THE CATHOLIC UNIVERSITY OF AMERICA
DEPARTMENT OF ELECTRICAL ENGINEERING

ADVANCED CONTROL SCHEMES AND KINEMATIC ANALYSIS FOR A KINEMATICALLY REDUNDANT 7 DOF MANIPULATOR

NAG5-1124

Charles C. Nguyen
Principal Investigator and Associate Professor
and
Zhen-Lei Zhou
Graduate Research Assistant

submitted to
Mr. Gary E. Mosier
Code 712.1
Goddard Space Flight Center (NASA)
Greenbelt, Maryland

February 1990

SUMMARY OF THE REPORT

This report presents the research results obtained from the research grant entitled "Development of Advanced Control Schemes for Telerobot Manipulators," funded by the Goddard Space Flight Center (NASA) under a research grant with Grant Number NAG 5-1124, for the period between August 1st, 1990 and February 1st, 1990.

This report deals with the kinematic analysis and control of a kinematically redundant manipulator, which is the slave arm of a telerobot system recently built at Goddard Space Flight Center (GSFC) to serve as a testbed for investigating research issues in telerobotics. A forward kinematic transformation is developed in its most simplified form, suitable for real-time control applications, and the manipulator Jacobian is derived using the vector cross product method. Using the developed forward kinematic transformation and quaternion representation of orientation matrices, we perform computer simulation to evaluate the efficiency of the Jacobian in converting joint velocities into Cartesian velocities and to investigate the accuracy of Jacobian pseudo-inverse for various sampling times. The equivalence between Cartesian velocities and quaternion is also verified using computer simulation. Three control schemes are proposed and discussed for controlling the motion of the slave arm end-effector.

1 Introduction

Recently research in the area of kinematically redundant[1] manipulators has been very active [1,8] because they have many advantages as compared to non-redundant manipulators. A robot manipulator is classified as *kinematically redundant* if its number of degrees of freedom (**DOF**) is greater than that of task space coordinates. The extra DOFs enable the redundant manipulator to avoid singularities and obstacles, to keep the joint variables within their physical limitations, to minimize kinetic energy and to provide greater dexterity. Consequently the above advantages have motivated robot designers to adopt redundant manipulators for telerobots which will replace or assist astronauts in performing operations in space. Goddard Space Flight Center (**GSFC**) is developing a Flight Telerobot Servicer (**FTS**) to carry out a variety of tasks including assembly of NASA space station and platforms, inspection, servicing and maintenance on the space station etc. An integral part of the research facilities for the FTS project is a dual-arm telerobot system which consists mainly of a pair of mini-master controllers and a pair of slave arms, each of which is a redundant manipulator possessing 7 DOFs. The telerobot system serves as a testbed for investigating research issues in telerobotics such as zero-g operation, teleoperated and autonomous control, dual-arm manipulators, advanced control of redundant manipulators, hierarchical control etc. [9].

In this report, we present some mathematical developments which will be employed in computer simulations and real-time control of the slave arm motion. In particular, we will focus on the manipulator forward kinematics, differential motion analysis and propose three control schemes for the slave arms. This report is organized as follows. Next section will give an overview of the GSFC telerobot system and briefly describe the structure of the slave arm. Then the forward kinematic transformation for the manipulator is derived in its most simplified form using Denavit-Hartenberg notation. After that, we obtain the manipulator Jacobian using the vector cross product method and then discuss the pseudo-inverse of the Jacobian. Computer simulation is then conducted to evaluate the efficiency of the Jacobian in converting joint velocities into Cartesian velocities, to investigate the accuracy of the Jacobian pseudo-inverse for various sampling times and to verify the equivalence between Cartesian velocities and quaternion. Finally three control schemes are proposed and discussed for controlling the position and/or force the slave arm end-effector performing compliant and non-compliant motion.

2 The Redundant Manipulator

GSFC recently has developed a dual-arm telerobot system which serves as a testbed [9] for investigation of research issues in telerobotics such as robot control algorithms, dual-arm teleoperation, dynamic simulation techniques, collision avoidance, end-effector

[1]The term "redundant" is often used instead of "kinematically redundant"

design, and hierarchical control using high-level programming languages, etc. The main components of the system include a pair of slave arms, each of which is a Robotics Research Corporation (**RRC**) K-1607 manipulator and a pair of 6-DOF Kraft Mini-Master (**KMM**) hand controllers manufactured by Kraft Telerobotics, Inc. Figure 1 shows the RRC K-1607 manipulator which is a kinematically redundant manipulator possessing 7 DOFs. The slave arm motion can be controlled in a teleoperation mode by a human operator using the master arm system or autonomously controlled by a set of computer programs. Following the convention in [10], 8 coordinate frames are assigned to the manipulator as illustrated in Figure 2 showing the manipulator in its home configuration with all joint angles being zero. Each ith frame {i} is characterized by its coordinate axes $\mathbf{x}_i$, $\mathbf{y}_i$, $\mathbf{z}_i$ and its origin $\mathbf{0}_i$ for i = 0,1,2,...,7. The Denavit-Hartenberg parameters for the assigned coordinate frames are listed in Table 1 given below:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0^o	0.000in	0.0in	θ_1
2	-90^o	0.000in	0.0in	θ_2
3	90^o	5.625in	27.0in	θ_3
4	-90^o	4.250in	0.0in	θ_4
5	90^o	-4.250in	27.0in	θ_5
6	-90^o	3.125in	0.0in	θ_6
7	90^o	-3.125in	0.0in	θ_7

Table 1: D-H parameters of the RRC K-1607 manipulator.

