THE CATHOLIC UNIVERSITY OF AMERICA
DEPARTMENT OF ELECTRICAL ENGINEERING

Semiannual Progress Report

EXPERIMENTAL STUDY OF TRAJECTORY PLANNING AND CONTROL OF A HIGH PRECISION ROBOT MANIPULATOR

Grant Number NAG 5-780

Charles C. Nguyen
Principal Investigator and Associate Professor
and
Sami S. Antrazi
Graduate Research Assistant

submitted to
Dr. Charles E. Campbell, Jr.
Code 714.1
Goddard Space Flight Center (NASA)
Greenbelt, Maryland

August 1991

1 Introduction

After its introduction as an aircraft simulator, the Stewart Platform [1] has been employed in the design of various robot manipulators, robotic end-effectors and robotic devices [2,27] for high precision assembly tasks where the requirements of accuracy and sturdiness are more essential than these of large workspace and maneuverability. A Stewart Platform-based manipulator consists mainly of two platforms coupled together by six parallel linear actuators driven by electrical drives such as servomotors or fluid power drives such as hydraulic or pneumatic systems. The motion of one platform with respect to the other can be produced by shortening or extending the actuator lengths. Conventional robot manipulators are anthropomorphic open-kinematic chain (**OKC**) mechanisms whose joints and links are actuated in series. OKC manipulators generally have long reach, large workspace and are capable of entering small spaces because of their compactness. However they have low stiffness and undesired dynamic characteristics, especially at high speed and large payload mainly due to the cantilever-like structure. Moreover, the nonuniform distribution of payload to actuators causes OKC manipulators to have low strength-to-weight ratios. Finally serial accumulation of joint errors throughout the OKC mechanism results in relatively large position error on the last link of the manipulator and suggests that OKC manipulators are not suitable for high precision tasks. On the other hand, Stewart Platform-based manipulators whose mechanism are parallel, have been proven to be capable of high precision positioning due to their high structural rigidity and non-serial accumulation of joint errors.

One of the first Stewart Platform applications is reported in [2], where an aircraft simulator was built at NASA Langley Research Center to train operators and kinematic transformations were developed for the motion control of the simulator. A finite element program was employed in [3] to simulate the motion of the Stewart Platform whose mechanism was later applied in [4] to design an automatic assembly table. A systematic study of in-parallel-actuated robot manipulators was conducted in [5] and the structural kinematic problem this type of manipulators was presented in [7]. A general technique was obtained in [6] to describe the instantaneous link motion of a single closed-loop mechanism by applying linear algebra elements to screw systems. In [8], the Stewart Platform mechanism was applied to construct a passive compliant robot end-effector which served as a testbed for studying autonomous part assembly. The problem of active compliance control of this end-effector was later investigated in [9]. The kinematic problems and practical construction of the Stewart Platform were considered in [10,11] and in [12], respectively. Kinematics and dynamics of parallel manipulators were studied in [13] and dynamical equations for a 3 degree-of-freedom (**DOF**) parallel manipulator were derived in [14]. Static force analysis using screw theory was applied in [15] to treat Stewart Platform-based manipulators. The Stewart Platform mechanism was applied to design a micromanipulator [17] performing fine motion, and to manufacture a force/torque sensor [19]. Other problems of Stewart Platform-based manipulators such as adaptive force/torque control, kinematics, dynamics and workspace computation were treated in [20,23]. Closed-form solutions of forward kinematics were derived in [21] and [24] for a class of Stewart Platforms. Learning control scheme and trajectory planning schemes were developed in [25] and [26], respectively for parallel manipulators. Experimental results obtained for a Stewart Platform manipulator were reported in [27].

This report deals with the trajectory planning and motion control of a Stewart Platform-based robotic end-effector built to study robotic assembly of part in space. This report is

organized as follows. The next section presents the main components of the end-effector and describes briefly their operations. Forward and inverse kinematic transformations are then developed for the end-effector. After that, the development of three trajectory planning schemes, two for fine motion and one for gross motion, are presented. Results of experiments conducted to evaluate the performance of the developed trajectory planning schemes in tracking several test paths are finally presented and discussed.

2 The Passive Compliant End-Effector

This section is devoted to briefly describe the main components of the end-effector. As illustrated in Figures 1-2, the end-effector, a prototype whose size is about ten times the size that of an end-effector proposed in [22,23] for in-space assembly of NASA hardwares, mainly consists of a lower base platform, an upper payload platform, a compliant platform, a gripper and six linear actuators. The movable payload platform is supported above the stationary base platform by six axially extensible rods with ballnuts and ballscrews providing the extensibility. The ballscrews are driven by stepper motors to extend or shorten the actuator *legs* whose variations will in turn create the motion of the payload platform with respect to the base platform. Each end of the actuator links is mounted to the platforms by 2 rotary joints whose axes intersect and are perpendicular to each other. As seen in Figure 2, passive compliance is provided through the compliant platform, which is suspended from the payload platform by six spring-loaded pistons arranged in a geometry similar to the Stewart Platform mechanism, by permitting strain on two opposing springs acting in the pistons. Thus the pistons are compressed and extended when resistive and gravitational forces are applied on the gripper. The rotation of each stepper motor is controlled by sending out proper commands to an indexer which then transmits proper pulse sequences to the stepper motor drive. Therefore, the motion of the gripper attached to the compliant platform can be precisely produced by properly controlling the motions of the six end-effector legs. The planning and control scheme employed to control the motion of the end-effector gripper is presented in Figure 3. A Cartesian trajectory planning scheme converts a desired Cartesian path specified by desired starting and ending velocities and accelerations is converted into 6 Cartesian trajectories. Then based upon the desired Cartesian trajectories, joint-space trajectories will be determined by a joint-space planner which sends proper commands through the RS232 port of a personal computer to the indexers. The indexer will then transmit pulses to the stepper motor drives where microstepping permits each revolution (360^0) of the stepper motor to be equivalent to 25,000 steps. The drive rotates the stepper motor one angular increment of $\frac{360^0}{25,000} = 0.0144^o$, each time it receives one step pulse. Furthermore, through the linear motion converter system consisting of the ballnut and the ballscrew, each angular increment (=1step) is converted into 8 μ-inches of linear translation of the end-effector leg. Consequently a revolution corresponds to 0.2 inch of linear translation.

3 Kinematic Transformations

This section presents the development of the kinematic transformations for the end-effector. First using vector analysis, a closed-form solution for the end-effector inverse kinematic transformation is obtained. Then Newton Raphson iterative method will be employed to obtain a numerical solution for the end-effector forward kinematic transformation.

3.1 Inverse Kinematic Transformation

The inverse kinematic transformation deals with the determination of the required actuator lengths for a given pose[1] of the payload platform with respect to the base platform. As seen in Figure 4, two coordinate frames {**P**}, and {**B**} are assigned to the payload and base platforms, respectively. The origin of Frame {**P**} is located at the centroid P of the payload platform, the $\mathbf{z}_P$-axis is pointing outward and the $\mathbf{x}_P$-axis is perpendicular to the line connecting the two attachment points P_1 and P_6. The angle between P_1 and P_2 is denoted by θ_P. A symmetrical distribution of joints on the payload platform is achieved by setting the angles between P_1 and P_3 and between P_3 and P_5 to 120°. Similarly, Frame {**B**} has its origin at the centroid B of the base platform. The $\mathbf{x}_B$-axis is perpendicular to the line connecting the two attachment points B_1 and B_6 the angle between B_1 and B_2 is denoted by θ_B. Also the angles between B_1 and B_3 and between B_3 and B_5 are set to 120° in order to symmetrically distribute the joints on the base platform. The Cartesian variables are chosen to be the relative position and orientation of Frame {**P**} with respect to Frame {**B**} where the position of Frame {**P**} is specified by the position of its origin with respect to Frame {**B**}. Now if we denote the angle between PP_i and $\mathbf{x}_P$ by λ_i, and the angle between BB_i and $\mathbf{x}_B$ by Λ_i for i=1,2,...,6, then by inspection we obtain

$$\Lambda_i = 60i - \frac{\theta_B}{2};\ \lambda_i = 60i - \frac{\theta_P}{2},\ \text{for i} = 1,3,5 \tag{1}$$

and

$$\Lambda_i = \Lambda_{i-1} + \theta_B;\ \lambda_i = \lambda_{i-1} + \theta_P,\ \text{for i} = 2,4,6 \tag{2}$$

where all angles are expressed in degrees (°).

