UCRL-21243

Final Progress Report to

Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, California 94550

UCRL--21243

DE90 002977

on

FEASIBILITY STUDY OF USING INDUSTRIAL ROBOT FOR ULTRASOUND TESTING

Principal Investigator: T.C. Hsia
Department of Electrical Engineering and Computer Science
University of California
Davis, California 95616

February 13, 1989

FINAL PROGRESS REPORT ON FEASIBILITY STUDY OF USING INDUSTRIAL ROBOT FOR ULTRASOUND TESTING

Research Objective

To study the feasibility of employing an articulated industrial robot arm to perform ultrasound scanning of a work piece. The robot arm suitable for such application is the Unimation PUMA 560 manipulator with six degrees of freedom.

This research is intended to study the basic properties of several issues which arise in the operations of the UTB facility at LLNL. These issues involve the understanding of control algorithms for collision-free motions of the robot arm with and without redundant kinematics, trajectory generation schemes for automated surface tracking based on 3D CAD/CAM model of the work piece, and the effective use of computer vision and ultrasound sensor to achieve position calibration of a work piece and on-line trajectory modification for accurate surface tracking.

The robotics research laboratory at UC Davis has a research robotic system based on a six degrees-of-freedom Unimation PUMA 560 arm and a VAL-II controller. This system is used as a research tool for the investigation.

This study was funded by LLNL under Purchase Order: LLNL BO34336.

Research Progress

A study on the Unimation PUMA 560 VAL-II controller has been made and its potential capability for implementing continuous 3D trajectory defined by the surface of a solid object has been examined. The findings are summarized below.

The VAL-II Controller is capable of generating smooth and continuous trajectories of the end effector by specifying appropriately positioned via-points along the trajectory as well as the speed of motion. The trajectory is specified by a six-dimensional vector consisting of the end effector (or tool tip) Cartesian space position and orientation. If an object has a known 3-D surface model, the trajectory via-points can then be appropriately generated off-line based on a certain desired scan path and speed. For an object with unknown and arbitrary surface geometry, trajectory via-points would have to be taught using a teach pendant. This teaching process can be very time consuming.

In most cases, we probably can assume that the surface geometry is approximately known with some detailed local information lacking. In order to avoid the manual teaching process described above, it is advantageous to first construct an approximate trajectory, and then to modify this trajectory on-line as the scanning proceeds. The error between the assumed trajectory and the corresponding actual surface can be detected by some appropriate sensor installed at the end effector. In the case of the ultrasound scanning problem being studied, the same ultrasound signal can be utilized as an error detector. An algorithm which implements such an on-line modification task based on the position and orientation errors can be devised to achieve accurate surface tracking. This type of algorithm is essentially the same as that used in Cartesian space robot end effector controllers.

The VAL-II controller provides a real-time trajectory control facility, called ALTER, which allows modification of a nominal trajectory in real-time. The controller has a serial path control through which incremental Cartesian position commands may be sent to the manipulator. These position increments are interpreted in either TOOL or WORLD coordinates, and may be either cumulative (robot location is modified by the sum of all the past ALTER data) or non-cumulative (robot location is modified only by the most recent data).

ALTER has two modes; one is called the external mode and the other is the internal mode. External mode requires a host (or some type of smart sensor) system to provide the path alteration data. The internal mode enables software (an ALTOUT instruction would be incorporated into this software) to provide the path alteration data. In this mode, ALTOUT mimics the role of a host in the external ALTER mode. The ALTOUT instruction will be executed every 28 msec. Since external mode requires quite a lot of hardware support, it is desirable to investigate the internal mode at this time.

In the course of our study, a problem arose when we tried to command the controller to execute a continuous spiral trajectory simulating the task of scanning the surface of a cylinder (or a solid rod) standing in upright position. The orientation of the end effector is required to be always normal to the surface. VAL-II specifies this orientation as 0, A, and T angles, all of which are limited to within the range from –180° to +180° by the design of the inverse kinematics software. If any one of the angles exceeds this limit, VAL-II will reset the angle to be within the range [–180°, 180°]. For example, if angle T has an initial angle of 170° and the command angle of T is 190°, VAL-II will reassign the 190° to –170°, which causes the end effector to turn backward 340° instead of forward 20° as desired. This behavior makes the generation of a continuous spiral trajectory impossible. In other words, the O, A, T orientation angles are constrained to vary no more than 360° even though the three wrist joints can rotate any direction continuously as there are no hardware limits. Thus, the use of a spiral trajectory for scanning a cylinder is not feasible for a VAL-II controller.

A viable alternative in this case is to use a trajectory which is made of a set of circles which are uniformly spaced. When one circle is scanned in one direction (say clockwise), the next circle will be scanned in the opposite direction (counter-clockwise). The circles can be connected to form a continuous scan path covering the surface of the cylindrical object. This scheme has been experimentally tested, and has proven to be very successful.

During the same time period, we have studied the trajectory generation problem for robot arms with redundant joints (more than six joints as in the case of the PUMA 560). Redundant arms have the advantage of facilitating the collision avoidance problem in trajectory planning. Interest in understanding the kinematic behaviors of redundant arms is growing among robotics researchers, as such manipulators are becoming increasingly available commercially.

An attractive joint trajectory generation scheme for redundant robots has been recently proposed by this investigator and his graduate student. A paper describing this result has been prepared and is attached at the end of this report.

The progress reported above has addressed only certain aspects of the issues mentioned in the beginning of this report. Obviously, further studies would be needed in order to gain additional insight concerning the feasibility of using industrial robots for ultrasound testing.

JOINT TRAJECTORY GENERATION FOR REDUNDANT ROBOTS

T. C. Hsia and Z. Y. Guo
Robotics Research Laboratory
Department of Electrical Engineering and Computer Science
University of California
Davis, CA 95616

ABSTRACT: A joint trajectory generation scheme for redundant robots is proposed which uses the redundancy to improve motion performance according to certain objective functions. The objective function can be either analytical or nonanalytical. For nonanalytical objective functions, a least squares scheme is proposed to estimate the gradient vector. In addition, an approximation scheme is developed to compute the pseudo-inverse of the Jacobian. Application of the scheme to a 4 link revolute planar robot manipulator is demonstrated through simulation. Several motion performance criteria are considered and their results analyzed.

I. INTRODUCTION

The problem of inverse kinematics is one of converting Cartesian space position and orientation to joint space displacement. The conversion can be obtained both analytically and numerically. As is well known, the joint trajectory is very important to robot control [6, 7], and many robotics researchers have been attracted to this problem for a long time. It is only in recent years that attention has been paid to robot joint trajectory generation from Cartesian space trajectory specifications through dynamic approach [1-5, 13, 17, 18].

Manipulators with redundant joints are of interest in that they can achieve singularity avoidance and work space obstacle avoidance, improve robot dexterity, and other advantages [9, 10, 11, 12, 15, 19]. The inverse kinematic problem for redundant arms is in general much more complicated than that of non-redundant arms.