3 The Manipulator Forward Kinematics

In this section, we consider the forward kinematics of the above redundant manipulator. Forward kinematics is useful if one employs a Cartesian-space control scheme and measurements of joint variables are available from joint sensors. A forward kinematic transformation is developed to convert the 7 joint angles θ_i for i=1,2,...,7 of the manipulator into the corresponding position and orientation, referred here to as *configuration* of the manipulator end-effector frame, Frame {7}, with respect to the base frame, Frame {0}. The configuration of the ith frame with respect to the (i-1)th frame is represented by the following homogeneous transformation matrix:

$$ {}^{i-1}_{i}\mathbf{T} = \begin{bmatrix} {}^{i-1}_{i}\mathbf{R} & {}^{i-1}_{i}\mathbf{p} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{1} $$

$$ = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i \cos\alpha_{i-1} & \cos\theta_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i \sin\alpha_{i-1} \\ \sin\theta_i \sin\alpha_{i-1} & \cos\theta_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i \cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

for i=1,2,...,7 where ${}^{i-1}_{i}\mathbf{R}$ and ${}^{i-1}_{i}\mathbf{p}$ represent the orientation and position of the ith frame expressed in the (i-1)th frame, respectively. The transformation ${}^{0}_{7}\mathbf{T}$ consisting of the orientation matrix ${}^{0}_{7}\mathbf{R}$ and the position vector ${}^{0}_{7}\mathbf{p}$ expresses the configuration of Frame {7} with respect to Frame {0} and is computed by

$$ {}^{0}_{7}\mathbf{T} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T}\,{}^{3}_{4}\mathbf{T}\,{}^{4}_{5}\mathbf{T}\,{}^{5}_{6}\mathbf{T}\,{}^{6}_{7}\mathbf{T}. \tag{3} $$

Carrying out the matrix multiplications in (3) and performing intensive trigonometric simplifications we obtain

$$ {}^{0}_{7}\mathbf{T} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

where

$$ \left.\begin{aligned} n_x &= s_7 h_1 + c_7 j_1 \\ s_x &= c_7 h_1 - s_7 j_1 \\ a_x &= s_6 h_2 + c_6 g_2 \\ p_x &= a_6 j_1 + a_5 h_2 + d_5(s_4 f_1 + c_1 s_2 c_4) \\ &\quad + a_4 g_1 + a_3 f_2 + c_1 j_3 \end{aligned}\right\} \tag{5} $$

$$ \left.\begin{aligned} n_y &= s_7 h_3 + c_7 j_2 \\ s_y &= c_7 h_3 - s_7 j_2 \\ a_y &= s_6 h_4 + c_6 g_4 \\ p_y &= a_6 j_2 + a_5 h_4 + d_5(s_4 f_4 + s_1 s_2 c_4) \\ &\quad + a_4 g_3 + a_3 f_4 + s_1 j_3 \end{aligned}\right\} \tag{6} $$

$$ \left.\begin{aligned} n_z &= s_7 g_5 + c_7 h_5 \\ s_z &= c_7 g_5 - s_7 h_5 \\ a_z &= s_6 g_6 + c_6 f_6 \\ p_z &= a_6 h_5 + a_5 g_6 + d_5(c_2 c_4 - s_2 c_3 s_4) \\ &\quad + a_4 f_5 - a_3 s_2 c_3 - a_2 s_2 + d_3 c_2 \end{aligned}\right\} \tag{7} $$

$$ \left.\begin{aligned} f_1 &= -c_1 c_2 s_3 - s_1 c_3 \\ f_2 &= c_1 c_2 c_3 - s_1 s_3 \\ f_3 &= -s_1 c_2 s_3 - c_1 c_3 \\ f_4 &= s_1 c_2 c_3 + c_1 s_3 \\ f_5 &= -s_2 c_3 c_4 - c_2 s_4 \\ f_6 &= -s_1 c_3 s_4 + c_2 c_4 \end{aligned}\right\} \tag{8} $$

$$ \left.\begin{aligned} g_1 &= -c_1 s_2 s_4 + c_2 f_2 \\ g_2 &= c_1 s_2 c_4 + s_4 f_2 \\ g_3 &= -s_1 s_2 s_4 + c_4 f_4 \\ g_4 &= s_1 s_2 c_4 + s_4 f_4 \\ g_5 &= s_2 s_3 c_5 - s_5 f_5 \\ g_6 &= s_2 s_3 s_5 + c_5 f_5 \end{aligned}\right\} \tag{9} $$

$$\left.\begin{aligned} h_1 &= c_5 f_1 - s_5 g_1 \\ h_2 &= s_5 f_1 + c_5 g_1 \\ h_3 &= c_5 f_3 - s_5 g_3 \\ h_4 &= s_5 f_3 + c_5 g_3 \\ h_5 &= c_6 g_6 - s_6 f_6 \end{aligned}\right\} \tag{10}$$

$$\left.\begin{aligned} j_1 &= c_6 h_2 - s_6 g_2 \\ j_2 &= c_6 h_4 - s_6 g_4 \\ j_3 &= d_3 s_2 + a_2 c_2, \end{aligned}\right\} \tag{11}$$

and we have used the compact notations, $c_i \equiv \cos\theta_i$ and $s_i \equiv \sin\theta_i$. We also note that in (5)-(11), a_{i-1} and d_i for i=1,2,...,7 are manipulator parameters listed in Table 1. Since matrix multiplications are avoided in (5)-(11), the computation time for the forward kinematics is greatly reduced. As a consequence, the derived forward kinematic equations are highly suitable for real-time control implementation.

4 Differential Motion Analysis

This section is devoted to the analysis of the manipulator differential motion. In the following, we first compute the manipulator Jacobian using the vector cross product method and then discuss its inverse computation using the method of Moore-Penrose pseudo-inverse. After that, we review the quaternion representation of orientation which will be used in the computer simulation study.

4.1 The Manipulator Jacobian

To be compatible with the coordinate frame assignments according to the convention given in [10], the vector cross product method [11] is slightly modified and is applied to derive the manipulator Jacobian. According to [11] the manipulator Jacobian is obtained by

