

THE UNIVERSITY OF MICHIGAN

COMPUTING RESEARCH LABORATORY¹

**RESOLVED MOTION ADAPTIVE CONTROL
FOR MECHANICAL MANIPULATORS**

**C.S.G. Lee
and
B.H. Lee**

CRL-TR-22-83

JUNE 1983

**Room 1079, East Engineering Building
Ann Arbor, Michigan 48109
USA
Tel: (313) 763-8000**

¹This research was supported by the National Science Foundation Grant ECS-8106954 and the Air Force Office of Scientific Research/AFSC, United States Air Force under AFOSR contract F49620-82-C-0089. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

RESOLVED MOTION ADAPTIVE CONTROL FOR MECHANICAL MANIPULATORS

C.S.G. Lee and B.H. Lee

Department of Electrical and Computer Engineering

The University of Michigan

Ann Arbor, Michigan 48109

ABSTRACT

This paper presents the development of a resolved motion adaptive control which adopts the ideas of "Resolved Motion Rate Control" [9] and "Resolved Motion Acceleration Control" [10] to control a manipulator in Cartesian coordinates for various loading conditions. The proposed adaptive control is performed at the hand level and is based on the linearized perturbation system along a desired hand trajectory. A recursive least square identification scheme is used to perform on-line parameter identification of the linearized perturbation system. The controlled system is characterized by feedforward and feedback components which can be computed separately and simultaneously. The feedforward component resolves the specified positions, velocities, and accelerations of the hand into a set of values of joint positions, velocities, and accelerations from which the nominal joint torques are computed using the Newton-Euler equations of motion to compensate all the interaction forces among the various joints. The feedback component computes the variational joint torques which reduce the manipulator hand position and velocity errors along the nominal hand trajectory. This adaptive control strategy reduces the manipulator control problem from a nonlinear control to controlling a linear control system about a desired hand trajectory. The feasibility of implementing the proposed adaptive control using present day low-cost microprocessors is discussed.

1. Introduction

The purpose of resolved motion control is to maintain a prescribed motion for the manipulator along a desired time-based hand trajectory in Cartesian coordinates by applying corrective compensation torques to the actuators to adjust for any deviations of the manipulator from the hand trajectory. Most of the existing control schemes [1]-[7] control the arm at the joint level and emphasize nonlinear compensations of the interaction forces among the various joints. For most applications, resolved motion control, which commands the manipulator hand to move in a desired Cartesian direction in a coordinated position and rate control, may be more appropriate. Most existing resolved motion control algorithms [8]-[12] control the arm at the hand level with or without external sensory feedback information. The above joint motion and resolved motion control algorithms are inadequate because they neglect the changes of the load in a task cycle. These changes in the payload of the controlled system may be significant enough to render conventional feedback control strategies ineffective. The result is reduced servo response speed and damping, which limits the precision and speed of the end-effector. Any significant performance gain in this require the consideration of adaptive control techniques.

Recently various adaptive control algorithms [13]-[15] have been proposed. Dubowsky [13] proposed a model referenced adaptive control which uses a linear second-order time invariant differential equation as the referenced model for each degree of freedom of the robot arm. The manipulator is controlled by adjusting the position and velocity feedback gains to follow the model. A steepest decent method is used to update the feedback gains. Koivo [14] proposed an adaptive self-tuning controller using an autoregressive model to fit the input-output data from the manipulator. Both control algorithms assume that the interaction forces among the joints are negligible. Lee [15] proposed an adaptive control based on the perturbation equations in the vicinity of a desired joint trajectory. The highly coupled dynamic equations of a manipulator are linearized about a preplanned joint trajectory to obtain the

perturbation equations. A recursive least square identification scheme is used to identify the system parameters in the perturbation equations. The torques for the joint actuators consist of the nominal torques computed from the Newton-Euler equations of motion and the variational torques computed from the controller of the linearized system.

This paper adopts the ideas of resolved motion rate and acceleration control [8]-[10] and extends the above adaptive control concept to control the manipulator hand in Cartesian coordinates. All the feedback information are performed at the hand level. In this paper, we only investigate the resolved motion adaptive control for a *six-joint manipulator* and assume that the desired hand positions, velocities and accelerations along a path/trajectory in Cartesian coordinates are given.

2. Kinematics of the Manipulator Hand

In general, the desired motion of a manipulator is specified in terms of a time-based hand trajectory in Cartesian coordinates, while the servo control system requires the reference inputs specified in joint coordinates. The mathematical relation between these two coordinate systems have been investigated by several authors [8]-[12]. We shall briefly describe the basic kinematic theory relating these two coordinate systems that will lead us to derive the equations of motion of the manipulator hand in Cartesian coordinates.

2.1. Position and Euler Angles of the Manipulator Hand

The location of the manipulator hand with respect to a fixed reference coordinate system can be realized by establishing an orthonormal coordinate frame at the hand (the hand coordinate frame) [16] (See Figure 1). The problem of finding the location of the hand is reduced to finding the position and orientation of the hand coordinate frame with respect to the inertial frame of the manipulator. This can be conveniently achieved by a 4×4 homogeneous transformation matrix [18], [16]:

$$T_{base}^{hand} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \left[\begin{array}{c|c} \text{Rotation} & \text{Position} \\ \text{matrix} & \text{vector} \\ \hline - & - \\ 0 & 0 & 0 & 1 \end{array} \right] \quad (1)$$

where \mathbf{p} is the position vector of the hand; and $\mathbf{n}, \mathbf{s}, \mathbf{a}$ are the unit vectors along the principal axes of the hand coordinate frame describing the orientation of the hand. Instead of using the rotation submatrix $[\mathbf{n}, \mathbf{s}, \mathbf{a}]$ to describe the orientation, we can use three Euler angles, yaw α , pitch β , and roll γ , which are defined as rotations of the reference frame about the x, y , and z axes of the reference frame respectively. One can obtain the elements of $[\mathbf{n}, \mathbf{s}, \mathbf{a}]$ from the Euler rotation matrix resulting from a rotation of α angle about the x_0 axis, then a rotation of β angle about the y_0 axis, and a rotation of γ angle about the z_0 axis of the reference frame. Thus:

$$\mathbf{R} = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} = \begin{bmatrix} C\gamma & -S\gamma & 0 \\ S\gamma & C\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C\beta & 0 & S\beta \\ 0 & 1 & 0 \\ -S\beta & 0 & C\beta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix} \quad (2)$$

$$= \begin{bmatrix} C\gamma C\beta & -S\gamma C\alpha + C\gamma S\beta S\alpha & S\gamma S\alpha + C\gamma S\beta C\alpha \\ S\gamma C\beta & C\gamma C\alpha + S\gamma S\beta S\alpha & -C\gamma S\alpha + S\gamma S\beta C\alpha \\ -S\beta & C\beta S\alpha & C\beta C\alpha \end{bmatrix}$$

where $\sin \alpha \equiv S\alpha, \cos \alpha \equiv C\alpha, \sin \beta \equiv S\beta, \cos \beta \equiv C\beta, \sin \gamma \equiv S\gamma, \cos \gamma \equiv C\gamma$.