The most direct approach to solve the inverse kinematics problem is certainly to obtain a closed form solution. However, this is either not always possible or evaluation of the analytical solution is too time consuming. For a redundant manipulator, the problem can be even more complex since there are usually an infinite number of solutions to a given set of Cartesian coordinates. Hence, an efficient and sophisticated numerical algorithm for solving the inverse kinematics will be greatly appreciated.

Wolovich and Elliot [1] proposed, and Vaccaro and Hill [2-4] subsequently studied, a scheme to generate joint trajectories from Cartesian coordinates using a nonlinear feedback system. This scheme is called the "joint-space command generator". This command generator is depicted in Fig. 1a. An adaptive version of the scheme was also proposed by Hsia, Guo and Li [5].

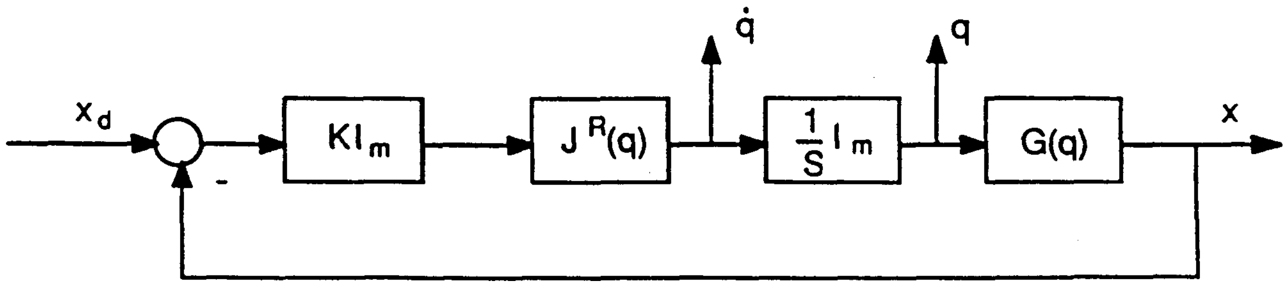


Figure 1a. Position-Velocity (PV) Command Generator

As is well known, the end-effector position $X \in R^m$ of a robot is related to the vector of joint coordinates $q \in R^n$ $(n \geq m)$ by the kinematic equation

$$X = G(q). \tag{1}$$

Most research on the resolution of redundancy has been based on the relationship between joint velocity $\dot{q}$ and Cartesian velocity $\dot{X}$, i.e.,

$$\dot{X} = J(q)\dot{q} \tag{2}$$

where $J(q) = \frac{\partial G}{\partial q} \in R^{mxn}$ is the Jacobian matrix. Assuming the Jacobian is of full rank, Vaccaro and Hill [2] used the right inverse $J^R(q)$ of $J(q)$ to resolve redundancy by letting $\dot{q} = KJ^R \dot{X}$ which yields a local minimum norm least squares joint velocity vector by choosing $J^R = J^T(JJ^T)^{-1}$. Using this approach, a joint position-velocity (PV) command generator can be reduced to a set of decoupled first order linear feedback systems as shown in Fig. 2. The error decay rate of each system can be separately adjusted. Joint acceleration can also be generated by inserting a first order high gain system in the feedforward path of the PV command generator [2].

The system in Fig. 1a will fail to work if J is degenerated since J^R does not exist. Furthermore, this command generator does not attempt to make full use of the redundancy to prevent J from becoming degenerated. Therefore it is desirable to propose an alternative solution. This motivates the extension of the above joint trajectory generator by the authors. The general solution to (2) can be written as

$$\dot{q} = J^- \dot{X} + (I - J^- J)Z \tag{3}$$

where J^- is any generalized inverse defined by $JJ^-J = J$ and Z is an arbitrary vector with proper dimension. Liegeois [8] used Z to avoid joint limits. Many other researchers [9-12] have used Z to improve system performance by associating it with certain performance criteria.

II. REDUNDANCY UTILIZATION

For simplicity, only the PV command generator (Fig. 1a)is examined in the analysis below. To extend the range of applicability of the system in Fig. 1a, we replace J^R with J^- as shown in Fig. 1b. This PV command generator can also be represented by the decoupled linear system in Fig. 2 if and only if the error $e = X_d - X$ is in the range space of J(q), i.e., $e \in \mathcal{R}(J(q))$, where X_d is the desired Cartesian trajectory. This can be seen from the following: In Fig. 1b, we obtain

$$\dot{q} = KJ^- e \tag{4}$$

and then multiply both sides by J to obtain

$$\dot{X} = KJJ^- e \tag{5}$$

From matrix theory [14], $JJ^-e = e$, or $\dot{X} = Ke$, if and only if $e \in \mathcal{R}(J)$. This condition is less restrictive than that of J being of full rank.

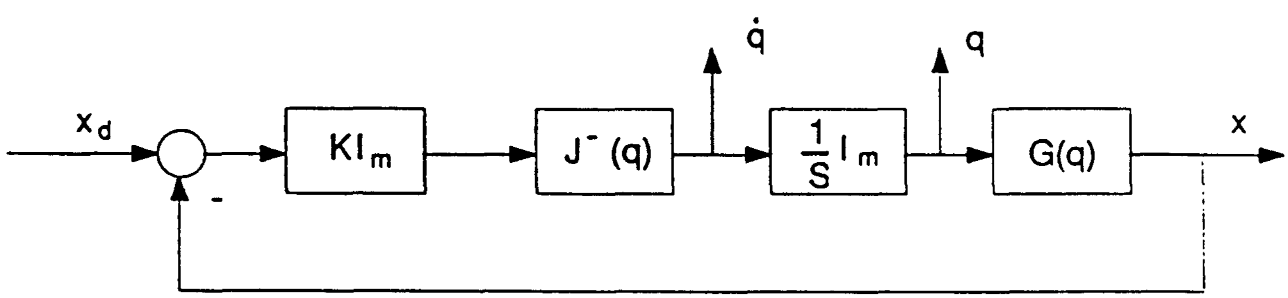


Fig. 1b. General Position-Velocity (PV) Command Generator

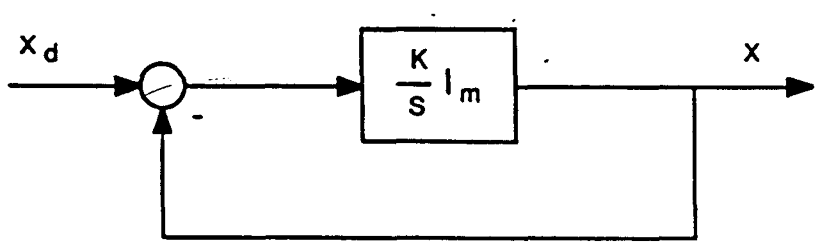


Figure 2. Equivalent Linear System for PV Command Generator

Two important observations can be made with regard to the above result: (i) Equation (5) allows us to generalize the framework of the linearity property of the command generator to include the case where J is degenerated. (ii) The above result also implies that the linearity property (5) can be preserved even if we add a term defined in the null space of J to the right side of (4). This vector term can be used by the command generator to facilitate the utilization of the redundancy property of the manipulator.

