Computers Elect. Engng Vol. 17, No. 3, pp. 191–203, 1991
Printed in Great Britain

0045-7906/91 $3.00 + 0.00
Pergamon Press plc

ANALYSIS AND IMPLEMENTATION OF A 6 DOF STEWART PLATFORM-BASED ROBOTIC WRIST

CHARLES C. NGUYEN,[1] SAMI C. ANTRAZI,[1] ZHEN-LEI ZHOU[1] and
CHARLES E. CAMPBELL Jr[2]

[1]Robotics and Control Laboratory, Department of Electrical Engineering, Catholic University of America, Washington, DC 20064, U.S.A.
[2]NASA/Goddard Space Flight Center, Greenbelt, MD 20771, U.S.A.

(Received 1 December 1990; accepted in final revised form 10 April 1991)

Abstract—In this paper, we present the kinematic analysis and implementation of a 6 DOF robotic wrist which is mounted to a general open-kinematic chain manipulator to serve as a testbed for studying precision robotic assembly in space. The wrist design is based on the Stewart-Platform mechanism and consists mainly of two platforms and six linear actuators driven by d.c. motors. Position feedback is achieved by linear displacement transducers mounted along the actuators and force feedback is obtained by a 6 DOF force sensor mounted between the gripper and the payload platform. The robot wrist inverse kinematics which computes the required actuator lengths corresponding to Cartesian variables has a closed-form solution. The forward kinematics is solved iteratively using the Newton–Raphson method which simultaneously provides a modified Jacobian matrix which relates length velocities to Cartesian translational velocities and time rates of change of roll–pitch–yaw angles. Results of computer simulation conducted to evaluate the efficiency of the forward kinematics and modified Jacobian matrix are presented and discussed.

1. INTRODUCTION

Motions robots perform during a robotic operation in space can be divided into *gross motion* and *fine motion*. Gross motion permits low positioning accuracy, e.g. in obstacle avoidance, while fine motion requires very high positioning accuracies, usually of thousands of an inch, e.g. in mating and demating space-rated connectors. Traditional 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, their cantilever-like structure causes OKC manipulators to have low stiffness and consequently undesirable dynamic characteristics, especially at high speed and large payload. In addition, they have low strength-to-weight ratios due to the fact that the payload is not uniformly distributed to the actuators. Finally, the fact that relatively large position error occurs at the last link because the joint errors are accumulated throughout the mechanism suggest that OKC manipulators are not suitable for high-precision tasks. As a result, it was proposed in [1, 2] that a robotic end-effector capable of performing high-precision motion be mounted to a general OKC manipulator to perform fine motion while the OKC manipulator is solely responsible for carrying out gross motion during a telerobotic operation. Closed-kinematic chain (CKC) mechanism has been selected for the design of the end-effector because even though it has relatively small workspace and low maneuverability, it possesses high positioning capability produced by its high structural rigidity and noncumulative actuator errors. CKC mechanism also has higher strength-to-weight ratios as compared to the OKC mechanism because the payload is proportionally distributed to the links. In addition, the inverse kinematic problem of the CKC mechanism has simple closed-form solutions. Implementations of the CKC mechanism concept first appeared in the Stewart Platform [3] which was originally designed as an aircraft simulator. A typical Stewart Platform consists of two platforms driven by a number of parallel actuators and is often referred as *parallel mechanism* or *parallel manipulator*. The invention of the Stewart Platform has attracted tremendous robot designer's attention and its mechanism was used in many robotic applications [4, 18]. Dieudonne *et al.* [4] derived an actuator extension transformation and presented experimental results of a Stewart Platform-based simulator built at NASA Langley

Research Center to train aircraft operators. A finite element program was used by Hoffman and McKinnon [5] to simulate the motion of the Stewart Platform whose mechanism was later applied by McCallion and Truong [6] to design an automatic assembly table. Hunt [7] conducted a systematic study of in-parallel-actuated robot arms and presented the structural kinematic problem of this type of manipulators [9]. Sugimoto and Duffy [8] developed a general technique to describe the instantaneous link motion of a single closed-loop mechanism by employing linear algebra elements to screw systems. In order to study autonomous robotic assembly, Premack *et al.* [10] employed the Stewart Platform mechanism to build a passive compliant robot end-effector whose control problem was investigated by Nguyen *et al.* [11]. Kinematic problems and practical construction of the Stewart Platform were later considered by Yang and Lee [12] and Fichter [13], respectively. Sugimoto [14] studied kinematics and dynamics of parallel manipulators and Lee *et al.* [15] derived dynamical equations for a 3 DOF CKC manipulator. Nguyen and Pooran [16] developed a learning control scheme for a 2 DOF CKC manipulator performing repetitive tasks. Trajectory planning schemes were developed by Nguyen *et al.* [19] for Stewart Platform-based manipulators whose actuators are driven by stepper motors.

Recently a robotic wrist possessing 6 DOFs was designed and built at the Goddard Space Flight Center (NASA) based on the mechanism of the Stewart Platform to serve as a testbed for studying high-precision robotic operations in space. This paper presents the development and implementation of kinematic transformations for the robotic wrist. This paper is organized as follows. The next section describes the main components of the robotic wrist. Then a kinematic analysis is performed to provide a closed-form solution to the inverse kinematic transformation. After that, a computationally efficient solution is derived for the forward kinematic transformation using the Newton–Raphson method which simultaneously provides a modified Jacobian matrix. Finally evaluation of the forward kinematic transformation and modified Jacobian matrix is done by means of a computer simulation whose results are presented and discussed.

