N90-23033

An Inverse Kinematics Algorithm For A Highly Redundant Variable-Geometry-Truss Manipulator

Frank Naccarato
Peter Hughes

University of Toronto Institute for Aerospace Studies
4925 Dufferin St., Downview, Ontario, Canada M3H 5T6

Abstract

A new class of robotic arm consists of a periodic sequence of truss substructures, each of which has several variable-length members. Such *variable-geometry-truss manipulators* (VGTMs) are inherently highly redundant and promise a significant increase in dexterity over conventional anthropomorphic manipulators. This dexterity may be exploited for both obstacle avoidance and controlled deployment in complex workspaces. The inverse kinematics problem for such unorthodox manipulators, however, becomes complex because of the large number of degrees of freedom, and conventional solutions to the inverse kinematics problem become inefficient because of the high degree of redundancy. This paper presents a solution to this problem based on a spline-like reference curve for the manipulator's shape. Such an approach has a number of advantages: (a) direct, intuitive manipulation of shape; (b) reduced calculation time; and (c) direct control over the effective degree of redundancy of the manipulator. Furthermore, although the algorithm has been developed primarily for variable-geometry-truss manipulators, it is general enough for application to a number of manipulator designs.

1 Introduction

A new class of robotic arm consists of a periodic sequence of truss substructures containing variable-length members. *Variable-geometry-truss manipulators* (VGTMs) have emerged from various designs for deployable/controllable trusses for space-based applications [10,12,16]. An example of a VGTM is illustrated in Figure 1. The manipulator is a statically determinate truss comprised of linear members hinged together at joints. Some members are actuated and can change their length; control of these lengths determines the manipulator's shape as well as the position and orientation of the end effector.

Truss manipulators promise a number of advantages over conventional anthropomorphic robot arms and space cranes, not the least of which is a significant increase in dexterity. This increased dexterity, however, has a price: a complex and unorthodox kinematic description. This paper will outline a kinematic methodology for VGTMs and set up a solution to the general

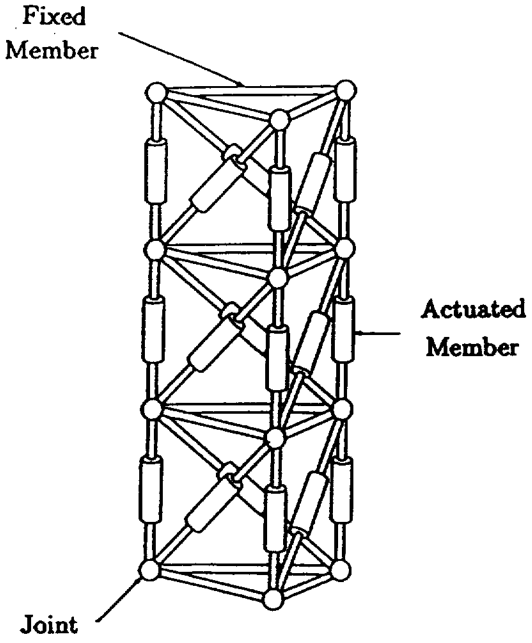

Figure 1: Example of a VGTM

inverse kinematics problem. Furthermore, a new inverse kinematics algorithm — based on spline-like reference shape curves — will be presented and compared with the more conventional solution.

2 Background to the Inverse Kinematics Problem

The purpose of any robotic arm is to manipulate an end effector along a desired trajectory; this motion depends on the combined action of the manipulator's actuators. The task of determining the correct actuator motion for a given end effector trajectory is the *inverse kinematics problem.*

2.1 Direct Kinematics

The end effector is defined by a set of n_e coordinates, $\mathbf{x}_e(t)$. This vector may contain up to six elements: three position and three orientation coordinates. The manipulator's *configuration* is defined by the vector $\mathbf{q}(t)$ containing n_q elements representing the manipulator's internal degrees of freedom. In general, we may state the *direct kinematics* relationship in the form

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) \quad (1)$$

$$n_e \leq n_q \quad (2)$$

The inequality represents the possibility of *redundancy.*

2.2 Differential Kinematics

In general, the direct kinematics relationship (1) cannot be inverted directly to give the inverse kinematics solution. Instead, we resort to *differential kinematics* [1,15], whereby we arrive at a

