Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

14th International Conference on Applied Mechanics and Mechanical Engineering.

Path Planning of High Speed Manipulators according to Actuator Dynamics

Mohamed H. Mabrouk* and Hussien M. Mahgoub*

Abstract:

The equations of motion for a robot manipulator with high speed are highly non-linear and coupled. The combined equations are developed, piecewise linearized and discretized to form a set of state difference equations by which a mathematical basis is provided for computer control and simulations. A mathematical model for describing the complete dynamics of a robot manipulator is obtained by combining the non-linear and coupled dynamic equations of motion of a manipulator linkage with those for its actuators. The adaptive control scheme and on-line computational scheme are employed to drive robot manipulators to follow high trajectories specified by continuous time functions. These schemes are briefly described and the control strategy involves a combination of feedforward and feedback control. Based on the control strategy, a program has been developed for simulations using FORTRAN. The inclusion of actuator dynamics in the modeling is shown to influence the controller design substantially. With appropriate controller design procedures, the tracking performance of the manipulators is shown to be very good. The introduced control strategy has been used to develop a planar robot manipulator with three degrees of freedom planar robot (RRR) case study using MATLAB. The result of the case study shows the capability of the control strategy to assist in controlling a manipulator using a GUI.

KEYWORD
Intelligent control and robotics, industrial robot, digital processing, digital signal processing, MATLAB

* **Egyptian Armed Forces**

1. Introduction:

Industrial robots have their origins in both numerically controlled machine tools and remote manipulator or so-called teleoperators. A robot manipulator is required to follow a trajectory specified by continuous time functions in many industrial applications. Arc welding, spray-painting, machining and material cutting are examples. The ideal or intended trajectory can be taught by demonstration or it can be defined mathematically. Knowledge of the trajectory and possession of a mathematical model of the robot provide a basis for calculating the joint torques which, in an error free environment, would allow the robot to follow the trajectory more or less precisely. To deal with modeling inaccuracy, parameter uncertainties and imperfect joint torque production, feedback control is necessary. With sufficient quality in the feedforward control, the feedback control problem is one involving small perturbations from known conditions and it can be treated by linear system theory. A common further simplification is to decouple the control of each joint from the remainder of the problem, position and velocity feedbacks being employed within each local controller [1-9] to provide joint position control. In fact, the individual joint feedback control problem involves parameters which in general vary widely depending on the positions, velocities and accelerations of the other joints. If the local controllers have fixed parameters, the design is likely to involve heavy compromise and if the parameters are to vary, some scheme for the adaptation [2,10-13] must be provided and arrangements must be made for the supply of any necessary information.

Many adaptive control schemes operate in conjunction with on-line parameter estimation [2,11]. An alternative scheme for combining feedforward control with adaptive feedback control was described in [12] and further developed in [13]. According to this scheme, the small perturbation feedback control problem around any nominal condition on the ideal trajectory is treated by linear discrete optimal control theory. A so called State and Control Prediction Strategy was introduced in [13] to accommodate time delays associated with A/D, D/A conversions and with signal processing. The essence of tin's strategy is that joint position and velocity errors used for feedback control are predicted far enough ahead of the actual motion for the appropriate control to be applied at each step in the trajectory. In most previous research, controller designs and performance computations arc based on the ideal actuator assumption, namely, that the control variables are joint torques/forces. In reality, actuator performance limitations are likely to exert a strong influence on the controller design and the robot performance and it is advantageous to include realistic actuator properties in the modeling.

This paper contains an extension of the work of [13] to include actuator limitations. A mathematical description of a simplified robot is utilized to form a set of discretized state difference equations, which are the basis for the controller design and performance calculations. The emphasis is on the effectiveness of the control strategy and on the influence of actuator dynamics. Simulation results are compared with those of perfect actuators.

2. Dynamic Model:

2.1 Equations of motion of a manipulator linkage

A manipulator is defined to be a kinematic chain of rigid links connected through joints. It is assumed that each degree of freedom of the manipulator is powered by an independent torque source. Using the Lagrange formulation, the equations of motion of an n degrees of freedom manipulator [1] can be written in matrix form as

$$\tau = \mathbf{M}(q)\ddot{q} + \mathbf{C}(q,\dot{q}) \quad (1)$$

Where **M(q**) is the n x n generalized inertia matrix (n is the number of degrees of freedom). q, $\dot{q}$ and $\ddot{q}$ are n x 1 vectors representing generalized joint variables, joint velocities and joint accelerations. **C** is a n x 1 vector representing the combined effects of Coriolis, centrifugal and gravitational forces. τ is the n x 1 generalized joint force vector.