2. THE STEWART-PLATFORM BASED ROBOTIC WRIST

Figure 1 presents a robot manipulator of the Intelligent Robotic Laboratory (IRL) at the Goddard Space Flight Center (NASA), which consists of a 6 DOF Cincinnati T3 robot and a 6 DOF Stewart Platform-based robotic wrist mounted to the last link of the T3 robot. The manipulator has a total of 11 DOFs since 1 DOF of the wrist is identical to that of the T3 robot.

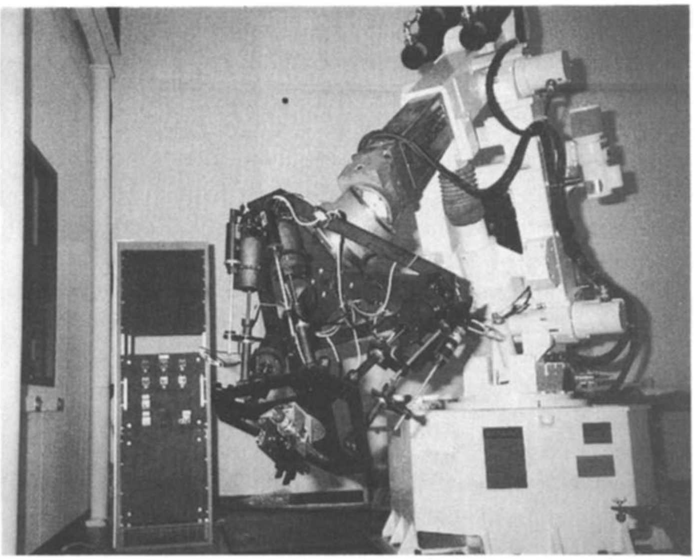


Fig. 1. The GSFC-IRL robot manipulator.

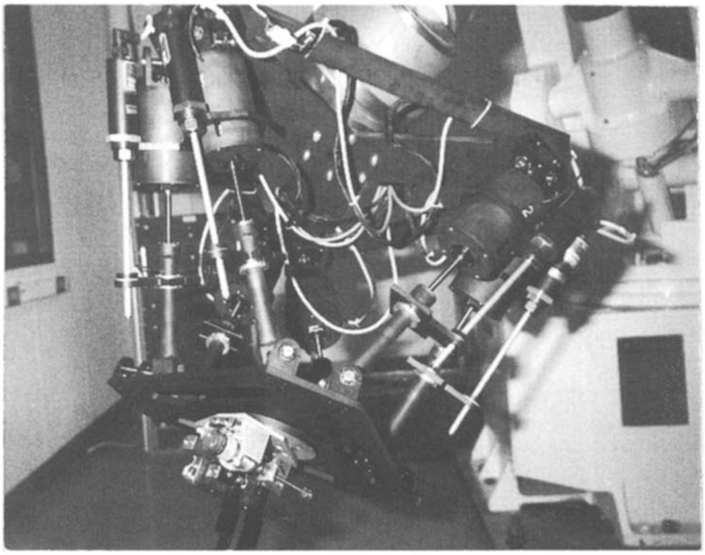


Fig. 2. The Stewart Platform-based robotic wrist.

The main function of the T3 robot is to perform gross motion, for example to bring the robotic wrist into its workspace and then let the wrist carry out fine motion required for high-precision operations such as assembly of parts, mating connectors, etc. As shown in Fig. 2, the design of the robotic wrist is based on the mechanism of the Stewart Platform. It mainly consists of a payload platform, a base platform, six linear actuators and a gripper attached to the payload platform. The payload platform is coupled with the base platform by the actuators each of which is composed of a NSK ballscrew assembly mounted axially with a PMI d.c. motor. The motors drive the ballscrews to extend or shorten the actuator lengths whose variations will in turn produce the motion of the payload platform relative to the base platform. The actuator lengths are measured by six BALLUFF linear displacement transducers (LDT) mounted along the actuators. Forces/torques exerted by the gripper are acquired through a JR^3 Universal Force–Moment Sensor System mounted between the gripper base and the payload platform. Each end of the acuator links is mounted to the platforms by 2 DOF universal joints. The LDT signals are sent to the IRL local area network (LAN) via an ethernet board and a data translation input board resided in a PC/386. An Apollo workstation will take the sensor signals off the LAN by means of another ethernet board, performs all necessary computations for the implementation of control schemes, coordinate transformations, etc., and sends the actuating signals to the PMI motor drives via a data translation output board.

3. THE INVERSE KINEMATIC TRANSFORMATION

This section develops an inverse kinematic transformation for the robot wrist, which determines the required actuator lengths for a given configuration composed of Caretsian position and orientation of the payload platform with respect to the base platform. Frame assignment to the robot wrist is illustrated in Fig. 3 where two coordinate frames $\{\mathbf{P}\}$ and $\{\mathbf{B}\}$ are assigned to the payload and base platforms, respectively. The origin of Frame $\{\mathbf{P}\}$ 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 $\{\mathbf{B}\}$ 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 $\{\mathbf{P}\}$ with respect to Frame $\{\mathbf{B}\}$ where the position of Frame $\{\mathbf{P}\}$ is specified by the position of its

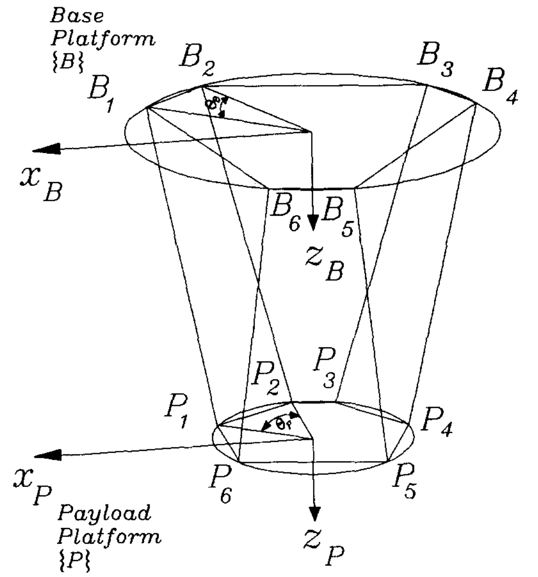


Fig. 3. Frame assignment for the robotic wrist.

