Kinematics of a Flexible-Link Manipulator

R. E. Ellis

COINS Technical Report 86-10

Laboratory for Perceptual Robotics
Department of Computer and Information Science
University of Massachusetts, Amherst MA 01003

ABSTRACT

Most robotic manipulators are composed of rigid members in serial linkage. A tendon-driven flexible-limb linkage is described, and its kinematics are developed. The linkage is unusual because the variable-stiffness bending of the limbs is the primary method of movement of the manipulator. This presents interesting problems in kinematics, dynamics, and obstacle avoidance.

Additionally, the kinematics appear not to have a closed-form inverse – which is an unusual situation in conventional robot linkages. The inverse kinematics are solved using a fast, simple iterative technique, which appears capable of solving the inverse position and velocity kinematics at typical servo sampling rates.

This research has been supported by the Natural Sciences and Engineering Research Council of Canada, and by the Office of Naval Research under Contract N00014-84-K-0564. Publication as a COINS Technical Report was recommended by Prof. Allen R. Hanson.

1. Introduction

Most industrial manipulators are designed to be fast and have good repeatability. As a result of these and other considerations, they are often composed of rigid members in serial linkage. Such manipulator design can lead to well-specified kinematics and reasonably well-behaved dynamics, if appropriate assumptions are made.

The manipulator examined here violates these design constraints. It is a tendon-driven parallel linkage, in which the limb tip is moved by bending the limb (with variable stiffness control available during the bend). It also appears that the inverse kinematics are sufficiently complex to rule out a simple closed-form solution.

This paper describes an *iterative* approach to the inverse kinematic problem. The convergence is such that, if there were separate processors for solving the kinematics and for controlling the manipulator, then it would be possible to employ the results of an early iteration as a first approximation to the final control variable values, and to complete the solution while the manipulator is beginning to perform the movement.

1.1 Mechanical Considerations

The basic element of the manipulator is a flexible beam of circular cross section. Because the movement of such an object is reminiscent of an elephant's trunk, this is referred to as a **Trunk** limb. Consider a pair of cables running longitudinally through a homogeneous beam, the cables diametrically opposed and at the same distance from the beam centre line. Let one of the cables be pulled shorter than the other (both are anchored at one end, pulled from the other). Disregarding gravity, the beam will be curved into the arc of a circle, the curvature of which depends *only*

upon the difference in length between the two cables. The sum of the tension in the two cables will (in part) determine the total stiffness of the beam. The stiffness and position are thus independently controllable.

Note, however, that this is an unusual manipulator limb. Unlike a rotary joint, the distance from the base of the limb to the tip is not fixed; and unlike a prismatic or sliding joint the orientation of the tip changes. What *is* constant is the arc length of the beam, the centre arc that lies between the two tendons. Figure 1.1 gives a simple schematic representation of a non-homogeneous (but kinematically very similar) design for such a limb.

Suppose now that a second pair of tendons is added, orthogonal to the first set. Once again, if the tendons are pulled differentially, then the beam will arc; but the actions of the first and second set can no longer be separated. The beam must still arc if both sets are pulled, and the direction and total arc depend upon the vector sum of the tensions of the two pairs. The stiffnesses, being independently controllable, produce a beam of anisotropic cross-sectional stiffness. (If the stiffnesses are equal, the system is mechanically equivalent to a beam of circular cross section fixed at one end, with a bending moment applied to the tip.)

Such a limb has two degrees of freedom. Suppose that another **Trunk** limb was arranged serially with it; the base of one limb would be attached to the tip of the other. The manipulator would thus have four degrees of freedom, and it is this arrangement that will be considered and solved.[1] The *positional problem* is to find the values of the control variables which move the manipulator tip to a desired point (possibly with other constraints, e.g. orientation); the *velocity problem* is to find the

[1] In actual use, a standard three-degree-of-freedom wrist would likely be mounted on the manipulator tip, for a total of seven degrees of freedom. A good wrist is, by and large, kinematically independent of the arm on which it is mounted; the two problems are amenable to parallel computation. See [Hollerbach & Sahar, 1983] for a discussion of wrist-partitioned inverse kinematics.

rate of change of control variables which result in the desired tip velocity.

1.2 Notation

The kinematics of a manipulator will be assumed throughout to be translations and rotations from the origin of the base coordinate system to the manipulator tip. All such transformations will be represented by *homogeneous transforms*, which are 4×4 matrices. The first three columns represent the orientation transformation, and the last column – the P vector – represents the translation (or, alternatively, the position with respect to the base coordinate frame). Homogeneous transforms and their use in kinematics are described in [Paul, 1981]. The first three components of the P vector are denoted as P_X, P_Y, and P_Z respectively; the fourth component is always 1.

When only two axes are needed, the $[X, Z]$ pair will be the default, rather than the more conventional $[X, Y]$ pair. This will be notationally convenient when extra degrees of freedom are added to the elementary limb model.

Table 1.1: Notation

P	:	Position vector in homogeneous coordinates
P_X	:	**X** component of position
$\mathbf{V}$	:	Velocity vector in homogeneous coordinates
$\mathbf{V_X}$	:	**X** component of velocity
A, T	:	4×4 homogeneous transform matrices
$\alpha, \beta, \gamma, \omega$	:	Control variables
$\dot{\alpha}$	:	Rate of change of α
S_α	:	$\sin(\alpha)$
C_α	:	$\cos(\alpha)$
V_α	:	$\text{versine}(\alpha) \equiv 1 - \cos(\alpha)$