Furthermore, if Vector ${}^P\mathbf{p}_i = (p_{ix}\ p_{iy}\ p_{iz})^T$ describes the position of the attachment point P_i with respect to Frame {**P**}, and Vector ${}^B\mathbf{b}_i = (b_{ix}\ b_{iy}\ b_{iz})^T$ the position of the attachment point B_i with respect to Frame {**B**}, then they can be written as

$${}^P\mathbf{p}_i = \begin{bmatrix} r_P cos(\lambda_i) & r_P sin(\lambda_i) & 0 \end{bmatrix}^T \tag{3}$$

and

$${}^B\mathbf{b}_i = \begin{bmatrix} r_B cos(\Lambda_i) & r_B sin(\Lambda_i) & 0 \end{bmatrix}^T \tag{4}$$

for i=1,2,...,6 where r_P and r_B represent the radii of the payload and base platforms, respectively.

We proceed to consider the vector diagram for an ith actuator given in Figure 5. The position of Frame {**P**} is represented by Vector ${}^B\mathbf{d} = [x\ y\ z]^T$ which contains the Cartesian coordinates x, y, z of the origin of Frame {**P**} with respect to Frame {**B**}. The length vector ${}^B\mathbf{q}_i = (q_{ix}\ q_{iy}\ q_{iz})^T$, expressed with respect to Frame {**B**} can be computed by

$${}^B\mathbf{q}_i = {}^B\mathbf{x}_i + {}^B\mathbf{p}_i \tag{5}$$

where

$${}^B\mathbf{x}_i = {}^B\mathbf{d} - {}^B\mathbf{b}_i \tag{6}$$

[1] In this report *pose* means *position and orientation.*

$$= \begin{bmatrix} x - b_{ix} \\ y - b_{iy} \\ z - b_{iz} \end{bmatrix} = \begin{bmatrix} x - b_{ix} \\ y - b_{iy} \\ z \end{bmatrix} = \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \\ \bar{z}_i \end{bmatrix} \tag{7}$$

which is a shifted vector of ${}^B\mathbf{d}$ and

$${}^B\mathbf{p}_i = {}^B_P\mathbf{R}\ {}^P\mathbf{p}_i \tag{8}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{bmatrix} = \begin{bmatrix} r_{11}p_{ix} + r_{12}p_{iy} \\ r_{21}p_{ix} + r_{22}p_{iy} \\ r_{31}p_{ix} + r_{32}p_{iy} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} \tag{9}$$

which is the representation of ${}^B\mathbf{p}_i$ in Frame $\{\mathbf{B}\}$ and ${}^B_P\mathbf{R}$ is the *Orientation Matrix* representing the orientation of Frame $\{\mathbf{P}\}$ with respect to Frame $\{\mathbf{B}\}$.

Thus the length l_i of Vector ${}^B\mathbf{q}_i$ can be computed from its components as

$$l_i = \sqrt{q_{ix}^2 + q_{iy}^2 + q_{iz}^2}. \tag{10}$$

or

$$l_i = \sqrt{(\bar{x}_i + u_i)^2 + (\bar{y}_i + v_i)^2 + (\bar{z}_i + w_i)^2} \tag{11}$$

We obtain from (3)-(4)

$$p_{ix}^2 + p_{iy}^2 + p_{iz}^2 = r_P^2, \tag{12}$$

$$b_{ix}^2 + b_{iy}^2 + b_{iz}^2 = r_B^2. \tag{13}$$

and from the properties of orientation matrix

$$r_{11}^2 + r_{21}^2 + r_{31}^2 = r_{12}^2 + r_{22}^2 + r_{32}^2 = r_{13}^2 + r_{23}^2 + r_{33}^2 = 1 \tag{14}$$

and

$$\begin{aligned} r_{11}r_{12} + r_{21}r_{22} + r_{31}r_{32} &= 0 \\ r_{11}r_{13} + r_{21}r_{23} + r_{31}r_{33} &= 0 \\ r_{11}r_{13} + r_{22}r_{23} + r_{32}r_{32} &= 0. \end{aligned} \tag{15}$$

Employing (12)-(15), (10) can be rewritten as

$$\begin{aligned} l_i^2 \;=\; & x^2 + y^2 + z^2 + r_P^2 + r_B^2 + 2(r_{11}p_{ix} + r_{12}p_{iy})(x - b_{ix}) \\ & +2(r_{21}p_{ix} + r_{22}p_{iy})(y - b_{iy}) + 2(r_{31}p_{ix} + r_{32}p_{iy})z - 2(xb_{ix} + yb_{iy}), \end{aligned} \tag{16}$$

for i=1,2,...,6.

Equation (16) represents a *closed-form* solution to the inverse kinematic problem in the sense that required actuator lengths l_i for i=1,2,...,6 can be determined using (16) to yield a given Cartesian configuration composed of Cartesian position and orientation of Frame $\{\mathbf{P}\}$ with respect to Frame $\{\mathbf{B}\}$.

The orientation of Frame $\{\mathbf{P}\}$ with respect to Frame $\{\mathbf{B}\}$ can be described by the orientation matrix ${}^B_P\mathbf{R}$ as shown in (9) which requires nine variables r_{ij} for i,j=1,2,3 from which six are redundant because only three are needed to specify an orientation [29]. There exist several ways to specify an orientation by three variables, but the most widely used one is the Roll-Pitch-Yaw angles α, β, and γ, which represent the orientation of Frame $\{\mathbf{P}\}$, obtained after the following sequence of rotations from Frame $\{\mathbf{B}\}$:

1. First rotate Frame {B} about the $\mathbf{x}_B$-axis an angle γ (*Yaw*)

2. Then rotate the resulting frame about the $\mathbf{y}_B$-axis an angle β (*Pitch*)

3. Finally rotate the resulting frame about the $\mathbf{z}_B$-axis an angle α (*Roll*).

The orientation represented by the above Roll-Pitch-Yaw angles is given by[2]

$$ {}^B_P\mathbf{R} = \mathbf{R}_{RPY} = \begin{bmatrix} c\alpha\, c\beta & c\alpha\, s\beta\, s\gamma - s\alpha\, c\gamma & c\alpha\, s\beta\, c\gamma + s\alpha\, s\gamma \\ s\alpha\, c\beta & s\alpha\, s\beta\, s\gamma + c\alpha\, c\gamma & s\alpha\, s\beta\, c\gamma - c\alpha\, s\gamma \\ -s\beta & c\beta\, s\gamma & c\beta\, c\gamma \end{bmatrix}. \tag{17} $$

3.2 Forward Kinematic Transformation

This section considers the development of the forward transformation which transforms the actuator lengths l_i for i=1,2,...,6 into the pose of the payload platform with respect to the base platform. The forward kinematic problem can be formulated as to find a Cartesian position specified by x, y, z and an orientation specified by Roll-Pitch-Yaw angles α, β, and γ to satisfy Equation (16) for a given set of actuator lengths l_i for i=1,2,...,6. In general, there exists no closed-form solution for the above problem since Equation (16) represents a set of 6 highly nonlinear simultaneous equations with 6 unknowns. Consequently iterative numerical methods must be employed to solve the above set of nonlinear equations. In the following we will present the implementation of Newton-Raphson method for solving the forward kinematic problem.

In order to apply the Newton-Raphson method, first from (11) we define 6 scalar functions

$$ f_i(\mathbf{a}) = (\bar{x}_i + u_i)^2 + (\bar{y}_i + v_i)^2 + (\bar{z}_i + w_i)^2 - l_i{}^2 = 0 \tag{18} $$

for i=1,2,...,6, where the vector **a** is defined as

$$ \mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}^T = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^T, \tag{19} $$

and then employ the following algorithm [28] to solve for **a**:

Algorithm 1: Forward Kinematic Transformation

1. Select an initial guess **a**.

2. Compute the elements r_{ij} of ${}^B_P\mathbf{R}$ using (17) for i, j=1,2,...,6.

3. Compute $\bar{x}_i$,$\bar{y}_i$, $\bar{z}_i$, using (7) and u_i, v_i, w_i using (9) for i=1,2,...,6.

4. Compute $f_i(\mathbf{a})$ and $A_{ij} = \frac{\partial f_i}{\partial a_j}$ using (18) for i, j=1,2,...,6.

5. Compute B_i = -$f_i(\mathbf{a})$ for i=1,2,...,6. If $\sum_{j=1}^{6} \mid B_j \mid <$ *tolf* (tolerance), stop and select **a** as the solution.