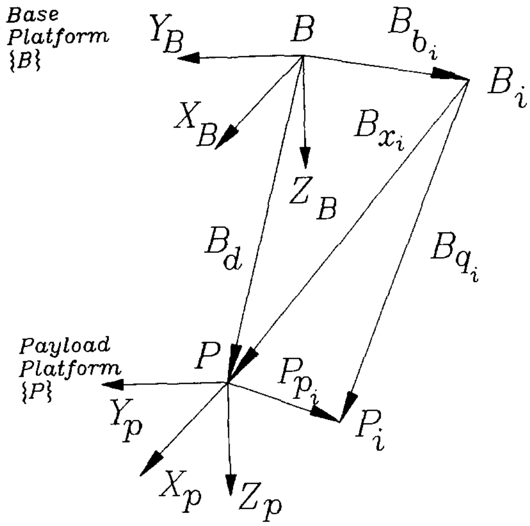


Fig. 4. Vector diagram for the ith actuator.

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 $i = 1, 2, \ldots, 6$, then by inspection we obtain:

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

and

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

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:

$$^P\mathbf{p}_i = [r_P\cos(\lambda_i) \quad r_P\sin(\lambda_i) \quad 0]^T \tag{3}$$

and

$$^B\mathbf{b}_i = [r_B\cos(\Lambda_i) \quad r_B\sin(\Lambda_i) \quad 0]^T, \tag{4}$$

for $i = 1, 2, \ldots, 6$ 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 Fig. 4. 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

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

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

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

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

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

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

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

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 $i = 1, 2, \ldots, 6$.

Equation (16) represents the *closed-form* solution to the inverse kinematic problem in the sense that required actuator lengths l_i for $i = 1, 2, \ldots, 6$ 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 $\{\mathbf{B}\}$.

Specification of the payload platform orientation

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 [21]. 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 $\{\mathbf{B}\}$ 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*:

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

*$c\alpha \equiv \cos\alpha$ and $s\alpha \equiv \sin\alpha$.

4. THE FORWARD KINEMATIC TRANSFORMATION

This section considers the development of the forward transformation which transforms the actuator lengths l_i for $i = 1, 2, \ldots, 6$ measured by six LDTs into the Cartesian position and orientation 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 $i = 1, 2, \ldots, 6$. In general, there exists no closed-form solution for the above problem since equation (16) represents a set of six highly nonlinear simultaneous equations with six 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 six 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 $i = 1, 2, \ldots, 6$, where the vector $\mathbf{a}$ is defined as:

$$\mathbf{a} = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6]^T = [x \quad y \quad z \quad \alpha \quad \beta \quad \gamma]^T \tag{19}$$

and then employ the following algorithm [20] to solve for $\mathbf{a}$:

Newton–Raphson Algorithm

Step 1 Select an initial guess $\mathbf{a}$.

Step 2 Compute the elements r_{ij} of ${}^B_P\mathbf{R}$ using (17) for $i, j = 1, 2, 3$.

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

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

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

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

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

It is still unsolved how to select an initial guess which ensures convergence of the algorithm. However according to the experiences gained from computer simulation presented later, any nonzero initial guess within the reachable workspace of the robotic wrist will make the algorithm converge. Perhaps this is one of the properties of the Stewart Platform mechanism. The Newton–Raphson algorithm is expected to work well in a real-time tracking problem where it is used to compute the actual position/orientation of the payload platform with respect to the base platform based on the actuator lengths measured by the LVDTs because the current guess is based on the previous actual position/orientation which is very close to the correct solution provided that the wrist is tracking the desired path very closely.

Computation of partial derivatives

In order to minimize the computational time of the Newton–Raphson algorithm, the expressions 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 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:

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

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

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

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

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

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

Modified Jacobian matrix

Conventionally the manipulator Jacobian matrix is defined as a matrix relating joint velocities to Cartesian velocities composed of translational velocities and rotational velocities. For the robot wrist, since actuator lengths are selected as joint variables, the time rates of change of actuator lengths $\dot{l}_1, \dot{l}_2, \ldots, \dot{l}_6$ are joint velocities. However in order to utilize the partial derivatives computed for the forward kinematic transformation, we define here the velocities of Cartesian positions of the payload platform with Frame {**B**}, namely $\dot{x}$, $\dot{y}$ and $\dot{z}$ as the translational velocities and the velocities of the roll–pitch–yaw angles $\dot{\alpha}$, $\dot{\beta}$ and $\dot{\gamma}$ as the *rotational velocities*. The matrix **J** which relates the length velocities to translation velocities and roll–pitch–yaw angle velocities is therefore called *The modified Jacobian matrix*. Denoting:

$$\dot{\mathbf{a}} = (\dot{a}_1\, \dot{a}_2\, \dot{a}_3\, \dot{a}_4\, \dot{a}_5\, \dot{a}_6)^{\mathrm{T}} = (\dot{x}\, \dot{y}\, \dot{z}\, \dot{\alpha}\, \dot{\beta}\, \dot{\gamma})^{\mathrm{T}}, \tag{30}$$

and

$$\dot{\mathbf{l}} = (\dot{l}_1\, \dot{l}_2\, \dot{l}_3\, \dot{l}_4\, \dot{l}_5\, \dot{l}_6)^{\mathrm{T}}, \tag{31}$$

we obtain

$$\dot{\mathbf{a}} = \mathbf{J}_M \dot{\mathbf{l}}, \tag{32}$$

or

$$\dot{\mathbf{l}} = \mathbf{J}_M^{-1}\, \dot{\mathbf{a}} . \tag{33}$$

where $\mathbf{J}_M$ is the modified Jacobian matrix. Calling $k_{ij} = \frac{\partial l_i}{\partial a_j}$, the ij-element of $\mathbf{J}_M^{-1}$, from (33) we have:

$$\dot{l}_i = \sum_{j=1}^{6} k_{ij} \dot{a}_j = \sum_{j=1}^{6} \frac{\partial l_i}{\partial a_j} \dot{a}_j . \tag{34}$$