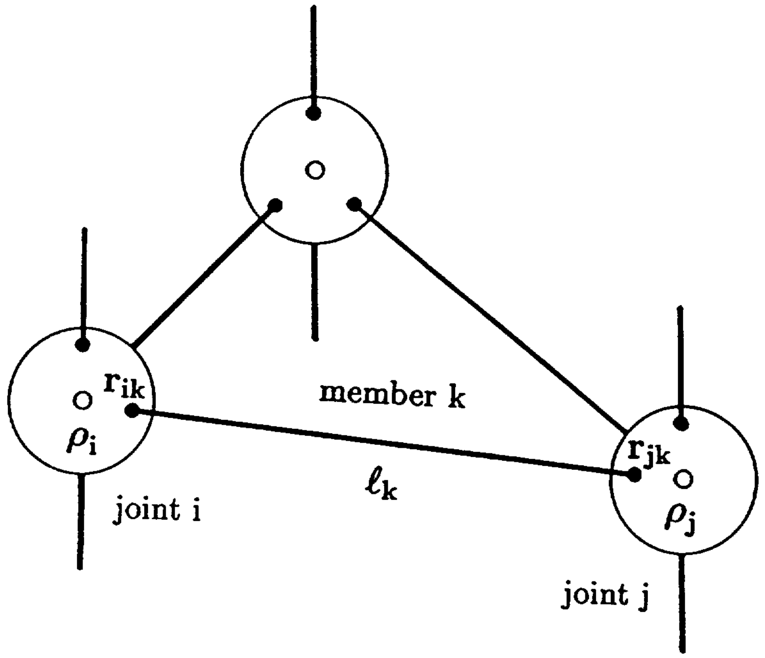

Figure 2: Geometric Representation of a VGTM

rate-linear system by expanding the time derivative of $\mathbf{x}_e(t)$:

$$\dot{\mathbf{x}}_e(t) = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}(t) \tag{3}$$

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}^T} \tag{4}$$

The *Jacobian matrix* $\mathbf{J}$ has dimensions $n_e \times n_q$.

2.3 Inverse Kinematics Solution with Redundancy

For a redundant manipulator, the Jacobian is not square and a solution to (3) is not straightforward. A common formulation of a solution takes the form [6,7]

$$\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{x}}_e + (\mathbf{1} - \mathbf{J}^+\mathbf{J})\dot{\mathbf{q}}_s \tag{5}$$

$$\mathbf{J}\mathbf{J}^+ = \mathbf{1} \tag{6}$$

The *Moore-Penrose pseudoinverse* [13], $\mathbf{J}^+$, represents a least-squares solution to (3); alternative solutions have used weighted pseudoinverses [5]. The vector $\dot{\mathbf{q}}_s$ is an arbitrary configuration velocity that may be used to minimize a cost function, avoid obstacles or fulfil some other objective [3,8,11]. The operator $(\mathbf{1} - \mathbf{J}^+\mathbf{J})$ projects this velocity onto the null space of $\mathbf{J}$ so that the desired motion of the end effector is not affected.

3 Kinematic Description of VGTMs

Generalized techniques exist to develop kinematic descriptions for conventional anthropomorphic manipulators [2]. These techniques, however, are not well suited to truss manipulators. In this section, a general kinematic methodology for VGTMs will be developed.

3.1 VGTM Geometry

Figure 2 shows a geometric representation of a truss manipulator. The linear members of the truss (actuated and nonactuated) are represented by straight lines having lengths ℓ_k. Embedded in the joint mechanisms are the member *endpoints*, represented by the position vectors $\mathbf{r}_{ik}$. Associated with each joint — or truss node — is a ***node coordinate vector***, $\boldsymbol{\rho}_i$; each node vector represents a fixed point within the joint mechanism that defines the joint's *position* in space. Since the truss structures are assumed to be statically determinate, we may calculate the *orientation* of each joint given the set of node vectors $[\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \cdots, \boldsymbol{\rho}_N]$, where N is the number of joints in the truss.

3.2 Configuration Variables

For conventional robotic arms, the configuration variables are identical to the actuator variables, i.e., revolute joint angles and prismatic joint lengths. For VGTMs, however, the actuator variables — the nonfixed member lengths — are unsuitable when writing the direct kinematics; an intermediate set of variables must be used. The most appropriate choice for configuration variables are the *node coordinate vectors* $\boldsymbol{\rho}_i$, $(i = 1, 2, \ldots, N)$. These variables have the following properties that make them suitable:

1. Relationships may be found between the node vectors and useful external parameters. For instance, we may be interested in the position and orientation of a particular triangular face of the truss (where an instrument platform may be attached); these coordinates may be found if the node vectors describing the vertices of the triangle are known.

2. Since the truss is statically determinate, we may calculate the endpoint vectors $\mathbf{r}_{ik}$ given $\boldsymbol{\rho}_i$, $(i = 1, 2, \cdots, N)$; once the endpoints are known, we may determine the length of each actuator. Hence, the actuator coordinates for the truss may be arrived at through this set of configuration variables.

