(q[1]+q[2]+q[3]

That is, we have two equations, or equality constraints, in three variables, q[1], q[2], q[3]. The degree of redundancy of the system is one; intuitively, the solution set is the locus of points lying on a space-curve in configuration space.

Lagrangean Methods

One method for dealing with redundant systems is to propose an objective function to be minimized and to apply the method of Lagrange Multipliers [60]. This results in a perfectly constrained sytem which will both satisfy the constraints and minimize the objective [19,63,64].

Suppose we are given an objective function:

(22) f(q^)

where q^ is the vector (q[1],...q[n]), subject to constraints:

 c1(q^) = 0
(23) :
 :

$$cm(q\hat{}) = 0$$

We introduce a vector of variables $u\hat{} = (u[1], \ldots, u[m])$, called Lagrange multipliers, and write the Lagrangean:

$$\text{(24)} \quad L(q\hat{},u\hat{}) = f(q\hat{}) - (u[1]*c1(q\hat{}) + u[2]*c2(q\hat{}) + \ldots + u[m]*cm(q\hat{}))$$

We then generate n new constraints, by setting to zero the partial derivatives of L with respect to q[1] through q[n], respectively:

$$\text{(25)} \quad \begin{array}{l} dL/dq[1] = 0 \\ \vdots \\ \vdots \\ dL/dq[n] = 0 \end{array}$$

This results in a system of m+n equations (eqs. 23 and eqs. 25) in m+n unknowns (q's and u's), which usually must be solved numerically.

It is sometimes possible to avoid solving such a large system of equations by performing algebraic manipulation directly on vector valued functions.

We rewrite the constraints as a single vector valued function:

$$\text{(26)} \quad c\hat{}(q\hat{}) = 0\hat{}$$

The Lagrangean, which is scalar-valued, may also be rewritten as:

(27) L(q^,u^) = f(q^) - u^'*c^(q^)

where u^' is just u^ written as a row vector. We now set to 0^ the derivative with respect to the vector q^, obtaining the vector equation:

(28) dL/dq^ = d/dq^(f(q^)) - d/dq^(u^'*c^(q^))

Equations 26 and 28 above constitute two vector equations in two vector unknowns (q^ and u^). Working with velocities, rather than position, and using a suitable objective function, Whitney [60,62] has demonstrated that u^ may be eliminated from these two equations, resulting in a single vector equation consisting of n scalar equations in the original n unknowns, q[1], ..., q[n]. The manipulation requires the use of the generalized or psuedo-inverse for rectangular matrices, which has been the subject of books by Boullion and Odell [10] and Ben-Israel [8]. Similar techniques have been proposed for robot hand control by Asada [1].

Extensions for Inequality Constraints

The techniques for handling redundant systems which have been discussed do not account for inequality constraints. Those arising from joint limits are of particular concern. Lagrangean methods may be made to cope with inequality constraints in several different ways.

The Lagrange approach discussed earlier will find all minima for the objective, subject to the equality constraints only. Those which do not satisfy the joint limit inequalities may be discarded immediately. Any new minima which arise from the inequalities must lie on the boundary of the region defined by those limits; that is, when one or more of the joint variables take extreme values. Therefore, these minima may be found by solving 2n smaller problems, each involving one fewer variables than the original. The number of levels of recursion to solve a problem in this manner is just the degree of redundancy of the original system.

Another method is to introduce a new "slack" variable for each inequality, transforming it into an equality constraint. If all n joints have upper and lower limits, the resulting system contains 2n additional variables [64].

A somewhat different method is the use of a penalty function, which is incorporated into the objective function. The penalty causes the objective to increase as joints approach their limits. The desired result is that the objective function itself effectively prohibits joint limit violations. Discussions of penalty functions may be found in Gill [19] and Wismer [64].

Another method is to ignore the inequalities until an iterative solution procedure run into one of the boundaries of the solution space. The boundary is then followed in the feasible direction closest to that of the objective function gradient [64].

Objective Functions

The choice of objective function is application dependent. Some possible objectives are minimization of discomfort, time, work, disturbance to the linkage. We must bear in mind that if a Lagrangean method is to be used, the objective function must be differentiable. Moreover, functions of degree larger than quadratic carry a larger computational cost for numerical methods such as Newton-Raphson [64]. Whitney [60,61] chose to minimize instantaneous kinetic energy, expressed as a function of velocity, which proved computationally expedient for the resolved rate control method.