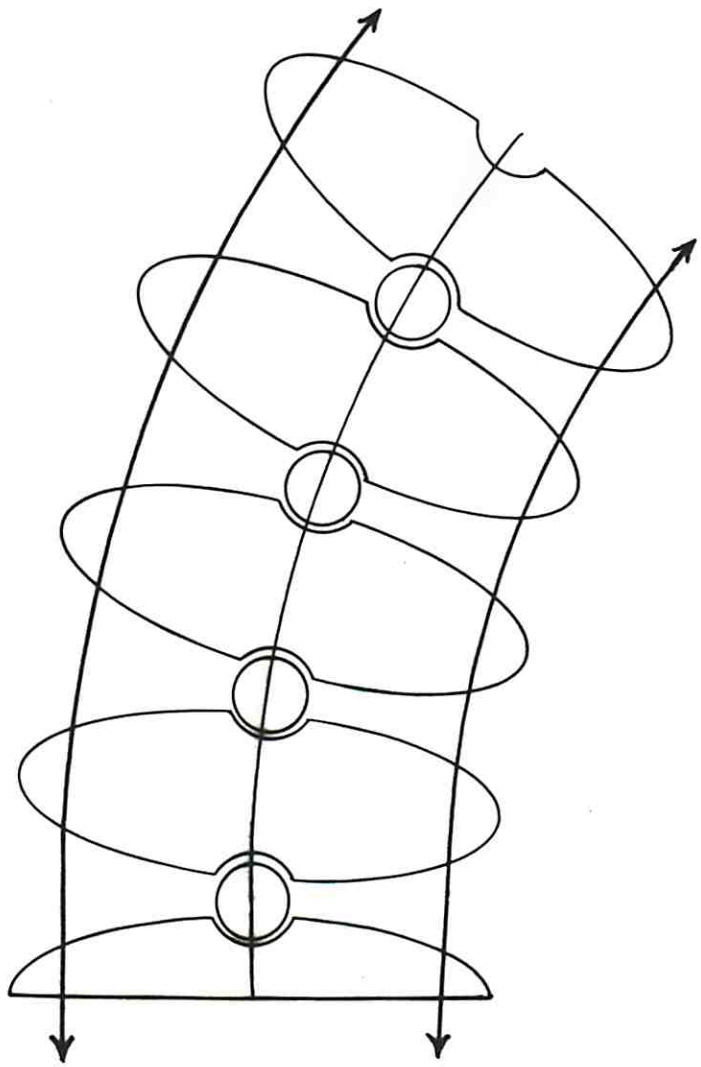

Figure 1.1: A possible construction of the limb.

2. Positional Kinematic Structure

The kinematic structure of a **Trunk** limb is not difficult to analyse if the limb is stiff, but becomes very complicated if the limb is compliant. The problem of a stiff limb is the one addressed here.

2.1 One-Degree-of-Freedom Case

Assume that the limb is sufficiently stiff so that gravity loading produces no significant deflection of the tip (bending, compressive, and shear forces are negligible). The stresses on each of the cables will be approximately equal at any two points; assume that the mechanical structure allows the strains to also be equal at any two points along a cable. Under such conditions, each cable (and thus the central axis of symmetry of the limb) must describe an arc of a circle, the radius depending on the relative lengths of the cable.

Suppose that, when straight, the limb is of length L, and that the cables are separated by a distance d. Let the lengths of the two cables be L_1 and L_2; without loss of generality, let $L_1 \geq L_2$. See Figure 2.1.

Observe that when $L_1 > L_2$, both describe arcs of concentric circles; furthermore, both arcs must subtend the same angle. Let this angle be α, and let the radius of the inner, L_2 circle be $r = d/2$. Then we know that $L_2 = \alpha \cdot (r - d/2)$, and also that $L_1 = \alpha \cdot (r + d/2)$; hence,

$$\alpha = \frac{L_1 - L_2}{d}$$

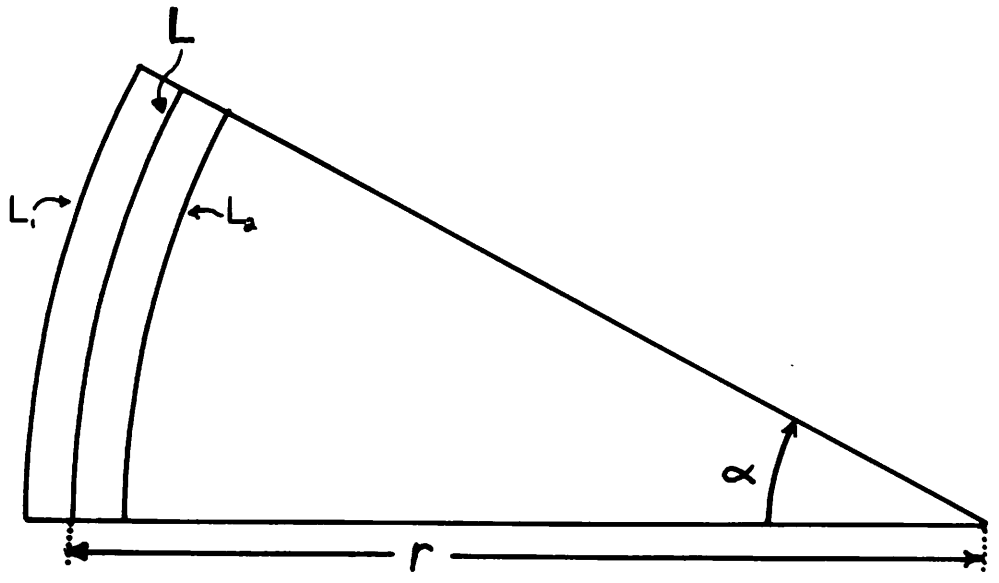

Figure 2.1: Geometry of a 1-DOF limb.

and the radius of the limb arc is

$$r = \frac{L}{\alpha} \qquad (\alpha = 0 \Rightarrow r = \infty)$$

From these, the position of the tip of the limb can be derived as

$$P_X = r - r \cdot \cos\alpha$$

$$P_Z = r \cdot \sin\alpha$$

Figure 2.2 gives a geometric representation of these relationships. Note that all of these equations remain valid if $L_1 < L_2$, because of sign changes in r and $\sin\alpha$.

This describes the position of the limb tip. The tip's orientation will be that of the base coordinate frame, rotated by α. To establish a base coordinate frame, suppose that when $\alpha = 0$ the limb bottom is at $[X, Z] = [0, 0]$, with the limb tip at