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \mathbf{J}_3 & \mathbf{J}_4 & \mathbf{J}_5 & \mathbf{J}_6 & \mathbf{J}_7 \end{bmatrix} \tag{12}$$

where

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{b}_i \times \mathbf{p}_i \\ \mathbf{b}_i \end{bmatrix}, \quad \text{i=1,2,...,7} \tag{13}$$

and $\mathbf{b}_i$, defined as the unit vector pointing along the axis of motion of Joint i expressed in Frame $\{\mathbf{0}\}$, is given by

$$\mathbf{b}_i = {}^0_1\mathbf{R}\,{}^1_2\mathbf{R}\,\ldots{}^{i-1}_i\mathbf{R}\,\mathbf{b}_0, \quad \text{i=1,2,...,7} \tag{14}$$

with

$$\mathbf{b}_0 = [0\ 0\ 1]^T \tag{15}$$

and $\mathbf{p}_i$, defined as the vector pointing from the origin of the ith-frame to the origin of Frame $\{7\}$, expressed in Frame $\{0\}$, is obtained from

$$\begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} = {}^0_7\mathbf{T}\,\mathbf{x}_0 - {}^0_i\mathbf{T}\,\mathbf{x}_0, \quad i=1,2,\ldots,7 \tag{16}$$

with

$$\mathbf{x}_0 = [0\ 0\ 0\ 1]^T \tag{17}$$

and $\times$ indicates the vector cross product. A Fortran program was written to compute the manipulator Jacobian $\mathbf{J}$ and because of the space constraint in this report, only the first three columns of the Jacobian are given below:

$$\mathbf{J}_1 = \begin{bmatrix} -p_y \\ -p_x \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \quad \mathbf{J}_2 = \begin{bmatrix} c_1 p_z \\ s_1 p_z \\ -s_1 p_y - c_1 p_x \\ -s_1 \\ c_1 \\ 0 \end{bmatrix} \tag{18}$$

and

$$\mathbf{J}_3 = \begin{bmatrix} s_1 s_2(p_z + a_2 s_2 - d_3 c_2) - c_2(p_y - a_2 s_1 c_2 - d_3 s_1 s_2) \\ -c_1 s_2(p_z + a_2 s_2 - d_3 c_2) + c_2(p_x - a_2 c_1 c_2 - d_3 c_1 s_2) \\ c_1 s_2(p_y - a_2 s_1 c_2 - d_3 s_1 s_2) - s_1 s_2(p_x - a_2 c_1 c_2 - d_3 c_1 s_2) \\ c_1 s_2 \\ s_1 s_2 \\ c_2 \end{bmatrix}. \tag{19}$$

4.2 The Jacobian Inverse

The Cartesian velocity vector $\dot{\mathbf{x}}(t)$ are related to joint angle velocity vector $\dot{\mathbf{q}}(t)$ by the Jacobian $\mathbf{J}$ as

$$\dot{\mathbf{x}}(t) = \mathbf{J}\dot{\mathbf{q}}(t). \tag{20}$$

The inverse solution to (20) which minimizes the weighted quadratic form $\dot{\mathbf{q}}^T\mathbf{W}^{-1}\dot{\mathbf{q}}$, is given by [14]

$$\dot{\mathbf{q}}(t) = \mathbf{J}_W^{\dagger}\dot{\mathbf{x}}(t) + (\mathbf{I}_7 - \mathbf{J}_W^{\dagger}\mathbf{J})\mathbf{z} \tag{21}$$

where $\mathbf{J}_W^{\dagger}$, the *Weighted Pseudo-Inverse* of the Jacobian $\mathbf{J}$ is given by

$$\mathbf{J}_W^{\dagger} = \mathbf{W}\mathbf{J}^T[\mathbf{J}\mathbf{W}\mathbf{J}^T]^{-1} \tag{22}$$

$\mathbf{W}$, the *Weighting Matrix* is a symmetric matrix, $\mathbf{z}$ denotes an arbitrary joint velocity vector and the second term of (21) belongs to the null space of $\mathbf{J}$. Vector $\mathbf{z}$ can be

selected for optimization purposes. When $\mathbf{W} = \mathbf{I}$ and $\mathbf{z} = 0$, then (22) reduces to the well-known *Moore-Penrose Pseudo-Inverse* of the Jacobian given by

$$\mathbf{J}^{\dagger} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1} \tag{23}$$

which provides the minimum norm least-squares solution.

4.3 Quaternion Representation

The *Quaternion* consisting of a scalar η and a vector $\mathbf{s} = [\beta\ \gamma\ \xi]^T$, also called *Euler Parameters* of an orientation matrix $\mathbf{R}$ specified by

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{24}$$

is obtained by an operator $\mathbf{Q}$ defined by

$$(\eta, \mathbf{s}) = \mathbf{Q}\{\mathbf{R}\} \tag{25}$$

such that

$$\begin{array}{rcl} \eta & = & \sqrt{1 + r_{11} + r_{22} + r_{33}}/2 \\ \beta & = & (r_{32} - r_{23})/4\eta \\ \gamma & = & (r_{13} - r_{31})/4\eta \\ \xi & = & (r_{21} - r_{12})/4\eta. \end{array} \tag{26}$$

On the other hand, an orientation matrix R can be computed from its quaternion by the inverse operator defined by

$$\mathbf{R} = \mathbf{Q}^{-1}\{\eta, \mathbf{s}\} \tag{27}$$

so that

$$\mathbf{R} = (\eta^2 - \mathbf{s}^T\mathbf{s})\,\mathbf{I}_6 + 2\mathbf{s}\mathbf{s}^T - 2\eta\mathbf{s}^{\times} \tag{28}$$

where

$$\mathbf{s}^{\times} = \begin{bmatrix} 0 & -\xi & \gamma \\ \xi & 0 & -\beta \\ -\gamma & \beta & 0 \end{bmatrix}. \tag{29}$$

Now considering two orientation matrices ${}^0_1\mathbf{R}$ and ${}^0_2\mathbf{R}$ which represent the orientation of Frame $\{\mathbf{1}\}$ and $\{\mathbf{2}\}$ with respect to Frame $\{\mathbf{0}\}$, respectively, we can write

$${}^0_2\mathbf{R} = {}^0_1\mathbf{R}\ {}^1_2\mathbf{R}. \tag{30}$$

In (30), since ${}^1_2\mathbf{R}$ is postmultiplied to ${}^0_1\mathbf{R}$, ${}^1_2\mathbf{R}$ represents a rotation of Frame $\{\mathbf{1}\}$ about Frame $\{\mathbf{1}\}$ to move Frame $\{\mathbf{1}\}$ to Frame $\{\mathbf{2}\}$. ${}^1_2\mathbf{R}$ can also be interpreted as the orientation of Frame $\{\mathbf{2}\}$ with respect to Frame $\{\mathbf{1}\}$. However if the rotation is performed about Frame $\{\mathbf{0}\}$, then we should write

$${}^0_2\mathbf{R} = \overline{{}^1_2\mathbf{R}}\ {}^0_1\mathbf{R} \tag{31}$$

where $\overline{{}^1_2\mathbf{R}}$ represents the rotation of Frame $\{\mathbf{1}\}$ about Frame $\{\mathbf{0}\}$ to bring Frame $\{\mathbf{1}\}$ to Frame $\{\mathbf{2}\}$ and can be computed from (31) as

$$\overline{{}^1_2\mathbf{R}} = {}^0_2\mathbf{R}\ {}^0_1\mathbf{R}^{-1} = {}^0_2\mathbf{R}\ {}^0_1\mathbf{R}^T. \tag{32}$$