6. Solve $\sum_{j=1}^{6} A_{ij}\delta a_j = B_i$ for δa_j for i,j=1,2,...,6 using LU decomposition. If $\sum_{j=1}^{6} \delta a_j <$ *tola* (tolerance), stop and select **a** as the solution.

[2] $c\alpha \equiv \cos\alpha$, and $s\alpha \equiv \sin\alpha$.

7. Select $\mathbf{a}^{new} = \mathbf{a} + \delta\mathbf{a}$ and repeat Steps 1–7.

In order to minimize the computational time of Algorithm 1, the expressions needed for computing the partial derivatives in Step 4 of the algorithm should be simplified. First using (9) and (17), the partial derivatives of u_i, v_i, and w_i with respect to the angles α, β, and γ can be computed as follows:

$$\frac{\partial u_i}{\partial \alpha} = -v_i; \quad \frac{\partial u_i}{\partial \beta} = c\alpha\, w_i; \quad \frac{\partial u_i}{\partial \gamma} = p_{iy}\, r_{13}, \tag{20}$$

$$\frac{\partial v_i}{\partial \alpha} = u_i; \quad \frac{\partial v_i}{\partial \beta} = s\alpha\, w_i; \quad \frac{\partial v_i}{\partial \gamma} = p_{iy}\, r_{23}, \tag{21}$$

$$\frac{\partial w_i}{\partial \alpha} = 0; \quad \frac{\partial w_i}{\partial \beta} = -(c\beta\, p_{ix} + s\beta\, s\gamma\, p_{iy}); \quad \frac{\partial w_i}{\partial \gamma} = p_{iy}\, r_{33}. \tag{22}$$

From (7), we note that

$$\frac{\partial \bar{x}_i}{\partial x} = \frac{\partial \bar{y}_i}{\partial y} = \frac{\partial \bar{z}_i}{\partial z} = 1. \tag{23}$$

Employing (20)-(23), we obtain after intensive simplification

$$\frac{\partial f_i}{\partial a_1} = \frac{\partial f_i}{\partial x} = \frac{\partial f_i}{\partial \bar{x}_i} = 2(\bar{x}_i + u_i), \tag{24}$$

$$\frac{\partial f_i}{\partial a_2} = \frac{\partial f_i}{\partial y} = \frac{\partial f_i}{\partial \bar{y}_i} = 2(\bar{y}_i + v_i), \tag{25}$$

$$\frac{\partial f_i}{\partial a_3} = \frac{\partial f_i}{\partial z} = \frac{\partial f_i}{\partial \bar{z}_i} = 2(\bar{z}_i + w_i), \tag{26}$$

$$\frac{\partial f_i}{\partial a_4} = \frac{\partial f_i}{\partial \alpha} = 2(-\bar{x}_i v_i + \bar{y}_i u_i), \tag{27}$$

$$\frac{\partial f_i}{\partial a_5} = \frac{\partial f_i}{\partial \beta} = 2[(-\bar{x}_i\, c\alpha + \bar{y}_i\, s\alpha)w_i - (p_{ix}\, c\beta + p_{iy}\, s\beta\, s\gamma)\bar{z}_i] \tag{28}$$

$$\frac{\partial f_i}{\partial a_6} = \frac{\partial f_i}{\partial \gamma} = 2p_{iy}(\bar{x}_i r_{13} + \bar{y}_i r_{23} + \bar{z}_i r_{33}). \tag{29}$$

4 Trajectory Planning Schemes

Two types of motion occur in an assembly task, *fine motion* and *gross motion*. While fine motion requires very high positioning tolerance, up to thousands of an inch, gross motion allows relatively low positioning tolerance, e.g. in obstacle avoidance. Three trajectory planning schemes developed to control the motion of the end-effector gripper are presented in this section. The first two schemes, one for tracking straight lines and the other for arbitrary paths, are intended for fine motion while the third scheme is developed for gross motion.

4.1 Trajectory Planning For Straight-Line Motion

The stepper motor indexer has two main modes of operation: the *normal mode* and the *continuous mode*. In the normal mode, based on the information about the velocity v_f, acceleration a, and the distance to be traveled Δ_l, which are requested by the user and coded using the indexer commands, the indexer will determine the appropriate leg velocity profiles which are either a trapezoid or a triangle depending on the relationship between the given information. The trapezoidal profile is utilized in the development of the straight-line trajectory planning scheme. A typical trapezoidal velocity profile is shown in Figure 5, where t_a, t_c, and t_d denote the acceleration time, the constant velocity time, and the deceleration time, respectively. In addition, the indexer requires that $t_a = t_d$. By inspection, we found that

$$\Delta_l = v_f(t_c + t_d) \tag{30}$$

and

$$v_f = at_a. \tag{31}$$

To track a path in a 3-dimensional space, the positions of x- y- and z-coordinates must always be linearly related to each other anytime during the tracking. Intuitively, if the end-effector leg displacements are planned such that their velocities are linearly related to each other, then the resulting Cartesian motion of the end-effector gripper should be a linear path. Computer simulation utilizing the end-effector forward kinematic transformation developed in Section 3 was performed to verify the above fact and the simulation results have agreed with our intuition. As a result, the following algorithm is developed to plan the leg trajectories for straight-line motion.

Algorithm 2: Straight-Line Motion Trajectory Planning Scheme

1. Use the end-effector inverse kinematic transformation given in (16) to compute the leg lengths corresponding to the starting point $\mathbf{P}_s$ and the final point $\mathbf{P}_f$ of the straight line, namely l_{is} and l_{if} for i=1,2,...,6.

2. Compute $\Delta_{li} = l_{if} - l_{is}$ for i=1,2,...,6 and find Δ_{lk} which has the largest absolute value.

3. Select a_k and v_{fk} for the k-th leg such that $a_k \leq a_{max}$; and $v_{fk} \leq v_{max}$; $v_{fk} \leq a_k\sqrt{|\Delta_{lk}|}$ to ensure trapezoidal profile where a_{max} and v_{max} denote the maximum acceleration and velocity of the stepper motor, respectively, and then compute $t_a = \frac{v_{fk}}{a_k} = t_d$ and $t_c = \frac{\Delta_{lk}}{v_{fk}} - t_a$.

4. For i≠k; i=1,2,..,6 compute $a_i = \frac{\Delta_{li}}{t_a(t_a+t_c)}$ and $v_{fi}=t_a a_i$.

5. Use indexer commands to code v_{fi}, a_i, Δ_{li} for i = 1,2,...,6.

4.2 Trajectory Planning Scheme For Arbitrary Paths

In the *continuous mode* of operation, in addition to the acceleration a and the final velocity v_f, the stepper motor indexer must know about the rotation direction of the stepper motor, which determines the direction of the linear leg displacement. The indexer will transmit proper pulses

to the stepper motor drive which accelerates the stepper motor to velocity v_f. The stepper motor continues to run at this velocity until a new velocity and new acceleration are given in the same rotation direction. Leg trajectory planning for an arbitrary path is done by first dividing a the path into n segments and then planning the velocity profiles of the end-effector legs in the continuous mode so that each segment will be reached within a specified time. The planning is facilitated by using the following algorithm:

Algorithm 3: Arbitrary Path Trajectory Planning Scheme

1. Divide the desired path into n segments.
2. Use the end-effector inverse kinematic transformation given in (16) to compute the leg lengths corresponding to each segment point on the curve, namely l_{ij} for i=1,2,...,6 (leg number) and j=1,2,...,n+1 (segment point number)
3. Compute $\Delta_{ij} = l_{i,j+1} - l_{i,j}$ for i=1,2,...,6 and j=1,2,...n.
4. For each segment, select an appropriate travel time t_j for j=1,2,...,n, and compute the corresponding acceleration and final velocity at the end of each segment.

In general, the travel times for the segments are constant and equal to each other during the tracking of curves which do not require the change of leg direction. However when direction of any leg has to change, the travel time can be selected efficiently using the *look ahead method.* Using this method, the algorithm looks at the next segment point and determine if any change in leg direction is necessary. For example if the direction of a leg requires direction change, then its travel time will be recomputed to ensure that the velocity at the end of the segment will be zero to allow direction change. After that, the recomputed travel time will be set for the remaining legs for the next two segments. Finally the travel time of all legs will be set back to the old value before the leg direction change occurs. The above process can be repeated whenever a leg direction change is necessary.