2.2. Velocities and Accelerations of the Manipulator Hand

Let us define the position, orientation, linear velocity, and angular velocity vectors of the manipulator hand with respect to the reference frame respectively:

$$\begin{aligned} \mathbf{p}(t) &\stackrel{\Delta}{=} (p_x(t), p_y(t), p_z(t))^T ; & \Phi(t) &\stackrel{\Delta}{=} (\alpha(t), \beta(t), \gamma(t))^T \\ \mathbf{v}(t) &\stackrel{\Delta}{=} (v_x(t), v_y(t), v_z(t))^T ; & \Omega(t) &\stackrel{\Delta}{=} (\omega_x(t), \omega_y(t), \omega_z(t))^T \end{aligned} \quad (3)$$

where the superscript "T" denotes transpose operation on vectors and matrices. The linear velocity of the hand with respect to the reference frame is equal to the time derivative of the position of the hand:

$$\mathbf{v}(t) = \frac{d\mathbf{p}(t)}{dt} = \dot{\mathbf{p}}(t) \quad (4)$$

Since the inverse of a direction cosine matrix is equivalent to its transpose, the instantaneous angular velocities of the hand coordinate frame about the principal axes of the reference frame can be obtained from (2) as (see [21]):

$$\begin{aligned} \mathbf{R} \frac{d\mathbf{R}^T}{dt} &= - \frac{d\mathbf{R}}{dt} \mathbf{R}^T = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -S\beta\dot{\alpha} + \dot{\gamma} & -S\gamma C\beta\dot{\alpha} - C\gamma\dot{\beta} \\ S\beta\dot{\alpha} - \dot{\gamma} & 0 & C\gamma C\beta\dot{\alpha} - S\gamma\dot{\beta} \\ S\gamma C\beta\dot{\alpha} + C\gamma\dot{\beta} & -C\gamma C\beta\dot{\alpha} + S\gamma\dot{\beta} & 0 \end{bmatrix} \end{aligned} \quad (5)$$

From the above equation, the relation between the $(\omega_x, \omega_y, \omega_z)^T$ and $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})^T$ can be found by equating the non-zero elements in the matrices:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -C\gamma C\beta & S\gamma & 0 \\ -S\gamma C\beta & -C\gamma & 0 \\ S\beta & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (6)$$

Its inverse relation can be found easily:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \sec\beta \begin{bmatrix} -C\gamma & -S\gamma & 0 \\ S\gamma C\beta & -C\gamma C\beta & 0 \\ -C\gamma S\beta & -S\gamma S\beta & -C\beta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (7)$$

Or

$$\dot{\Phi}(t) \stackrel{\Delta}{=} \mathbf{M} \Omega(t)$$

Based on the moving coordinate frame concept [17], the linear and angular velocities of the hand can be obtained from the velocities of the lower joints:

$$\begin{bmatrix} \mathbf{v}(t) \\ \Omega(t) \end{bmatrix} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}(t) = [\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_6] \dot{\mathbf{q}}(t) \quad (8)$$

where $\dot{\mathbf{q}}(t) = (\dot{q}_1, \dots, \dot{q}_6)^T$ is the joint velocity vector of the manipulator, and $\mathbf{J}(\mathbf{q})$ is a 6×6 matrix whose i^{th} column vector \mathbf{J}_i can be found from (see [17]):

$$J_i = \begin{cases} \begin{bmatrix} z_{i-1} \times (p - p_{i-1}) \\ z_{i-1} \end{bmatrix} & ; \text{ if joint } i \text{ is rotational} \\ \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & ; \text{ if joint } i \text{ is translational} \end{cases} \quad (9)$$

where \times indicates cross product, p_{i-1} is the position of the origin of the $(i-1)^{th}$ coordinate frame with respect to the reference frame, z_{i-1} is the unit vector along the axis of motion of joint i , and p is the position of the hand with respect to the reference coordinate frame. (8) is the basis of resolved motion rate control [8]-[9], and $J(q)$ is the Jacobian matrix.

The accelerations of the hand can be obtained by taking the time derivative of the velocity vector in (8):

$$\begin{bmatrix} \dot{v}(t) \\ \dot{\Omega}(t) \end{bmatrix} = [J(q)]\dot{q}(t) + [J(q)]\ddot{q}(t) \quad (10)$$

where $\ddot{q}(t) = (\ddot{q}_1, \dots, \ddot{q}_6)^T$ is the joint acceleration vector of the manipulator.

If the inverse Jacobian matrix exists at $q(t)$, then the joint velocities $\dot{q}(t)$ of the manipulator can be computed from the hand velocities using (8):

$$\dot{q}(t) = [J^{-1}(q)] \begin{bmatrix} v(t) \\ \Omega(t) \end{bmatrix} \quad (11)$$

Using (8) and (11), (10) can be expressed as:

$$\begin{bmatrix} \dot{v}(t) \\ \dot{\Omega}(t) \end{bmatrix} = [J(q)][J^{-1}(q)] \begin{bmatrix} v(t) \\ \Omega(t) \end{bmatrix} + [J(q)]\ddot{q}(t) \quad (12)$$

and the joint accelerations $\ddot{q}(t)$ can be computed from the hand velocities and accelerations as:

$$\ddot{q}(t) = [J^{-1}(q)] \begin{bmatrix} \dot{v}(t) \\ \dot{\Omega}(t) \end{bmatrix} - [J^{-1}(q)][J(q)][J^{-1}(q)] \begin{bmatrix} v(t) \\ \Omega(t) \end{bmatrix} \quad (13)$$

The above kinematic relations between the joint coordinates and the Cartesian coordinates will be used in the next section to derive the equations of motion of the manipulator hand in Cartesian coordinates.