3.3 Kinematic Relationships

There are three kinematic relationships by which we may determine the node coordinate vectors:

Explicit External Constraints. This kinematic relationship involves node vectors that are known, explicit functions of time; these functions are defined independently of the truss configuration and hence are 'external.' If there are N_c such constrained nodes, we may group them into the vector $\boldsymbol{\rho}_c$ such that

$$\begin{aligned} \boldsymbol{\rho}_c &= \mathrm{col}[\boldsymbol{\rho}_i(t)] \qquad (7) \\ i &= 1, 2, \ldots, N_c \qquad (8) \end{aligned}$$

Typically, these form the base nodes of a VGTM. For completeness, we also group the unconstrained node vectors into $\boldsymbol{\rho}$:

$$\begin{aligned} \boldsymbol{\rho} &= \mathrm{col}[\boldsymbol{\rho}_i] \qquad (9) \\ i &= N_c + 1,\ N_c + 2, \ldots, N \qquad (10) \end{aligned}$$

Implicit External Constraints. This type of constraint describes the relationship of truss nodes to a set of external coordinates — such as the end effector coordinates $\mathbf{x}_e$ — and takes the form

$$\mathbf{f}(\boldsymbol{\rho}, \boldsymbol{\rho}_c) = \mathbf{x}_e(t) \tag{11}$$

where $\mathbf{f}(\boldsymbol{\rho}, \boldsymbol{\rho}_c)$ is a set of functions in the node coordinate vectors.

Internal Constraints. The n_l nonactuated members of a truss manipulator serve as hard constraints to the motion of the nodes. These internal constraints take the implicit form

$$\begin{aligned} (\mathbf{r}_{ik} - \mathbf{r}_{jk})^T(\mathbf{r}_{ik} - \mathbf{r}_{jk}) &= \ell_k^2 & (12)\\ \mathbf{r}_{ik} &= \mathbf{r}_{ik}(\boldsymbol{\rho}, \boldsymbol{\rho}_c) & (13)\\ k &= 1, 2, \ldots, n_l & (14)\\ i, j &= 1, 2, \ldots, N & (15) \end{aligned}$$

The length of fixed member k is expressed in terms of its *endpoint vectors*, $\mathbf{r}_{ik}$. If the truss nodes are perfect pin joints, the endpoint vectors become identical to the node vectors.

3.4 Inverse Kinematics

A unique solution for $\boldsymbol{\rho}$ may be impossible to determine because of redundancy. To resolve this redundancy, we first find rate-linear expressions based on the constraint equations. From the implicit external constraints (11), we form

$$\mathbf{J}_u(\boldsymbol{\rho}, \boldsymbol{\rho}_c)\dot{\boldsymbol{\rho}} + \mathbf{J}_c(\boldsymbol{\rho}, \boldsymbol{\rho}_c)\dot{\boldsymbol{\rho}}_c = \dot{\mathbf{x}}_e \tag{16}$$

where

$$\begin{aligned} \mathbf{J}_u &= \frac{\partial \mathbf{f}}{\partial \boldsymbol{\rho}^T} & (17)\\ \mathbf{J}_c &= \frac{\partial \mathbf{f}}{\partial \boldsymbol{\rho}_c^T} & (18) \end{aligned}$$

From the internal constraints we get a set of n_l equations, $(k = 1, 2, \ldots, n_l)$, that take the form

$$(\mathbf{r}_{ik} - \mathbf{r}_{jk})^T \left[(\mathbf{J}_{ik,u} - \mathbf{J}_{jk,u})\dot{\boldsymbol{\rho}} + (\mathbf{J}_{ik,c} - \mathbf{J}_{jk,c})\dot{\boldsymbol{\rho}}_c\right] = 0 \tag{19}$$

$$\begin{aligned} \mathbf{J}_{ik,u} &= \frac{\partial \mathbf{r}_{ik}}{\partial \boldsymbol{\rho}^T} & (20)\\ \mathbf{J}_{ik,c} &= \frac{\partial \mathbf{r}_{ik}}{\partial \boldsymbol{\rho}_c^T} & (21) \end{aligned}$$

(N.B. If the nodes are ideal pin joints, the nonzero portions of these Jacobians reduce to identity matrices, and we are left with a much simpler set of expressions [18]). We may group these n_l equations into a system:

$$\mathbf{J}_g(\boldsymbol{\rho}, \boldsymbol{\rho}_c)\dot{\boldsymbol{\rho}} = \dot{\mathbf{x}}_g(\boldsymbol{\rho}, \boldsymbol{\rho}_c) \tag{22}$$