Suppose $(\eta_1,\mathbf{s}_1)$ and $(\eta_2,\mathbf{s}_2)$ are the quaternions of ${}^0_1\mathbf{R}$ and ${}^0_2\mathbf{R}$, respectively. Then the quaternion of $\overline{{}^1_2\mathbf{R}}$ can be expressed in terms of those of ${}^0_1\mathbf{R}$ and ${}^0_2\mathbf{R}$ as follows:

$$\delta\eta = \eta_1\eta_2 + \mathbf{s}_1^T\mathbf{s}_2 \tag{33}$$

and

$$\delta\mathbf{s} = \eta_1\mathbf{s}_2 - \eta_2\mathbf{s}_1 + \mathbf{s}_1^\times\mathbf{s}_2. \tag{34}$$

Now we are interested in finding how the quaternion of $\overline{{}^1_2\mathbf{R}}$ are related to differential rotations introduced in [15]. According to [15], if the orientation difference between Frame $\{\mathbf{1}\}$ and Frame $\{\mathbf{2}\}$ is small then

$${}^0_2\mathbf{R} \approx \begin{bmatrix} 1 & -\delta_z & -\delta_y \\ \delta_z & 1 & -\delta_x \\ -\delta_y & \delta_x & 1 \end{bmatrix} {}^0_1\mathbf{R} \tag{35}$$

where δ_x, δ_y, and δ_z denote the differential rotations of Frame $\{\mathbf{1}\}$ made in any order about the x, y, and z axes of Frame $\{\mathbf{0}\}$, respectively to bring Frame $\{\mathbf{1}\}$ to Frame $\{\mathbf{2}\}$.

Comparing (31) and (35), we obtain

$$\overline{{}^1_2\mathbf{R}} \approx \begin{bmatrix} 1 & -\delta_z & -\delta_y \\ \delta_z & 1 & -\delta_x \\ -\delta_y & \delta_x & 1 \end{bmatrix}. \tag{36}$$

From (36) the differential rotations δ_x, δ_y, and δ_z can be computed from the quaternion of $\overline{{}^1_2\mathbf{R}}$ by taking the quaternion on both sides of (36) using (26) and solving for δ_x, δ_y, and δ_z as follows:

$$\begin{aligned} \delta_x &\approx 2\,\delta\beta \\ \delta_y &\approx 2\,\delta\gamma \\ \delta_z &\approx 2\,\delta\xi \end{aligned}. \tag{37}$$

Equations given in (37) provides a relatively accurate computation of the rotation velocities if the quaternion of $\overline{{}^1_2\mathbf{R}}$ is given.

5 Computer Simulation Study

This section presents the results of the computer simulation study conducted to verify the above mathematical developments. The study is composed mainly of three parts, the first part is devoted to investigate the efficiency of Jacobian in converting joint angle

velocities to Cartesian velocities, the second to evaluate the accuracy of the pseudo-inverse Jacobian and the third to verify the equivalence between Cartesian velocities and quaternion representation. Computer simulation is repeated for various sampling times so that a maximum permitted sampling time can be established for an acceptable conversion accuracy.

5.1 Part 1: Joint to Cartesian Velocities

Figure 3 represents the computer simulation scheme used for Part 1 and Part 2. In the upper loop, a set of test joint angle trajectories are converted to the corresponding Cartesian trajectories using the derived forward kinematic transformation. The orientation matrix ${}^0_7\mathbf{R}$ is used to compute the differential rotations by employing (35). In the lower loop, the joint velocities which are obtained by differentiating the test joint angle trajectories are supplied to the Jacobian which produces the corresponding Cartesian velocities. The Cartesian velocities obtained from the upper loop are then compared with those from the lower loop to compute the conversion errors. Figure 4 shows the error between the x-axis velocities $\dot{p}_{xJ}$ (from Jacobian) and $\dot{p}_x$ for two different sampling times. The maximum error is about 0.5 inch/sec for a sampling time of 10 msec (indicated by solid line) and about 6 inch/sec for a sampling time of 100 msec (indicated by asterisk line). Figure 5 presents the error between the x-axis angular velocities ω_{xJ} and ω_x for sampling times of 10 msec (solid line) and 100 msec (dotted line). The maximum angular velocity errors are about 0.5 minch/sec and 5 minch/sec for sampling times of 10 msec and 100 msec, respectively.

5.2 Part 2: Cartesian to Joint Velocities

In the lower loop of Figure 3, the Cartesian velocities provided by the Jacobian are supplied to the Jacobian pseudo-inverse which is computed by Equation (23) and whose outputs are compared with the joint velocities. Figures 6 and 7 show the joint angle velocities $\dot{\theta}_{1J}$ (from the pseudo-inverse) and $\dot{\theta}_1$ for sampling times of 10 msec and 100 msec, respectively. According to the obtained results, the pseudo-inverse does not provide adequate conversion of Cartesian velocities to joint velocities at a sampling time of 100 msec. The velocity conversion is excellent at a sampling time of 10 msec.