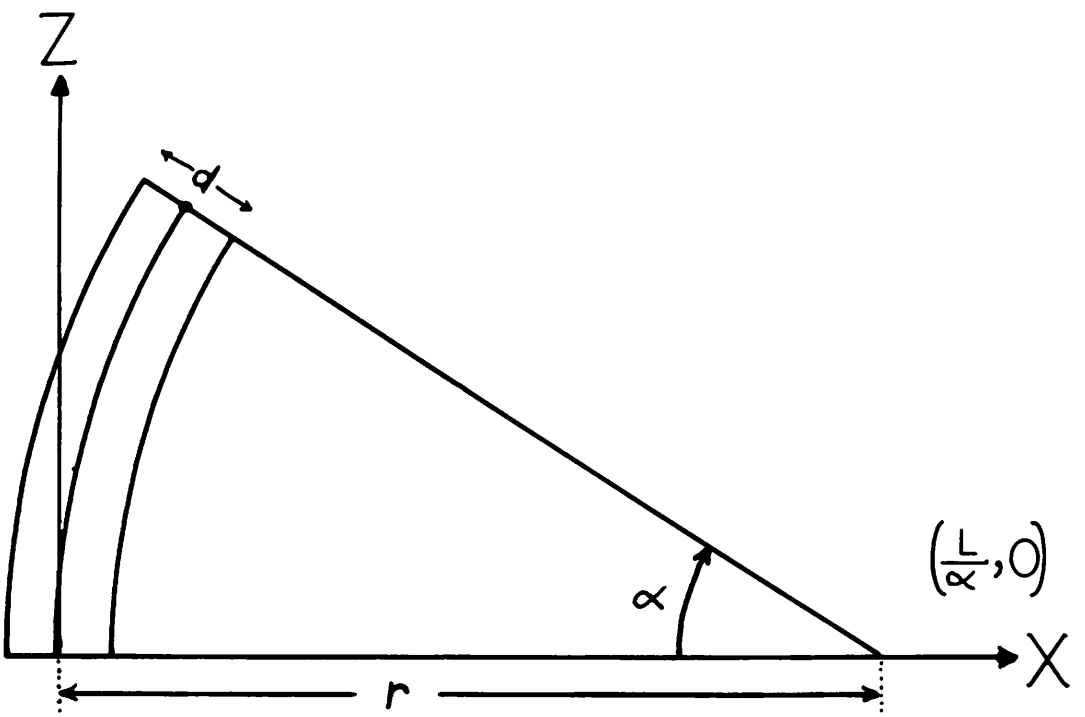

Figure 2.2: Coordinate system for a 1-DOF limb.

$[0, L]$; the limb is thus pointing along the **Z** axis. The total transformation from limb base to tip is, in homogeneous coordinates,

$$A_1 = \begin{vmatrix} \cos\alpha & 0 & \sin\alpha & L\left(\frac{1-\cos\alpha}{\alpha}\right) \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & L\frac{\sin\alpha}{\alpha} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

where

$P_X = 0$ if $\alpha = 0$

and

$P_Z = L$ if $\alpha = 0$

Figure 2.3 is a representation of the workspace line described by the limb tip as it is arced through $0 \leq \alpha \leq \pi$. Note that the orientation is not simply the normal to the dashed line; it is a parametric function of α, as given in the above homogeneous transform. (The orientation is the direction in which the tip is pointing.)

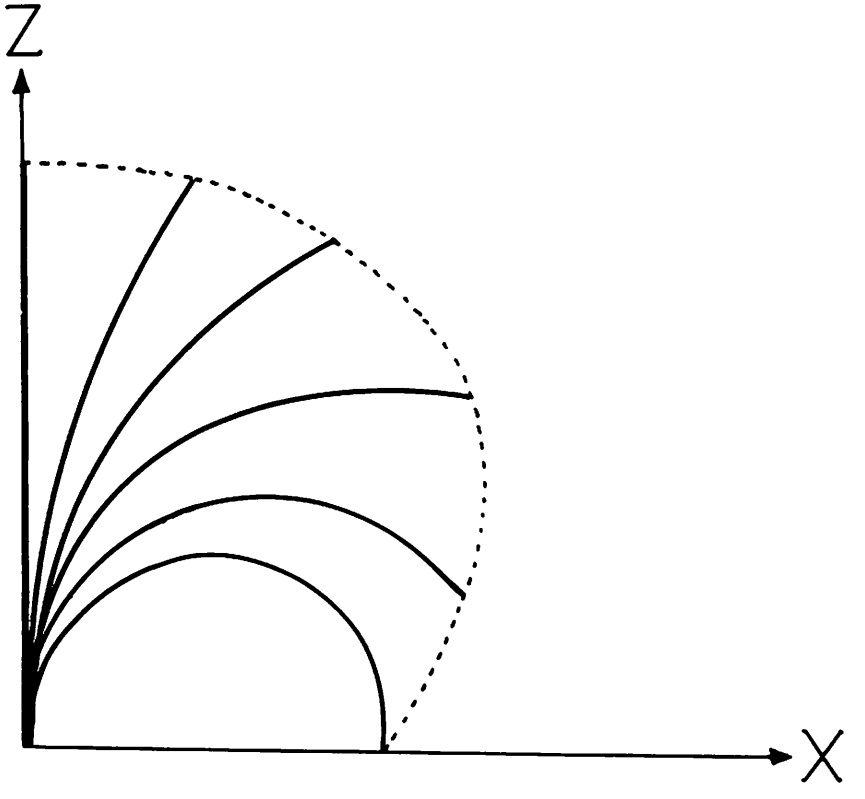

Figure 2.3: Workspace of a 1-DOF limb.

2.2 Two-Degree-of-Freedom Case

Consider now an augmentation to the 1-DOF limb. Assume that a second pair of cables is run parallel to the first pair, such that pulling on the second pair results in arcing orthogonal to that of the first pair (holding the first pair at $\alpha = 0$). Denote the control angle of this second pair by β. As in the 1-DOF case, β is simply related to the cable separation and the differences in lengths of the pair. Letting the cable separation be the same **d** as for the α pair, and denoting the respective cable differences by D_α and D_β, then the relations are $\alpha = D_\alpha/d$ and $\beta = D_\beta/d$.

When both cable sets are actuated, the assumption of uniform force distribution ensures that the limb will still describe an arc (or a line, in the limiting case of $D_\alpha = D_\beta = 0$). This arc will be in a plane Ω that is orthogonal to the **X-Y** plane; denote the angle between these two planes by ω. The arc in the Ω plane may be described by a virtual control variable γ, which is exactly analogous to α in the 1-DOF, **X-Z** plane case. We can continue this analogy by postulating a pair of virtual control cables for γ, which will have a difference in length of D_γ. As with the other control variables, this virtual one obeys the relationship $\gamma = D_\gamma/d$ (same **d** as for α and β). Figure 2.4 shows the coordinate system for a 2-DOF limb, and the angle ω.

The relationships between $\{D_\alpha, D_\beta\}$ and $\{D_\gamma, \omega\}$ are not complex. First, suppose that γ is held constant at some non-zero value **k**, i.e. the limb height is constant and the limb is arced. Then as D_α and D_β are varied such that $\gamma = \mathbf{k}$, the limb tip will describe a circle. Second, if ω is held constant, the magnitude of D_γ will be the distance from the point $[\mathbf{0}, \mathbf{0}]$ to $[\mathbf{D}_\alpha, \mathbf{D}_\beta]$ in the control space.