The solution $\dot{q}$ obtained from (4) lacks physical interpretation provided that J^- is chosen specifically. The most important solution is obtained by replacing J^- with J^+, the Penrose-Moore generalized inverse, which gives the minimum norm least squares solution. J^+ is defined by $JJ^+J = J$, $J^+JJ^+ = J^+$, $(JJ^+)^+ = JJ^+$, and $(J^+J)^+ = J^+J$. When J has full row rank, $J^+ = J^T(JJ^T)^{-1}$. When J has less than full row rank, J^+ can be computed by singular value decomposition, i.e., $J^+ = V\Sigma^+U^T$ [14, 10].

Let $X_{\mathcal{R}(J)}$ denote the vector obtained by projecting X onto $\mathcal{R}(J)$, the range space of J, and $X_{\mathcal{N}(J)}$ denote the vector obtained by projecting X onto $\mathcal{N}(J)$, the null space of J. Any vector $X \in \mathbf{R}^m$ can be written as $X = X_{\mathcal{R}(J)} + X_{\mathcal{N}(J^T)}$ since $\mathbf{R}^m = \mathcal{R}(J) \oplus \mathcal{N}(J^T)$, where $\oplus$ denotes direct sum.

Lemma: $\mathcal{N}(J^+) = \mathcal{N}(J^T)$

The lemma is true since by matrix singular value decomposition [14, 20], $J = U\Sigma V^T$, $J^T = V\Sigma^T U^T$, and $J^+ = V\Sigma^+ U^T$. Any vector X yielding $J^TX = 0$ will make $J^+X = 0$, and vice versa.

Theorem: The PV system in Fig. 1b with J^- replaced by J^+ can be described by $\dot{X} = K(X_d - X)_{\mathcal{R}(J)}$.

The theorem is true since from (5)

$$\begin{aligned} \dot{X} &= KJJ^+(X_d - X) \\ &= KJJ^+(X_d - X)_{\mathcal{R}(J)} + KJJ^+(X_d - X)_{\mathcal{N}(J^T)} \\ &= KJJ^+(X_d - X)_{\mathcal{R}(J)} \\ &= K(X_d - X)_{\mathcal{R}(J)} \end{aligned}$$

The third equality is true by invoking the Lemma.

The above theorem generalizes further the framework of the command generator in the sense that the null space component (in $\mathcal{N}(J^T)$) of input vector X_d will be completely suppressed. Note that we cannot assert that $X(t) \in \mathcal{R}(J)$ for all $t \in [0, \infty)$ because of the time varying property of $\mathcal{R}(J)$; i.e., J(q) may change its rank. Therefore we provide a workable command generator even if J is degenerated. The system now is not "quite" linear.

Since the null space component in $\mathcal{N}(J)$ has no effect on the end-effector motion, we can use it to improve system performance according to some objective function, e.g., maximize the minimal singular value to prevent J from becoming singular. The proposed redundancy utilization command generator (PV type) is depicted in Fig. 3. From this figure, we can see that

$$\dot{q} = KJ^{+}(X_d - X) + K\dot{q}_m$$

where $$\dot{q}_m = \alpha(I - J^{+}J)\nabla\Phi(q)$$

in which $\Phi(q)$ is the objective function of interest, $\nabla\Phi(q) = \frac{\partial\Phi(q)}{\partial q}$, and α is an arbitrary scaling factor governing the speed of the performance improvement.

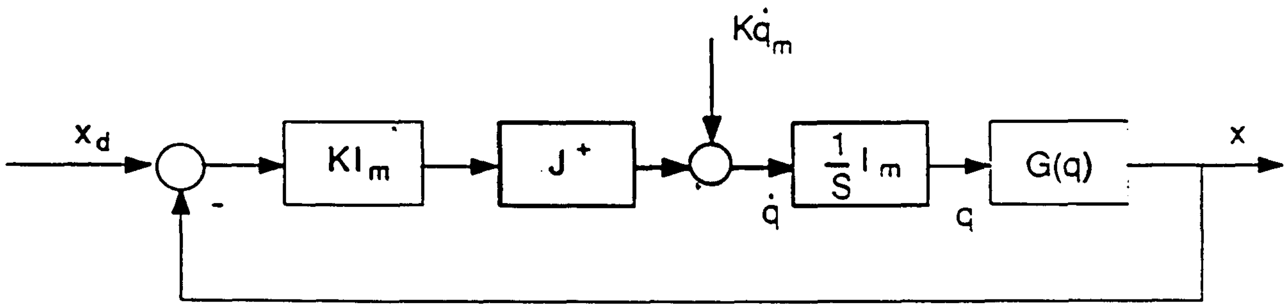


Figure 3. Redundancy Utilization Command Generator

If α is small, the performance improvement effect is very limited. However, if α is too large, $\dot{q}$ may become excessively large, causing the robot to jump to the mirror-image configuration resulting in large chattering in q. It has been shown that a heuristically selected fixed α would work well for some problems [10], but it would fail to work well in a certain "range" of the configuration. In practice it is desirable to say something about the speed of the performance improvement. In the following, we propose a way of selecting α which guarantees a prespecified performance improvement speed wherever possible.

From above, we know that

$$\dot{q} = KJ^{+}(X_d - X) + K\alpha(I - J^{+}J)\nabla\Phi(q). \tag{6}$$

The first term, called the primary term, is used to satisfy Cartesian trajectory requirements. The second term, called the secondary term, is used to improve the robot performance. It is seen that the primary term will change the optimal value of $\Phi(q)$ as the trajectory evolves, and the secondary term will try to track the optimal value wherever possible.

Since

$$\frac{d\Phi}{dt} = [\nabla\Phi(q)]^T \dot{q} \tag{7}$$

From (6), we have

$$\frac{d\Phi}{dt} = K\left\{[\nabla\Phi(q)]^T J^+(X_d - X) + \alpha\,[\nabla\Phi(q)]^T (I - J^+J)\,\nabla\Phi(q)\right\} \tag{8}$$

The second term vanishes, $[\nabla\Phi(q)]^T (I - J^+J)\,\nabla\Phi(q) = 0$, if and only if (i) $\nabla\Phi(q) = 0$, or (ii) $\nabla\Phi(q) \perp \mathcal{N}(J)$ since $(I - J^+J)$ is positive semi-definite. Case (i) simply implies that a local minimum or maximum is reached. Case (ii) tells us that the projection of gradient $\nabla\Phi$ onto $\mathcal{N}(J)$ is zero, hence no performance improvements can be made. In both cases, α can be set to zero. When $[\nabla\Phi(q)]^T (I - J^+J)\,\nabla\Phi(q) \neq 0$, $\Phi(q)$ can be further improved and we can set the performance improvement rate at some "arbitrary" level. For example we can let $\frac{d\Phi}{dt} = \pm\lambda$ (where λ is a positive constant, and the "+" and "−" sign should be chosen based on whether $\Phi(q)$ is to be maximized or minimized) by choosing α as

$$\alpha = \frac{\pm\lambda/K - [\nabla\Phi(q)]^T J^+ (X_d - X)}{[\nabla\Phi(q)]^T (I - J^+J)\,\nabla\Phi(q)} \tag{9}$$

With such a choice of α, the objective function $\Phi(q)$ will be increasing or decreasing linearly. Another possibility is to let $\frac{d\Phi}{dt} = -\lambda(\Phi - M)$ (where M is a constant), by selecting α as

$$\alpha = \frac{-\lambda[\Phi(q) - M]/K - [\nabla\Phi(q)]^T J^+ (X_d - X)}{[\nabla\Phi(q)]^T (I - J^+J)\,\nabla\Phi(q)} \tag{10}$$

In this case $\Phi(q)$ is converging to an arbitrary constant M exponentially. This scheme can provide a solution, to some extent, to the problem of workspace reduction [19].