2.2 Equations of motion of manipulator actuators

The actuators are taken to be separately excited DC motors as they are the most common actuators for robots. A separately excited DC motor is schematically drawn in Fig. 1. The control is applied at the armature terminals in the form of the control voltage $e_a(i)$. It is assumed that magnetic field voltage ef(t) is applied sufficiently long so that the field current $i_f(t)$ is constant, I_f. The further assumptions are: (1) The air gap flux, $\mathbf{\Phi_f}(t)$. is proportional to $i_f(t)$, that is $\mathbf{\Phi_f}(t) = K_f i_f(t) = K_f I_f$ = constant; (2) The torque developed by the motor. $\tau_m(t)$ is proportional to $\mathbf{\Phi_f}$ and armature current, $i_a(t)$, i. e. $\tau_m = K_m \mathbf{\Phi_f}\, i_a(t) = K_i\, i_a(t)$, where $K_i = K_m K_f I_f$ is the torque constant in Nm/amp. For a single motor, the following set of equations results [14]:

$$L_a \frac{di_a}{dt} = e_a - R_a i_a - K_b \dot{\theta}_m \quad (2)$$

$$J_m \ddot{\theta}_m = K_i i_a - \tau_L - B_m \dot{\theta}_m \quad (3)$$

where L_a is armature inductance, R_a armature resistance, K_b back electro magnetic force constant, θ_m motor shaft angular position, J_m rotor inertia of motor, τ_L motor output torque, B_m viscous frictional coefficient. The equations for n motors are written in vector form as:

$$[\mathbf{L}_a]\frac{di_a}{dt} = e_a - [\mathbf{R}_a] i_a - [\mathbf{K}_b]\dot{\theta}_m \quad (4)$$

$$[J_m]\ddot{\theta}_m = [K_i] i_a - \tau_L - [B_m]\dot{\theta}_m \quad (5)$$

Where vectors are defined as:

$$i_a = [i_{a1}, i_{a2}, \Lambda, i_{an}]^T \quad (6)$$

$$e_a = [e_{a1}, e_{a2}, \Lambda, e_{an}]^T \quad (7)$$

$$\theta_m = [\theta_{m1}, \theta_{m2}, \Lambda, \theta_{mn}]^T \quad (8)$$

and $[\mathbf{L}_a]$, $[\mathbf{R}_a]$,$[\mathbf{K}_b]$, $[\mathbf{K}_i]$, $[\mathbf{J}_m]$, $[\mathbf{B}_m]$ are all n x n diagonal matrices with the i^{th} non-zero element corresponding to the relevant parameter value for the i^{th} motor.

2.3 Equations of motion of the complete robot manipulator system

The dynamic model of a whole robot system can be completed by combining the equations of motion of motors, eq. (4), and eq. (5), with those for a robot linkage, eq. (1). Substituting the variables q, $\dot{q}$ and $\ddot{q}$ and τ in eq. (1) by $\left[\frac{1}{K_g}\right]\theta_m, \left[\frac{1}{K_g}\right]\dot{\theta}_m, \left[\frac{1}{K_g}\right]\ddot{\theta}_m$, and $[K_g]\tau_L$ the equations transformed into motor shaft space and can be written:

$$e_a = [L_a]\frac{di_a}{dt} + [R_a]i_a + [K_b]\dot{\theta}_m \quad (9)$$

$$M'(\theta_m)\ddot{\theta}_m + C'(\theta_m, \dot{\theta}) - [K_i]i_a = 0 \quad (10)$$

where $[K_g]$, $\left[\frac{1}{K_g}\right]$ are n x n diagonal matrices with the value of the i^{th} joint gear ratio and its inverse as the i^{th} non-zero elements respectively, and

$$M'(\theta_m) = [J_m] + \left[\frac{1}{K_g}\right] M(q) \left[\frac{1}{K_g}\right] \quad (11)$$

$$C'(\theta_m, \dot{\theta}) = \left[\frac{1}{K_g}\right] C(q, \dot{q}) + [B_m]\dot{\theta}_m \quad (12)$$