These relations are shown geometrically in Figure 2.5. Algebraically, they are just

$$D_\gamma^2 = D_\alpha^2 + D_\beta^2$$

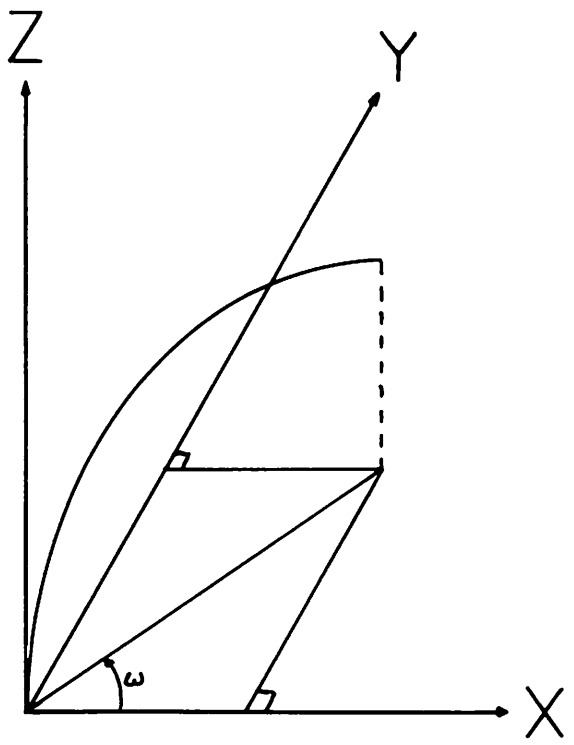

Figure 2.4:. Coordinate system for a 2-DOF limb.

and

$$\omega = \tan^{-1} \frac{D_\beta}{D_\alpha}$$

Because of the common divisor **d**, these can be expressed in terms of the control angles α and β, and the virtual control angle γ, as

$$\gamma^2 = \alpha^2 + \beta^2$$

and

$$\omega = \tan^{-1} \frac{\beta}{\alpha}$$

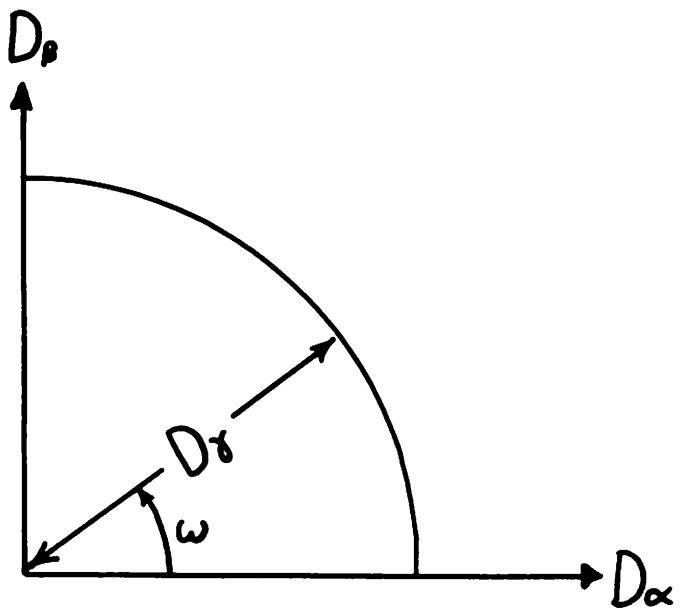

Figure 2.5: Control space for a 2-DOF limb.

From these, the translation and rotation components of the transformation are readily determined. The original coordinate frame is rotated by the angle γ in the Ω plane, in a manner analogous to the 1-DOF case (in which the rotation was about the Y-axis). The axis of rotation is about the vector $[-\sin\omega, cos\omega, 0]$. Calculating each, and composing first the translation, then the rotation, yields the two-degree-of-freedom transform matrix

$$T_2 = \begin{vmatrix} S_\omega S_\omega V_\gamma + C_\gamma & -S_\omega C_\omega V_\gamma & C_\omega S_\gamma & C_\omega L\frac{V_\gamma}{\gamma} \\ -S_\omega C_\omega V_\gamma & C_\omega C_\omega V_\gamma + C_\gamma & S_\omega S_\gamma & S_\omega L\frac{V_\gamma}{\gamma} \\ -C_\omega S_\gamma & -S_\omega S_\gamma & C_\omega & L\frac{S_\gamma}{\gamma} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

The workspace of this 2-DOF limb is the surface that results from the revolution of the curve of Figure 2.3 about the **Z** axis. Once again, note that the orientation of the limb tip is not the normal to the surface.

The above kinematics assume that a **Trunk** limb may be modelled as a continuous beam, but the physical structure that is easiest to build consists of a series of discs stacked and hinged in various manners. As a confirmatory test, the limb was modelled in a simple way, using various numbers of discs, and the finite model compared to the continuous model (limb lengths were identical in the continuous and discrete models). For a series of 50 discs, the error between the position vectors for the continuous and finite models was about one part in a thousand; for 100 discs, the error dropped to about three parts in ten thousand. The latter involved successive multiplication of 100 4×4 homogeneous matrices, requiring significant computation; slightly smaller error achieved by using continuous model is probably not worth the computational effort. This small difference in models is thus taken to be sufficient justification for continuing to model a **Trunk** limb as a continuous beam.

3. Inverse Positional Kinematics

The inverse kinematic problem is, given some parameters specifying the position and orientation of the manipulator, to find values of the control variables which will result in the manipulator tip satisfying the given parameters. If there are fewer parameters than control variables, the problem is *under-constrained*; if more, the problem is *over-constrained.* Even if the number of parameters equals the number of control variables, there may be more than one set of control variables which satisfy the parameters; in such a case, the parameters are said to represent a *kinematic degeneracy.*

The inverse positional kinematics of the **Trunk** manipulator will be dealt with in three successive passes: first, a single 1-DOF limb; second, a single 2-DOF limb; and lastly, a pair of 2-DOF limbs in serial linkage (a total of four degrees of freedom). The first two passes will be over-constrained, and the third will be exactly constrained. The over-constrained cases are interesting both because with the constraints given the control variable can be found with very little calculation, and because the exactly-constrained case readily reduces to the algebraically simpler over-constrained cases.