Reach Hierarchy

A rather different approach to the specific problem of reaching a point goal has been proposed by Korein [28]. The procedure relies on precomputed workspaces for the chain and each of its distal subchains. A distal subchain is a chain extending from any joint in the chain to its distal end.

Let the chain be Chi[1] and its workspace W[1]. Let the subchain with just the most proximal joint and link deleted be Chi[2] with workspace W[2], etc. Given these workspaces, the algorithm proceeds as follows:

- If the goal is not in W[1], it is not reachable: give up.

- Otherwise, adjust q[1] only as much as is neccesary to bring the goal into W[2].

- Proceed down the chain, at each step adjusting q[i] only as much as is necessary to bring the goal into q[i+1].

This algorithm will work with exact workspace descriptions, and with with approximations which satisfy the condition that

(29) approx(W[i]) is a subset of sweep(approx(W[i+1]),q[i])

where the sweep is the volume generated by sweeping the approximation to W[i+1] about joint q[i].

This method requires precomputation and storage of workspace descriptions. While it is theoretically extensible to position-orientation goals, the cost of storing high dimensional workspaces is prohibitive. A number of papers on the nature and construction of workspaces have been published in the context of robotics [13, 21, 29, 30, 50, 54, 55].

The implicit "objective function" minimizes adjustment of proximal joints. This may be advantageous for chains imbedded in tree structures, since disturbance to the tree is minimized.

With respect to time efficiency, each adjustment requires finding the intersection between a workspace surface and a line or circle [28]. The final links of the chain, which comprise a perfectly constrained system, may be solved algebraically.

Conclusion

A number of techniques relevant to goal directed motion of kinematic linkages have been discussed. The intent has been to publicize linkage positioning tools which may be useful for the development of computer animation systems which are easier to use.

References

[1] Asada, Haruhiko, Studies on Prehension and Handling by Robot Hands with Elastic Fingers, Ph.D. Thesis, Automation Research Laboratory, Kyoto University, Japan, 1979.

[2] Badler, N. I. and O'Rourke, J., "Representation of Articulable, Quasi-Rigid, Three-Dimensional Objects," _Proceedings of the Workshop on Representation of Three-Dimensional Objects_, Phil., PA, May 1979.

[3] Badler, N. I., O'Rourke, J., Smoliar, S. W. and Weber, L., "The Simulation of Human Movement by Computer," University of Pennsylvania, Dept. of Computer and Information Science, Tech. Rep. MS-CIS-78-36, 1978.

[4] Badler, N. I. and Smoliar, S. W., "Digital Representation of Human Movement," Computing Surveys, Vol. 11, No. 1, March 1979, pp. 19-38.

[5] Badler, N. I., O'Rourke, J. and Toltzis, H., "A Spherical Representation of a Human Body for Visualizing Movement," Proceedings of the IEEE, Vol. 67, No. 10, Oct. 1979, pp. 1397-1403.

[6] Badler, N. I., O'Rourke, J. and Kaufman, B., "Special Problems in Human Movement Simulation," Computer Graphics, Vol. 14, No. 3, 1980, pp. 189-197.

[7] Badler, N. I., O'Rourke, J. and Kaufman, B., "Special Problems in Human Movement Simulation," _Computer Graphics_, Vol. 14, No. 3, 1980, pp. 189-197.

[8] Ben-Israel, Adi and Greville, Thomas, Generalized Inverses: Theory and Applications, John Wiley and Sons, New York, 1974.

[9] Bottema, O. and Roth, B., Theoretical Kinematics, North Holland Press, 1978.

[10] Boullion, T. and Odell, P., Generalized Inverse Matrices, John Wiley and Sons, New York, 1971.

[11] Cooper, L. and Steinberg, D., Introduction to Methods of Optimization, W. B. Saunders Company, Philadelphia, 1970.

[12] Denavit, J. and Hartenberg, R.S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," Journal of Applied Mechanics, Vol. 77, June, 1955, pp. 215-221.

[13] Derby, Stephen, "The Maximum Reach of a Revolute Jointed Manipulator," Mechanism and Machine Theory, Vol. 16, No. 5, 1981, pp. 255-261.

[14] Dijksman, E. A., Motion Geometry of Mechanisms, Cambridge University Press, London; New York, 1976.

[15] Duffy, Joseph, Analysis of Mechanisms and Robot

Manipulators, Edward Arnold Ltd., London, 1980.