Now solving for l_i^2 in (18) yields:

$$l_i^2 = (\bar{x}_i + u_i)^2 + (\bar{y}_i + v_i)^2 + (\bar{z}_i + w_i)^2 = \bar{f}_i, \tag{35}$$

for $i = 1, 2, \ldots, 6$. Recognizing that $\bar{f}_i$ is a function of $\bar{x}_i$, $\bar{y}_i$, $\bar{z}_i$, α, β and γ, and using (23) we differentiate both sides of (34) with respect to time to obtain:

$$2 l_i \dot{l}_i = \sum_{j=1}^{6} \frac{\partial \bar{f}_i}{\partial a_j} \dot{a}_j, \tag{36}$$

from which solving for $\dot{l}_i$ yields

$$\dot{l}_i = \sum_{j=1}^{6} \frac{1}{2l_i}\frac{\partial \bar{f}_i}{\partial a_j}\dot{a}_j. \tag{37}$$

Now comparing (34) and (37) and noting from (35) and (18) that $\frac{\partial \bar{f}_i}{\partial a_j} = \frac{\partial f_i}{\partial a_j}$, we arrive at:

$$k_{ij} = \frac{1}{2l_i}\frac{\partial f_i}{\partial a_j} \tag{38}$$

where $\partial f_i / \partial a_j$ can be obtained from Step 4 of the Newton–Raphson algorithm using (24–29). In other words, we just showed that the inverse of the modified Jacobian matrix can be computed using the results of the forward kinematic transformation.

5. COMPUTER SIMULATION STUDY

In this section we will report results obtained from the computer simulation conducted to study the efficiency of the developed inverse and forward kinematic transformations as well as the modified Jacobian matrix. The simulation scheme employed in the study is illustrated in Fig. 5. In the upper loop, a set of Cartesian test trajectories comprised by vector **a** are converted to the corresponding actuator length trajectories comprised by vector **l** via the inverse kinematic transformation. The Newton–Raphson algorithm implementing the forward kinematic transformation is then applied to convert **l** to $\mathbf{a}_c$, a vector composed of computed Cartesian trajectories corresponding to **l**. The computed Cartesian trajectories are then compared with the Cartesian test trajectories and the resulting errors are recorded. In addition, the test length velocities contained by $\dot{\mathbf{l}}$ are obtained by differentiating **l** with respect to time. In the lower loop, the Cartesian test velocities comprised by vector $\dot{\mathbf{a}}$ are obtained by differentiating **a** with respect to time. The Cartesian test velocities $\dot{\mathbf{a}}$ are then converted to the corresponding length velocities $\dot{\mathbf{l}}_c$ using the inverse modified Jacobian matrix $\mathbf{J}_M^{-1}$. Errors in length velocities are then obtained by comparing the computed length velocities with the test length velocities. The developed transformations are implemented in C and the graphical facility is provided by MATLAB. Computer simulation results for two test cases are presented and discussed below, and English units will be used. The average number of iterations used in the Newton–Raphson algorithm is 2. The wrist parameters used in the simulations are $r_P = 10.441$ in., $r_B = 13.838$ in., $\theta_P = 99.20°$, $\theta_B = 16°$.

Test Case 1. Straight line motion

Figures 6–8 present the computer simulation results of the case in which the Cartesian test trajectories specify a straight line in the x–y plane of the base frame. The straight line motion is described by:

$$x(t) = x_0 + 6.3\left[1 + 3\exp\left(-\frac{3.5}{3}t\right) - 4\exp\left(-\frac{3.5}{4}t\right)\right] \tag{39}$$

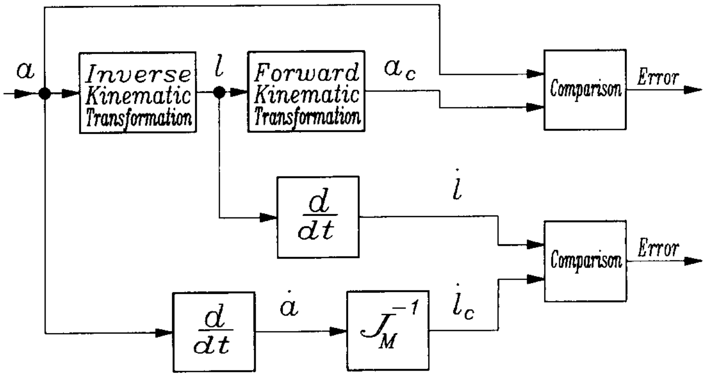

a
Inverse Kinematic Transformation
l
Forward Kinematic Transformation
a_c
Comparison
Error
d/dt
l̇
Comparison
Error
d/dt
ȧ
J_M^{-1}
l̇_c

Fig. 5. Computer simulation scheme.