3.1 1-DOF Inverse Kinematics

If the parameter given is the **Z** coordinate of the limb tip, then we could solve for α by noting that

$$P = L\frac{S_\alpha}{\alpha}$$

This, however, entails finding the inverse function of

$$f(x) = \frac{\sin(x)}{x}$$

for which there is no known closed form inverse. The inverse could be numerically approximated in many ways, e.g., series expansion and reversion, root-finding, spline approximation, or interpolation. The problem is similar if the solution parameter is P_X.

If, however, *both* P_X and P_Z are given, then α may be found by employing trigonometric or inverse trigonometric functions. Observe that

$$\left(\frac{P_X}{L}\right)^2 + \left(\frac{P_Z}{L}\right)^2 = \frac{\sin^2\alpha + 1 + \cos^2\alpha - 2\cos\alpha}{\alpha^2} = \frac{2}{\alpha}\frac{V_\alpha}{\alpha}$$

which implies that

$$\alpha = \frac{2LP_X}{P_X^2 + P_Z^2} \tag{3.1}$$

Formula 3.1 requires only one addition, three multiplications (assuming that $2 \cdot L$ is stored as a constant), and one division.

3.2 2-DOF Inverse Kinematics

The 2-DOF case is not very different from the 1-DOF case. The control variables to be solved for are γ and ω. If the three Cartesian coordinates of the limb tip are chosen as solution parameters – P_X, P_Y, and P_Z – then solution is possible in a manner similar to the 1-DOF case.

Observe that $\omega = \tan^{-1}(P_Y/P_X)$. The distance from the origin to the orthographic projection of the tip on the **X**-**Y** plane is $\sqrt{P_X^2 + P_Z^2}$; using this in Formula 3.1,

which gives α in terms of P_X and P_Z, and given the relations between α, β, γ, and ω from the kinematics of the 2-DOF limb, we have

$$\gamma = \frac{2L\sqrt{P_X^2 + P_Z^2}}{P_X^2 + P_Y^2 + P_Z^2} \tag{3.2}$$

$$\alpha^2 = \frac{\gamma^2}{1 + \omega^2}$$

and

$$\beta^2 = \gamma^2 - \alpha^2$$

The solution is completed by setting the sign of α to be that of $\gamma \cdot \cos\omega$, and the sign of β to that of $\gamma \cdot \sin\omega$. Should $P_X = P_Y = 0$, it is clear that $\alpha = \beta = 0$.

4. 4-DOF Serial Linkage Inverse Positional Kinematics

The next linkage to consider is one in which a 2-DOF **Trunk** limb has attached to its tip the base of another 2-DOF **Trunk** limb. Let the control cables for the upper limb be colinear and parallel with those of the lower limb, for simplicity. Assume for now that both limbs are of identical length and construction. The base of the lower limb is located at $[0,0,0]$, and it is the position of the upper limb tip which is of kinematic interest.

The control variables for the upper limb will be simple correlates of those for the lower limb:

Lower Limb		Upper Limb
α	$\longleftrightarrow$	ϕ
β	$\longleftrightarrow$	ψ
γ	$\longleftrightarrow$	η
ω	$\longleftrightarrow$	θ

Thus, if both limbs started straight, the arcing of the lower limb due to α would be in the same plane as that of the upper limb due to ϕ. Note that η and θ are measured and controlled with respect to coordinate frame at the tip of the lower limb, which in general is both translated and rotated from the base coordinate frame.

The parameters that constrain the inverse kinematic solution may now be selected. Three natural choices are the Cartesian coordinates of the tip; these will be P_X, P_Y, and P_Z of the final transform matrix. After much consideration, the fourth parameter selected was the direction of tilt of the lower limb. Such a parameter is useful in obstacle avoidance, and simplifies some kinematic equations. This fourth parameter is the control variable ω of the lower limb.

The initial approach taken involved setting up the transform matrix from the base to the tip in terms of homogeneous transforms, and inspecting the resulting position column. This is a rather imposing set of equations, and does not seem to admit of closed-form solution. Hence, a numerical approximation method is employed.

Observe that because ω is one of the parameters, the original 4-DOF problem is now a 3-DOF problem: if γ, and either ϕ and ψ *or* η and θ, can be found then the problem is solved. The first step is thus to rotate the base coordinate system about the **Z** axis by ω, and to modify P_X and P_Y accordingly. This results in a 1-DOF **Trunk** limb surmounted by a 2-DOF limb. The position of the tip in this new coordinate frame, given by composing the 1-DOF and 2-DOF limb transforms, is

$$\begin{aligned} P_X &= L\left(C_\gamma C_\theta \frac{V_\eta}{\eta} + S_\gamma \frac{S_\eta}{\eta} + \frac{V_\gamma}{\gamma}\right) \\ P_Y &= LS_\theta \frac{V_\eta}{\eta} \\ P_Z &= L\left(-S_\gamma C_\theta \frac{S_\eta}{\eta} + C_\gamma \frac{S_\eta}{\eta} + \frac{S_\gamma}{\gamma}\right) \end{aligned}$$

Note, however, that if γ (the control variable for the lower limb in this new system) is specified, the problem is simple; for, having γ, we can transform the base coordinate system to the tip of the lower limb, and solve for the upper limb control variables as a 2-DOF limb in which the three Cartesian coordinates of its tip are given.

The approach, then, is to *iteratively* find γ. From any initial value of γ that is sufficiently close to the final value, it is possible to approximate a correct value of γ if an error term can be found, and if this error term can be used to produce a new value of γ such that the limb tip can be moved closer to the desired point.

4.1 Iterative Solution of the 4-DOF Positional Problem

A *numerical approach* to the problem is an attempt to find the control variables so that the manipulator tip comes within some specified ϵ of the desired point. Several things are needed if an iterative numerical approach to this problem is to succeed:

- Given the desired point, a function must be developed which will give reasonable values for the control variables. Such a function must actually yield correct values if the point is within the limb workspace, and must also get the tip reasonably close (according to some measure) to the desired point if that point is *not* within the workspace.

- An error function must be found which will indicate the distance (by some metric) between the actual tip location and the desired location.

- A method must be developed which will use the error function to change the current value of the iteration variable (in this case, γ) to one which allows the manipulator tip to move closer to the desired location.

- The value(s) of the iteration variable must be within the convergence interval of the procedure, i.e. the starting point must be sufficiently close to the end point that the method works.

The approximation function used is the one which solves for γ given P_X and P_Z (Formula 3.2). It returns the correct result if the point is reachable, and simulation tests showed that it was not unreasonable when the desired point was near (but not on) the workspace line of the 1-DOF manipulator.