3. Equations of Motion of the Manipulator Hand

The dynamics of a six-joint manipulator can be derived either by Lagrange-Euler [1]-[3], [16] or Newton-Euler formulations [17], [16]. The resulting equations of motion are highly nonlinear and consist of inertia loading, coupling reaction forces (Coriolls and centrifugal) among the various joints, and gravity loading effects. In general, the Lagrange-Euler equations of motion of a six-joint manipulator, excluding the actuator dynamics, gear friction and backlash, can be expressed in vector matrix notation as [16]:

$$D(\mathbf{q}) \ddot{\mathbf{q}}(t) + H(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) = \boldsymbol{\tau}(t) \quad (14)$$

where $\boldsymbol{\tau}(t)$ is a 6×1 applied torque vector for joint actuators, \mathbf{q} is the angular positions, $\dot{\mathbf{q}}$ is the angular velocities, $\ddot{\mathbf{q}}(t)$ is a 6×1 joint acceleration vector, $G(\mathbf{q})$ is a 6×1 gravitational force vector, $H(\mathbf{q}, \dot{\mathbf{q}})$ is a 6×1 Coriolls and centrifugal force vector, and $D(\mathbf{q})$ is a 6×6 acceleration-related matrix.

Since $D(\mathbf{q})$ is always nonsingular, $\ddot{\mathbf{q}}(t)$ can be obtained from (14) and substituted into (12) to obtain the accelerations of the manipulator hand:

$$\begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\boldsymbol{\Omega}}(t) \end{bmatrix} = [\mathbf{J}(\mathbf{q})][\mathbf{J}^{-1}(\mathbf{q})] \begin{bmatrix} \mathbf{v}(t) \\ \boldsymbol{\Omega}(t) \end{bmatrix} + [\mathbf{J}(\mathbf{q})][\mathbf{D}^{-1}(\mathbf{q})] [\boldsymbol{\tau}(t) - H(\mathbf{q}, \dot{\mathbf{q}}) - G(\mathbf{q})] \quad (15)$$

For convenience, let us partition $\mathbf{J}(\mathbf{q})$, $\mathbf{J}^{-1}(\mathbf{q})$, and $\mathbf{D}^{-1}(\mathbf{q})$ into 3×3 submatrices and $H(\mathbf{q}, \dot{\mathbf{q}})$, $G(\mathbf{q})$, and $\boldsymbol{\tau}(t)$ into 3×1 submatrices:

$$[\mathbf{J}(\mathbf{q})] \triangleq \begin{bmatrix} \mathbf{J}_{11} & | & \mathbf{J}_{12} \\ \text{---} & | & \text{---} \\ \mathbf{J}_{21} & | & \mathbf{J}_{22} \end{bmatrix} ; \quad [\mathbf{J}^{-1}(\mathbf{q})] \triangleq [\mathbf{K}] \triangleq \begin{bmatrix} \mathbf{K}_{11} & | & \mathbf{K}_{12} \\ \text{---} & | & \text{---} \\ \mathbf{K}_{21} & | & \mathbf{K}_{22} \end{bmatrix} \quad (16)$$

$$[\mathbf{D}^{-1}(\mathbf{q})] \triangleq [\mathbf{E}] \triangleq \begin{bmatrix} \mathbf{E}_{11} & | & \mathbf{E}_{12} \\ \text{---} & | & \text{---} \\ \mathbf{E}_{21} & | & \mathbf{E}_{22} \end{bmatrix} ; \quad [H(\mathbf{q}, \dot{\mathbf{q}})] \triangleq \begin{bmatrix} \mathbf{H}_1 \\ \text{---} \\ \mathbf{H}_2 \end{bmatrix} ; \quad [G(\mathbf{q})] \triangleq \begin{bmatrix} \mathbf{G}_1 \\ \text{---} \\ \mathbf{G}_2 \end{bmatrix} ; \quad [\boldsymbol{\tau}] \triangleq \begin{bmatrix} \tau_1 \\ \text{---} \\ \tau_2 \end{bmatrix} \quad (17)$$

Using (16)-(17), (15) can be expressed as:

$$\begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\Omega}(t) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{J}}_{11}\mathbf{K}_{11} + \dot{\mathbf{J}}_{12}\mathbf{K}_{21} & \dot{\mathbf{J}}_{11}\mathbf{K}_{12} + \dot{\mathbf{J}}_{12}\mathbf{K}_{22} \\ \dot{\mathbf{J}}_{21}\mathbf{K}_{11} + \dot{\mathbf{J}}_{22}\mathbf{K}_{21} & \dot{\mathbf{J}}_{21}\mathbf{K}_{12} + \dot{\mathbf{J}}_{22}\mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ \Omega(t) \end{bmatrix} \quad (18)$$

$$+ \begin{bmatrix} \mathbf{J}_{11}\mathbf{E}_{11} + \mathbf{J}_{12}\mathbf{E}_{21} & \mathbf{J}_{11}\mathbf{E}_{12} + \mathbf{J}_{12}\mathbf{E}_{22} \\ \mathbf{J}_{21}\mathbf{E}_{11} + \mathbf{J}_{22}\mathbf{E}_{21} & \mathbf{J}_{21}\mathbf{E}_{12} + \mathbf{J}_{22}\mathbf{E}_{22} \end{bmatrix} \begin{bmatrix} -\mathbf{H}_1 - \mathbf{G}_1 + \tau_1 \\ -\mathbf{H}_2 - \mathbf{G}_2 + \tau_2 \end{bmatrix}$$

Combining (4), (7) and (18), we can obtain the state equations of the manipulator hand in Cartesian coordinates:

$$\begin{bmatrix} \dot{\mathbf{p}}(t) \\ \dot{\Phi}(t) \\ \dot{\mathbf{v}}(t) \\ \dot{\Omega}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & | & \mathbf{I}_{3 \times 3} & 0 \\ 0 & 0 & | & 0 & \mathbf{M} \\ \hline 0 & 0 & | & \dot{\mathbf{J}}_{11}\mathbf{K}_{11} + \dot{\mathbf{J}}_{12}\mathbf{K}_{21} & \dot{\mathbf{J}}_{11}\mathbf{K}_{12} + \dot{\mathbf{J}}_{12}\mathbf{K}_{22} \\ 0 & 0 & | & \dot{\mathbf{J}}_{21}\mathbf{K}_{11} + \dot{\mathbf{J}}_{22}\mathbf{K}_{21} & \dot{\mathbf{J}}_{21}\mathbf{K}_{12} + \dot{\mathbf{J}}_{22}\mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{p}(t) \\ \Phi(t) \\ \mathbf{v}(t) \\ \Omega(t) \end{bmatrix} \quad (19)$$

$$+ \begin{bmatrix} 0 & | & 0 \\ 0 & | & 0 \\ \hline \mathbf{J}_{11}\mathbf{E}_{11} + \mathbf{J}_{12}\mathbf{E}_{21} & | & \mathbf{J}_{11}\mathbf{E}_{12} + \mathbf{J}_{12}\mathbf{E}_{22} \\ \mathbf{J}_{21}\mathbf{E}_{11} + \mathbf{J}_{22}\mathbf{E}_{21} & | & \mathbf{J}_{21}\mathbf{E}_{12} + \mathbf{J}_{22}\mathbf{E}_{22} \end{bmatrix} \begin{bmatrix} -\mathbf{H}_1 - \mathbf{G}_1 + \tau_1 \\ \hline -\mathbf{H}_2 - \mathbf{G}_2 + \tau_2 \end{bmatrix}$$

(19) is the state equations of the manipulator hand and will be used later to derive an adaptive control scheme in Cartesian coordinates.