5.3 Part 3: Quaternion Representation

Figure 8 illustrates the computer simulation scheme used to verify the equivalence between Cartesian velocities and quaternion representation. In the upper loop of Figure 8, using Equations (33)-(34), we compute the quaternion of the orientation difference given by

$$\Delta\, {}^0_7\mathbf{R} = {}^0_7\mathbf{R}(t_i)\ {}^0_7\mathbf{R}^T(t_{i-1}) \tag{38}$$

where ${}^{0}_{7}\mathbf{R}(t_i)$ denotes the orientation matrix evaluated a the ith sampling during the computer simulation. In the lower loop, the quaternion can be computed from the output of the Jacobian by employing Equation (37) and then compared with the quaternion of the upper loop to determine the deviations. Figures 9 and 10 show the simulation results of the errors of $\delta\beta$ (solid line) and $\delta\gamma$ (asterisk line) for sampling times of 10 msec and 100 msec, respectively. In the case of 100 msec sampling time, the maximum errors for $\delta\beta$ and $\delta\gamma$ are 15 minch/sec and 0.15 minch/sec, respectively and are negligible in the case of 10 msec sampling time.

6 Proposed Control Schemes

In this section, we consider the problem of controlling the compliant and non-compliant motion of the slave arm end-effector. When the slave arm performs non-compliant motion, i.e. without being in contact with the environment, it is sufficient to employ *pure position control schemes* whose error-correcting forces are computed based only on the position errors. However during a compliant motion mode in which the slave arm end-effector is constantly in contact with the environment a *hybrid position/force*[2] *control scheme* which controls not only the position of the end-effector but also the contact forces it applies on the environment, should be applied. In the following, we present and discuss three control schemes which have been under study for controlling the slave arm motion and briefly report some preliminary findings.

6.1 Joint-Space Adaptive Control Scheme

Figure 11 shows the organization of a joint-space control scheme which has been considered for controlling the non-compliant motion of the slave arm. In the control scheme, actual joint angles measured by 7 joint sensors are compared with desired joint angles which are obtained from desired configuration of the slave arm end-effector through the inverse kinematics. The joint variable errors then serve as inputs to a set of proportional-derivative (**PD**)- controllers whose gains are adjusted by an adaptation law so that the error-correcting joint forces provided by the controllers track the slave arm end-effector along a desired path. The adaptation law was derived using the Lyapunov theory and the concept of model reference adaptive control (**MRAC**) under the assumption that the slave arm performs *slowly varying* motion. From the fact that the derived adaptation law does not have to evaluate the slave arm dynamics, it is computationally fast and very attractive to real-time control. Computer simulation results reported in [6] showed that the slave arm end-effector under the control of the above scheme can track several test paths with minimal tracking errors under sudden change in payload. The developed joint-space control scheme is currently implemented by GSFC for real-time control of the slave arm motion.

[2]In this report, "position" implies both "position and orientation" and "force" both "force and torque".

6.2 Cartesian-Space Adaptive Control Scheme

An adaptive control scheme in Cartesian space is presented in Figure 12. As the figure shows, feedback information of the actual joint variables are converted into the corresponding Cartesian variables by the forward kinematic transformation. The actual Cartesian variables are then compared with the desired Cartesian variables representing the desired configuration of the slave arm end-effector, and the corresponding Cartesian errors are supplied to a set of PD-controllers whose gains are adjusted by an adaptation law. The adaptation law is designed such that the joint forces which are obtained by transforming the Cartesian forces produced by the adaptive PD controllers using the Jacobian transpose will track the end-effector along desired paths. Extending the development in [6], an adaptation law was derived and presented in [5] under the assumption of slowly-varying motion. Computer simulation study is currently conducted to investigate the performance of the Cartesian-space control scheme and simulation results will be reported in [7]

6.3 Hybrid Position/Force Control Scheme

Figure 13 presents a hybrid position/force control scheme whose structure is similar to that introduced in [12] except that the controller gains of the current control scheme are adjusted by an adaptation law. As Figure 13 shows, the control scheme mainly consists of the two control loops, the upper loop for position and the lower for force control. A (6x6) diagonal *compliance selection matrix* S whose main diagonal elements s_{ii} for i = 1,2,...,6 assume either 1 or 0, allows the user to select which DOF to be position-controlled and which to be force-controlled by setting the element s_{ii} properly, namely $s_{ii} = 1$ for the ith DOF to be force-controlled and $s_{ii} = 0$ for the ith DOF to be position-controlled. In other words, the hybrid position/force control scheme allows independent and simultaneous control of position and force. The adaptation law which adjusts the gains of the PD-controllers of the position and force control loops so that the end-effector can follow a desired path while applying desired contact forces on the environment despite disturbances such as varying environment stiffness, is currently under intensive study. Results found for the adaptive hybrid position/force control will be reported in [8].

7 Conclusion

In this report, we have considered the kinematic analysis and control of a 7 DOF manipulator serving as a slave arm of a telerobot system developed at GSFC to investigate the feasibility of telerobotic applications in space. A forward kinematic transformation for the manipulator was derived and simplified for real-time implementation. Employing the method of vector cross product, we obtained the manipulator Jacobian and computed its inverse using the Moore-Penrose pseudo-inverse method. The concept of quater-

nion was reviewed for presenting the orientation of the manipulator end-effector and relationship between quaternion and differential rotations was developed. Computer simulation was performed to verify the efficiency of the Jacobian in converting joint velocities to Cartesian velocities and to investigate the accuracy of Jacobian pseudo-inverse. The equivalence between differential rotations and quaternion was also verified through computer simulation. Simulation results showed that the maximum sampling time which ensures the efficiency of the Jacobian, its pseudo-inverse, and the quaternion representation was about 10 msec. Three control schemes was proposed and discussed for controlling the motion of the slave arm. Current and future activities focus on the implementation of the proposed control schemes.

References

[1] Klein, C.A., and Huang, C.H., "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," *IEEE Trans. Sys., Man, and Cyber.*, pp. 245-250, March 1983.

[2] Hanafusa, H., Yoshikawa, T., and Nakamura, Y., "Analysis and Control of Articulated Robot Arms with Redundancy," *Proc. 8th IFAC Triennial World Congress*, Kyoto, Japan, pp. 1927-1932, 1981.

[3] Yoshikawa, T., "Analysis and Control of Robot Manipulators with Redundancy," *Proc. 1st Intern. Symp. on Robotics Research*, New Hampshire, pp. 735-747, 1983.

[4] Baillieul, J., Hollerbach, J., and Brockett, R., "Programming and Control of Kinematically Redundant Manipulators," *Proc. 23rd IEEE Conf. on Decision and Control*, pp. 768-774, 1984.