III. COMPUTATIONAL ISSUES

In this paper, a discrete version of the command generation scheme is implemented [5]. Simulation studies are carried out for a four link planar robot. Certain computational issues associated with the scheme are discussed in this section.

A. Approximation of Pseudoinverse: We can use the singular value decomposition method to compute J^+, but it is quite time consuming. For a redundant manipulator, there is an infinite number of configurations q for a given set of Cartesian coordinates. In general, we should always search for a robot configuration which possesses a full rank Jacobian. As a matter of fact, a primary reason for our interest in a redundant robot is to avoid configuration singularity by making use of certain dexterity measures [11]. Hence, in the "regular operation" situation, we can assume that J is of full row rank. Therefore, $J^+ = J^T(JJ^T)^{-1}$ exists at all times. In this case, we can apply the following approximation scheme for J^+ to speed up the computation.

Computationally, the matrix inverse operation in J^+ is not desirable for computational speed and efficiency [15]. However, if we implement the proposed command generator with sufficiently high sampling rate, we can assume $J(q(t_{K+1}))$ to be quite "close" to $J(q(t_K))$. From this assumption, an approximation scheme to the pseudoinverse can be developed as follows:

Let $G_K = (J_K J_K^T)^{-1}$, and $G_{K+1} = (J_{K+1} J_{K+1}^T)^{-1} = (J_K J_K^T)^{-1}(I + \Delta)^{-1}$.

Hence $$G_{K+1} = G_K(I + \Delta)^{-1} \tag{11a}$$

$$= G_K(I - \Delta + \Delta^2 - \Delta^3 + \ldots) \tag{11b}$$

The second equality (11b) is valid if $-1 < \lambda(\Delta) < 1$; $\lambda(\Delta)$ denotes the eigenvalue of Δ which is real since Δ is symmetric. Δ can be computed from (11a) as follows:

$$\Delta = (J_{K+1} J_{K+1}^T)\, G_K - I \tag{12}$$

With Δ computed in (12), G_{K+1} can be approximated by a truncated series in (11b). Note if $-1 < \lambda(\Delta) < 1$, the above approximation can be made arbitrarily precise by including a sufficient number of high order terms of Δ. Finally, J^+_{K+1} can be obtained from $J^+_{K+1} = J^T_{K+1} G_{K+1}$. We see that the pseudoinverse can be updated without matrix inversion.

B. <u>Gradient Computation of Objective Function</u>: Vectors in the null space of J, $\mathcal{N}(J)$, can be used to increase or decrease certain objective functions along the trajectory of a redundant robot. This vector can only reconfigurate the manipulator, leaving the Cartesian trajectory unaltered. Thus the gradient vector of an objective function can be projected onto the null space of J to serve as a mechanism to optimize the objective function. So, if we let $\Phi(q)$ be the objective function to be optimized, then (see (6))

$$\dot{q} = K\left[J^+(X_d - X) - \alpha(I - J^+J)\nabla\Phi\right] \tag{13}$$

Klein and Blaho [11] studied and compared several objective functions, i.e., dexterity measures and the associated inverse kinematic solutions. That is, given Φ and the desired fixed Cartesian coordinates, they searched for an optimal arm posture to optimize Φ. The functions investigated include the determinant of (JJ^T), the joint range availability criterion $\Phi(q) = \sum (q_i - \bar{q}_i)^2$, the condition number of J, and the minimal singular value of J. In our command generator case, it is not necessary to achieve an optimal value for the objective function at every instant of time, since doing so may require excessively large $\dot{q}$ if the value of the objective function is far away from the optimal initially. A large $\dot{q}$ is not desirable for robot control. In this paper, the value of α as introduced in (9) and (10) is clipped such that $\| \alpha(I - J^+J) \nabla\Phi \| \leq r \| J^+ (X_d - X) \|$ to make the trajectory solution "close" to the minimum norm least squares solution at each sampling instant, where r is a predetermined number that controls the upper bound of joint velocity and objective function improving rate.

Among various continuously differentiable objective functions, some are analytically expressible and some are not. The gradient of those analytical objective functions is easy to obtain. However, to obtain the gradient of other objective functions is quite involved (e.g., consider the minimal singular value of J as the objective function). In those cases, we propose to estimate the gradient on-line using the recursive least squares algorithm presented below.

Assume that there is a way to compute the value of the objective function and

$$\Delta\Phi(kT) = \left[\frac{\partial\Phi}{\partial q_1}, \frac{\partial\Phi}{\partial q_2}, \ldots, \frac{\partial\Phi}{\partial q_n}\right]_{q=q(kT)} \Delta q(kT) \qquad (14)$$

where $\left[\frac{\partial\Phi}{\partial q_i}\right]$ is the gradient vector, $\Delta\Phi(kT)$ and $\Delta q(kT)$ are incremental changes of objective function Φ and joint coordinates q at time $t = kT$. Let the estimated gradient vector be denoted as W:

$$\left[\frac{\partial\hat{\Phi}}{\partial q_1}, \frac{\partial\hat{\Phi}}{\partial q_2}, \ldots, \frac{\partial\hat{\Phi}}{\partial q_n}\right]^T = \left[w_1, w_2, \ldots, w_n\right]^T = W. \qquad (15)$$

Assume that the elements w_i are slowly varying, they can be tracked by using the following least squares updating law:

$$w(k+1) = w(k) + P(k+1)\,\Delta q(k+1)\left[\Delta\Phi(k+1) - W^T(k)\,\Delta q(k+1)\right] \qquad (16)$$

$$P(k+1) = \frac{1}{\beta}\left[P(k) - \frac{P(k)\,\Delta q(k+1)\,\Delta q^T(k+1)\,P(k)}{\beta + \Delta q^T(k+1)\,P(k)\,\Delta q(k+1)}\right] \tag{17}$$

where P(k) is a n x n symmetric matrix with initial condition $P(0) = \alpha I$, $\alpha \geq 10^3$, and β is the forgetting factor, $0 < \beta < 1$.