Several error functions were examined, and the choice of error function crucially affected the choice of iteration method. The one chosen was a signed form of the simple absolute-value metric $\mathbf{D}(x, y) = |x_1 - y_1| + |x_2 - y_2|$. This is easily calculated, and produces rapid convergence. Setting the sign of the error function according to

whether the **X** coordinate of the tip is less or greater in absolute value than the **X** coordinate of the desired point converts the problem to one of finding a simple real root, which is numerically more stable than optimisation.

4.2 Algorithm for Solving the 4-DOF Problem

The given parameters of the inverse kinematic problem are the desired tip position $P = [P_X, P_Y, P_Z]$, and the direction of tilt ω of the lower **Trunk** limb. First, the base coordinate system is rotated by ω about the **Z** axis, to convert the original 4-DOF problem into a 3-DOF problem. The rotation changes the original desired point P to the new desired point, denoted P':

$$P' = \begin{vmatrix} P_X \cos(-\omega) - P_Y \sin(-\omega) \\ P_X \sin(-\omega) + P_Y \cos(-\omega) \\ P_Z \\ 1 \end{vmatrix}$$

In order to ensure convergence, the manipulator workspace is broken into a number of **voxels**, or volume elements. Each voxel is seeded with a good value of the iteration variable γ; the seed values were found by extensive offline simulation, which involved changing the control variables in very small increments and calculating the position of the tip. Once it is determined in which the voxel the goal point of the tip lies, the seed and a value very close to the seed can be used as the initial values in the Secant method. The Secant method used is the one described in [Forsythe *et al.*, 1977], which attempts to find a root of the error function. (The error function is a function of the iteration variable γ.)

Observe that the desired point, if reachable, is the position column of the transform matrix $A_1 \cdot T_2$, where A_1 is the transform matrix of a 1-DOF limb, and T_2 is the

transform matrix of a 2-DOF limb. Denoting the inverse transform matrix of A_1 as A^{-T}, then the position vector $A^{-T} \cdot P'$ should be very close to the position vector of the transform T_2. Let this position vector be denoted as $P^{\sim}$ (it will be equal to the position vector of T_2 only in the iterative limit, when the error is vanishingly small).

Then, given γ, we can find the appropriate A^{-T}, find $P^{\sim}$, and form guesses for the upper limb tilt direction

$$\theta = \tan^{-1} \frac{P_Y^{\sim}}{P_X^{\sim}}$$

and for the upper limb arcing parameter

$$\eta = \frac{2L\sqrt{P_X^{\sim 2} + P_Y^{\sim 2}}}{P_X^{\sim 2} + P_Y^{\sim 2} + P_Z^{\sim 2}}$$

The error function returns the value $\mathbf{D}(P^*, P^{\sim})$, where P^* is the position vector of the T_2 matrix, formed from θ and η. Let the error be positive if $P_X^{\sim} < P_X^*$, and negative otherwise. This error is the function for which the Secant method attempts to find a zero. A solution has been found when the absolute value of this error is less than the specified ϵ.

To complete the solution, it is necessary to find the control variables from the intermediate variables γ, ω, η, and θ.

4.2.1 Convergence Results

The iterative method is seeded with good values of γ. This is done by modelling a large number of 3-DOF manipulator moves beforehand; for each set of control variable values, the voxel in which the tip is positioned receives the γ control variable value for the lower limb. Once calculated, this need not be changed; it can be read in from external memory prior to solving the inverse kinematics.

For a series of simulations, some 500,000 manipulator moves were modelled, and a table of 30,000 voxel seeds was created. The pre-calculation routine used about 22 CPU minutes on a VAX 11/750. In a test of 100,000 randomly-chosen points known to be in the workspace, over half were found within 3 iterations. For this entire set, with ϵ chosen to be $L/10000$, the mean convergence time was slightly less than 19 milliseconds, and 99% were complete within 50 milliseconds.

With such convergence rates, it would be possible to use a simple control scheme and sample positions at rates typically used in current robots. A more advanced scheme using a pair of processors is also possible, by using early iteration results (which are close, but not yet within ϵ) and updating the desired position on-line.

5. Kinematic Velocities

Control of a manipulator often entails controlling not only position, but also velocity. The velocity problem to be solved is, given a position and desired tip velocity, to find the values and rates of change of the control variables which achieve these constraints. This approach presupposes a solution to the positional kinematic problem, which for this manipulator has been solved above.

The kinematic velocities may be solved by finding the rate of change of the limb tip, i.e., differentiating the position vector. This is not difficult conceptually, but requires some lengthy algebra and liberal application of L'Hôpital's Rule. The general method is to proceed from the 1-DOF to the 4-DOF limbs, using in each case the rates of change of the constraint parameters employed in the corresponding positional problem. For the 4-DOF problem, the constraints are thus the three Cartesian velocities and the desired rate of change of the lower limb angle ω.

When some of the control variables are zero, kinematic degeneracies arise; these can be dealt with by taking the limit of the rates of change. In all cases, these limits are well-behaved, indicating that combined positional and velocity control of such a manipulator is feasible.

5.1 1-DOF Limb

The 1-DOF kinematic velocities are relatively easy to state and solve. Differentiating the position vector P with respect to time, the rate of change of the tip is a function of the present position, and of the rate of change $\dot{\alpha}$ of the control varialbe

α:

$$\mathbf{V} = \frac{d}{dt}(P) = L\frac{\dot{\alpha}}{\alpha} = \begin{vmatrix} S_\alpha - \frac{V_\alpha}{\alpha} \\ 0 \\ C_\alpha - \frac{S_\alpha}{\alpha} \\ 1 \end{vmatrix}$$

If $\alpha = 0$, it can be shown that $\mathbf{V}$ degenerates to

$$\mathbf{V} = \begin{vmatrix} \frac{\dot{\alpha}L}{2} \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

The inverse functions, where we seek $\dot{\alpha}$, are simple. Assume that we know the current value of α, and we wish to control one of the spatial velocities V_X or V_Z. If $\alpha \neq 0$, then

$$\dot{\alpha} = \frac{\mathbf{V_X}\alpha}{LS_\alpha - \frac{V_\alpha}{\alpha}} = \frac{\mathbf{V_Z}\alpha}{LC_\alpha - \frac{S_\alpha}{\alpha}}$$

and if $\alpha = 0$ then $\mathbf{V_Z}$ must also be 0, so

$$\dot{\alpha} = \frac{2\mathbf{V_X}}{L}$$

5.2 2-DOF Limb

The variables we seek to control are α, β, $\dot{\alpha}$, and $\dot{\beta}$. It will be convenient here, as it was with the positional kinematics, to first derive the relationship between $\dot{\omega}$ and $\dot{\gamma}$, and the variables thatwe seek to control.