4.3 Trajectory Planning Scheme for Gross Motion

We notice that Algorithm 3 requires a relatively large number of segments, n, and therefore is computationally intensive. To track a gross motion which does not require a very high positioning accuracy, the number of segments should be reduced so that the computation time of the trajectory planning scheme can be minimized. Unlike the development of Algorithm 2, the gross motion planning will use the triangular velocity profile in the normal mode of the stepper motor indexer. The following algorithm will facilitate the trajectory planning for gross motion.

Algorithm 4: Gross Motion Trajectory Planning Scheme

1. Divide the desired path into n segments.
2. Use the end-effector inverse kinematic transformation given in (16) to compute the leg lengths corresponding to each segment point on the curve, namely l_{ik} for i=1,2,...,6 (leg number) and k=1,2,...,n+1 (segment point number).

3. For the ith leg, locate the extreme (maximum and maximum) segment points and identify its sections each of which is located between two consecutive maximum and minimum points.

4. Compute the section lengths Δl_{ij}, for i=1,2,...,6 (leg number) and j=1,2,...,m_i where m_i is the number of sections in the ith leg and the section length is the absolute value of the difference between the extreme points of the section.

5. Select a travel time t_t for the desired path and compute

$$t_{ij} = \frac{\Delta l_{ij}}{\sum_{p=1}^{m_i} \Delta l_{ip}} t_t \tag{32}$$

for i=1,2,...,6 and j=1,2,...,m_i.

6. Compute

$$a_{ij} = 4\frac{\Delta l_{ij}}{t_{ij}^2} \quad \text{(acceleration)} \tag{33}$$

and

$$v_{ij}^{\max} = a_{ij}\,\frac{t_{ij}}{2} \quad \text{(maximum velocity)} \tag{34}$$

for i=1,2,..,6, and j=1,2,...,m_i, to ensure that triangular velocity profile is used for each section.

7. Use indexer commands to code $v_{ij}^{\max}$, a_{ij}, and Δl_{ij} for i = 1,2,...,6.

5 Experimental Study

In this section, we present the results obtained from experiments conducted to study the performance of the trajectory planning schemes developed in previous section. In particular, Algorithm 2 is used to plan the end-effector leg trajectories for tracking a triangle, Algorithm 3 for tracking a circular path and Algorithm 4 for tracking a spiral path. In the experiments, since the test paths are those to be tracked by the end-effector gripper and expressed with respect to the base platform Frame $\{\mathbf{B}\}$, the test paths must be transformed to the payload platform Frame $\{\mathbf{P}\}$ before Algorithms 2-4 can be applied. Moreover, the homogeneous transformation matrix ${}^P_G\mathbf{T}$, which represents the pose of the gripper with respect to the $\{\mathbf{P}\}$ can be assumed to be invariant because the test paths are planar path in the x-y plane of the base platform Frame $\{\mathbf{B}\}$. Thus the pose of the payload platform with respect to the base platform specified by ${}^B_P\mathbf{T}$, which corresponds to a desired gripper pose specified by ${}^B_G\mathbf{T}_{\text{des}}$, can be computed by

$${}^B_P\mathbf{T} = {}^B_G\mathbf{T}_{\text{des}}\,({}^P_G\mathbf{T})^{-1}. \tag{35}$$

The end-effector parameters are given below:

- Base Platform Radius r_B = 29.267 inches, $\theta_B = 52.14^o$
- Payload Platform Radius r_P = 22.238 inches, $\theta_P = 12.05^o$
- Gripper Platform Radius = 8.06 inches

5.1 Study Case 1: Tracking Straight Lines

Figures 7-8 present the results of tracking a triangle lying in the x-y plane of {B}. The path consists of three straight line segments modeled by

$$\begin{aligned} {}^B\mathbf{x}_g(t) = 0.4896\,t;\ {}^B\mathbf{y}_g(t) = {}^B\mathbf{x}_g(t) \ &;\ \text{for } 0 \text{ sec} \le t \le 8.17 \text{ sec} \\ {}^B\mathbf{x}_g(t) = -0.823\,t + 10.72;\ {}^B\mathbf{y}_g = 4 \ &;\ \text{for } 8.17 \text{ sec} \le t \le 17.89 \text{ sec} \\ {}^B\mathbf{x}_g(t) = 0.5304\,t - 13.49;\ {}^B\mathbf{y}_g = -{}^B\mathbf{x}_g(t) \ &;\ \text{for } 17.89 \text{ sec} \le t \le 25.43 \text{ sec.} \end{aligned} \tag{36}$$

Using Algorithm 2, the trapezoidal velocity profiles of the six legs are determined and illustrated in Figure 7. The path that the end-effector gripper actually tracked, is presented in Figure 8 together with the desired path. The average and maximum tracking errors were $7.79\text{x}10^{-3}$ inches and $10.5\text{x}10^{-3}$ inches, respectively.

5.2 Study Case 2: Tracking a Circular Path

The results of tracking a circular path modeled by

$$\left.\begin{array}{c} {}^B\mathbf{x}_g(t) = 1.6\cos\alpha(t) \\ {}^B\mathbf{y}_g(t) = 1.6\sin\alpha(t) \\ \alpha(t) = \frac{2\pi}{10}\,t \end{array}\right\} \text{for } 0 \text{ sec} \le t \le 10 \text{ sec} \tag{37}$$

are showed in Figures 9-10. Figure 9 illustrates the leg velocity profiles determined by Algorithm 3 and Figure 10 shows the actual and desired paths. The average and maximum tracking errors were found to be $0.469\text{x}10^{-3}$ inches and $1.2\text{x}10^{-3}$ inches, respectively.

5.3 Study Case 3: Tracking A Spiral Path

Figure 11 presents the triangular leg velocity profiles which were determined using Algorithm 4, and Figure 12 shows the actual and desired responses of tracking a spiral path modeled by

$$\left.\begin{array}{c} {}^B\mathbf{x}_g(t) = \mathrm{R}(t)\cos\alpha(t) \\ {}^B\mathbf{y}_g(t) = \mathrm{R}(t)\sin\alpha(t) \\ \alpha(t) = \frac{6\pi}{45}\,t \\ \mathrm{R}(t) = 0.16\mathrm{e}^{\sqrt{\alpha(t)}} \end{array}\right\} \text{for } 0 \text{ sec} \le t \le 45 \text{ sec.} \tag{38}$$

Experimental results showed that the average and maximum tracking errors were 0.125 inches and 0.514 inches, respectively, as expected for gross motion trajectory planning.

6 Concluding Remarks

In this report, we presented the kinematic and trajectory planning for a 6 DOF end-effector whose design was based on the Stewart Platform mechanism. The end-effector has been used as a testbed for studying robotic assembly of NASA hardwares with passive compliance. Vector analysis was employed to derive a closed-form solution for the end-effector inverse kinematic transformation. A computationally efficient numerical solution was obtained for the end-effector forward kinematic transformation using Newton-Raphson method. Three trajectory planning

schemes, two for fine motion and one for gross motion, were developed for the end-effector. Experiments conducted to evaluate the performance of the trajectory planning schemes showed excellent tracking quality with minimal errors. Current activities focus on implementing the developed trajectory planning schemes on mating and demating space-rated connectors and using the compliant platform to acquire forces/torques applied on the end-effector during the assembly task.

References

[1] Stewart, D., "A Platform with Six Degrees of Freedom," *Proc. Institute of Mechanical Engineering*, Vol. 180, Part 1, No. 5, pp. 371-386, 1965-1966.

[2] Dieudonne, J.E. et al, "An Actuator Extension Transformation for a Motion Simulator and an Inverse Transformation Applying Newton-Raphson's Method," *NASA Technical Report D-7067*, 1972.

[3] Hoffman, R., and McKinnon, M.C., "Vibration Modes of an Aircraft Simulator Motion System," *Proc. The Fifth World Congress for the Theory of Machines and Mechanisms, an ASME Publication*, pp. 603-606, 1979.

[4] McCallion, H., and Truong, P.D., "The Analysis of a Six-Degree-of-Freedom Work Station for Mechanised Assembly," *Proc. The Fifth World Congress for the Theory of Machines and Mechanisms, an ASME Publication*, pp. 611-616, 1979.

[5] Hunt, K. H., *Kinematic Geometry of Mechanisms*, Oxford University, London 1978.