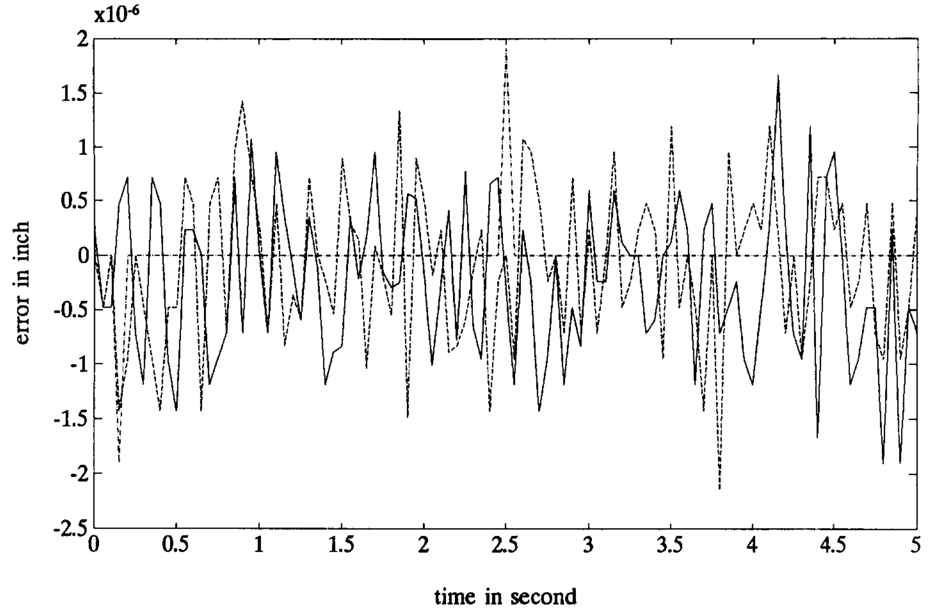


Fig. 6. Straight line motion, errors in x, y, z coordinates; x (——), y (– – –) and z (–·–·).

and

$$y(t)=y_0+9.45\left[1+3\exp\left(-\frac{3.5}{3}t\right)-4\exp\left(-\frac{3.5}{4}t\right)\right], \qquad (40)$$

where the initial position is denoted by $x_0=-3.5$ in., $y_0=-5$ in. The computer simulation was conducted with a sampling time of 0.05 s on a SUN workstation for 5 s. According to Fig. 6 which presents the errors in Cartesian positions x, y, z, a maximum error of $-2.146\,\mu$in. occurs in the y-position and a maximum root mean square (RMS) error of $0.7615\,\mu$in. occurs in the x-position. The errors in roll–pitch–yaw (RPY) angles are shown in Fig. 7 where a maximum error of $0.156\,\mu$rad and a maximum RMS error of $0.623\,\mu$rad occur in the roll angle. According to Fig. 8 which presents the errors in length velocities, relatively large errors exist at the beginning of the simulation and settle down almost to zero after about 1 s. A maximum error of 0.1619 in./s and a maximum RMS error of 0.0361 in./s occur in the second actuator length.

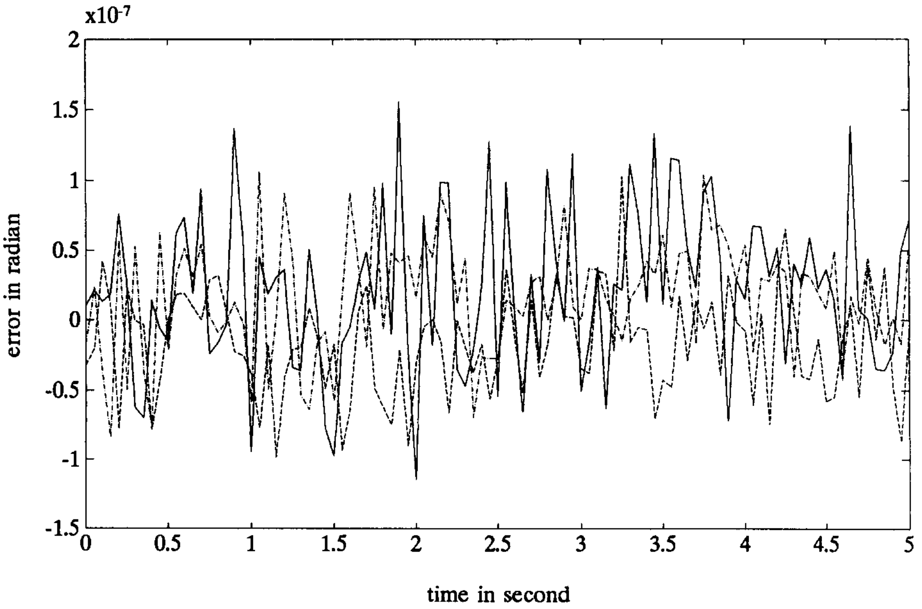


Fig. 7. Straight line motion, errors in RPY angles: α (——), β (– – –) and γ (–·–·).