4. Resolved Motion Adaptive Control Formulation

Defining the state vector for the manipulator hand as:

$$\begin{aligned} \mathbf{x}(t) &\stackrel{\Delta}{=} (x_1, x_2, \dots, x_{12})^T \\ &\stackrel{\Delta}{=} (p_x, p_y, p_z, \alpha, \beta, \gamma, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T \\ &\stackrel{\Delta}{=} (\mathbf{p}^T, \Phi^T, \mathbf{v}^T, \Omega^T)^T \end{aligned} \quad (20)$$

and the input torque vector as:

$$\mathbf{u}(t) \stackrel{\Delta}{=} (\tau_1, \dots, \tau_6)^T \stackrel{\Delta}{=} (u_1, \dots, u_6)^T \quad (21)$$

(19) can be expressed in state space representation as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (22)$$

where $\mathbf{x}(t) \in R^{2n}$, $\mathbf{u}(t) \in R^n$, $t \in R^+$, $\mathbf{f}(\cdot) : R^{2n} \times R^n \times R^+ \rightarrow R^{2n}$ and continuously differentiable, and $n = 6$ is the number of degree of freedom of the manipulator. (22)

can be expressed explicitly as:

$$\begin{aligned} \dot{x}_1 &= f_1(\mathbf{x}, \mathbf{u}, t) = x_7 \\ \dot{x}_2 &= f_2(\mathbf{x}, \mathbf{u}, t) = x_8 \\ \dot{x}_3 &= f_3(\mathbf{x}, \mathbf{u}, t) = x_9 \\ \dot{x}_4 &= f_4(\mathbf{x}, \mathbf{u}, t) = -\sec x_5 (x_{10} Cx_6 + x_{11} Sx_6) \\ \dot{x}_5 &= f_5(\mathbf{x}, \mathbf{u}, t) = \sec x_5 (x_{10} Cx_5 Sx_6 - x_{11} Cx_5 Cx_6) \\ \dot{x}_6 &= f_6(\mathbf{x}, \mathbf{u}, t) = -\sec x_5 (x_{10} Sx_5 Cx_6 + x_{11} Sx_5 Sx_6 + x_{12} Cx_5) \\ \dot{x}_{i+6} &= f_{i+6}(\mathbf{x}, \mathbf{u}, t) = g_{i+6}(\mathbf{x})x + b_{i+6}(\mathbf{x})\beta(\mathbf{x}) + b_{i+6}(\mathbf{x})\mathbf{u}(t) \quad ; \quad i = 1, \dots, 6 \end{aligned} \quad (23)$$

where $g_{i+6}(\mathbf{x})$ is the $(i+6)^{th}$ row of the matrix:

$$\left[\begin{array}{cc|ccc} 0 & 0 & & I_{3 \times 3} & 0 \\ 0 & 0 & & 0 & \mathbf{M} \\ \hline 0 & 0 & \dot{J}_{11}K_{11} + \dot{J}_{12}K_{21} & \dot{J}_{11}K_{12} + \dot{J}_{12}K_{22} \\ 0 & 0 & \dot{J}_{21}K_{11} + \dot{J}_{22}K_{21} & \dot{J}_{21}K_{12} + \dot{J}_{22}K_{22} \end{array} \right]$$

and $b_{i+6}(\mathbf{x})$ is the $(i+6)^{th}$ row of the matrix:

$$\left[\begin{array}{cc|cc} 0 & & 0 & \\ 0 & & 0 & \\ \hline J_{11}E_{11} + J_{12}E_{21} & & J_{11}E_{12} + J_{12}E_{22} \\ J_{21}E_{11} + J_{22}E_{21} & & J_{21}E_{12} + J_{22}E_{22} \end{array} \right]$$

and

$$\beta(\mathbf{x}) = \begin{bmatrix} -H_1(\mathbf{x}) - G_1(\mathbf{x}) \\ \hline -H_2(\mathbf{x}) - G_2(\mathbf{x}) \end{bmatrix}$$

With this formulation, the control problem is to find a feedback control law $\mathbf{u}(t) = \mathbf{g}(\mathbf{x}(t))$ such that the closed loop manipulator hand control system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{g}(\mathbf{x}(t)), t)$ is asymptotically stable and tracks a desired hand trajectory as closely as possible over a wide range of payloads for all times.

4.1. Perturbation Equations of Motion

Equation (23) describes the complete manipulator hand dynamics, the desired joint torques for each hand trajectory set point $(p^d(t), \phi^d(t), v^d(t), \Omega^d(t), \dot{v}^d(t), \dot{\Omega}^d(t))$ can be computed (in open-loop fashion) quite accurately as follows: (i) The hand trajectory set points are resolved into a set of values of desired joint positions, velocities and accelerations, (ii) The desired joint torques along the hand trajectory are computed from the Newton-Euler equations of motion [17] using the computed sets of values of joint positions, velocities and accelerations. These computed torques constitute the nominal torque values. Using the Taylor series expansion on (23) about the nominal hand trajectory, the associated linearized perturbation model for this control system can be obtained:

$$\delta \dot{\mathbf{x}}(t) = \mathbf{F}(t) \delta \mathbf{x}(t) + \mathbf{L}(t) \delta \mathbf{u}(t) \quad (24)$$

where $\mathbf{F}(t) \equiv \nabla_{\mathbf{x}} \mathbf{f} |_{\mathbf{x}_n}$ and $\mathbf{L}(t) \equiv \nabla_{\mathbf{u}} \mathbf{f} |_{\mathbf{u}_n}$ are the system parameters and are equivalent to the gradients of $f(\mathbf{x}(t), \mathbf{u}(t), t)$ evaluated at the nominal states, $\mathbf{x}_n(t)$, and inputs, $\mathbf{u}_n(t)$, respectively, $\delta \mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_n(t)$, and $\delta \mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_n(t)$. $\mathbf{F}(t)$ and $\mathbf{L}(t)$ are functionally defined by:

$$\mathbf{F}(t) = \begin{bmatrix} \mathbf{F}_{11} & | & \mathbf{F}_{12} \\ \text{---} & | & \text{---} \\ \mathbf{F}_{21} & | & \mathbf{F}_{22} \end{bmatrix}; \quad \mathbf{L}(t) = \begin{bmatrix} 0 & | & 0 \\ \text{---} & | & \text{---} \\ \mathbf{L}_{21} & | & \mathbf{L}_{22} \end{bmatrix} \quad (25)$$

where the submatrices are found explicitly as:

$$\mathbf{F}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial f_4}{\partial x_5} & \frac{\partial f_4}{\partial x_6} \\ 0 & 0 & 0 & 0 & \frac{\partial f_5}{\partial x_5} & \frac{\partial f_5}{\partial x_6} \\ 0 & 0 & 0 & 0 & \frac{\partial f_6}{\partial x_5} & \frac{\partial f_6}{\partial x_6} \end{bmatrix}; \quad \mathbf{F}_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial f_4}{\partial x_{10}} & \frac{\partial f_4}{\partial x_{11}} & 0 \\ 0 & 0 & 0 & \frac{\partial f_5}{\partial x_{10}} & \frac{\partial f_5}{\partial x_{11}} & 0 \\ 0 & 0 & 0 & \frac{\partial f_6}{\partial x_{10}} & \frac{\partial f_6}{\partial x_{11}} & \frac{\partial f_6}{\partial x_{12}} \end{bmatrix} \quad (26)$$

$$F_{21} = \begin{bmatrix} \frac{\partial f_7}{\partial x_1} & \frac{\partial f_7}{\partial x_2} & \dots & \frac{\partial f_7}{\partial x_6} \\ \frac{\partial f_8}{\partial x_1} & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \frac{\partial f_{12}}{\partial x_1} & \cdot & \dots & \frac{\partial f_{12}}{\partial x_6} \end{bmatrix}; F_{22} = \begin{bmatrix} \frac{\partial f_7}{\partial x_7} & \frac{\partial f_7}{\partial x_8} & \dots & \frac{\partial f_7}{\partial x_{12}} \\ \frac{\partial f_8}{\partial x_7} & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \frac{\partial f_{12}}{\partial x_7} & \cdot & \dots & \frac{\partial f_{12}}{\partial x_{12}} \end{bmatrix} \quad (27)$$

$$L_{21} = \begin{bmatrix} \frac{\partial f_7}{\partial u_1} & \frac{\partial f_7}{\partial u_2} & \frac{\partial f_7}{\partial u_3} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \frac{\partial f_{12}}{\partial u_1} & \frac{\partial f_{12}}{\partial u_2} & \frac{\partial f_{12}}{\partial u_3} \end{bmatrix}; L_{22} = \begin{bmatrix} \frac{\partial f_7}{\partial u_4} & \frac{\partial f_7}{\partial u_5} & \frac{\partial f_7}{\partial u_6} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \frac{\partial f_{12}}{\partial u_4} & \frac{\partial f_{12}}{\partial u_5} & \frac{\partial f_{12}}{\partial u_6} \end{bmatrix} \quad (28)$$

$F(t)$ and $L(t)$ are slowly time-varying and depend on the instantaneous manipulator hand position and velocity. The design of a feedback control law for the perturbation equations requires that the system parameters of (24) be known at all times. Thus parameter identification techniques must be used to identify the unknown elements in $F(t)$ and $L(t)$.

As a result of this formulation, the control problem is reduced to determining $\delta u(t)$ which drives $\delta x(t)$ to zero at all times. In other words, the position/orientation and velocity errors of the hand along the preplanned hand trajectory are minimized. The controlled system is characterized by a feedforward component and a feedback component. The feedforward component computes the nominal joint torques $u_n(t)$ from the Newton-Euler equations of motion and the feedback component computes the variational joint torques $\delta u(t)$. The main advantages of this formulation are two-fold. Firstly it reduces a nonlinear control problem to a linear control problem about a nominal hand trajectory, and secondly the computations of the nominal and variational joint torques can be performed separately and simultaneously. The proposed resolved motion adaptive control block diagram is shown in Figure 2.

4.2. Parameter Identification and Control of the Linearized Perturbation System

For implementation on digital computer, (24) needs to be discretized to obtain the following discrete-time linear equations appropriate for parameter identification:

$$\mathbf{x}((k+1)T) = \mathbf{A}(kT) \mathbf{x}(kT) + \mathbf{B}(kT) \mathbf{u}(kT) \quad ; \quad k = 0, 1, \dots \quad (29)$$

where T is the sampling period, $\mathbf{u}(kT) \in R^n$ is a piecewise constant control input vector of $\mathbf{u}(t)$ over the time interval between any two consecutive sampling instants for $kT \leq t \leq (k+1)T$, and $\mathbf{x}(kT) \in R^{2n}$ is a perturbed state vector which is given by:

$$\mathbf{x}(kT) = \Theta(kT, t_0) \mathbf{x}(t_0) + \int_{t_0}^{kT} \Theta(kT, t) \mathbf{L}(t) \mathbf{u}(t) dt \quad (30)$$

where $\Theta(kT, t_0)$ is the state-transition matrix of the system. $\mathbf{A}(kT)$ and $\mathbf{B}(kT)$ are respectively $2n \times 2n$ and $2n \times n$ matrices and are given by:

$$\mathbf{A}(kT) = \Theta((k+1)T, kT) \quad (31)$$

and

$$\mathbf{B}(kT) \mathbf{u}(kT) = \int_{kT}^{(k+1)T} \Theta((k+1)T, t) \mathbf{L}(t) \mathbf{u}(t) dt \quad (32)$$

With this model, a maximum of $6n^2$ (or 216) parameters need to be identified. Without confusion, the sampling period T will be dropped from the above equations for clarity.

In order to simplify the identification algorithm for real-time applications, a recursive least square parameter identification scheme is used. In the parameter identification scheme, we make the following assumptions: 1) The parameters of the system are slowly time-varying but the variation speed is slower than the adaptation speed, 2) Measurement noise is negligible, and 3) The state variables $\mathbf{x}(k)$ of (29) are measurable.