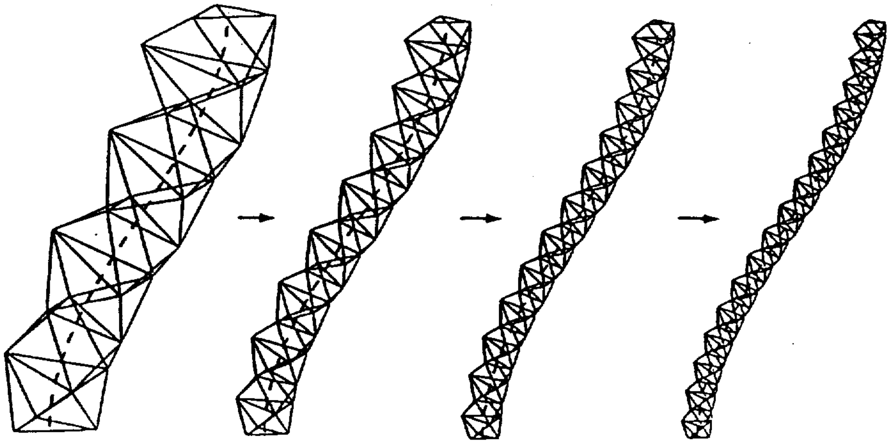

Figure 3: VGTMs Approach Continuum

where

$$\dot{\mathbf{x}}_g(\rho,\rho_c) = \mathrm{col}[-(\mathbf{r}_{ik}-\mathbf{r}_{jk})^T(\mathbf{J}_{ik,c}-\mathbf{J}_{jk,c})\dot{\rho}_c]_k \tag{23}$$

$$\mathbf{J}_g(\rho,\rho_c) = \mathrm{col}[(\mathbf{r}_{ik}-\mathbf{r}_{jk})^T(\mathbf{J}_{ik,u}-\mathbf{J}_{jk,u})]_k \tag{24}$$

Combining (16) and (22), and defining

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\mathbf{x}}_e - \mathbf{J}_c\dot{\rho}_c \\ \dot{\mathbf{x}}_g \end{bmatrix} \tag{25}$$

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_u \\ \mathbf{J}_g \end{bmatrix} \tag{26}$$

we arrive at the following system:

$$\mathbf{J}(\rho,\rho_c)\dot{\rho} = \dot{\boldsymbol{x}} \tag{27}$$

The inverse kinematic solution of (27) takes the conventional form

$$\dot{\rho} = \mathbf{J}^+\dot{\boldsymbol{x}} + (1-\mathbf{J}^+\mathbf{J})\dot{\rho}_s \tag{28}$$

4 Inverse Kinematics Using Reference Shape Curves

In the previous section, a solution to the inverse kinematics problem for VGTMs was found using the conventional methods outlined in §2. An alternative solution will now be presented, and a comparison will be made to the previous solution to illustrate some advantages associated with the new technique.

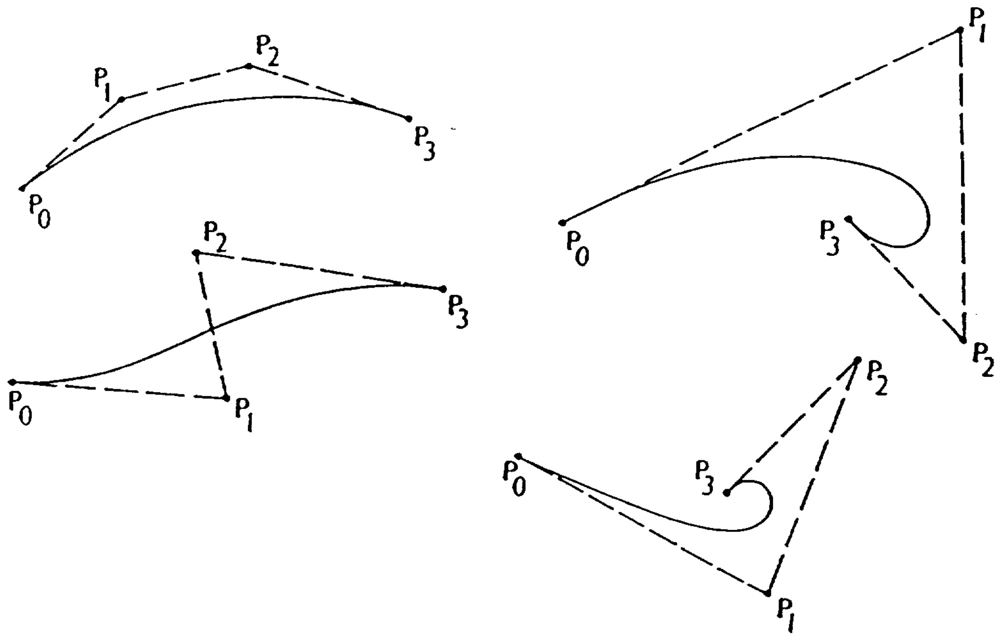

Figure 4: Space curves (Bezier)

4.1 Basic Concept

A distinctive feature of the VGTM design is its ability to assume curvilinear shapes [9]. Furthermore, as illustrated in figure 3, the shape of a truss manipulator may be made to conform *exactly* to an arbitrary space curve in the limit as the number of bays gets large. It is this serpentine property of VGTMs that has inspired an inverse kinematics technique based on the modelling of the manipulator's shape using *reference shape curves.*

The proposed solution algorithm is as follows:

1. When solving the inverse kinematics problem, *replace* the manipulator by a continuous space curve that satisfies the same external constraints.