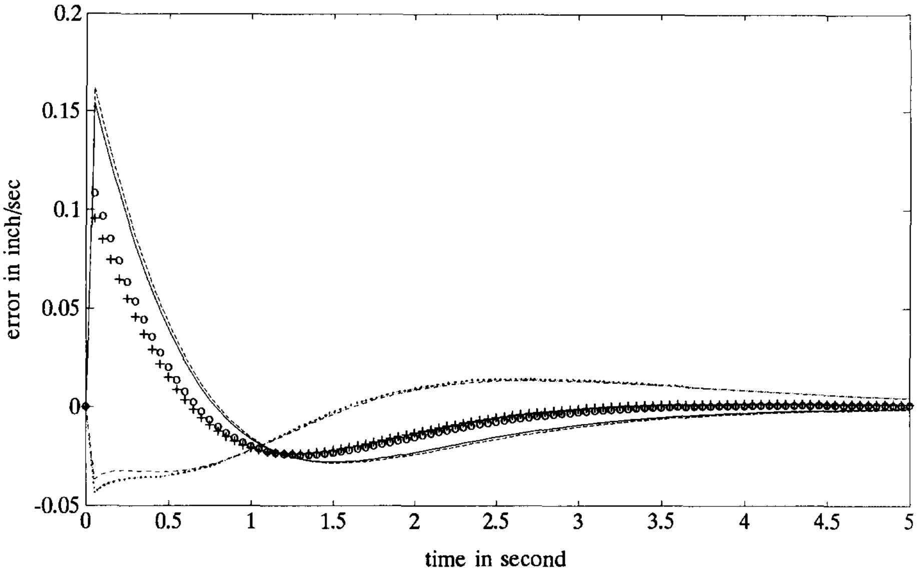


Fig. 8. Straight line motion, errors in length velocities: $\dot{l}_1$ (——), $\dot{l}_2$ (– – –), $\dot{l}_3$ (·····), $\dot{l}_4$ (+ + +), $\dot{l}_5$ (OOO) and $\dot{l}_6$ (–·–·).

Test Case 2. Circular motion

Computer simulation results of the case in which the Cartesian test trajectories specify a circular motion are presented in Figs 9–11. The circular motion consists of three segments described by:

$$x(t)=R\cos\Phi_i;\quad y(t)=R\sin\Phi_i \text{ for } t_{i-1}\leqslant t-<t_i \text{ for } i=1,2,3, \tag{41}$$

where the circular path radius $R=5$ in., and:

$$\Phi_1(t)=\phi_0+\frac{\beta}{2}t^2, \tag{42}$$

$$\Phi_2(t)=\phi_1+\omega(t-t_1), \tag{43}$$

$$\Phi_3(t)=\phi_0-\frac{\beta}{2}(t_3-t)^2 \tag{44}$$

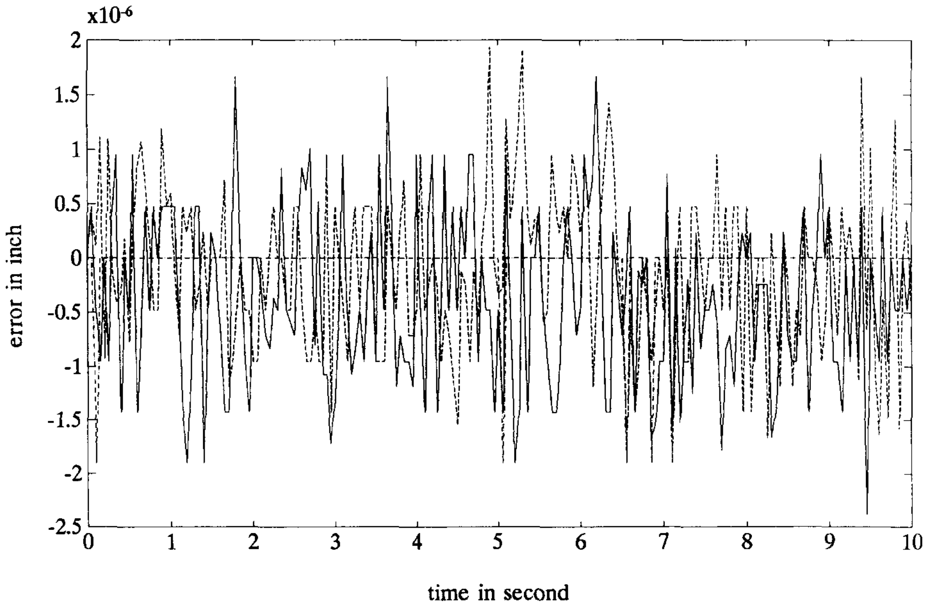


Fig. 9. Circular motion, errors in x, y, z coordinates: x (——), y (– – –) and z (–·–·).

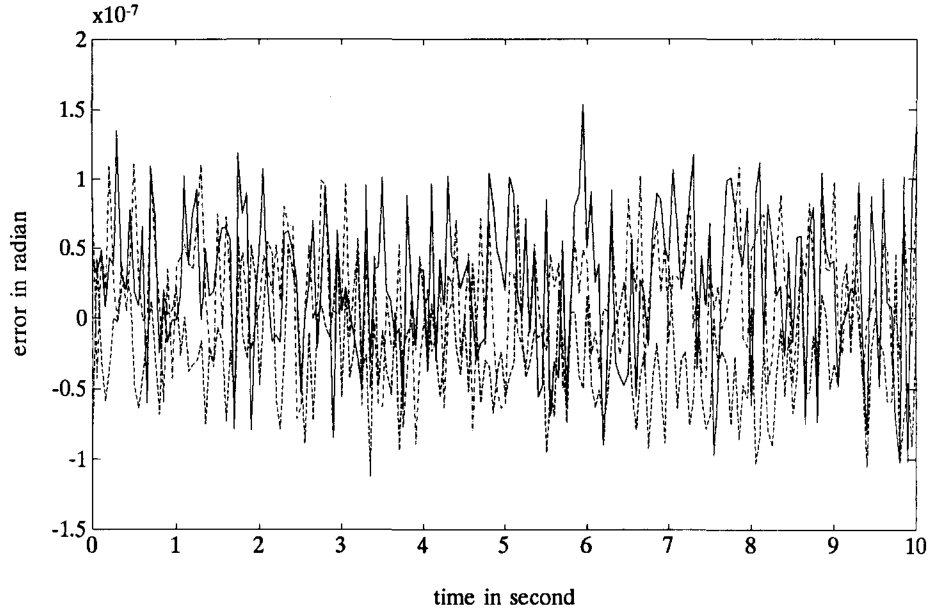


Fig. 10. Circular motion, errors in RPY angles: α (——), β (– – –) and γ (– · – ·).

with $\phi_0 = 0$ rad; $\phi_1(t_1) = \Phi_1(t_1)$ rad, angular velocity $\omega = \beta t_1$ rad s^{-1} and the angular acceleration $\beta = 2\pi/[t_1(t_3 - t_1)]$ rad s^{-1}. The computer simulation was conducted on a SUN workstation with a sampling time of 0.05 s for 10 s and with $t_1 = 1$ s, $t_2 = 9$ s and $t_3 = 10$ s.