Recall that γ is given by the relation $\gamma = \sqrt{\alpha^2 + \beta^2}$; elementary calculus and algebra show that

$$\frac{d}{dt}(\gamma) = \dot{\gamma} = \frac{\alpha\dot{\alpha} + \beta\dot{\beta}}{\gamma}$$

Similarly, from $\omega = \tan^{-1}(\beta/\alpha)$, we have

$$\frac{d}{dt}(\omega) = \dot{\omega} = \frac{\alpha\dot{\beta} + \beta\dot{\alpha}}{\gamma^2}$$

Thus, it is possible to transform from the basic control variables to a set which expresses the kinematics in simpler terms, and *vie versa*.

The tip velocity $\mathbf{V}$ is, as in the 1-DOF case, just the derivative of the position vector with respect to time:

$$\mathbf{V} = \frac{d}{dt}(P) = \frac{L}{\gamma} \cdot \begin{vmatrix} \dot{\gamma} C_\omega \left(S_\gamma - \frac{V_\gamma}{\gamma} \right) - \dot{\omega} S_\omega V_\gamma \\ \dot{\gamma} S_\omega \left(S_\gamma - \frac{V_\gamma}{\gamma} \right) - \dot{\omega} C_\omega V_\gamma \\ \dot{\gamma} \left(C_\gamma - \frac{S_\gamma}{\gamma} \right) \\ 0 \end{vmatrix}$$

If $\gamma = 0$, this degenerates to

$$\mathbf{V} = \begin{vmatrix} \frac{L\dot{\gamma} C_\omega}{2} \\ \frac{L\dot{\gamma} S_\omega}{2} \\ 0 \\ 0 \end{vmatrix}$$

The inverse velocity kinematics can now be derived. Suppose that we are given the desired velocity component $\mathbf{V_Z}$, and that $\gamma \neq 0$. Then

$$\dot{\gamma} = \frac{\mathbf{V_Z} \cdot \gamma}{L\left(C_\gamma - \frac{S_\gamma}{\gamma}\right)}$$

We can derive $\dot{\omega}$ in a similar manner, either directly by specifying the desired rotational speed, or by giving the $\mathbf{V_X}$ or $\mathbf{V_Y}$ component and solving from the definition of the velocity vector and the value of $\dot{\gamma}$ just found. If $\gamma = 0$, i.e. if the limb is vertical, we may arbitrarily declare its tilt direction and rotational velocity to be zero; in this degenerate case, $\omega = \dot{\omega} = 0$, and $\gamma = (2\mathbf{V_X}/L)$.

Finally, from the velocities of the virtual control variables and the relations between them and the basic control variables, we have

$$\dot{\alpha} = \frac{\alpha\dot{\gamma}}{\gamma} - \beta\dot{\omega}$$

and

$$\dot{\beta} = \frac{\gamma\dot{\gamma} - \alpha\dot{\alpha}}{\beta}$$

which are the velocities we seek to control.

5.3 3-DOF Limb

As with the positional kinematics, the control variable of the lower limb will be γ and those of the upper limb will be ϕ and ψ. As was shown in § 5.2, there will be no difficulties raised by manipulating the virtual control variables θ and η in place of the basic control variables.

Once again, we differentiate the position of the tip with respect to time. The position vector of the serially composed 1-DOF and 2-DOF transforms is

$$P = L \cdot \begin{vmatrix} C_\gamma C_\theta \frac{V_\eta}{\eta} + S_\gamma \frac{S_\eta}{\eta} + \frac{V_\gamma}{\gamma} \\ S_\theta \frac{V_\eta}{\eta} \\ -S_\gamma C_\theta \frac{S_\eta}{\eta} + C_\gamma \frac{S_\eta}{\eta} + \frac{S_\gamma}{\gamma} \\ 1 \end{vmatrix}$$

The vector $\mathbf{V}$, which is the rate of change of P, is linear in terms of the velocities $\dot{\gamma}$, $\dot{\theta}$, and $\dot{\eta}$. When we let the coefficients be

$$\begin{aligned}
a &= S_\gamma C_\theta \frac{V_\eta}{\eta} + C_\gamma \frac{S_\eta}{\eta} + \frac{1}{\gamma}\left(S_\gamma - \frac{V_\gamma}{\gamma}\right) \\
b &= -C_\gamma S_\theta \frac{V_\eta}{\eta} \\
c &= \frac{C_\gamma C_\theta \left(S_\eta - \frac{V_\eta}{\eta}\right)}{\eta} + \frac{S_\gamma \left(C_\eta - \frac{S_\eta}{\eta}\right)}{\eta} \\
d &= C_\theta \frac{V_\eta}{\eta} \\
e &= \frac{S_\theta \left(S_\eta - \frac{V_\eta}{\eta}\right)}{\eta} \\
f &= -C_\gamma C_\theta \frac{S_\eta}{\eta} - S_\gamma \frac{S_\eta}{\eta} + \frac{C_\gamma - \frac{S_\gamma}{\gamma}}{\gamma} \\
g &= S_\gamma S_\theta \frac{S_\eta}{\eta} \\
h &= \frac{S_\gamma C_\theta \left(S_\eta - \frac{V_\eta}{\eta}\right)}{\eta} + \frac{C_\gamma \left(C_\eta - \frac{S_\eta}{\eta}\right)}{\eta}
\end{aligned}$$

the tip velocity can be expressed as

$$\mathbf{V} = \frac{d}{dt}(P) = L \cdot \begin{vmatrix} a\dot{\gamma} + b\dot{\theta} + c\dot{\eta} \\ d\dot{\theta} + e\dot{\eta} \\ f\dot{\gamma} + g\dot{\theta} + h\dot{\eta} \\ 0 \end{vmatrix}$$

This can be considered to be a linear system from the desired control variables to the known velocity values. Applying Gaussian elimination, and back-substituting, we can express the desired control variables as

$$\begin{aligned} \dot{\eta} &= \frac{d\left(a\frac{\mathbf{V_Z}}{\mathbf{L}} - f\frac{\mathbf{V_X}}{\mathbf{L}}\right) - (ag - bf)\frac{\mathbf{V_Y}}{\mathbf{L}}}{adh - cdf - aeg + bef} \\ \dot{\theta} &= \frac{\frac{\mathbf{V_Y}}{\mathbf{L}} - e\dot{\eta}}{d} \\ \dot{\gamma} &= \frac{\frac{\mathbf{V_X}}{\mathbf{L}} - b\dot{\theta} - c\dot{\eta}}{a} \end{aligned}$$