[6] Sugimoto, K. and Duffy, J., "Application of Linear Algebra to Screw Systems," *Mech. Mach. Theory*, Vol. 17, No. 1, pp. 73-83, 1982.

[7] Hunt, K. H., "Structural Kinematics of in-parallel-actuated Robot Arms," *Trans. ASME, J. Mech., Transmis., Automa. in Des.*, Vol. 105, pp. 705-712, 1983.

[8] Premack, Timothy et al, "Design and Implementation of a Compliant Robot with Force Feedback and Strategy Planning Software," *NASA Technical Memorandum 86111*, 1984.

[9] Nguyen, C.C., Pooran, F.J., and Premack, T., "Control of Robot Manipulator Compliance," in *Recent Trends in Robotics: Modeling, Control and Education*, edited by M. Jamshidi, J.Y.S. Luh, and M. Shahinpoor, North Holland, New York, pp. 237-242, 1986.

[10] Yang, D. C. and Lee, T. W., "Feasibility Study of a Platform Type of Robotic Manipulators from a Kinematic Viewpoint," *Trans. ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 106. pp. 191-198, June 1984.

[11] Mohammed, M. G. and Duffy, J., "A Direct Determination of the Instantaneous Kinematics of Fully Parallel Robotic Manipulators," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 107 pp. 226-229, 1985.

[12] Fichter, E.F., "A Stewart Platform-Based Manipulator: General Theory and Practical Construction," *Int. Journal of Robotics Research*, pp. 157-182, Summer 1986

[13] Sugimoto, K., "Kinematic and Dynamic Analysis of Parallel Manipulators by Means of Motor Algebra," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, pp. 1-5, Dec. 1986.

[14] Lee, K. M., Chao, A., and Shah, D. K., "A Three Degrees of Freedom In-parallel Actuated Manipulator," *Proc. IASTED Int. Conf.*, pp. 134-138, 1986.

[15] Rees-Jones, J., "Cross Coordinate Control of Robotic Manipulators," in *Proceedings of the International Workshop on Nuclear Robotic Technologies and Applications, Present and Future*, University of Lancaster, UK, June 29-July 1, 1987.

[16] Behi, F., "Kinematic Analysis for a Six-Degree-of-Freedom 3-PRPS Parallel Mechanism," *IEEE Journal of Robotics and Automation*, Vol. 5, No. 5, pp. 561-565, October 1988.

[17] Sharon, A., Hogan, N., Hardt, D., "High-Bandwidth Force Regulation and Inertia Reduction Using a Macro/Micro Manipulator System," *Proc. IEEE International Conference on Robotics and Automation*, pp. 261-266, Philadelphia, PA, April 1988.

[18] Sugimoto, K., "Computational Scheme for Dynamic Analysis of Parallel Manipulators," in *Trends and Developments in Mechanisms, Machines, and Robotics-1988, ASME Proceedings of the 20th Biennial Mechanisms Conference*, 1988.

[19] Kerr, D. R., "Analysis, Properties, and Design of a Stewart-Platform Transducer,"in *Trends and Developments in Mechanisms, Machines, and Robotics-1988, ASME Proceedings of the 20th Biennial Mechanisms Conference*, 1988.

[20] Nguyen, C.C., Pooran, F.J., "Adaptive Force/Position Control of Robot Manipulators with Closed-Kinematic Chain Mechanism," in *Robotics and Manufacturing: Recent Trends in Research, Education, and Application*, edited by M. Jamshidi et al, ASME Press, New York, pp. 177-186, 1988.

[21] Griffis, M., Duffy, J., "A Forward Displacement Analysis of a Class of Stewart Platforms," *Journal of Robotic Systems*, Vol. 6, pp. 703-720, 1989.

[22] Nguyen, C.C., and Pooran, F.J., "Kinematic Analysis and Workspace Determination of A 6 DOF CKCM Robot End-Effector," *Journal of Mechanical Working Technology*, Vol. 20, pp. 283-294, 1989.

[23] Nguyen, C.C., and Pooran, F.J., "Dynamical Analysis of 6 DOF CKCM Robot End-Effector for Dual-Arm Telerobot Systems," *Journal of Robotics and Autonomous Systems*, Vol. 5, pp. 377-394, 1989.

[24] Nanua, P., Waldron, K.J., Murthy, V., "Direct Kinematic Solution of a Stewart Platform," *IEEE Trans. Robotics and Automation*, Vol. 6, No. 4, pp. 438-444, 1990.

[25] Nguyen, C.C., and Pooran, F.J., "Learning-Based Control of a Closed-Kinematic Chain Robot End-Effector Performing Repetitive Tasks," *International Journal of Microcomputer Applications*, Vol. 9, No. 1, pp. 9-15, 1990.

[26] Nguyen, C.C., Antrazi, S., Zhou, Z-L, "Trajectory Planning and Kinematic Control of a Stewart Platform-Based Manipulator," *Proc. 5th International Conference on CAD/CAM Robotics and Factories of the Future*, Norfolk, Virginia, December 1990.

[27] Nguyen, C.C., Antrazi, S., Zhou, Z-L, Campbell, Jr., C.E., "Experimental Study of Motion Control and Trajectory Planning for a Stewart Platform Robot Manipulator," *Proc. IEEE International Conference on Robotics and Automation*, Sacramento, California, April 1991.

[28] Press, W.H., et al, "Numerical Recipes in C: The Art of Scientific Computing," *Cambridge University Press*, 1988.

[29] Fu, K.S. et.al., *Robotics: Control, Sensing, Vision, and Intelligence* , McGraw-Hill, New York, 1987.