Defining and putting the j^{th} row of the unknown system parameters in (29) at the k^{th} instant of time in a vector as:

$$\mathcal{V}_{jk} = (a_{j1}(k), \dots, a_{jp}(k), b_{j1}(k), \dots, b_{jn}(k))^T \quad (33)$$

where $p = 12$ and $n = 6$, and the outputs and inputs at the k^{th} instant of time in a

vector as:

$$\varphi_k = (x_1(k), \dots, x_p(k), u_1(k), \dots, u_n(k))^T \quad (34)$$

and the state vector at the k^{th} instant of time as:

$$x(k) = (x_1(k), \dots, x_p(k))^T = (x_{1k}, \dots, x_{pk})^T \quad (35)$$

(28) can be rewritten as follow:

$$x_{lk+1} = \varphi_k^T \hat{\vartheta}_{lk} \quad ; \quad l = 1, 2, \dots, 2n. \quad (36)$$

where x_{lk+1} is the l^{th} state variable at the $(k+1)^{\text{th}}$ sampling instant.

Based on the input-output relation in (36), a recursive least squares parameter identification algorithm can be found to be [19]:

$$\begin{aligned} \hat{\vartheta}_{lk+1} &= \hat{\vartheta}_{lk} - P_k \varphi_k [\varphi_k^T P_k \varphi_k + r]^{-1} [\varphi_k^T \hat{\vartheta}_{lk} - x_{lk+1}] \quad ; \quad l = 1, 2, \dots, 2n \\ P_{k+1} &= [P_k - P_k \varphi_k [\varphi_k^T P_k \varphi_k + r]^{-1} \varphi_k^T P_k] r^{-1} \quad ; \quad 0 < r \leq 1 \end{aligned} \quad (37)$$

where "hat" is used to indicate the estimate of the parameters and P_k is a $3n \times 3n$ symmetric positive definite matrix. $P_k = r [\Psi_k \Psi_k^T]^{-1}$ and $\Psi_k = [\varphi_1, \varphi_2, \dots, \varphi_k]$ is the measurement matrix up to the k^{th} sampling instant, if input-output data are used to identify the system parameters non-recursively. The P_k matrix has the similar effect as the error covariance matrix in stochastic identification with zero mean modeling error.

The estimate of the parameters $\hat{\vartheta}_{lk+1}$ at the $(k+1)^{\text{th}}$ sampling period is equal to the previous estimate $\hat{\vartheta}_{lk}$ corrected by the term proportional to $\varphi_k^T \hat{\vartheta}_{lk} - x_{lk+1}$. The $\varphi_k^T \hat{\vartheta}_{lk}$ is the prediction of the value x_{lk+1} . The parameter r is a weighting factor and is commonly used for tracking slowly time-varying parameters. If $r \ll 1$, a large weighting factor is placed on the more recent sampled data by rapidly weighing out the previous samples. We can compromise between fast adaptation capabilities and loss of accuracy in parameter identification by adjusting the weighting factor r .

With the determination of $A(k)$ and $B(k)$ from the parameter identification scheme, proper control laws can be designed to obtain the required correction torques to reduce the position/orientation and velocity errors of the manipulator hand

along a nominal hand trajectory. This can be done by finding an optimal control, $u^*(k)$, which minimizes the performance index, $J(k)$, while satisfying the constraints of (29):

$$J(k) = \frac{1}{2} \left[x^T(k+1)Qx(k+1) + u^T(k)Ru(k) \right] \quad (38)$$

where Q is a $2n \times 2n$ semi-positive definite weighting matrix and R is an $n \times n$ positive definite weighting matrix. The one-stage performance index in (38) indicates that the objective of the optimal control is to drive the position/orientation and velocity errors of the manipulator hand to zero along the nominal hand trajectory in a coordinated position and rate control per interval step, while at the same time, a cost is attached to the use of control efforts. The optimal control solution which minimizes the functional in (38) subject to the constraints of (29) is well-known and is found to be [20]:

$$u^*(k) = - \left[R + \hat{B}^T(k) Q \hat{B}(k) \right]^{-1} \hat{B}^T(k) Q \hat{A}(k) x(k) \quad (39)$$

where $\hat{A}(k)$ and $\hat{B}(k)$ are the system parameters obtained from the identification algorithm at the k^{th} sampling instant.

The above identification and control algorithms in (37) and (39) do not require complex computations. In (37), $(\varphi_k^T P_k \varphi_k + r)$ gives a scalar value which simplifies its inversion. Although the weighting factor r can be adjusted for each i^{th} state variable as desired, this requires excessive computations in the P_{k+1} matrix. For real-time robot arm control, such adjustment is not desirable. P_{k+1} is computed only once at each sampling time using the same weighting factor r . Moreover, since P_k is a symmetric positive definite matrix, only the upper diagonal matrix of P_k needs to be computed. The computational requirements of the identification and control algorithms for a six-joint manipulator is tabulated in Table 1.

5. Computational Complexity of the Adaptive Control

The overall adaptive control system is characterized by a feedforward component and a feedback component. Such formulation has the advantage of employing

parallel computation schemes in computing these components. A feasibility study of implementing the adaptive controller using present day low-cost microprocessors is conducted by looking at the computational requirements in terms of mathematical multiplication and addition operations. The study assumes that multi-processors are available for parallel computation of the proposed controller.

5.1. Computations of Feedforward Control Component

The feedforward component which computes the nominal joint torques along a desired hand trajectory can be computed serially in four separate stages. The first three stages resolve the preplanned hand trajectory set points $(\mathbf{p}^d(t), \Phi^d(t), \mathbf{v}^d(t), \Omega^d(t), \dot{\mathbf{v}}^d(t), \dot{\Omega}^d(t))$ into a set of values of desired joint positions, velocities and accelerations. The first stage involves the computations of the inverse kinematics solution routine. The second stage involves the computation of the inverse of the 6×6 Jacobian matrix which can be computed using the methods stated in [22].

The third stage utilizes the computed joint positions and velocities from the previous stages to compute the desired joint accelerations from the hand accelerations. In this stage, the time derivative of the Jacobian matrix is computed by first-order approximation. Finally the fourth stage utilizes the computed values of joint positions, velocities, and accelerations from the previous stages to compute the nominal joint torques along the desired hand trajectory using the Newton-Euler equations of motion [17]. Computational requirements in terms of multiplications and additions for the feedforward component are tabulated in Table 1. It requires a total of 1341 multiplications and 973 additions.

5.2. Computations of Feedback Control Component

The feedback control component which computes the variational joint torques can be conveniently computed serially in three separate stages. The first stage

computes the actual manipulator hand position and velocity from the measured values of joint position and velocity. It involves the computations of: (i) The direct kinematics routine, (ii) The 6×6 Jacobian matrix which can be computed using (9). The above computations can be performed in parallel and the maximum number of mathematical operations is taken as the time requirement for this stage.