2. Force the manipulator to *track* this new desired shape.

The reference shape curve may be thought of as the next dimensional extension of the end effector coordinates, $\mathbf{x}_e$: while the latter is defined at a point, the reference curve is a continuous coordinate *distribution* over a line. The second-order case will be a reference shape *surface* that may be used to model plate-like variable geometry trusses.

The primary reasons for pursuing such a technique are

1. A simplified, analytic expression for the manipulator's shape will help in modelling interactions with the robot's environment (e.g., collision detection, obstacle avoidance, controlled deployment).

2. Less complex kinematic relations, resulting from the simplified shape model, will improve the computational efficiency of the inverse kinematics solution.

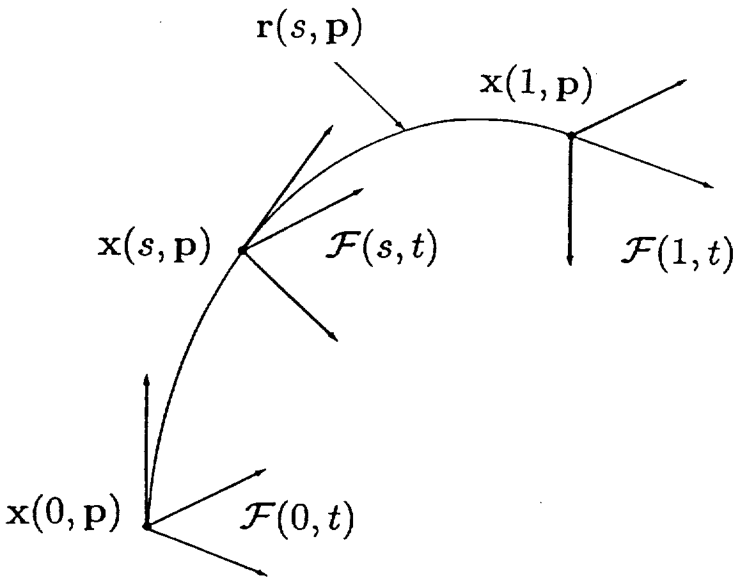

Figure 5: Coordinate Distribution Associated with Space Curve

4.2 Type of Curve

A certain class of space curve, found in computer graphics applications [17], has a number of properties that are desirable for this application. These spline-like curves — parameterized by s $(0 \leq s \leq 1)$ — have the form

$$\mathbf{r}(s,t) = \sum_{i=0}^{N} w_i(s)\mathbf{p}_i(t) \tag{29}$$

$$1 = \sum_{i=0}^{N} w_i(s) \tag{30}$$

$$\tag{31}$$

Example of such curves are ***Bezier curves, B-splines and Beta-splines.*** Because the ***weighting functions*** $w_i(s)$ are a partition of unity, a point on the curve $\mathbf{r}$ is a weighted average of the ***control vertices*** $\mathbf{p}_i$. Joining the control verticies sequentially forms an open ***control polygon*** that approximates the actual shape of the curve; closeness of fit between the control polygon and the curve depends on the form of the weighting functions. Figure 4 illustrates this relationship for a Bezier curve.

4.3 Coordinate Distribution

Associated with the curve is a distribution of position and orientation coordinates, $\mathbf{x}(s,\mathbf{p})$:

$$\mathbf{x}(s,\mathbf{p}) = \begin{bmatrix} \mathbf{r}(s,\mathbf{p}) \\ \theta(s,\mathbf{p}) \end{bmatrix} \tag{32}$$

$$\mathbf{p} = \mathrm{col}\,[\mathbf{p}_i] \tag{33}$$

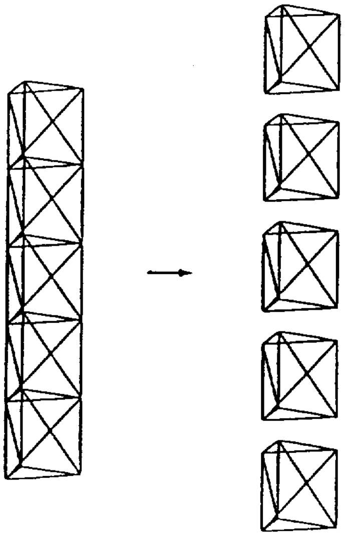

Figure 6: Partitioning of VGTM

Figure 5 shows that $\mathbf{x}(s,\mathbf{p})$ describes a frame of reference $\mathcal{F}(s,t)$ that has two properties:

1. Its origin intersects the curve at $\mathbf{r}(s,\mathbf{p})$.
2. One of its axes is tangent to the curve at the origin.