[5] Nguyen, C.C., Zhou, Z.L., and Mosier, G.E., "Lyapunov-Based Direct Adaptive Control of Kinematically Redundant Telerobot Manipulators," *Proc. IASTED International Symposium on Adaptive and Knowledge-Based Control and Signal Processing*, Honolulu, Hawaii, pp. 14-18 , August 1989.

[6] Nguyen, C.C., Zhou, Z.L., and Mosier, G.E., "Joint-Space Adaptive Control of a Redundant Telerobot Manipulator,"*Proc. 4th IEEE International Symposium on Intelligent Controls*,Albany, New York, pp. 59-65, September 1989.

[7] Nguyen, C.C., Zhou, Z.L., and Mosier, G.E., "A Computationally Efficient Error-Based Adaptive Control Scheme for Kinematically Redundant Manipulators," *Proc. 3rd International Symposium on Robotics and Manufacturing*, Vancouver, Canada, July 1990.

[8] Nguyen, C.C., Zhou, Z.L., and Mosier, G.E., "Compliant Control of a Redundant Telerobot Manipulator via a Position/Force Control Scheme," *Proc. 3rd International Symposium on Robotics and Manufacturing*, Vancouver, Canada, July 1990.

[9] Schnurr, R., O'brien, M., and Cofer, S., "The Goddard Space Flight Center (GSFC) Robotics Technology Testbed," *NASA Conference on Space Telerobotics*, Pasadena, January 1989.

[10] Craig, J.J., Introduction to Robotics, 2nd Edition, *Addison-Wesley, Inc.*, Reading, Ma, 1989.

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

[12] Raibert, M.H. and Craig, J.J., "Hybrid Position/Force Control of Manipulators," *Trans. ASME, Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, pp. 126-133, 1981.

[13] Yuan, J.S.C., "Closed-Loop Manipulator Control Using Quaternion Feedback," *IEEE Journal of Robotics and Automation*, Vol. 4., No. 4, pp. 434-440, August 1988.

[14] Burdick J. and Seraji, H., "Characterization and Control of Self-Motions in Redundant Manipulators," Proc. NASA Conf. on Space Telerobotics, Pasadena, CA, 1989.

[15] Paul, R.P., Robot Manipulators: Mathematics, Programming, and Control, *MIT Press*, 1981.

[16] Seraji, H., "Configuration Control of Redundant Manipulators: Theory and Implementation," to appear in *IEEE Journal of Robotics and Automation*, 1989.

[17] Colbaugh, R.D., "Adaptive Position and Force Control of Redundant Robot Manipulators," in *Robotics and Manufacturing: Recent Trends in Research, Education, and Application*, edited by M. Jamshidi et al, ASME Press, New York, pp. 319-328, 1988.

[18] Egeland, O., "Cartesian Control of a Hydraulic Redundant Manipulator," *Proc. IEEE Intern. Conf. on Robotics and Automation*, Raleigh, pp. 1081-1087, April 1987.

[19] Hsu, P., Hauser, J., Sastry, S., "Dynamic Control of Redundant Manipulators," *Proc. IEEE Intern. Conf. on Robotics and Automation*, pp. 1081-1087, Philadelphia, 1988.

[20] Kazerounian, K., Wang, S., "Global versus Local Optimization in Redundancy Resolution of Robotic Manipulators," *International Journal of Robotics Research* Vol. 7, No. 5, pp. 3-12, 1988.

[21] Nguyen, C.C., Pooran, F.J., "Joint-Space Adaptive Control of Robot End-Effectors Performing Slow and Precise Motions," *Proc. 21st Southeastern Symposium on System Theory*, Florida, pp. 547-552, March 1989.

[22] 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*, Chapter 4, edited by M. Jamshidi et al, ASME Press, New York, pp. 177-186, 1988.