The second stage involves the computations of: (i) The hand position and velocity errors which represent the deviations of the actual hand position and velocity from the preplanned hand trajectory, and (ii) The least square identification scheme for the linearized perturbation system. The third stage involves the computation of the dead-beat controller for the linearized system utilizing the parameters identified in the previous stage. Computational requirements for the feedback component are listed in Table 1. It requires about 3427 multiplications and 3037 additions. Since the feedforward and feedback components can be computed in parallel, the proposed adaptive controller requires a total of 3427 multiplications and 3037 additions in each sampling period.

Based on the specification sheet of INTEL 8087 microprocessor, an integer multiply requires $19 \mu s$, an addition requires $17 \mu s$, and a memory fetch or store requires $9 \mu s$, the proposed controller can be computed in about $175 ms$ which is not fast enough for closing the servo loop. Similarly looking at the specification sheet of PDP 11/45 computer, an integer multiply requires $3.3 \mu s$, an addition requires $300 ns$, and a memory fetch or store requires $450 ns$, the proposed controller can be computed in about $18 ms$ which translates to approximately a sampling frequency of 55 Hz. However, the PDP 11/45 is a uniprocessor machine and parallel computation assumption is not valid. But it does give us an indication of the required processing speed of the microprocessors. We anticipate that faster microprocessors will be just around the corner that will be able to compute the proposed adaptive controller within $10 ms$.

6. Conclusion

A resolved motion adaptive control based on the perturbation theory has been presented with an analysis of computational requirements for implementation. The adaptive control system is characterized by a feedforward component and a feedback component. The feedforward component computes the nominal joint torques $u_n(t)$ from the Newton-Euler equations of motion using the resolved joint information, and the feedback component consisting of recursive least square identification and control algorithms for the linearized system computes the variational joint torques $\delta u(t)$. Since the computations of the nominal and variational torques can be performed in parallel, the computations of the adaptive control for a six-joint robot arm may be implemented in low-cost microprocessors. Present day microprocessors still do not have the required speed to compute the proposed controller within 10 ms. This may be changing in the next year or two. The above analysis presented an ideal system study because it neglected such nonlinear effects as gear friction and backlash. The physical implementation of the proposed adaptive control may require further investigation on the effects of gear friction, backlash, control device dynamics, and flexible link structure to the controller.

7. References

1. M. E. Kahn, B. Roth, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," *Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 93, Sep. 1971, pp 164-172.
2. R. P. Paul, "Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm," Stanford Artificial Intelligence Laboratory, A.I. Memo 177, Sep. 1972.
3. A. K. Bejczy, "Robot Arm Dynamics and Control," Technical Memorandum 33-669, Jet Propulsion Laboratory, Feb. 1974.

4. H. Hemani, P. C. Camana, "Nonlinear Feedback in Simple Locomotion System," *IEEE Transactions on Automatic Control*, Vol. AC-19, Dec. 1976, pp 855-860.
5. K. K. D. Young, "Controller Design for a Manipulator Using Theory of Variable Structure Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-8, No. 2, Feb. 1978, pp. 101-109.
6. C. S. G. Lee, T. N. Mudge, et al, "Hierarchical Control Structure using Special Purpose Processors for the Control of Robot Arms," *Proceedings of the 1982 Pattern Recognition and Image Processing Conference*, Las Vegas, Nevada, June 14-17, 1982, pp 634-640.
7. B. R. Markiewicz, "Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator," Technical Memo 33-601, Jet Propulsion Lab, March 1973.
8. D. E. Whitney, "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, Dec. 1972, pp 303-309.
9. D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transactions on Man-Machine System*, Vol. MMS-10, No. 2, June 1969, pp 47-53.
10. J. Y. S. Luh, M. W. Walker, R. P. Paul, "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 3, June 1980, pp 468-474.
11. M. Takegaki, S. Arimoto, "A New Feedback Method for Dynamic Control of Manipulators," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, June 1981, pp 119-125.
12. C. H. Wu, R. P. Paul, "Resolved Motion Force Control of robot Manipulator," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-12, No. 3, June 1982, pp

266-275.

13. S. Dubowsky, D. T. DesForges, "The Application of Model Referenced Adaptive Control to Robotic Manipulators," *Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 101, Sep. 1979, pp 193-200.
14. A. J. Koivo, T. H. Guo, "Adaptive Linear Controller for Robotic Manipulators," *IEEE Transactions on Automatic Control*, Vol. AC-28, No. 1, February 1983, pp 162-171.
15. C. S. G. Lee, M. J. Chung, "An Adaptive Control Strategy for Computer-Based Manipulators," *Proceedings of the 21st Conference on Decision and Control*, Dec. 8-10 1982, Orlando, Florida, pp 95-100.
16. C. S. G. Lee, "Robot Arm Kinematics, Dynamics, and Control", *IEEE Computer*, Vol. 15, No. 12, December 1982, pp 62-80.
17. J. Y. S. Luh, M. W. Walker, R. P. C. Paul, "On-line Computational Scheme for Mechanical Manipulators," *ASME Journal of Dynamics Systems, Measurement and Control*, Vol. 102, June 1980, pp. 69-76.
18. R. P. Paul, *Robot Manipulators - Mathematics, Programming, and Control*, MIT Press, 1981.
19. K. J. Astrom, P. Eykhoff, "System Identification - A Survey," *Automatica*, Vol. 7, 1971, pp 123-162.
20. G. N. Saridis, R. N. Lobbia, "Parameter Identification and Control of Linear Discrete-Time Systems," *IEEE Transactions on Automatic Control*, Vol. AC-17, No.1 February 1972, pp 52-60.
21. R. A. Frazer et al, *Elementary Matrices*, Cambridge University Press, 1960, pp 250-256.
22. H. Melouah, P. Andre, "High Speed Computation of the Inverse Jacobian Matrix and of Servo Inputs for Robot Arm Control," *Proceedings of the 21st Conference*

ASME

RESOLVED MOTION ADAPTIVE CONTROL

on Decision and Control, Dec. 8-10 1982, Orlando, Florida, pp 89-94.