IV. SIMULATION STUDIES

Application of the algorithm derived above to a four link planar robot shown in Fig. 4 is demonstrated through simulations. The kinematics G(q) and Jacobian J(q) of the robot are given by

$$G(q) = \begin{bmatrix} L_1C_1 + L_2C_{12} + L_3C_{123} + L_4C_{1234} \\ L_1S_1 + L_2S_{12} + L_3S_{123} + L_4S_{1234} \end{bmatrix} \tag{18}$$

$$J(q) = \begin{bmatrix} -L_1S_1 - L_2S_{12} - L_3S_{123} - L_4S_{1234} & -L_2S_{12} - L_3S_{123} - L_4S_{1234} & -L_3S_{123} - L_4S_{1234} & -L_4S_{1234} \\ L_1C_1 + L_2C_{12} + L_3C_{123} + L_4C_{1234} & L_2C_{12} + L_3C_{123} + L_4C_{1234} & L_3C_{123} + L_4C_{1234} & L_4C_{1234} \end{bmatrix} \tag{19}$$

where

$C_1 = \cos(q_1)$, $C_{12} = \cos(q_1 + q_2)$, $C_{123} = \cos(q_1 + q_2 + q_3)$, $C_{1234} = \cos(q_1 + q_2 + q_3 + q_4)$

$S_1 = \sin(q_1)$, $S_{12} = \sin(q_1 + q_2)$, $S_{123} = \sin(q_1 + q_2 + q_3)$, $S_{1234} = \sin(q_1 + q_2 + q_3 + q_4)$.

L_1, L_2, L_3, and L_4 are the link lengths of links 1, 2, 3, and 4 respectively. In the simulation study here, $L_1 = L_2 = L_3 = L_4 = 1$ m.

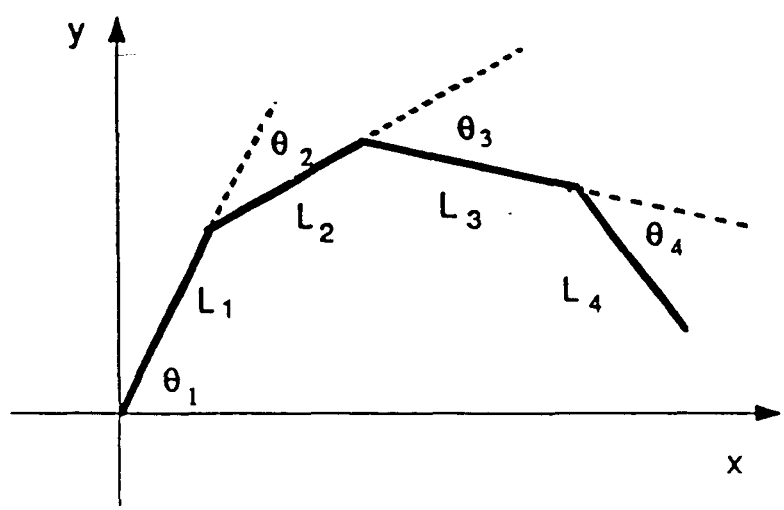


Figure 4. Four Link Planar Robot

Tustin's approximation scheme was implemented for numerical integration. The integration interval is 0.002 sec. The updating interval of the estimated gradient is 0.01 sec. For the least squares gradient estimate W, previous experience [5] has shown that estimation accuracy is not satisfactory when using the joint increments Δq, due to its lack of strong perturbation. Thus W is estimated in the following way: At each iteration, $\Phi(kT)$ is evaluated on the basis of $q(kT)$. Then $q(kT)$ is perturbed with a small random vector $\Delta\tilde{q}$ (zero mean and small variance) by which the corresponding perturbation $\Delta\tilde{\Phi}$ is computed. Then we apply $\Delta\tilde{q}$ and $\Delta\tilde{\Phi}$ to the updating law (16) (17) to estimate the gradient $W(kT)$.

In the simulation, we used $K = 100$, $\beta = 0.95$, and $r = 8$. Initially, the robot is at a configuration expressed in joint space coordinates $q_0 = [179^\circ, 0^\circ, -178^\circ, 0^\circ]^T$, or in Cartesian space coordinates $X_0 = [0, 0.07]^T$m. The desired Cartesian trajectory is a straight line connecting X_0 to $X_f = [2.5, -2.43]^T$m with a BBPB (Bang-Bang Parabolic Blend) [16] time profile. Transfer time is 1.5 seconds.

Part of the dexterity measures studied in [11] are used in this paper as objective functions. For comparison purposes the minimum norm least squares joint trajectory and error in x and y coordinates are shown in Fig. 5. Fig. 6 shows the trajectory with the joint range availability objective function $\Phi(q) = \sum_{i=1}^{4} q_i^2$, which makes the joint angles cluster around zero degree. In this case the gradient $\nabla\Phi$ is analytically available. In this simulation, we have chosen $\frac{d\Phi}{dt} = -20$ rad/sec.

In our next experiment, we used an analytically unexpressible objective function, the minimum singular value of J, to test the gradient estimation scheme. The results are shown in Fig. 7 with $\frac{d\Phi}{dt} = 1.6$. . By examining both Fig. 6 and Fig. 7, it is obvious that the objective functions increase or decrease linearly wherever possible, and system performance is decoupled as expected (we purposely choose x to increase and y to decrease the same distance). Note that the value of the objective function depends on joint angles. As the joint trajectory evolves, the objective function may encounter local minima or maxima. Thus, when different performance improvement rates are used for a given objective function, the objective function may converge to a different local optimal solution. This fact is illustrated by Fig. 8 when $\frac{d\Phi}{dt} = 1.2$ was used for the same objective function studied in Fig. 7. So an interesting open question to be asked is how to choose a performance improvement rate to arrive at the global minimum or maximum of an objective function for a given Cartesian trajectory.

The results obtained above have been compared to the results using approximate J^+. Simulation shows that in order to obtain identical results. approximation of J^+ using the first