The effects of the armature inductances are neglected, as is common, based on the fact that the inductance has a very small influence at the operating frequencies of concern. After neglecting the inductances and eliminating the variable $\mathbf{i}_a$, the two equations eq.(9) and eq.(l0) are merged into one as follows:

$$e_a = M''(\theta_m)\ddot{\theta}_m + C''(\theta_m, \dot{\theta}) \quad (13)$$

$$M''(\theta_m) = [R_a][K_i]^{-1} M'(\theta_m) \quad (14)$$

$$C''(\theta_m, \dot{\theta}_m) = [R_a][K_i]^{-1} C'(\theta, \dot{\theta}_m) + [K_b]\dot{\theta}_m \quad (15)$$

and all the following study is based on this set of equations.

2.4 Piecewise linearization and discretization of the equations:

The equations of motion are firstly piecewise linearized about a nominal trajectory and then the linearized equations are transformed into discrete time versions for the application of digital control. Assuming $(\theta_{m0}, \dot{\theta}_{m0}, \ddot{\theta}_{m0})$ is the nominal point in motor shaft space corresponding to a point $(q_0, \dot{q}_0, \ddot{q}_0)$ on a nominal trajectory which satisfies equations (13), the equations of motion around this point are linearized by using Taylor series expansions and neglecting all the terms with order higher than one. Then, defining a state vector (2n x 1)

$$X = [X_1 \quad X_2]^T \quad (16)$$

where

$$\mathbf{X}_1 = \theta_m = [\theta_{m1}, \theta_{m2}, \Lambda, \theta_{mn}]^T, \qquad \mathbf{X}_2 = \dot{\theta}_m = [\dot{\theta}_{m1}, \dot{\theta}_{m2}, \Lambda, \dot{\theta}_{mn}]^T \quad (17)$$

the linearized equations are expressed in state space form:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u} \quad (18)$$

the coefficient matrices **A** (2n x 2n) and **B** (2n x n) are updated step by step with the updating of the nominal point to accommodate the variation of robot dynamic properties. The details for the evaluation of the matrices **A** and **B** and the definition of the control **u** (n x 1) can be found in [14].
The linear equations are discretized for one step length, T, yielding the following state difference equations [15]:

$$\mathbf{X}(k+1) = \Phi(k)\mathbf{X}(k) + \Gamma(k)\mathbf{u}(k) \tag{19}$$

where **X**(k) and **X**(k+l) are 2n x 1 state vectors at time instants (k-l)T and kT and **u**(k) is a n x 1 control vector which is constant within one period between (k-l)T and kT. The 2n x 2n state transition matrix **Φ** (k) and the 2n x n control gain matrix **Γ**(k) are constant within each time step but are updated for successive steps. The mathematical basis is provided by eq. (19) not only for digital computer control but also for simulations.

3. Control Strategy:

Robot manipulators are often required to follow desired trajectories which are specified by continuous time functions. Given a trajectory with reference to time, i.e. $\theta = \theta(t)$, $\dot{\theta} = \dot{\theta}(t)$, $\ddot{\theta} = \ddot{\theta}(t)$, in joint space it is the aim to control a robot manipulator to follow that trajectory as closely as possible. Here, it is assumed that over the closed interval [0,t_f] the first derivative $\dot{\theta}$ is continuous and the second derivative $\ddot{\theta}$ is bounded. In general, it is not possible to drive a manipulator to follow a trajectory exactly because of the complexity of the dynamics and the existence of modeling errors. In this particular case, the problem is discretized such that a sequence of points on the trajectory is to be tracked. In fact, because the control **u**(k) is constant within a sampling period, it is generally not possible to make the manipulator move from the current state to exactly the next desired state. Therefore, the problem cam be stated as: to find the discrete time control **u**(k) so that the trajectory can be followed as closely as possible. The control and computational schemes are briefly described as follows.