Since the tangent is a function of $\mathbf{p}$, two of the three attitude coordinates are explicit functions of the control vertices. The third attitude coordinate — representing a torsion about the tangent — may be specified independently of the control vertices, and hence depends on the parameter s alone.

4.4 Kinematics of a Curve

The relationship between the control verticies and the overall curve shape allows direct, intuitive manipulation of the curve. Hence, the control vertices $\mathbf{p}$ will be defined as the configuraton variables for the curve. The direct kinematics relation follows from the fact that the coordinate distribution, $\mathbf{x}$, is an extension of the original end effector coordinates, i.e.,

$$\mathbf{x}_e(t) = \mathbf{x}(1,t) \tag{34}$$

The rate-linear kinematic expression for any point on the curve is

$$\mathbf{J}_s(s,\mathbf{p})\,\dot{\mathbf{p}} = \dot{\mathbf{x}}(s,t) \tag{35}$$

where $\mathbf{J}_s$ — the *curve Jacobian* — is defined as

$$\mathbf{J}_s(s,\mathbf{p}) = \begin{bmatrix} \frac{\partial \mathbf{r}(s,\mathbf{p})}{\partial \mathbf{p}^T} \\ \frac{\partial \boldsymbol{\theta}(s,\mathbf{p})}{\partial \mathbf{p}^T} \end{bmatrix} \tag{36}$$

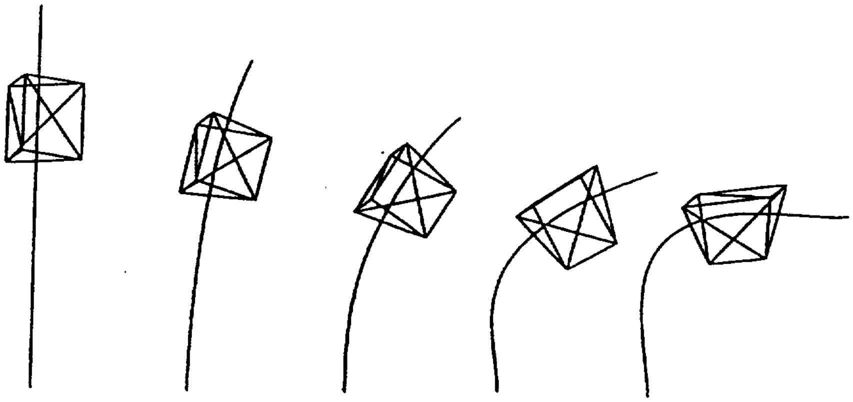

Figure 7: Tracking the Curve

To impose the implicit external constraint expressed in (34), we set

$$\mathbf{J}_s(1,\mathbf{p})\,\dot{\mathbf{p}} = \dot{\mathbf{x}}_e(t) \tag{37}$$

Following (5), the solution for the curve kinematics becomes

$$\dot{\mathbf{p}} = \mathbf{J}_s^+\dot{\mathbf{x}}_e + (1 - \mathbf{J}_s^+\mathbf{J}_s)\dot{\mathbf{p}}_s \tag{38}$$

As was the case for the general manipulator in §2 and for a VGTM in §3, the vector $\dot{\mathbf{p}}_s$ may be formulated to achieve a variety of objectives including obstacle avoidance.

4.5 Tracking the Reference Curve

The second step in the algorithm is to force the manipulator to track the reference curve. To do this, the manipulator is *partitioned* into smaller VGTMs (see figure 6); each segment may be described kinematically in the same manner as the whole VGTM. The implicit external constraints $\mathbf{x}_i(t)$ corresponding to each segment are then read off the reference curve at discrete points — $[s_0, s_1, \ldots, s_N]$ — called *curve nodes*, i.e.

$$\mathbf{x}_i(t) = \mathbf{x}(s_i, \mathbf{p}) \tag{39}$$

The VGTM segments may be solved in one of two ways:

1. *Recursively*, whereby the implicit constraints for the kth bay are given by $\mathbf{x}(s_k, \mathbf{p})$, and the explicit constraints are provided by the $(k-1)$st bay, or,

2. *In parallel*, whereby each bay is solve independantly using only the implicit information provided by the curve.

Figure 7 shows a manipulator segment tracking a curve that is executing a maneuver.