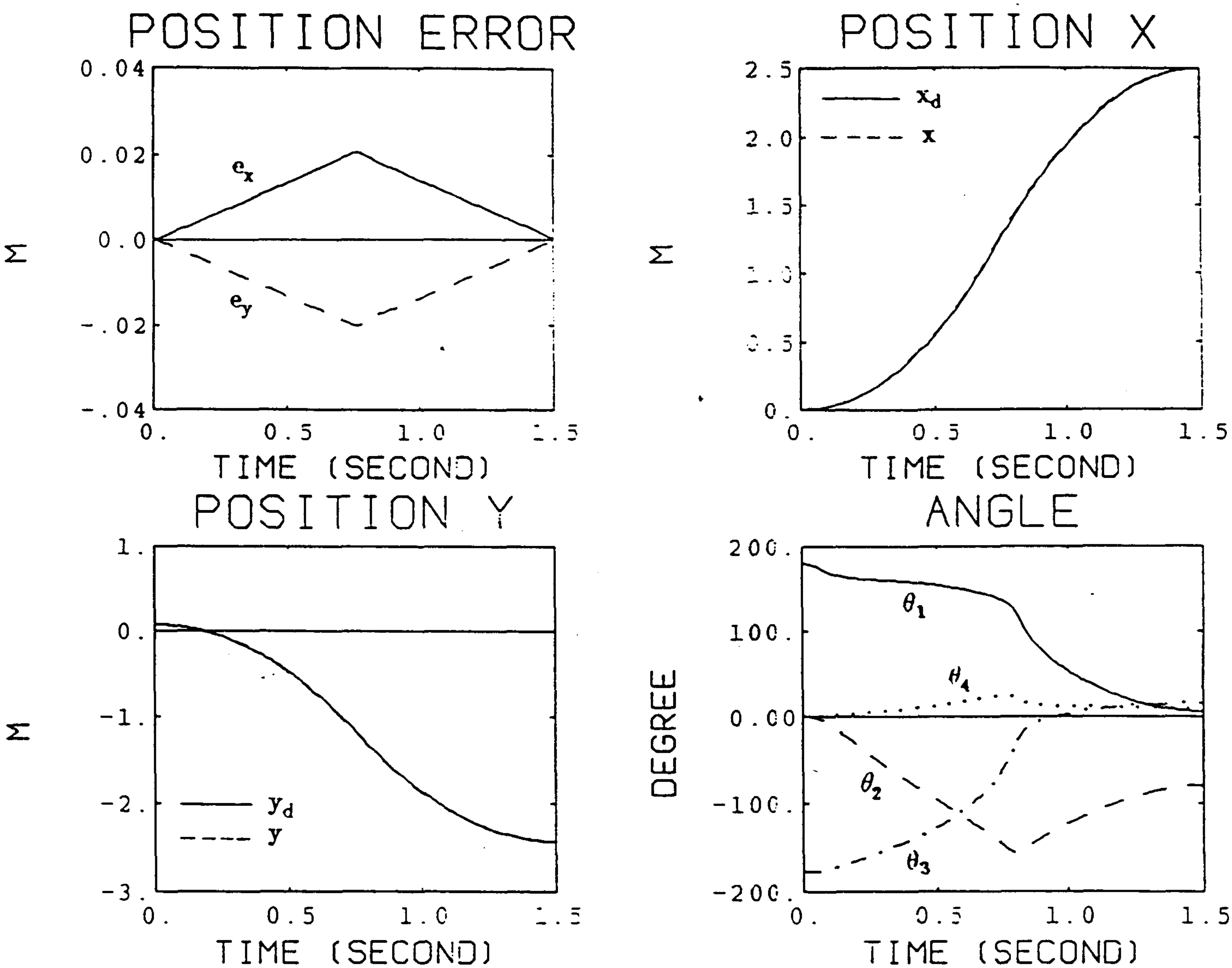


Figure 5. Least Squares Minimal Norm Trajectories

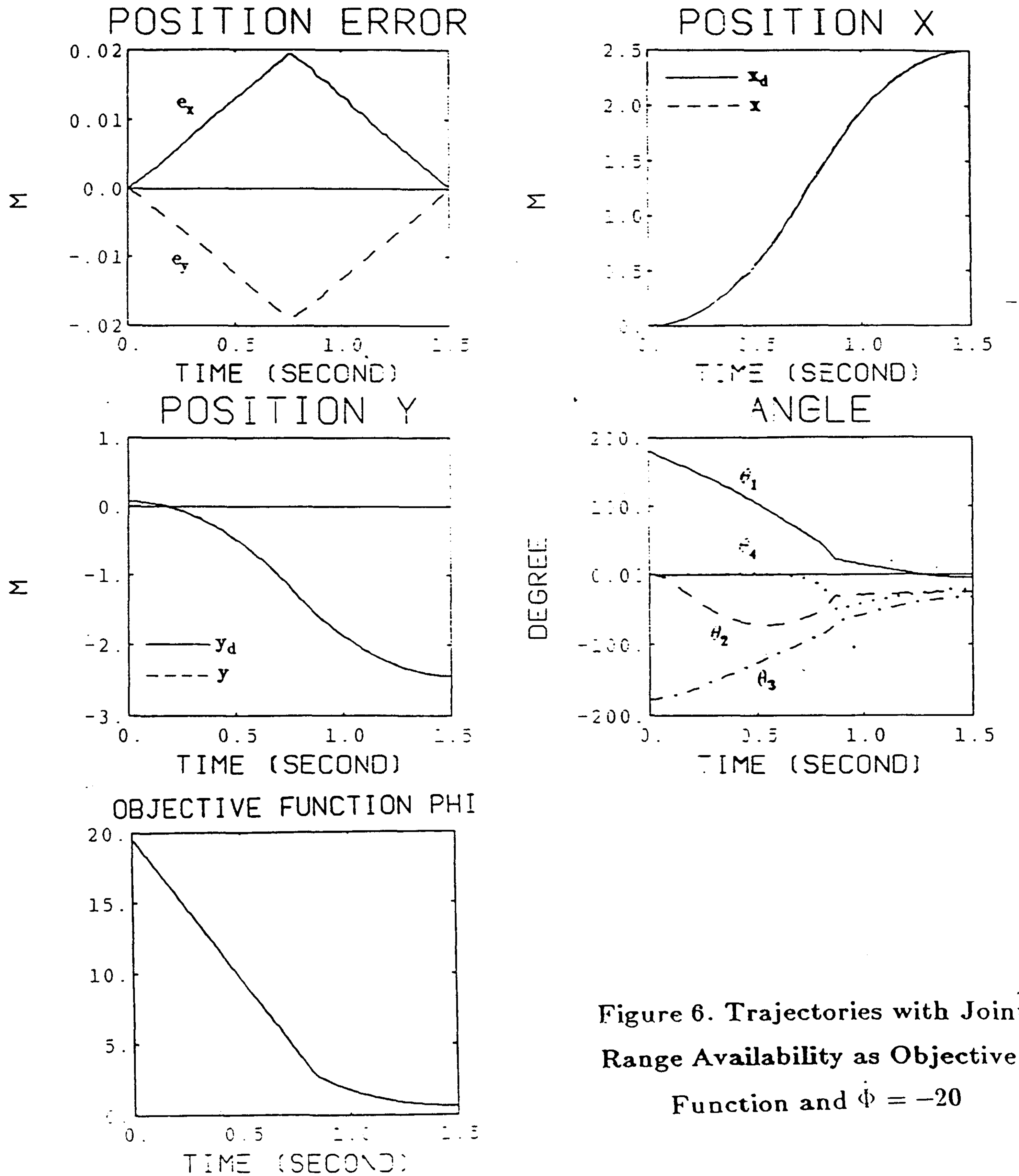
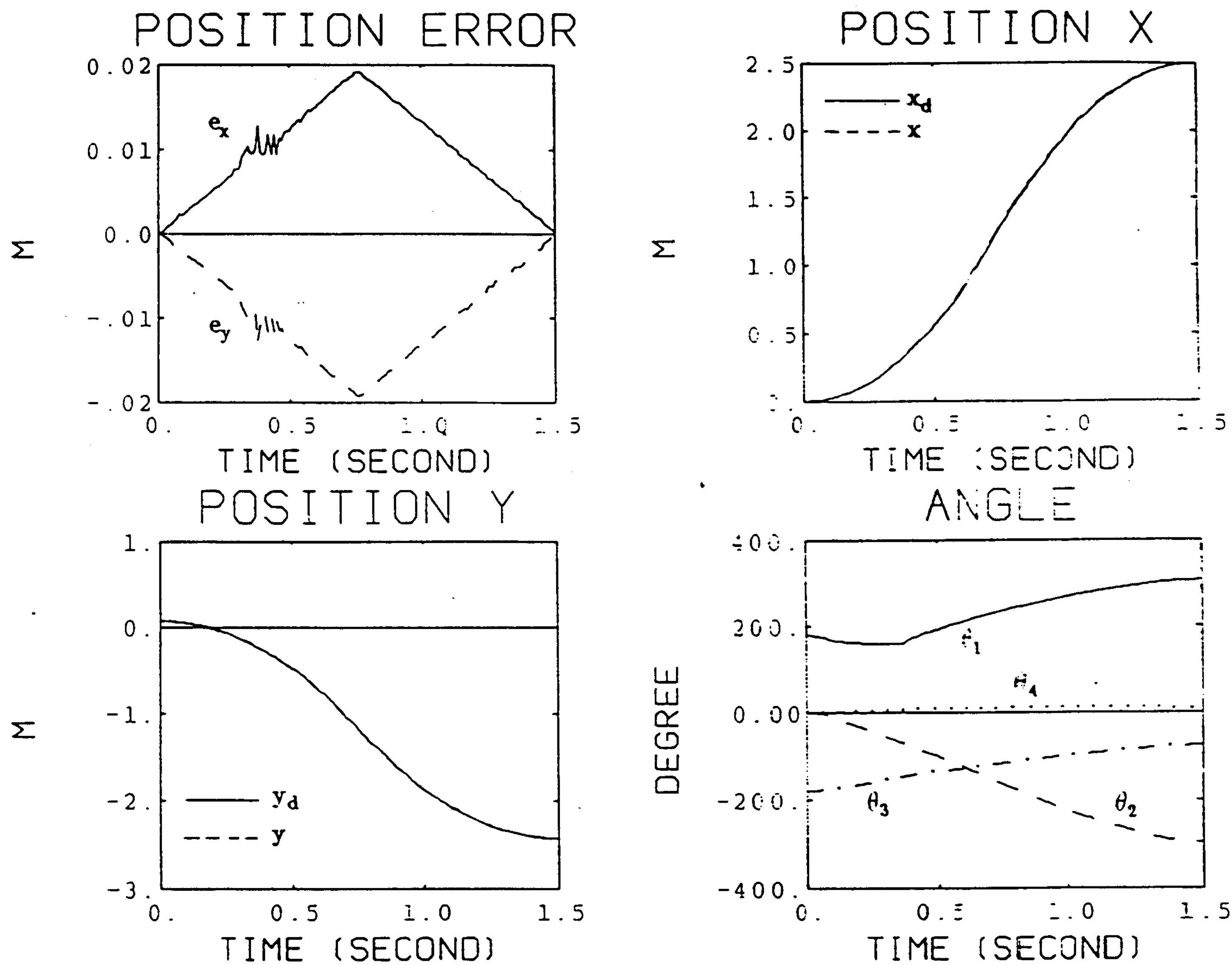