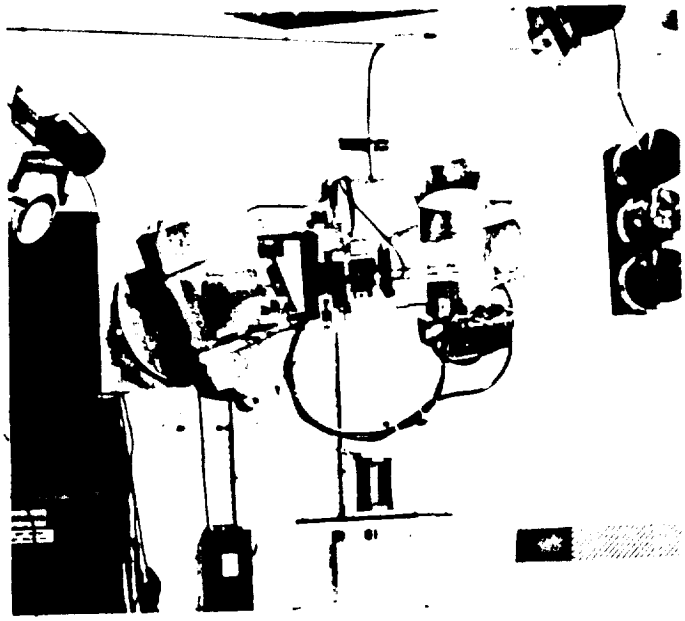

Figure 1: The RRC K-1607 slave arm

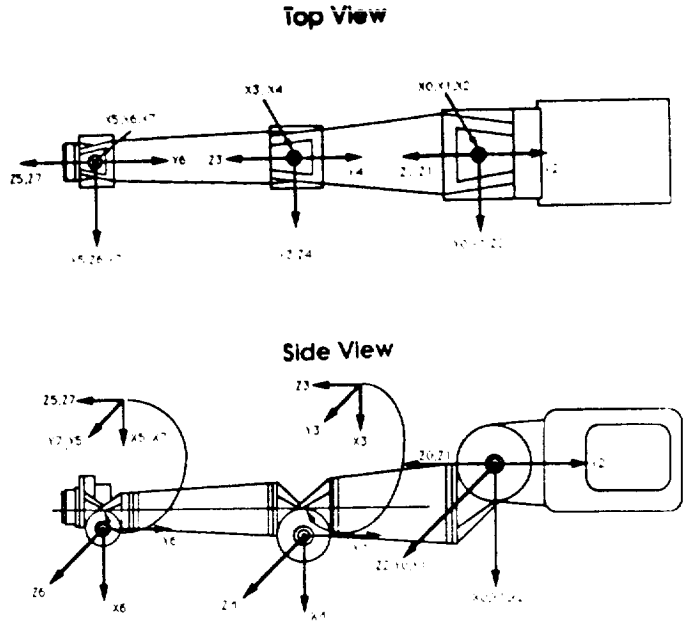

Figure 2: Assignment of the coordinate frames

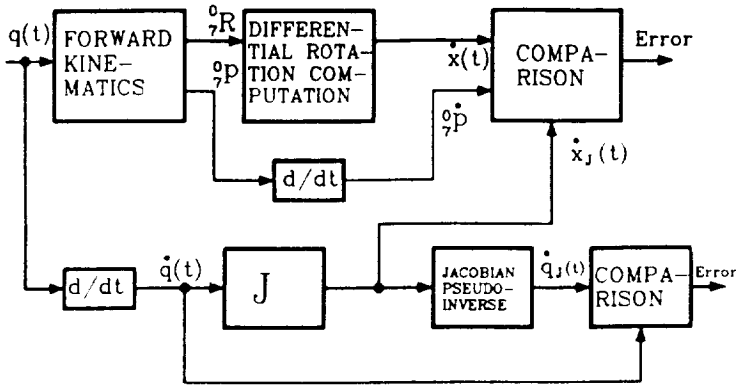

Figure 3: Computer simulation scheme for parts 1 and 2

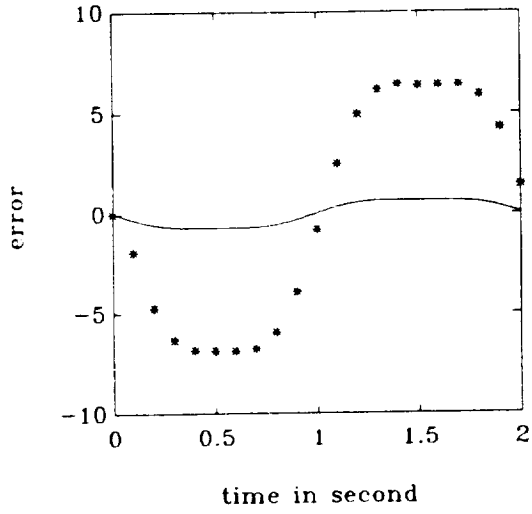

Figure 4: Errors of x-axis velocities
*sampling times: 10 msec (solid line), 100 msec (**-line)*

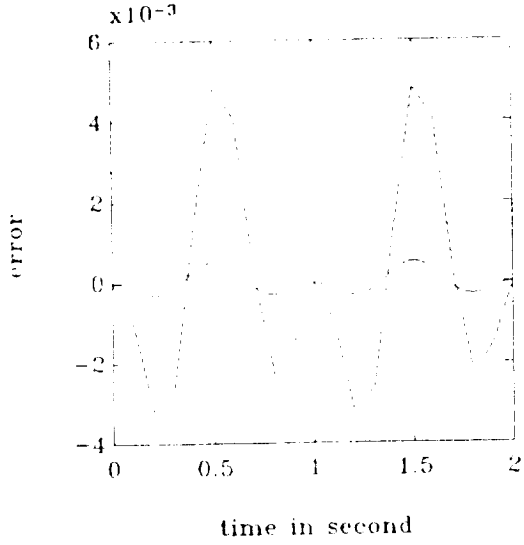

Figure 5: Errors of x-axis angular velocities
sampling times: 10 msec (solid line), 100 msec (dotted line)

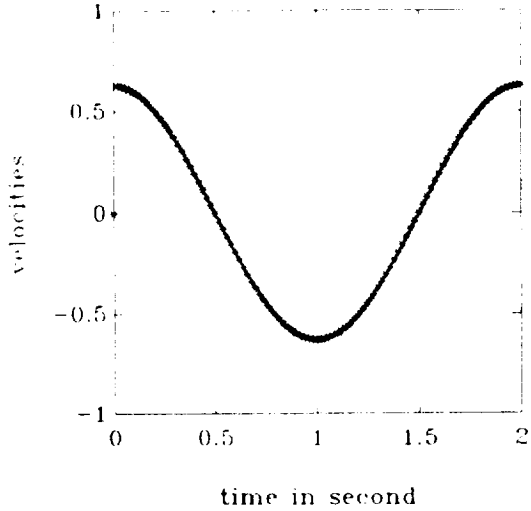

Figure 6: Velocities of joint angle 1 for sampling time of 10 msec, $\dot{\theta}_{1J}$ *(**-line)*, $\dot{\theta}_1$ *(dotted line)*

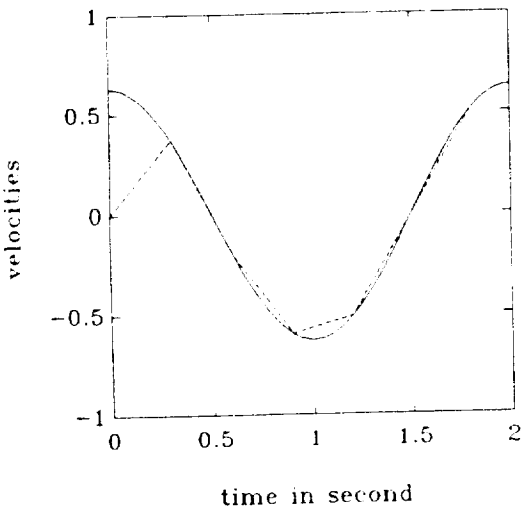

Figure 7: Velocities of joint angle 1 for sampling time of 100 msec, $\dot{\theta}_{1J}$ *(dotted line)*, $\dot{\theta}_1$ *(solid line)*