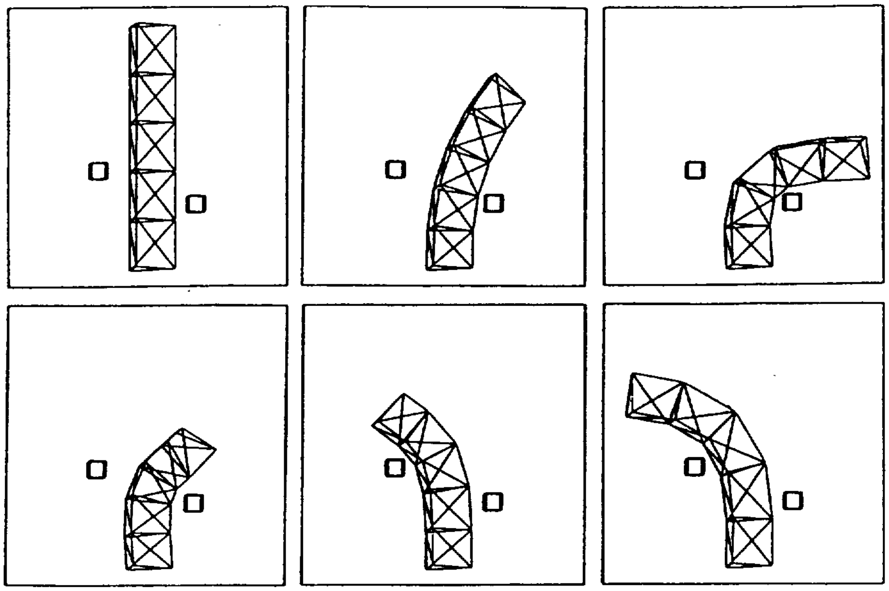

Figure 8: Example Maneuver

4.6 Example Maneuver

Figure 8 shows sample frames from a 50-step maneuver involving a five-bay VGTM. The arm is directed to avoid two cube-shaped obstacles. The inverse kinematics has been solved using the reference curve technique; clearly, the technique works adequately in that the desired end effector trajectory is tracked, and a safe clearance is maintained between the robot arm and the obstacles.

4.7 Advantages to the Reference Curve Technique

Some of the advantages of the reference curve technique are summarized below:

Improved Computational Efficiency To evaluate any improvement in computation time, a test trajectory was designed and *both* techniques — conventional and reference shape curve (recursive) — were used to solve the inverse kinematics for VGTMs of different lengths. The maneuver was partitioned into 50 time-steps, and the runs were carried out on an APOLLO^{TM} DN 4000 workstation. Figure 9 shows a comparison of the run-times for both methods; clearly, there is a marked improvement in computational efficiency when using the reference shape curve technique. This improvement may be attributed to the recursive nature of the tracking problem, as discussed in §4.5.

Parallel Structure of Problem As stated earlier, the tracking problem may be solved *in parallel* as well. This natural parallel structure makes the reference curve technique conducive to a *multiprocessing* environment [14]; in such an architecture, dedicated processors may be used to solve the inverse kinematics of the VGTM segments concurrently, thereby providing a quantum improvement in efficiency.

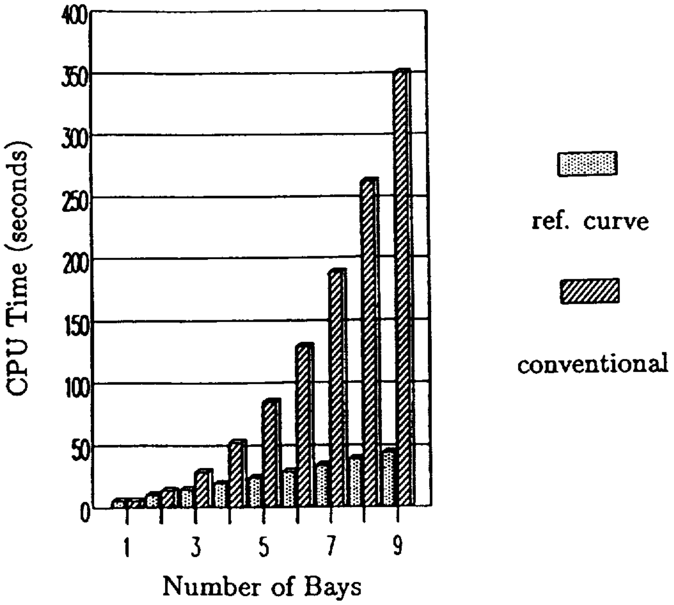

Figure 9: Run-time Comparison

Analytical Description of Manipulator Shape An analytical expression for the manipulator's shape is helpful when describing robot/environment interactions. An important part of many obstacle avoidance routines involves finding the closest point to the obstacle on the manipulator [4,5,11]. Using the shape curve equation and quickly-converging iterative solvers, we may readily solve for these critical points. An analytical description may also be incorporated into a mathematical model of the manipulator's workspace to predict collisions.