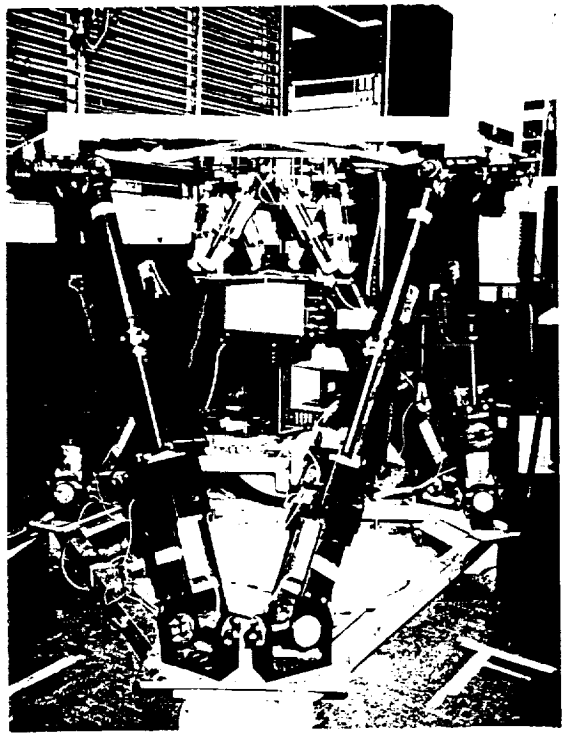

Figure 1: The Stewart Platform-based end-effector

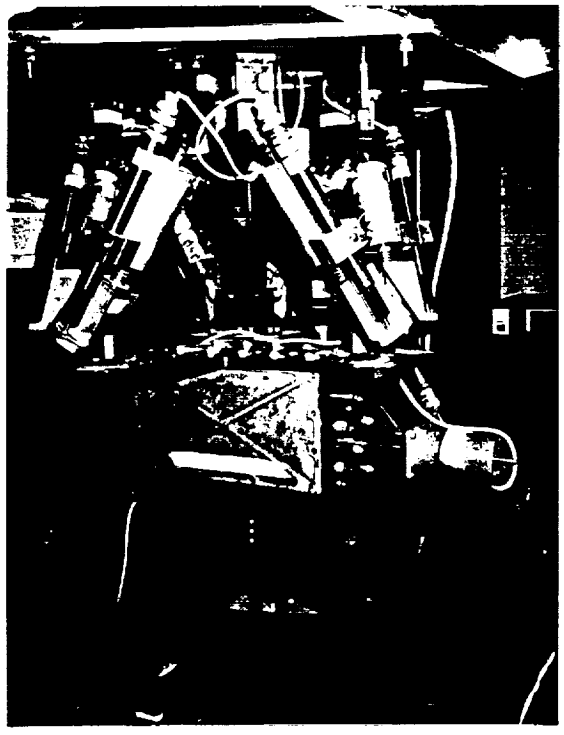

Figure 2: The compliant platform

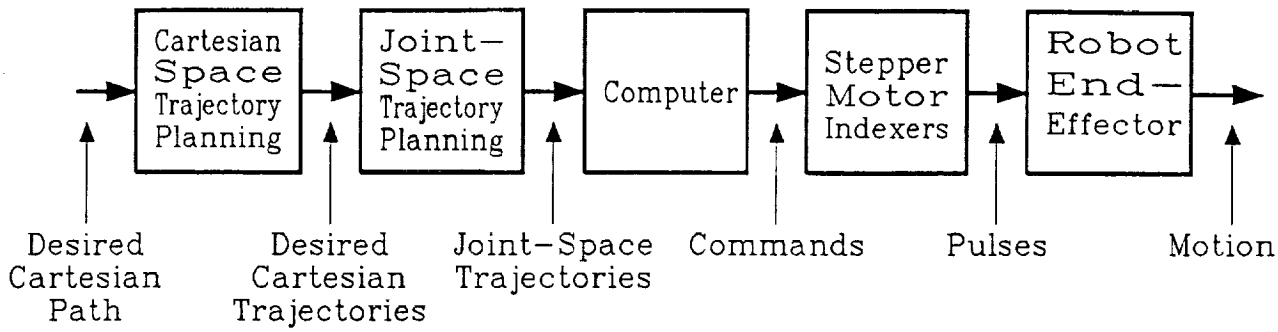

Figure 3: Trajectory planning and control scheme for the end-effector

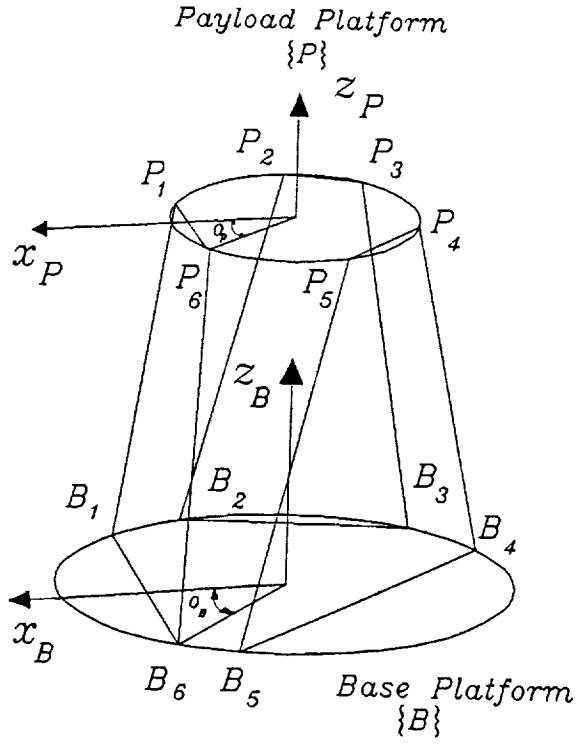

Figure 4: Frame assignment of the platforms

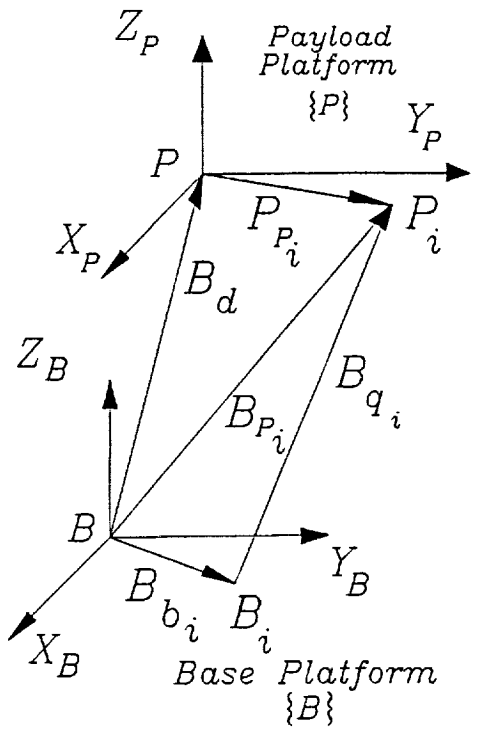

Figure 5: Vector diagram for the ith actuator

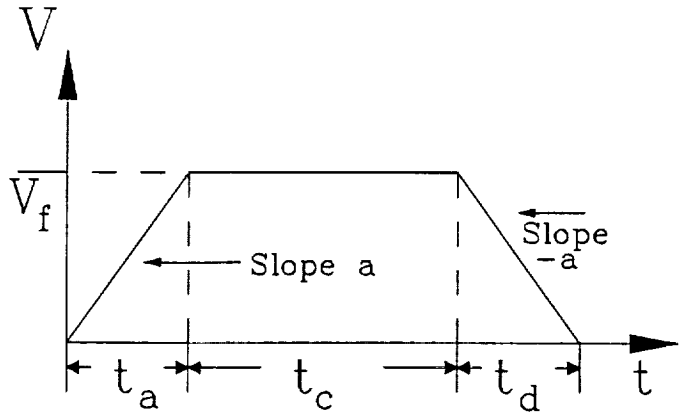

Figure 6: Trapeziodal velocity profile

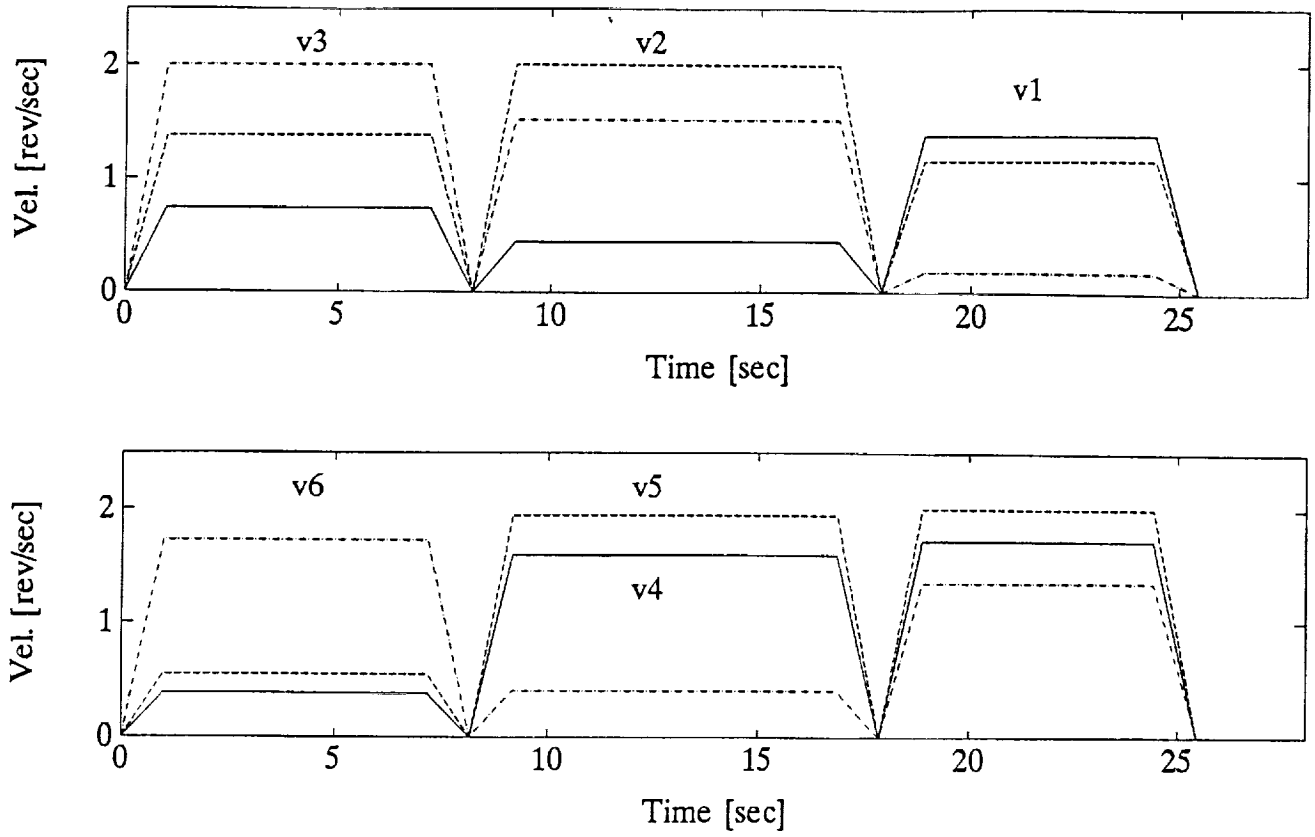

Figure 7: Leg velocity profiles (Study Case 1)
v_i = ith leg velocity; 1 rev. = 0.2 inch