a) Feed forward and Feedback Control

Feed forward control compensates the coupled forces among all the various joints. With this control strategy, the feedforward control law is eq. (20). It provides gross control efforts (voltages) for the nominal model and the nominal trajectory. The control law is obtained by using a step-by-step optimization technique with respect to minimizing the objective function eq. (22).

$$\mathbf{u}^*(k) = \mathbf{K}(k)[\Phi(k)\mathbf{X}(k) - \mathbf{X}_d(k+1)] \tag{20}$$

where

$$\mathbf{K}(k) = -[\Gamma^{\mathrm{T}}(k)\mathbf{Q}_1\Gamma(k) + \mathbf{Q}_2]^{-1}\Gamma^{\mathrm{T}}(k)\mathbf{Q}_1 \tag{21}$$

is the open-loop optimal control gain matrix (n x 2n).

$$\mathbf{J} = [\mathbf{X}(k+1) - \mathbf{X}_d(k+1)]^{\mathrm{T}}\mathbf{Q}_1[\mathbf{X}(k+1) - \mathbf{X}_d(k+1)] + [\mathbf{u}(k)]^{\mathrm{T}}\mathbf{Q}_2\mathbf{u}(k) \tag{22}$$

where Q_1, and Q_2 are 2n x 2n and n x n positive definite weighting matrices respectively. A feedback control scheme based on the on-line measurement of state information is used to correct state deviation caused by any sources of modeling errors such as parameter uncertainty, external disturbances and linearization of the non-linear system. Given error state equations

$$\mathbf{X}_e(k+1) = \Phi(k)\mathbf{X}_e(k) + \Gamma(k)\mathbf{u}_e(k) \tag{23}$$

the control variation $u_e(k)$ is to be found in such a way that $X_e(k+l)$ is as near to zero as possible, where $X_e(k)$ is the 2n x 1 state error vector at instant k, $u_e(k)$ is the small variation on the control vector u(k), and $X_e(k+l)$ is the terminal value of the state error $X_e(k)$ at instant k+1. Using a similar derivation procedure as for feedforward control, the feedback control law is given by

$$\mathbf{u}_e^*(k) = \mathbf{K}_e(k)\mathbf{X}_e(k) \tag{24}$$

where the optimal feedback control gain matrix is

$$\mathbf{K}_e(k) = -[\Gamma^{\mathrm{T}}(k)\mathbf{Q}_1\Gamma(k) + \mathbf{Q}_2]^{-1}\Gamma(k)\mathbf{Q}_1\Phi(k) \tag{25}$$

The optimal control gain matrices K(k) and $K_e(k)$ are constant in each sampling period but vary from period to period.

b) State-And-Control Prediction Computation Strategy

To solve the on-line computational problem for facilitating real time control, the State-And-Control Prediction Strategy [13-14] is used to account for all the time needed for on-line computation and A/D, D/A conversions Fig. 2. The effect of time delay associated with the on-line computation, which is very significant at high operational speed, is, therefore, eliminated. The principle of this method is to predict the terminal state of each current control step, for one step ahead prediction, for example, and then the predicted state is used to evaluate the control required for the next step, instead of computing the control when the robot has actually arrived. The control for the $(k+1)^{th}$ step is predicted according to the formula below [13-14].

$$\mathbf{u}_{ep}^*(k+1) = \mathbf{K}_e(k+1)[\mathbf{X}_p(k+1) - \mathbf{X}_d(k+1)] \tag{26}$$

if the one step ahead prediction is required, where $X_p(k+l)$ is the predicted state at the end of k^{th} step, i. e. the starting state vector of the $(k+l)^{th}$ step.

4. Simulations and Discussion:

The FORTRAN computer program used in [13-14] has been developed for simulations using FORTRAN for either of the two cases, model with ideal actuators and model with real actuators. The program is general for a robot with any joints and any number of degrees of freedom providing that the inertia matrix M and the term C (eq. 13) and their partial derivatives can be obtained in algebraic form and are supplied by subroutines. Here, a planar robot manipulator with two degrees of freedom [13-14] is used as an example. In all the simulations a desired trajectory is specified in the joint space and tracking performance indices are MSQERR1 and MSQERR2. They are defined to be