In calculating the coefficients, it is necessary to ensure that the degeneracy conditions (variables equalling zero) are properly handled. Fortunately, all of the functions necessary have well-behaved limits about zero:

$$\begin{aligned} \lim_{x\to 0} \frac{\mathrm{vers}(x)}{x} &= 0 \\ \lim_{x\to 0} \frac{\sin(x)}{x} &= 1 \\ \lim_{x\to 0} \frac{1}{x}\left(\sin(x) - \frac{\mathrm{vers}(x)}{x}\right) &= \frac{1}{2} \\ \lim_{x\to 0} \frac{1}{x}\left(\cos(x) - \frac{\sin(x)}{x}\right) &= 0 \end{aligned}$$

The solution is completed by tranforming from the virtual control variables to the basic control variables.

5.4 4-DOF Limb

The solution of the 4-DOF kinematic velocity problem is analogous to the solution of the 4-DOF kinematic position problem. In the latter, we rotated the frame of reference to remove the non-zero ω tilt direction, and reducing it to a 3-DOF problem. Here, we must not only remove the rotation, but also the *rate* of rotation.

There are four virtual control variables to be found, and thus four constraints to be supplied. Let us choose as the four constraints the desired Cartesian velocity of the tip, and the rotational speed of the lower limb. The velocity induced by the lower limb rotation, em at the manipulator tip, is directed along the rotated **Y** axis; its magnitude is the **X** coordinate of the tip in the rotated frame, times the rotational speed $\dot{\omega}$. Hence, the constraining tip velocity for solution of the 3-DOF inverse kinematic velocity problem is the desired Cartesian tip velocity, plus a correction for the magnitude of the rotation of the rotated frame, plus a correction for the rate at which this frame is rotating:

$$\mathbf{V_{tip}} = \begin{vmatrix} \mathbf{V_X}\cos(-\omega) - \mathbf{V_Y}\sin(-\omega) - P_Y\dot{\omega} \\ \mathbf{V_X}\cos(-\omega) - \mathbf{V_Y}\sin(-\omega) - P_X\dot{\omega} \\ \mathbf{V_Z} \\ 0 \end{vmatrix}$$

This new constraining tip velocity may then be used in the 3-DOF solution, as described above. Note, however, that in transforming from the virtual control variables to the basic control variables it is necessary to remove the effect of *both* the rotated

frame of reference *and* its rate of rotation. Once these effects have been removed, the transformation is straightforward.

6. Extensions to the Simple Model

One of the simplifications made in the above kinematics was the assumption that the base of one limb attached directly to another, and that the limb tip was of kinematic importance. In constructing such a manipulator, however, there would almost certainly have to be a physical extension on the end of the limbs, for the purpose of attaching end effectors, drive units, cabling, etc.

This is no great problem, if it is assumed that there is a fixed translatory segment on the end of the limb, one which is parallel and colinear with the central arc line. (Only the changes in the 1-DOF kinematics are presented; 2-DOF changes are similar.) The homogeneous transform for the 1-DOF limb, with an extension of length λ on its tip, is

$$A_1^* = \begin{vmatrix} \cos\alpha & 0 & \sin\alpha & L\frac{V_\alpha}{\alpha} + \lambda S_\alpha \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & L\frac{S_\alpha}{\alpha} + \lambda C_\alpha \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Sparing the reader the details, it can be shown that for such a linkage

$$\alpha = \frac{2LP_X}{P_X^2 + P_Z^2 - \lambda^2}$$

provided that $P_X^2 + P_Z^2 \neq \lambda^2$. If λ is relatively small, e.g., less than $L/5$, there is no significant change in either the convergence rate or the success rate of the iterative solution.

This extension to the model has no great effect on the inverse velocity problem. A few more terms are added, and the algebra is slightly more complicated, but there is no significant change required in the problem-solving approach.

Another problem – which is far more difficult – is that of compliant limbs. The complexity of the kinematics (all dynamics aside) increases dramatically if the limbs deflect noticeably under gravity. This is a great difficulty because the deflection of the lower limb depends not merely upon the mass of the upper limb, but also upon the orientation – the upper limb generates not merely a downward force, but also a torque that is a function of its orientation as well as its own configuration. Worse still, the exact configuration of the upper limb depends upon the orientation due to the lower limb; so there is a dual feedback in the beam deflections of the two limbs. The slightly simpler case of a stiff lower limb and a compliant upper limb is currently under investigation.

In any case, it is probably wise to make the cross-sectional stiffness of a **Trunk** limb as uniform as possible. Since there are independent, orthogonal stiffness controls, a limb can mimic the stiffness of a wide variety of uniform rectangular beams, from ones with square cross-sections to ones which are much wider than they are thick. The problems of a varying-stiffness beam whipping through space are expected to be of sufficient computational complexity that on-line calculation and control employing conventional hardware will require some ingenuity.

7. Conclusions

It is possible to solve iteratively a small but interesting kinematic problem for which no closed-form solution seems to exist. The iterative approach is accurate (one part in 20000, which is considerably more accurate than the mechanical system would likely be), reliable, and sufficiently fast to be a real-time algorithm. Being an iterative approximation employing a stable numerical method, it is reasonably insensitive to kinematic degeneracies, which show up as multiple roots in the error function – a situation readily handled by a good implementation of the Secant method.

More important, though, is the fact that that the method worked at all. There are a large number of possible manipulator designs with kinematics not solvable by ordinary means; it is possible that many of them are amenable to such an iterative approach. Some alternative designs are now being considered, and it is hoped that this simple numerical method is as successful in subsequent applications as it was in its first.

Acknowledgements

Many thanks are due to Steve Begej for detailed checking of the equations, to Ken Overton for his valuable encouragement, and to Al Hanson for comments on the manuscript.

REFERENCES

Forsythe, G. E., Malcolm, M. A., and Moler, C. B.: 1977. *Computer Methods for Mathematical Computations.* Prentice-Hall: Englewood Cliffs, NJ.

Hollerbach, J. M. and Sahar,G.: 1983. Wrist-partitioned inverse kinematic accelerations and manipulator dynamics. *A.I. Memo 717, MIT AI Laboratory,* Massachusetts Institute of Technology: Cambridge, MA.

Paul, R. P.: 1981. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press: Cambridge, MA.