The errors in Cartesian positions x, y, z are shown in Fig. 9 where there exists a maximum error of $-2.384\,\mu$in and a maximum RMS error of 0.8737 μin. in the x-position. According to Fig. 10 which presents the errors in RPY angles, a maximum error of 0.154 μrad and a maximum RMS error of 0.0607 μrad occur in the roll angle. The errors in length velocities are reported in Fig. 11 where a maximum error of -0.049 in./s occurs in the fifth actuator length and a maximum RMS error of 0.025 in./s occurs in both the first and the sixth actuator lengths. The complete simulation results are tabulated in Table 1.

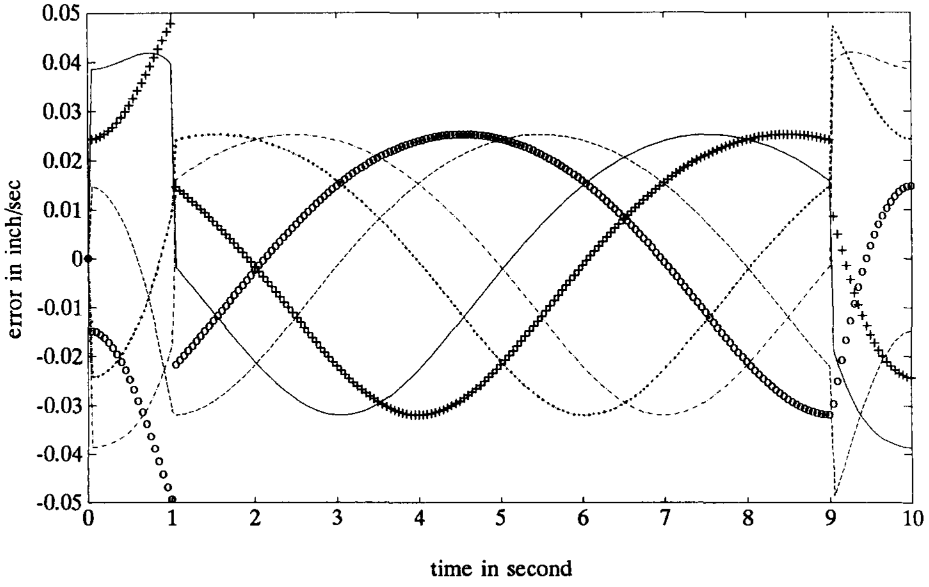


Fig. 11. Circular motion, errors in length velocities: $\dot{l}_1$ (——), $\dot{l}_2$ (– – –), $\dot{l}_3$ (· · · · · ·), $\dot{l}_4$ (+ + +), $\dot{l}_5$ (○○○) and $\dot{l}_6$ (– · – ·).

Table 1. Computer simulation results

	Straight line motion		Circular motion	
	Max error	RMS error	Max error	RMS error
x (μin.)	−1.907	0.7615	−2.384	0.8737
y (μin.)	−2.146	0.7313	−1.9374	0.7367
z (μin.)	−1.907	0.2684	−1.907	0.2691
α (μrad)	0.156	0.0623	0.154	0.0607
β (μrad)	0.103	0.0407	−0.113	0.0461
γ (μrad)	0.106	0.0445	0.111	0.0473
$\dot{l}_1$ (in./s)	0.1546	0.0343	0.0419	0.0250
$\dot{l}_2$ (in./s)	0.1619	0.0361	−0.0486	0.0196
$\dot{l}_3$ (in./s)	−0.0433	−0.0177	0.0473	0.0215
$\dot{l}_4$ (in./s)	0.0955	0.0212	0.0478	0.0216
$\dot{l}_5$ (in./s)	0.1084	0.0239	−0.0494	0.0197
$\dot{l}_6$ (in./s)	−0.0369	−0.0166	0.4919	0.0250

6. CONCLUSION

This paper presented a 6 DOF robotic wrist built at the Goddard Space Flight Center (NASA) to investigate the feasibility of autonomous robotic operations in space. Designed based on the mechanism of the Stewart Platform, the wrist mainly consists of two platforms, six linear actuators and a sensor system and is mounted to a Cincinnati T3 robot to study high-precision robotic assembly. Using vector analysis and coordinate frame assignment, a closed-form solution was obtained for the inverse kinematic transformation to convert Cartesian variables into required actuator lengths. The inverse kinematic equations were then extensively simplified and then applied to develop an iterative solution for the forward kinematic transformation converting actuator lengths to Cartesian variables using the Newton–Raphson method. It was proved that a modified Jacobian matrix relating length velocities to translational velocities and velocities of RPY angles can be obtained as part of the forward kinematic transformation. Results of computer simulation conducted to evaluate the developed transformations and modified Jacobian matrix showed that the conversion accuracies were excellent with very negligible errors. Current research activities focus on implementing the developed transformations for use in real-time control of the robot wrist motion. Control schemes such as a fixed-gain PID controller and an adaptive controller are also currently developed in the IRL to control the motion of the wrist during a high-precision assembly of NASA hardware.

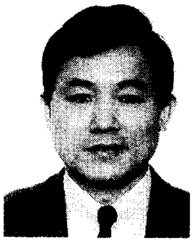
Acknowledgement—The research presented in this paper was conducted at the Catholic University of America under Research Grant Number NAG 5-780, funded by the Goddard Space Flight Center (NASA).