$$\text{MSQERR1} = \frac{1}{N_t + 1} \sum_{K=1}^{N_t+1} \mathbf{X}_{1e}^{T}(k)\mathbf{X}_{1e}(k) \tag{27}$$

the sum of mean squared joint position errors over all the joints and

$$\text{MSQERR1} = \frac{1}{N_t + 1} \sum_{K=1}^{N_t+1} \mathbf{X}_{2e}^{T}(k)\mathbf{X}_{2e}(k) \tag{28}$$

the sum of mean squared joint velocity errors over all the joints, where $X_{1e}(k)$, $X_{2e}(k)$ are respective state errors in positions and velocities at instant k, and $N_t = t_f / T_1$. Here t_f is the final time and T_1 is the step length for integration. It is assumed that for the purpose of precise trajectory tracking, the actuators have sufficient capacity to correct the effects of modeling errors after the nominal dynamic properties have been compensated. Therefore, the elements of the weighting matrix $\mathbf{Q}_2$ are set to zero. The detailed simulations are presented and discussed below.

a) Simulations Using a simple Sinusoidal Trajectory with Payload Errors

Trajectories of the two joints are specified by a simple sinusoidal function, θ= -2.0cos(2t), $\dot{\theta}$= 4.0 sin(2t), $\ddot{\theta}$ =8.0cos(2t), to see the effect of the modeling errors in the payload on the tracking ability of the robot manipulator. Simulation results are listed in Table 1. Note that the control sampling period is denoted by T and the prediction step length by T_2. The trajectories are so well followed that tracking errors can not seen from plotting the desired and the controlled trajectories together even when the sampling period T=0.05 s. Therefore, the graphs are not presented here.

Table 1: Simulation Results with errors in Payload (Units: MSQERR1, rad.^{2}, MSQERR2, (rad./s)^2)

Payload Error	$T=T_2=0.05$s		$T=T_2=0.025$s	
	MSQERR1	MSQERR2	MSQERR1	MSQERR2
5%	0.000012	0.000447	0.000001	0.000118
10%	0.000031	0.000868	0.000004	0.00015
20%	0.000088	0.001926	0.000019	0.000382

b) Simulations Using Sinusoidal Trajectories with Different Velocity Amplitudes

The tracking ability of the robot manipulator to trajectories with different velocities is studied by varying velocity amplitudes of joint motion (10% payload error) and keeping the frequency the same; i.e. $\dot{\theta}(t)$ =A sin(2t). The results are shown in Fig. 3(a) and (b) ($T=T_2=0.05$s).

c) Simulations Using Sinusoidal Trajectories with Various Velocities

The simulation for investigating tracking ability of the robot manipulator to trajectories $\dot{\theta}(t)$ = 4.0sin(2πft) with various frequencies f are investigated. Fig. 4 shows the simulation results (10% payload error, $T=T_2=0.05$s).

The dynamics and control of a robot manipulator arm are complicated in nature mainly because of the complexity of the equations of motion. Simplifying the equations of motion, by linearization and discretization, control strategy is found to follow desired trajectories. Simulations based on the model with ideal actuators [13] showed that specified trajectories with high velocities can be tracked accurately by using the control and computational strategy proposed, even when parameter inaccuracies and disturbances exist. However, much better tracking ability is obtainable when actuator dynamics are included.

In most cases, the state errors measured by MSQERR1 and MSQERR2 are reduced greatly through the inclusion of actuator dynamics. The reductions in the errors of angular velocities are more significant than those of angular positions. This indicates the stabilizing effect of real actuators. From Fig. 3 (a) and (b), it can be seen that the errors measured in MSQERR1 and MSQERR2 increase with joint velocities. However, actuator dynamics have a very good effect on improving the robot stability for a wide range of operational speeds. At the lower range of the trajectory frequencies (for example below 4.0 rad/s. Fig. 4) the effect of actuators on the tracking performance is very satisfactory. The tracking errors are significantly reduced.

The control strategy mentioned in section 3 has been used to develop a planar manipulator (RRR) case study using MATLAB. The developed GUI, shown in Fig. 5, is prepared to assist the process of determining the coordinates of the tip of a three DOF planar manipulator (X_T, Y_T) as well as the degrees of rotation of the three actuators ($\theta 1$, $\theta 2$, $\theta 3$). The model is prepared to act in two modes the first mode, shown in Fig. 6(a), is to control the three actuator angles while the second mode, shown in Fig. 6(b), is to control the radius and the angle of rotation of the manipulator tip. The goal of the case study is to show the capability of the introduced control strategy to assist in controlling the manipulator.