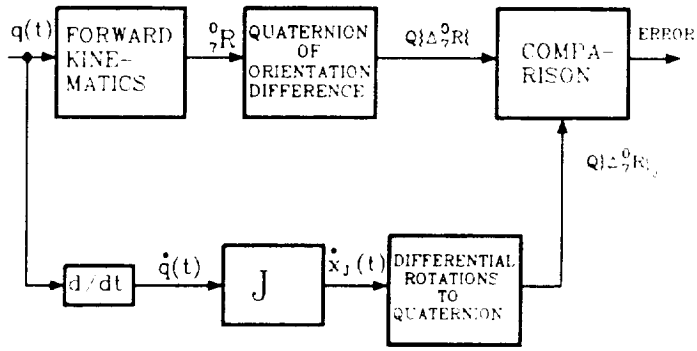

Figure 8: Computer simulation scheme for Part 3

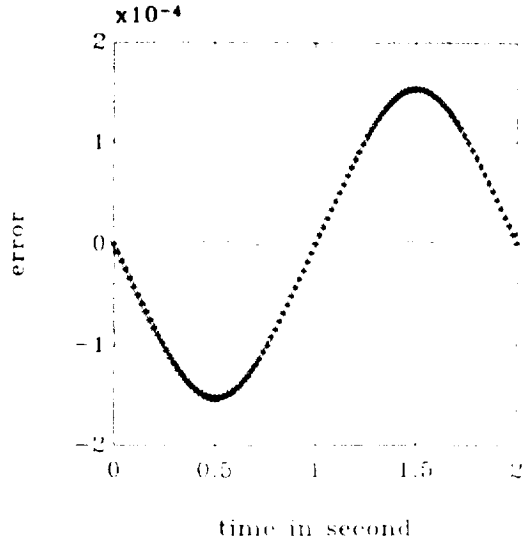

Figure 9: Quaternion errors for sampling time of 10 msec. $\delta\beta$ *(solid line)*, $\delta\gamma$ *(**-line)*

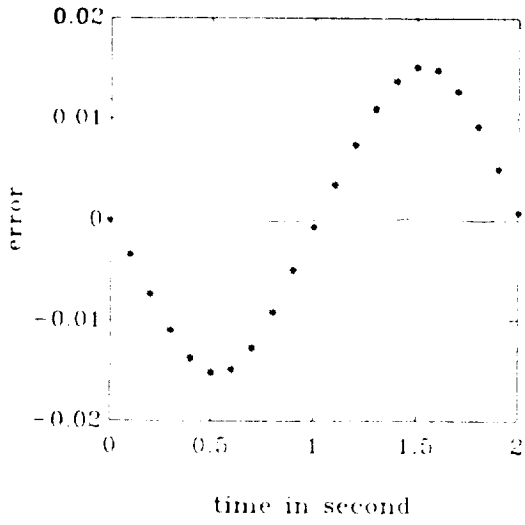

Figure 10: Quaternion errors for sampling time of 100 msec. $\delta\beta$ *(solid line)*, $\delta\gamma$ *(**-line)*

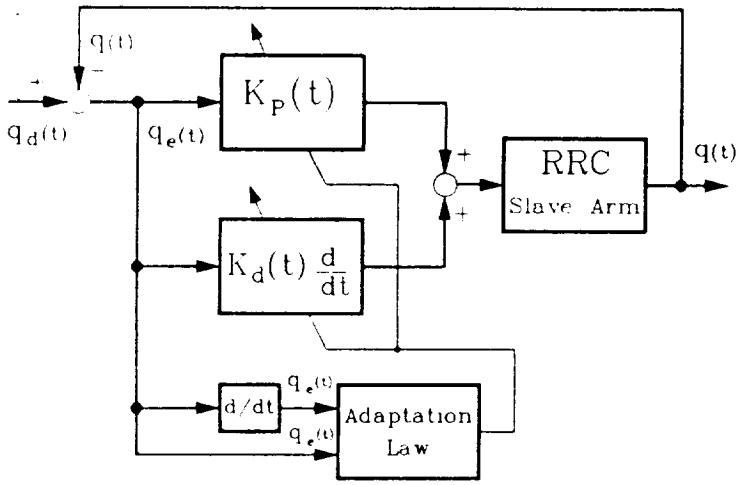

Figure 11: The joint-space adaptive control scheme

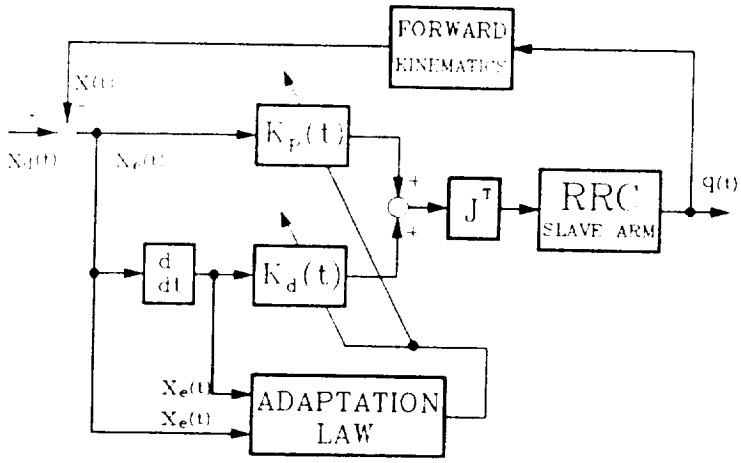

Figure 12: The Cartesian-space adaptive control scheme

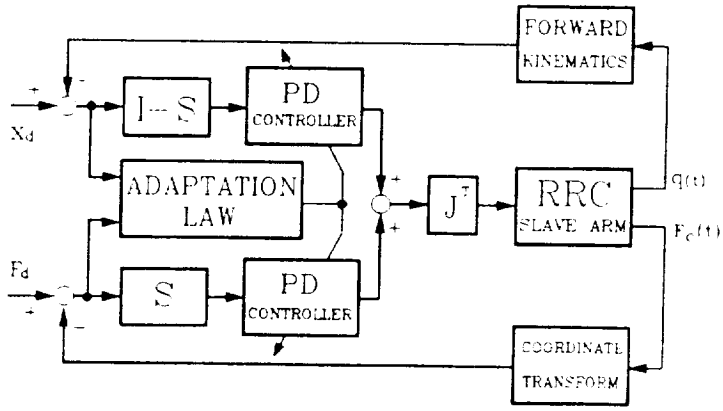

Figure 13: The hybrid adaptive control scheme