REFERENCES

1. C. C. Nguyen and F. J. Pooran, Kinematic analysis and workspace determination of a 6 DOF CKCM robot end-effector. *J. Mech. Wrkg Technol.* **20,** 283–294 (1989).
2. C. C. Nguyen and F. J. Pooran, Dynamical analysis of 6 DOF CKCM robot end-effector for dual-arm telerobot systems. *J. Robot. Autonom. Syst.* **5,** 377–394 (1989).
3. D. Stewart, A platform with six degrees of freedom. *Proc. Inst. Mech. Engng* **180,** 371–386 (1965–66).
4. J. E. Dieudonne *et al.*, An actuator extension transformation for a motion simulator and an inverse transformation applying Newton–Raphson method. NASA Technical Report D-7067 (1972).
5. R. Hoffman and M. C. McKinnon, Vibration modes of an aircraft simulator motion system. *Proc. The World Congr. for the Theory of Machines and Mechanisms*, pp. 603–606 ASME, New York (1979).
6. H. McCallion and P. D. Truong, The analysis of a six-degree-of-freedom work station for mechanised assembly. *Proc. The Fifth World Congr. for the Theory of Machines and Mechanisms*, pp. 611–616. ASME, New York (1979).
7. K. H. Hunt, *Kinematic Geometry of Mechanisms.* Oxford University Press, London (1978).
8. K. Sugimoto and J. Duffy, Application of linear algebra to screw systems. *Mech. Mach. Theory* **17,** 73–83 (1982).
9. K. H. Hunt, Structural kinematics of in-parallel-actuated robot arms. *Trans. ASME J. Mech. Transmis. Autom. Des.* **105,** 705–712 (1983).
10. T. Premack *et al.*, Design and implementation of a compliant robot with force feedback and strategy planning software. NASA Technical Memorandum 86111 (1984).
11. C. C. Nguyen, F. J. Pooran and T. Premack, Control of robot manipulator compliance. In *Recent Trends in Robotics: Modeling, Control and Education* (M. Jamshidi *et al.*, Eds), pp. 237–242. North-Holland, New York (1986).
12. D. C. Yang and T. W. Lee, Feasibility study of a platform type of robotic manipulators from a kinematic viewpoint. *Trans. ASME J. Mech. Transmis. Autom. Des.* **106,** 191–198 (1984).
13. E. F. Ficher, A Stewart Platform-based manipulator: general theory and practical construction. *Int. J. Robot. Res.* 157–182 (1986).

14. K. Sugimoto, Kinematic and dynamic analysis of parallel manipulators by means of motor algebra, *ASME J. Mech. Transmis. Autom. Des.* **108,** 1–5 (1986).
15. K. M. Lee, A. Chao and D. K. Shak, A three degrees of freedom in-parallel actuated manipulator. *Proc. IASTED Int. Conf.,* pp. 134–138 (1986).
16. C. C. Nguyen and F. J. Pooran, Learning-based control of a closed-kinematic chain robot end-effector performing repetitive tasks. *Int. J. Microcomput. Applic.* **9,** 9–15 (1990).
17. C. C. Nguyen and F. J. Pooran, Adaptive force/position control of robot manipulators with closed-kinametic chain mechanism. In *Robotics and Manufacturing: Recent Trends in Research, Education and Application* (M. Jamshidi *et al.,* Eds), pp. 177–186. ASME Press, New York (1988).
18. F. Behi, Kinematic analysis for a six-degree-of-freedom 3-PRPS parallel mechanism. *IEEE J. Robot. Autom.* **5,** 561–565 (1988).
19. C. C. Nguyen, S. Antrazi and Z.-L. Zhou, Trajectory planning and kinematic control of a Stewart Platform-based manipulator. *5th Int. Conf. on CAD/CAM Robotics and Factories of the Future,* Norfolk, VA (1990).
20 W. H. Press *et al., Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press (1988).
21. K. S. Fu *et al., Robotics: Control, Sensing, Vision and Intelligence.* McGraw-Hill, New York (1987).

AUTHORS' BIOGRAPHIES*



Zhou Zhen-Lei—Zhou Zhen-Lei received the Master of Science degree in Modern Physics from the University of Science & Technology of China (USTC), Hefei, in 1981; and the Master of Science degree in Electrical Engineering from the Catholic University of America (CUA), Washington, DC, in 1988, respectively. He taught at electronics courses and laboratories from 1973 to 1978, and was a lecturer of nuclear electronics and computer applications from 1981 to 1984 at USTC. Afterwards, he first worked at the Johns Hopkins University and then at CUA as a visiting scholar until he became a Graduate Student at CUA, in 1987. He has extensive experience in electronics and computer applications, has designed several electronic instruments and published several papers in the above areas. Since 1989 he has been a Graduate Research Assistant in the Robotics & Control Laboratory of the Electrical Engineering Department at CUA and has been involved with several NASA projects in control and robotics. He has published more than 10 technical papers in journals and at conferences. Currently, he is engaged in a project implementing robot control schemes for a U.S. Air Force contract and completing his Ph.D. dissertation in the Electrical Engineering Department, CUA. His research interests include kinematics, dynamics and control in robotics computer control and real-time applications and digital instrumentation.

Sami C. Antrazi—Sami C. Antrazi received his B.E. degree in Electrical Engineering from the American University of Beirut, Lebanon in 1989. He is currently completing his Master of Science degree at the Catholic University of America, Washington, DC. Since 1990 he has been a Research Assistant in the Robotics and Control Laboratory at the Catholic University of America. He has been involved with NASA sponsored research projects in the area of robotics and control and has published several journal and conference papers in this research area. His research interests include kinematics, and control and design of robot manipulators.

*To avoid duplication, the biographies of Charles C. Nguyen and Charles E. Campbell Jr, the other authors of this paper, are given elsewhere in this issue.