[16] Faux, I. D. and Pratt, M. J., Computational Geometry for Design and Manufacture, Ellis Horwood Limited, John Wiley and Sons, New York, 1979.

[17] Finkel, Rapheal A., _Constructing and Debugging Manipulator Programs_, Ph.D Thesis, Stanford University, Aug., 1976.

[18] Finkel, R., Taylor, R., Bolles, R., Paul, R. and Feldman, J., "AL, A Programming System for Automation," Stanford University Artificial Intelligence Laboratory Memo AIM-243, Nov., 1974.

[19] Gill, P. E. and Murray, W., Numerical Methods for Constrained Optimization, Academic Press, 1974.

[20] Gottfried, B. and Weisman, J., Introduction to Optimization Theory, Prentice Hall, Englewood Cliffs, N. J., 1973.

[21] Gupta, K. C. and Roth, B., "Design Considerations for Manipulator Workspace," ASME Paper No. 81-DET-79, Sept., 1981.

[22] Herbison-Evans, Donald, "NUDES 2: A Numeric Utility for Displaying Ellipsoid Solids, Version 2," _Computer Graphics_, Vol. 12, No. 3, Aug. 1978, pp. 354-356.

[23] Hollerbach, John M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," _IEEE Transactions on Systems, Man, and Cybernetics_, Vol. SMC-10, No. 11, Nov. 1980, pp. 730-736.

[24] Horn, B. K. P. and Inoue, H., "Kinematics of the MIT-AI-VICARM Manipulator," M.I.T. Artificial Intelligence Laboratory Working Paper 69, May, 1974.

[25] Horn, Berthold K. P., "Kinematics, Statics, and Dynamics of Two-Dimensional Manipulators," Artificial Intelligence: An MIT Perspective, Vol. 2, ed. P. H. Winston, MIT Press, Cambridge, Mass., 1979.

[26] Hunt, K. H., Kinematic Geometry of Mechanisms, Clarendon Press, Oxford; New York, 1978.

[27] Joint Motion: Method of Measuring and Recording, American Academy of Orthopedic Surgeons, Chicago, Ill., 1965.

[28] Korein, James, "Using Reach Descriptions to Position Kinematic Chains," Proceedings of the Fourth National Conference of the Canadian Society for Computational Studies of Intelligence, May 1982.

[29] Kumar, A., Characterization of Manipulator Geometry, Ph.D. Thesis, University of Houston, Feb., 1980.

[30] Kumar, A. and Waldron, K., "The Workspaces of a Mechanical

Manirulator," Journal of Mechanical Desisn, Vol. 103, July, 1981, rr. 665-672.

[31] Lanrana, N. A., "Spatial Kinematic Analysis of the Urrer Extremity Usins a Biplanar Videotapins Method," Journal of Biomechanical Ensineerins, Vol. 103, Feb., 1981, rr. 11-17.

[32] Lewis, R. A., "Autonomous Manipulation on a Robot: Summary of Manipulator Software Functions," Jet Propulsion Laboratory Tech. Memo 33-679, March, 1974.

[33] Lootsma, F. A. (ed.), Numerical Methods for Nonlinear Optimization, Academic Press, New York, 1972.

[34] Lozano-Perez, T. and Wesley, M. A., "An Alsorithm for Plannins Collision-free Paths amons Polyhedral Obstacles," _Communications of the ACM_, Vol. 22, Oct., 1979, rr. 560-570.

[35] Lozano-Perez, Tomas, "Automatic Plannins of Manipulator Transfer Movements," _IEEE Transactions on Systems, Man, and Cybernetics_, Vol SMC-11, No. 10, Oct. 1981, rr. 681-698.

[36] McGhee, Robert B., "Control of Lessed Locomotion Systems," _Proceedinss of the 1977 Joint Automatic Control Conference_, Vol. 1, rr. 205-213.

[37] McGhee, Robert B. and Iswandhi, Geoffrey I., "Adaptive Locomotion of a Multilessed Robot Over Roush Terrain," _IEEE Transactions on Systems, Man, and Cybernetics_, Vol. SMC-9, No. 4, April 1979, rr. 176-182.

[38] Mujtaba, M. S., "Discussion of Trajectory Calculation Methods," Chapter 8 in "Exploratory Study of Computer Intesrated Assembly Systems," by Binford, T.O. et al, Stanford University Artificial Intellisence Laboratory Memo AIM-285, March, 1977.

[39] O'Rourke, Joseph F., Imase Analysis of Human Motion, Ph.D. Thesis, University of Pennsylvania, 1980.