Variable Degree of Redundancy Since the global inverse kinematics problem has been shifted from the manipulator to the reference curve, we may specify the degree of redundancy *a priori* by choosing the number of control vertices used to describe the space curve. For instance, if it is known that the VGTM is to operate in a very simple, obstacle-free workspace, then only a minimum number of degrees of freedom — enough to satisfy the external constraints — need be included; conversely, a complex workspace will demand the use of more control vertices. There is a limit, of course: it will be impossible for the manipulator to track a shape curve that has more degrees of freedom than itself. Yet, this flexiblity in choosing the degree of redundancy may be used to reduce computational overhead.

4.8 Generality

While developed for application to truss manipulators, the reference curve method is general enough to find use as a redundancy resolution technique with more conventional robotic arms. The first step in the algorithm remains the same since the kinematics of the reference curve is independant of the manipulator. The second step — the tracking problem — may be applied to any robotic arm that can be easily segmented into smaller manipulators.

5 Conclusions

A generalized kinematic description has been presented for variable-geometry-truss manipulators. From this description, a solution to the inverse kinematics problem was found by conventional means. An alternative technique — based on reference shape curves — was developed and applied to VGTMs in the hopes of improving the efficiency of the straightforward approach, as well to inject some geometric insight in the description of the manipulator's shape. The new technique succeeded in solving the inverse kinematics problem with a significantly decreased computation time. Further advantages were noted:

- Technique is attractive to obstacle avoidance and collision detection applications;
- Problem structure is applicable to parallel processing;
- Variable degree of redundancy
- Generality

The preliminary success of this new method is encouraging for the development of real-time-control robotic facilities based on truss manipulators.

6 Acknowledgements

The authors would like to acknowledge research support from The Natural Sciences and Engineering Research Council of Canada, and would like to thank the following for their help: V. Pugliese, members of the Space Dynamics and Control Group, and Dynacon Enterprises Ltd.

7 References

1. CRAIG, J. *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, 1986.

2. GOLDENBERG, A. ET AL. "A Complete Generalized Solution to the Inverse Kinematics of Robots" *IEEE Journal of Robotics and Automation*, Vol. Ra, No.1, March 1985.

3. HOLLERBACH, J.M. & SUH, K.C. "Redundancy Resolution of Manipulators Through Torque Optimization," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, August 1987.

4. KHATIB, O. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotics Research*, Vol. 5, No. 1, Spring 1986.

5. KIRCANSKI, M. & VUKOBRATOVIC, M. "Contribution to Control of Redundant Robotic Manipulators in an Environment with Obstacles," *Intl. Journal of Robotics Research*, Vol.5, No. 4, Winter 1986.

6. KLEIN, C.A. & HUANG, C.-H. "Review of Pseudoinverse Control for Use With Kinematically Redundant Manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No. 3, March/April 1983.

7. LIEGEOIS, A. "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanics," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-7, No. 12, Dec. 1977.

8. MACIEJEWSKI, A.A. & KLEIN, C.A. "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotics Research*, Vol. 4, No. 3, Fall 1985.

9. MIURA, K. & FURUYA, H. "Variable Geometry Truss and Its Application to Deployable Truss and Space Crane Arms," *35th Congress of the International Astronautical Federation*, Lausanne, Switzerland, Oct. 7-13, 1984 [IAF-84-394].

10. MIURA, K. & FURUYA, H. "An Adaptive Structure Concept for Future Space Applications," *36th Congress of the International Astronautical Federation*, Stockholm, Sweden, Oct. 7-12, 1985 [IAF-85-211].

11. NAKAMURA, Y. & HANAFUSA, H. "Optimal Redundancy Control of Robot Manipulators," *International Journal of Robotics Research*, Vol. 6, No. 1, Spring 1987.

12. NAYFEH, A.H. "Kinematics of Foldable Discrete Space Cranes," *NASA CR176360*, 1985.

13. NOBLE, B. & DANIEL, J.W. *Applied Linear Algebra*, 2nd Ed., Prentice-Hall, New Jersey 1977.

14. OSTERHAUG, A. *Guide to Parallel Programming*, Sequent Computer Systems, Oregon, 1987.

15. PAUL, R. *Robot Manipulators*, MIT Press, Cambridge, 1981.

16. RHODES, M.D. & MIKULAS, M.M. "Deployable Controllable Geometry Truss Beam," *NASA TM-86366*, 1985.

17. ROGERS, D.F. & ADAMS, J.A. *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York, 1976.

18. SINCARSIN, W.G. & HUGHES, P.C. *Trussarm: Candidate Geometries*, Dynacon Enterprises Ltd., Toronto, 1987.