Adaptive Controller		Number of Multiplications	Number of Additions
stage 1	Compute q^d (Inverse Kinematics)	39	32
stage 2	Compute \dot{q}^d (Eq. 11)	516	242
stage 3	Compute \ddot{q}^d (Eq. 13)	108	102
stage 4	Compute τ (see [16])	678	597
Total Feedforward Computations		1341	973
stage 1	Compute $(p^T, \phi^T)^T$ (Eqs. 1 & 2)	48	22
	Compute $(v^T, \Omega^T)^T$ (Eqs. 8 & 9)	168	138
stage 2	Compute Hand Errors $(x(k) - x_n(k))$ and Identification Scheme (Eq. 37)	0 + 1111	12 + 1097
stage 3	Compute Dead-beat Controller (Eq. 39)	2148	1790
Total Feedback Computations		3427	3037
Total Mathematical Operations		3427	3037

**Table 1 Parallel Computations of the Proposed Adaptive Controller
For a Six-Joint Manipulator**

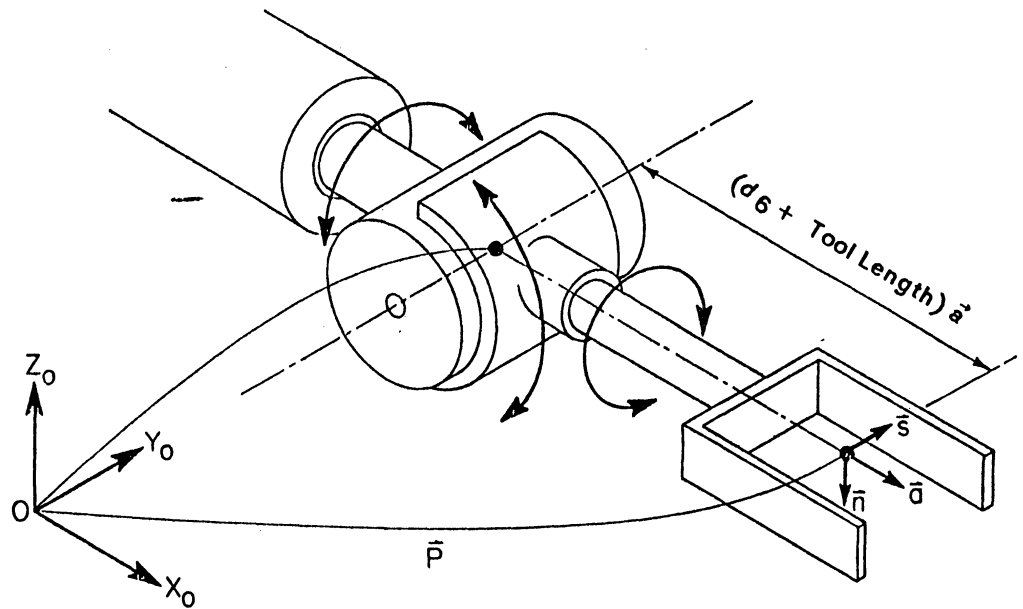


Figure 1 Hand and Reference Coordinate Frames

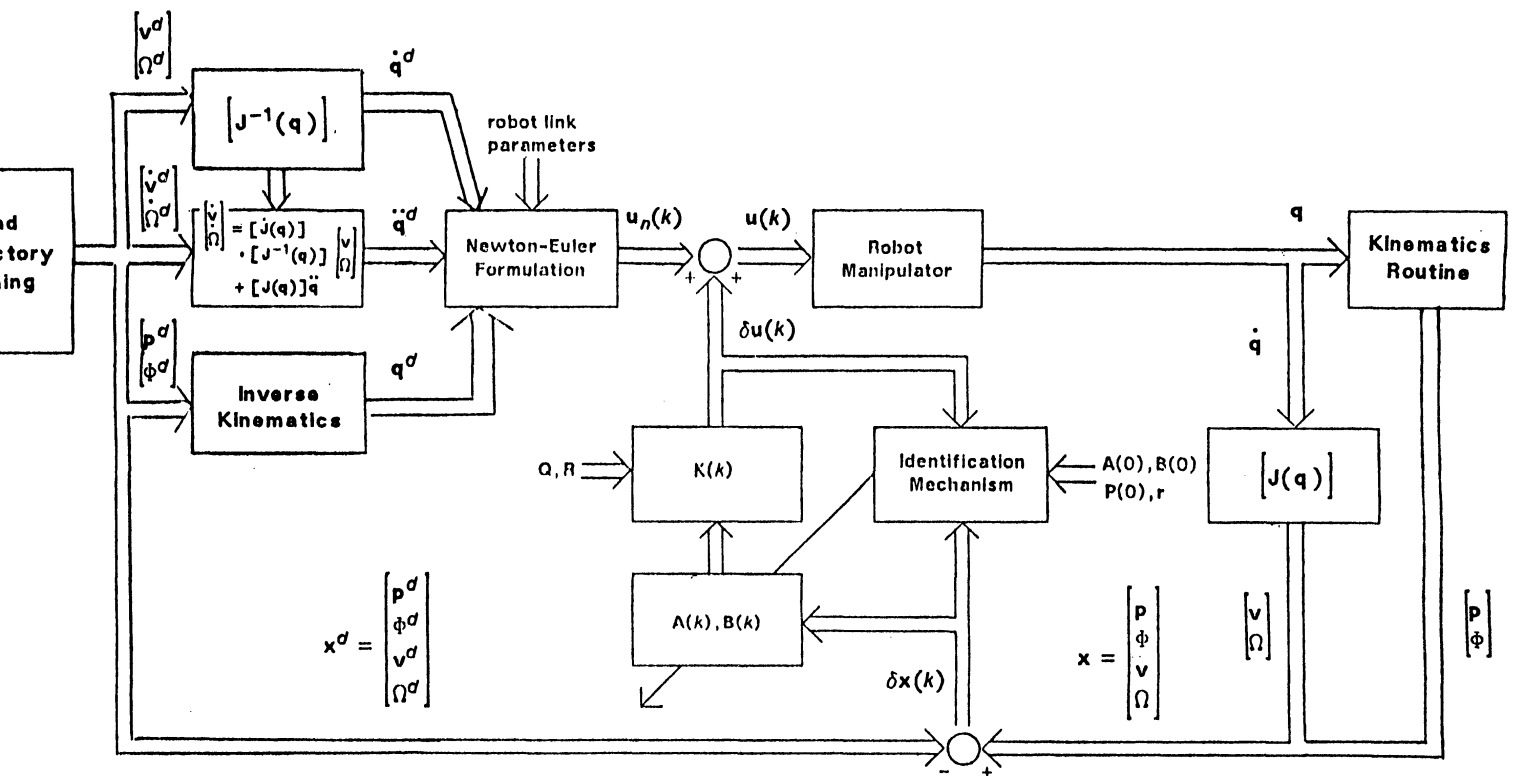


Figure 2 Proposed Resolved Motion Adaptive Control

UNIVERSITY OF MICHIGAN



3 9015 03023 0331