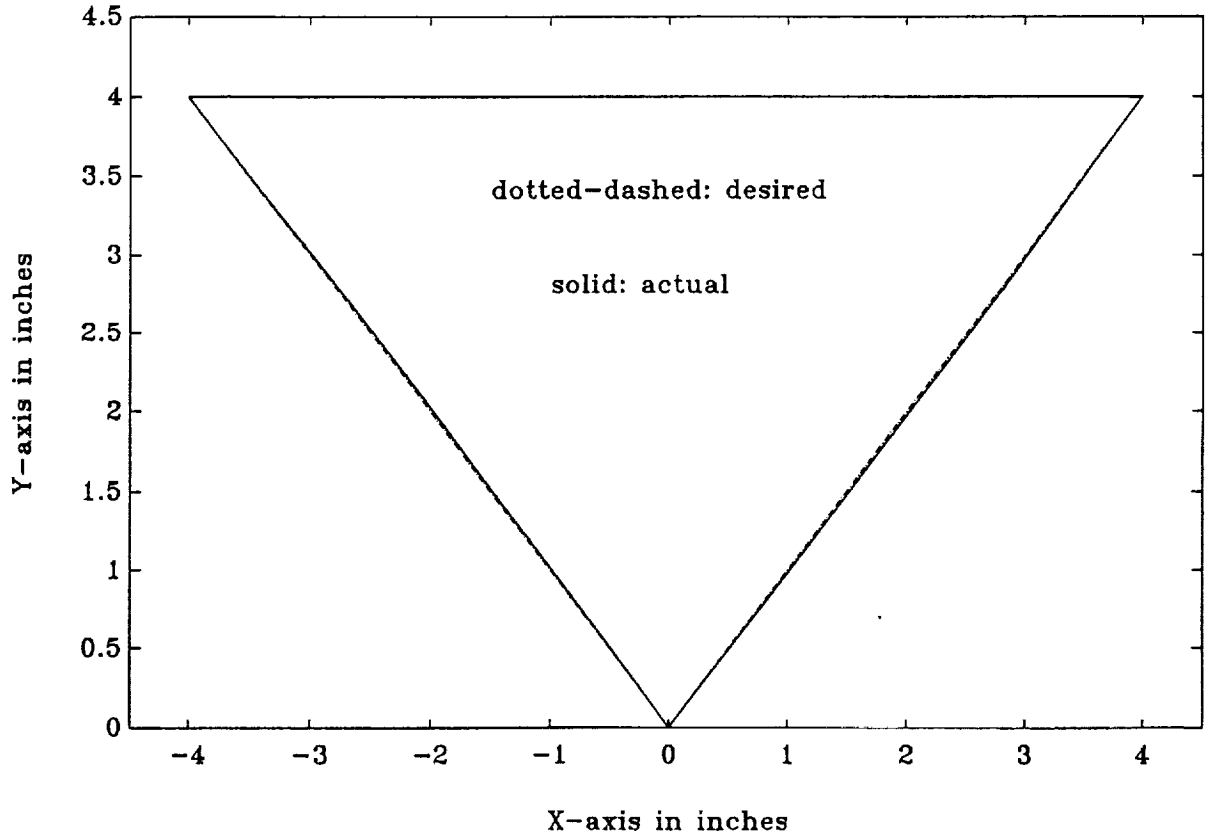

Figure 8: Tracking a triangle (Study Case 1)

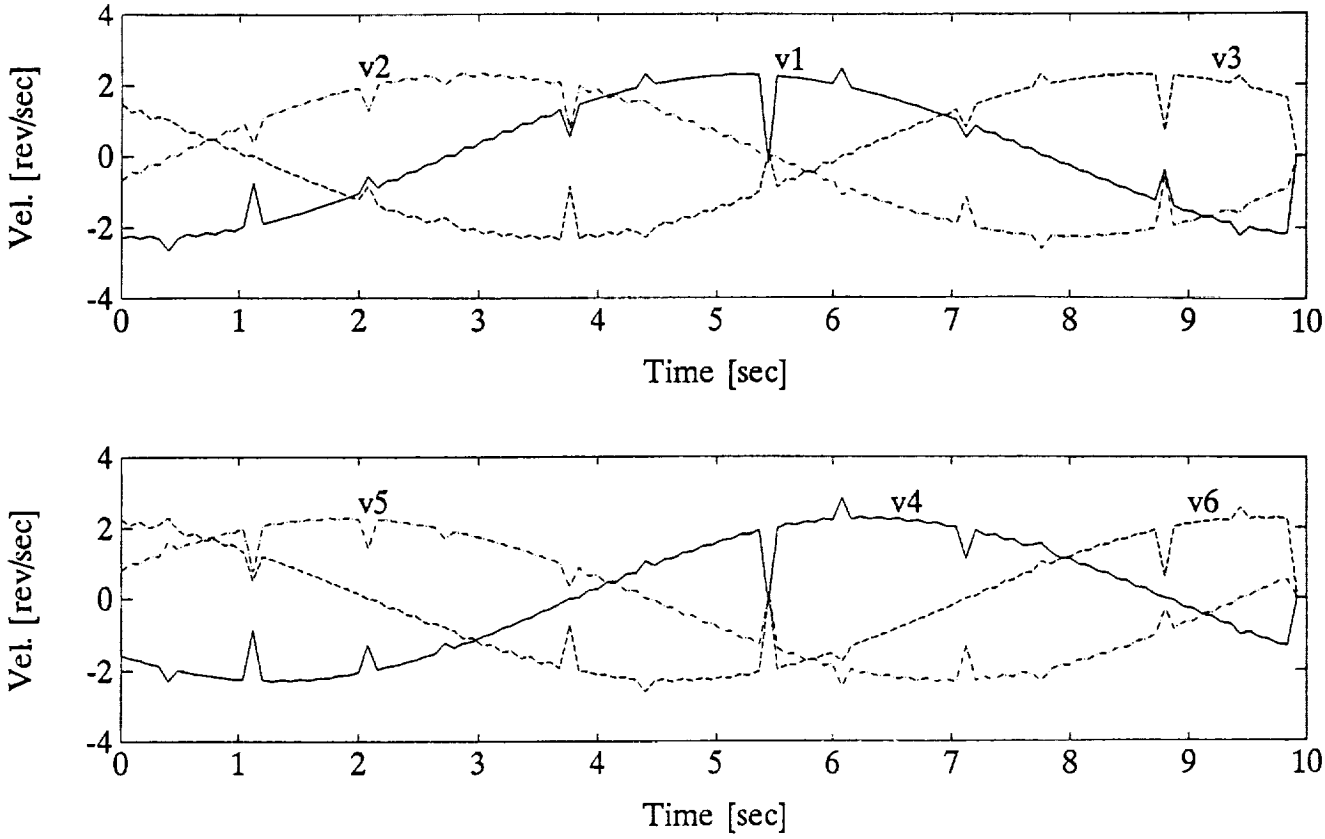

Figure 9: Leg velocity profiles (Study Case 2)
v_i = ith leg velocity; 1 rev. = 0.2 inch