[40] Ortesa, J. M. and Rheinboldt, W. C., Iterative Solution of Non-linear Equations in Several Variables, Academic Press, 1970.

[41] Paul, Burton, Kinematics and Dynamics of Planar Machinery, Prentice-Hall, 1979.

[42] Paul, Richard C., Modellins, Trajectory Calculation and Servoins of a Computer Controlled Arm, Ph.D Thesis, Stanford University, 1972.

[43] Paul, R., Luh, J., Bender, J., Brown, R., Reminston, M. and Walker, M., "Advanced Industrial Robot Systems," First Report, Purdue University School of Electrical Ensineerins Tech Rep. TR-EE 78-25, May, 1978.

[44] Paul, Richard, "Manipulator Cartesion Path Control," IEEE

Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, No. 11, Nov., 1979, pp. 702-711.

[45] Paul, R., Shimano, B. and Mayer, G. E., "Kinematic Control Equations for Simple Manipulators," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 6, June, 1981.

[46] Paul, Richard P., Robot Manipulators, MIT Press, Cambridge, Mass., 1981.

[47] Pieper, Donald L., The Kinematics of Manipulators under Computer Control, Ph.D. Thesis, Stanford University, Oct., 1968.

[48] Roberts, L. G., "Homogeneous Matrix Representation and Manipulation of N-Dimensional Constructs," MIT Lincoln Laboratory Document MS1045, May, 1965.

[49] Roberts, L. G., "Machine Perception of Three Dimensional Solids," Optical and Electro-Optical Information Processing, Tippett, J. T., ed., MIT Press, Cambridge, Mass., 1965.

[50] Roth, Bernard, "Performance Evaluation of Manipulators from a Kinematic Viewpoint," _Performance Evaluation of Programmable Robots and Manipulators_, National Bureau of Standards Special Publication 459, 1976, pp. 39-62.

[51] Roth, Bernard, "Robots," Applied Mechanics Reviews, Vol. 31, No. 11, Nov., 1978, pp. 1511-1519.

[52] Shimano, Bruce, "Improvements in the AL Run-Time System," Chapter 3 in "Exploratory Study of Computer Integrated Assembly Systems: Progress Report 3," Stanford University Artificial Intelligence Laboratory Memo AIM-285, 1976.

[53] Spegel, Marjan, _Programming of Mechanism Motion_, Tech. Report No. CRL-43, Division of Applied Science, New York University, 1975.

[54] Sugimoto, K. and Duffy, J., "Determination of Extreme Distances of a Robot Hand - Part 1: A General Theory," Journal of Mechanical Design, Vol. 103, July, 1981.

[55] Sugimoto, K. and Duffy, J., "Determination of Extreme Distances of a Robot Hand - Part 2: Robot Arms with Special Geometry," Journal of Mechanical Design, Vol. 103, Oct., 1981.

[56] Suh, C. H. and Radcliffe, C. W., Kinematics and Mechanisms Design, John Wiley and Sons, New York, 1978.

[57] Waters, Richard C., "A Mechanical Arm Control System," MIT Artificial Intelligence Laboratory Memo AIM-301, Jan., 1974.

[58] Waters, Richard C., "Mechanical Arm Control," MIT Artificial Intelligence Laboratory Memo AIM-549, Oct., 1979.

[59] Weber, L., Smoliar, S. W. and Badler, N. I., "An Architecture for the Simulation of Human Movement," Proceedings of the ACM National Conference, 1978, pp. 737-745.

[60] Whitney, Daniel, "Resolved Motion Rate Control of Manipulators and Human Prostheses," IEEE Transactions on Man-Machine Systems, Vol. MMS-10, No. 2, June, 1969, pp. 47-53.

[61] Whitney, Daniel, "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," Journal of Dynamic Systems, Measurement, and Control, Vol. 94, Series G, No. 4, Dec., 1972, pp. 303-309.

[62] Whitney, D. E., C. S. Draper Laboratories, Cambridge, Mass., Personal Communication.

[63] Williamson, R. E., Crowell, R. H. and Trotter, H. F., Calculus of Vector Functions, Prentice-Hall, Englewood Cliffs, N.J., 1972.

[64] Wismer, David A. and Chattergy, R., Introduction to Nonlinear Optimization, North Holland, New York, 1978.

[65] Zeltzer, David, "Motor Control Techniques for Figure Animation," _Computer Graphics and Applications_, Oct. 1982.