Figure 6. Trajectories with Joint Range Availability as Objective Function and $\dot{\Phi} = -20$



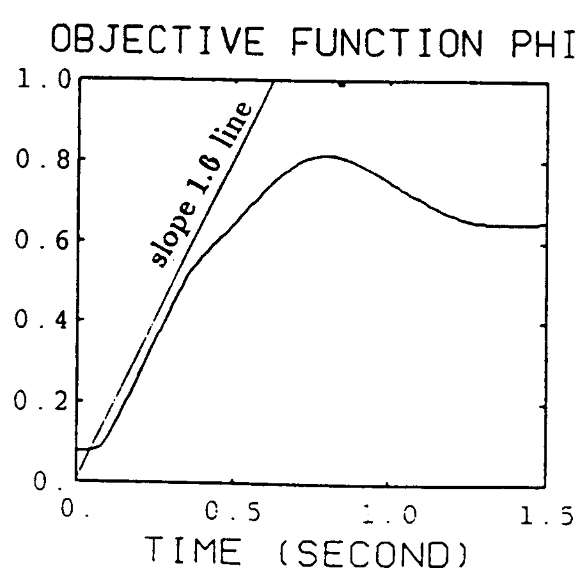
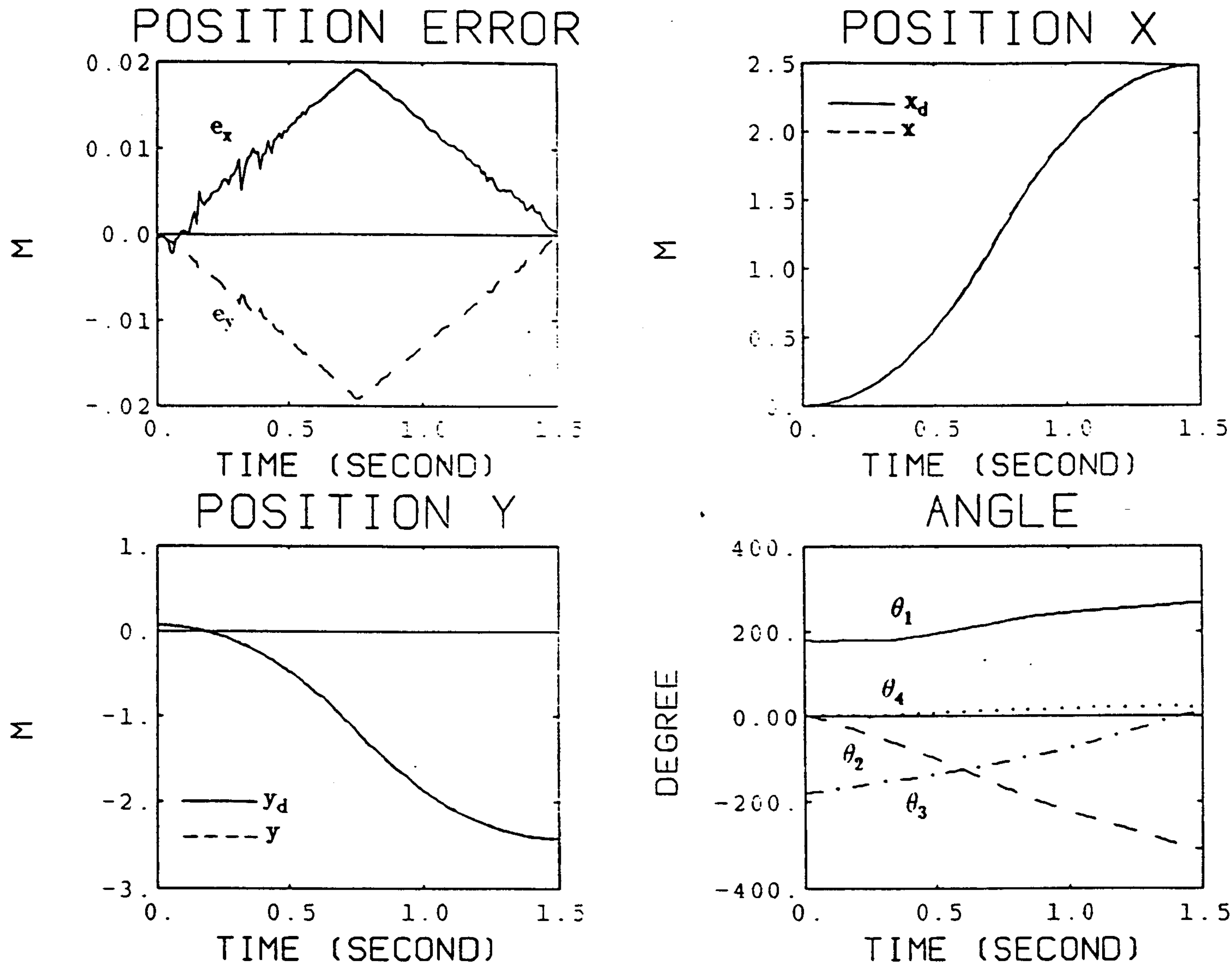