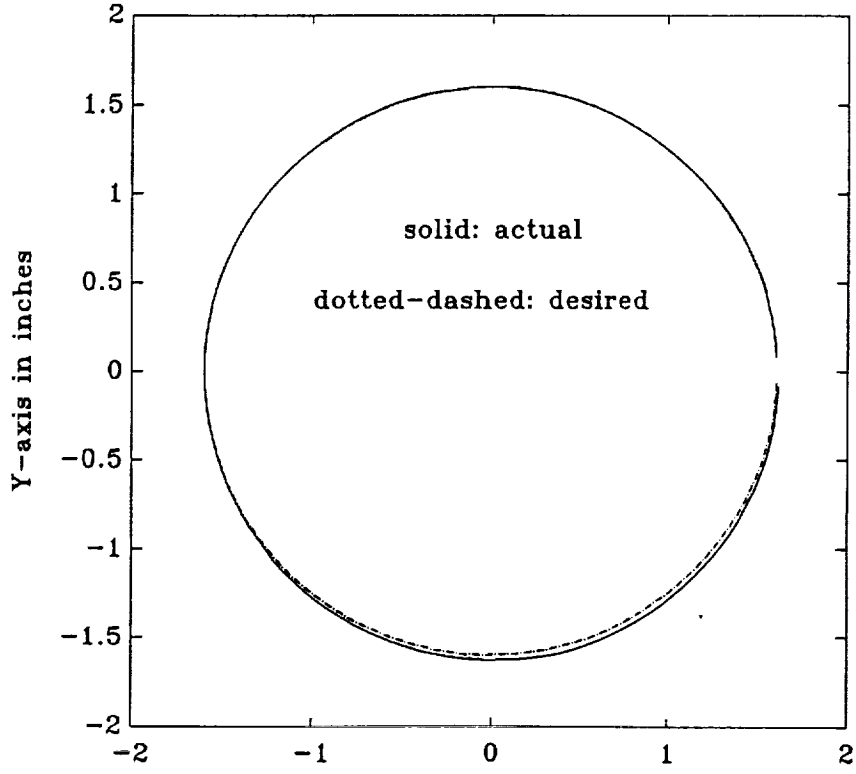

Figure 10: Tracking a circular path (Study Case 2)

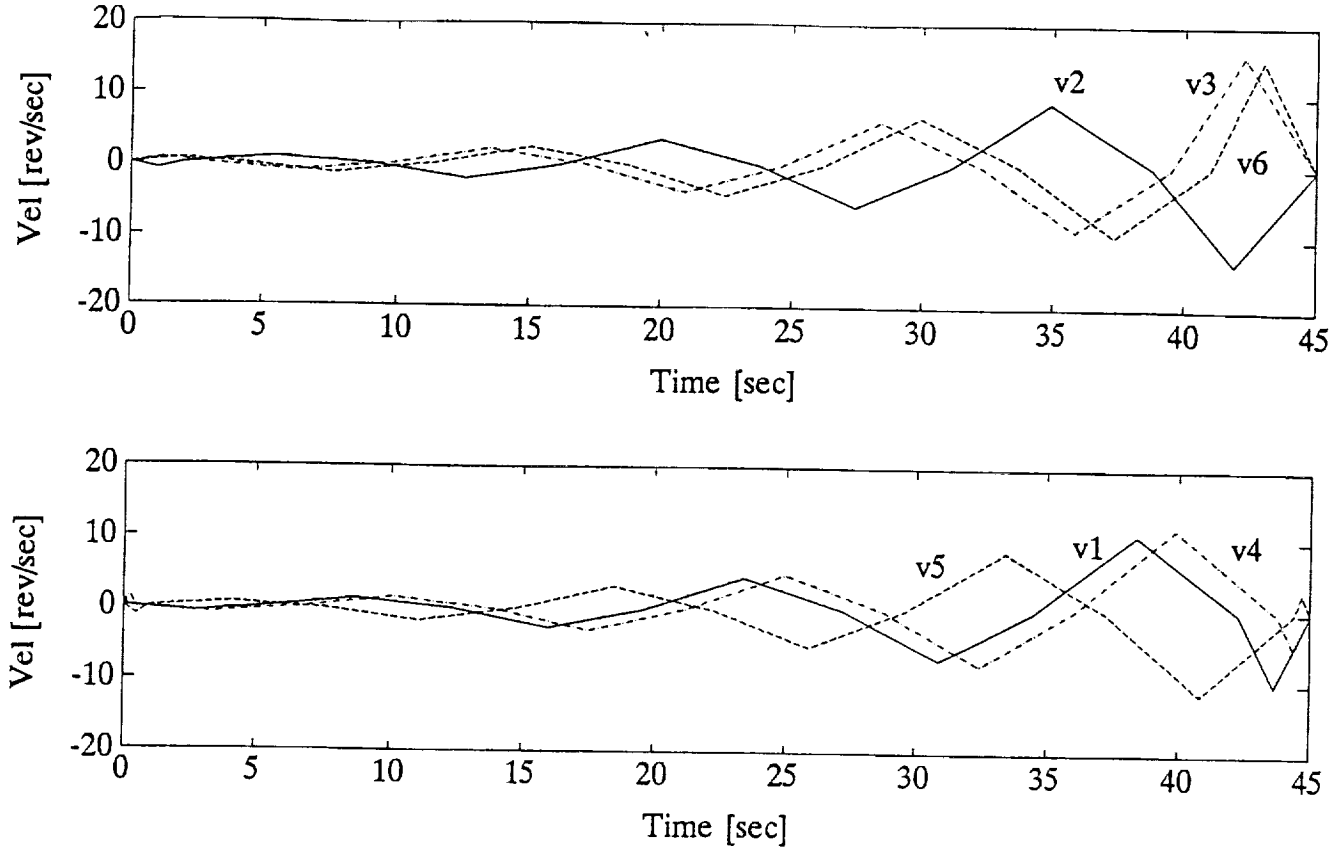

Figure 11: Leg velocity profiles (Study Case 3)
v_i = ith leg velocity; 1 rev. = 0.2 inch

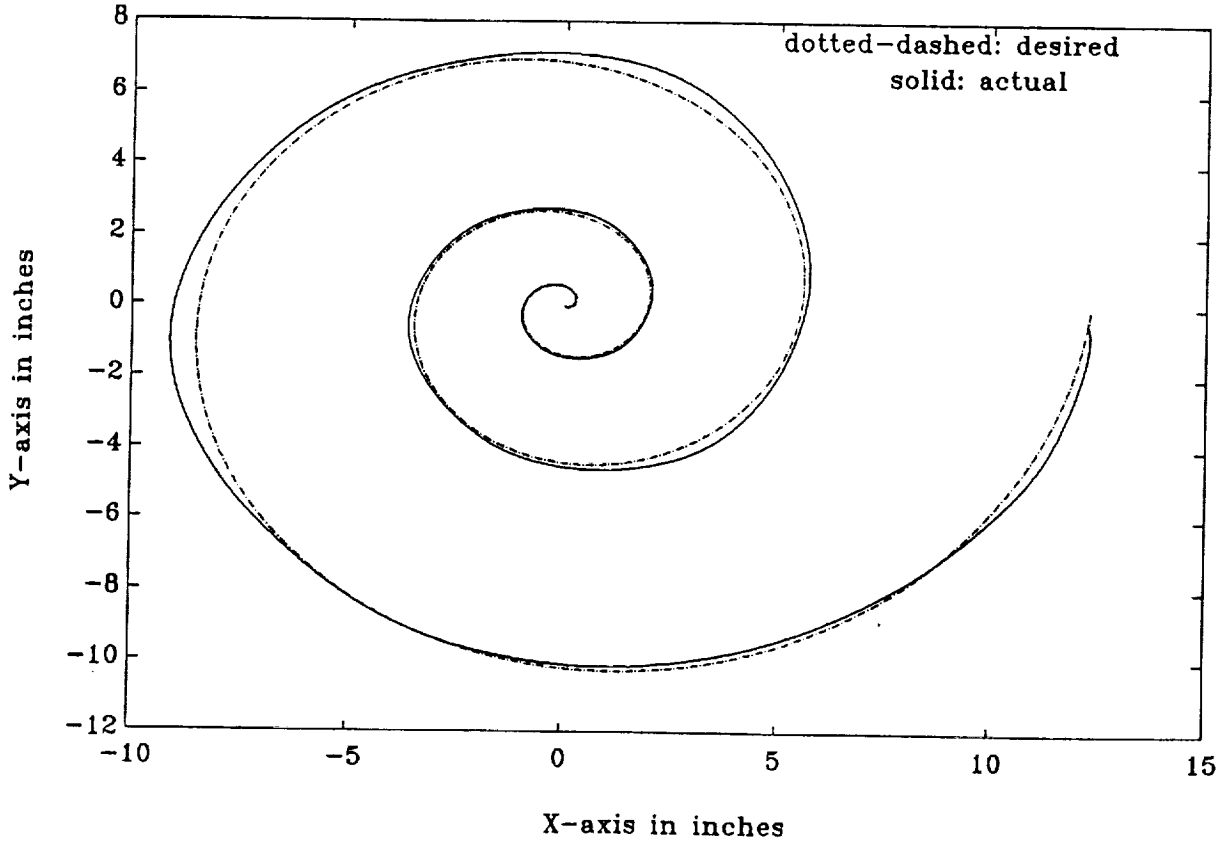

Figure 12: Tracking a spiral path (Study Case 3)