Figure 7. Trajectories with Minimal Singular Value as Objective Function and $\dot{\Phi} = 1.6$



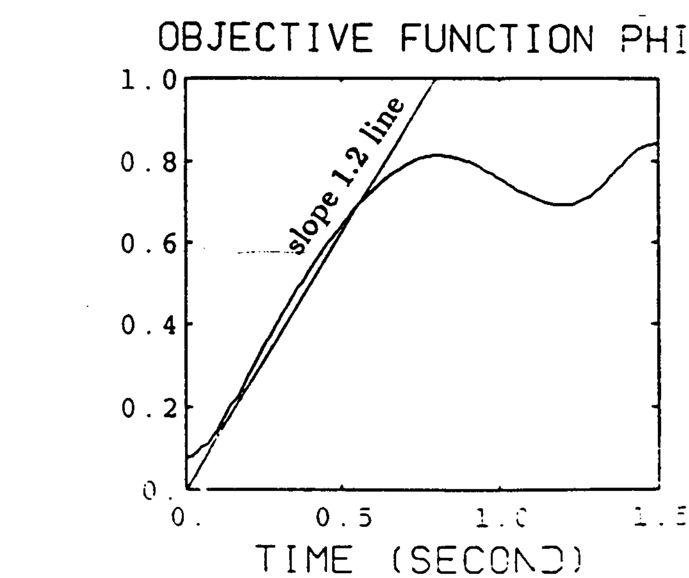


Figure 8. Trajectories with Minimal Singular Value as Objective Function and $\dot{\Phi} = 1.2$

three terms in (11b) is sufficient. The effects of the approximation scheme are experimentally proven to be quite satisfactory.

V. CONCLUSION

The joint command generator concept is extended to improve the redundant robot motion performance along the path of the trajectory by using the vector in null space $\mathcal{N}(J)$. Due to the upper bound of joint velocity, the goal here is not to obtain the optimum of the objective function at every point on the Cartesian trajectory but rather to improve the performance step by step by a guaranteed rate wherever possible. If the gradient of the continuously differentiable objective function cannot be easily obtained, a least squares algorithm is proposed to estimate the gradient vector. Furthermore, on-line matrix inversion of the Jacobian is avoided by using an approximation scheme. The proposed trajectory generation concept has been demonstrated through simulation and proven to be quite useful. The scheme proposed here has great potential for obstacle avoidance. This possibility is being investigated actively.

VI. REFERENCES

1. W. A. Wolovich and H. Elliot, "A Computational Technique for Inverse Kinematics," Proc. 23rd Conf. Decision and Control, Las Vegas, pp. 1359-1363, 1984.
2. R. J. Vaccaro and S. D. Hill, "A Joint-Space Command Generator for Cartesian Control or Robotic Manipulators," IEEE Journal of Robotics and Automation, Vol. RA-4, No. 1, pp. 70-76, February 1988.
3. S. D. Hill and R. J. Vaccaro, "A Command Generator for 6 D-o-F Robotic Manipulators," Proc. 1987 Int. Conf. Robotics and Automation, Raleigh, pp. 1104-1109, March 1987.
4. S. D. Hill and R. J. Vaccaro, "A Discrete-Time Robotic Command Generator," Proc. 1988 IEEE Conf. Robotics and Automation, Raleigh, Philadelphia, pp. 675-680, 1988.
5. T. C. Hsia, Z. Y. Guo, and L. Li, "An Adaptive Joint Trajectory Generation Scheme for Cartesian Robot Control," Proc. Eleventh IASTED Int. Symposium on Robotics and Automation, Santa Barbara, pp. 120-124, May 25-27, 1988.
6. T. J. Tarn, A. K. Bejczy, A. Isidori, and Y. L. Chen, "Nonlinear Feedback in Robot Arm Control," Proc. 23rd IEEE Conf. on Decision and Control, Las Vegas, pp. 736-751, 1984.
7. T. C. Hsia, T. A. Lasky and Z. Y. Guo, "Robust Independent Robot Joint Control: Design and Experimentation," Proc. 1988 IEEE Conf. on Robotics and Automation, Philadelphia, pp. 1329-1334, April 1988.
8. A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Trans. Syst. Man. Cybern., Vol. SMC-7, No. 12, pp. 868-871, 1977.
9. T. Yoshikawa, "Dynamic Manipulability of Robot Manipulators," Proc. 1985 IEEE Conf. on Robotics and Automation, St. Louis, pp. 1033-1038, March 1985.
10. R. Dubey and J. Y. S. Luh, "Redundant Robot Control for Higher Flexibility," Proc. 1987 IEEE Conf. on Robotics and Automation, Raleigh, pp. 1066-1072, March 1987.
11. C. A. Klein and B. E. Blaho, "Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators," Int. Jour. of Robotics Research, Vol. 6, No. 2, pp. 72-83, Summer 1987.
12. C. A. Klein and C. H. Huang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," IEEE Trans. Syst. Man. Cybern., Vol. SMC-13, pp. 245-250, 1983.
13. L. Sciavicco and B. Siciliano, "A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators," IEEE Jour. of Robotics and Automation, Vol. RA-4, No. 4, pp. 403-410, August 1988.

14. C. R. Rao and S. K. Mitra, Generalized Inverse of Matrices aned Its Application, John Wiley and Sons, New York, 1971.

15. R. V. Dubey, J. A. Euler and S. M. Babcock, "An Efficient Gradient Projection Optimization Scheme for a Seven Degree-of-Freedom Redundant Robot with Spherical Wrist," IEEE Int. Conf. on Robotics and Automation, Philadelphia, pp. 28-36, April 24-29, 1988.

16. W. A. Wolovich, Robotics: Basic Analysis and Design, Holt, Rinehart and Winston, New York, 1987.

17. C. Chevallerean and W. Khalil, "A New Method for the Solution of the Inverse Kinematics of Redundant Robots," IEEE 1988 Conf. on Robotics and Automation, Philadelphia, pp. 37-42, April 24-29, 1988.

18. H. Das, J-J. E. Slotine and T. B. Sheridan, "Inverse Kinematic Algorithms for Redundant Systems," IEEE 1988 Conf. on Robotics and Automation, Philadelphia, pp. 43-48, April 24-29, 1988.

19. I. D. Walker and S. L Marcus, "Subtask Performance by Redundance Resolution for Redundant Robot Manipulators," IEEE Jour. of Robotics and Automation, Vol. RA-4, No. 3, pp. 350-354, June 1988.

20 G. H. Golub and C. F. Van Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, MD, 1983.