5. Conclusions:

A systematic procedure for driving the closed form equations of motion of a robot manipulator is developed. Based on the piecewise linearized and discretized mathematical model, a set of linear state difference equations is provided for the application of a digital computer controller. By comparing the simulation results in this work with those for the corresponding model with ideal actuators [13,14], it is clear that although the control and computational schemes in [13,14] work perfectly well, the controller design, based on the robot model with ideal actuators is not sufficient.

The procedure for designing a robot controller should consider the actuator dynamics, otherwise instability or performance deterioration would occur in a real system, were such a controller design to be implemented. Much better tracking performance can be obtained if the controller is properly designed by taking actuator dynamics into account. DC motor actuators have the effect of low-pass filtering the control efforts and thereby stabilizing the controlled robot system. Oscillations in the tracking performance can be smoothed and the tracking errors can be reduced greatly. On the other hand, the tracking ability in trajectories with higher frequencies is degraded due to the limited bandwidth of the DC motors. This implies that actuator bandwidths should be matched to the path following demands which are to be placed on the manipulator.

6. References:

[1] R. P. Paul, Robot Manipulators: Mathematics, Programming, and Control, the MIT Press, Cambridge, Mass., 1981.
[2] Q. Zhang, A. J. Koivo, and T. H. Guo, Industrial Manipulator Control Using Kalman Filter and Adaptive Controller, Int. J. 'Robotics and Automation, Vol.2, No. 1 1987
[3] R. P. Paul, Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm, Stanford Artificial Intelligence Lab., A. I. Memo 177, Sept. 1972.
[4] A.K. Bejczy, Robot Arm Dynamics and Control, Jet Propulsion Lab., Tech. Memo, Feb. 1974
[5] E. G. Gilbert, I. I. Ha, An Approach to Nonlinear Feedback Control with Applications to Robotics, Proc. Of the IEEE Conf. on Decision and Control, San Antonio, Texas, Dec. 1983.
[6] C. Samson, Robust Nonlinear Control of Robotic Manipulators. Proc. of IEEE Conf. on Decision and Control, San Antonio. Texas, Dec. 1983.
[7] V. D. Tourassis, C. P. Neuman, Robust Nonlinear Feedback Control for Robotic Manipulators, IEE Proc., Vol. 132. No.4, July 1985.
[8] H. Asada, T. Kanade, and I. Takeyama, Control of a Direct-Drive Ann. ASME Trans., J. Dyn. Syst., Meas., and Contr., Vol.105, No.3. 1983.
[9] S. Jayasuriya, C. N. Hwang. Tracking Controllers for Robot Manipulators: A High Gain Perspective. ASME Trans., J. Dyn. Syst.. Meas., and Contr., Vol.1 10, March 1988.
[10] S. H. Wang., I. Horowitz, CREATE - A New Adaptive Technique. Proc. of Conf. on Information Sciences and Systems, Johns Hopkines Univ., March 1985,
[11] C. S. G. Lee, M. J. Cluing, and B. H. Lee. An Approach of Adaptive Control for Robotic Manipulators, J. Robotic Systems. Vol.1 Part 1, John Wiley and sons. Inc.. 1984.
[12] C. W. DeSilva, J. Van Winssen, Least Squares Adaptive Control for Trajectory Following Robots, ASME Trans., J. Dyn. Syst., Meas., and Contr., Vol 109, June 1987.
[13] H. M. Mahgoub, I. A. Craighead, Discrete-Time Control of Fast Non-linear Motion of 3DOF Robot Manipulator, 7th Conf. on Applied Mechanics and Mechanical Engineering, May 1996.
[14] H. M. Mahgoub, I. A. Craighead, Robot Actuation Using Air Motors, Int. J. of advanced Manufacturing Technology, Vol.11, 1996.
[15] G. F. Franklin, J. D. powell, Digital Control of Dynamic Systems, Wesley Publishing Company, 1980.

Figures

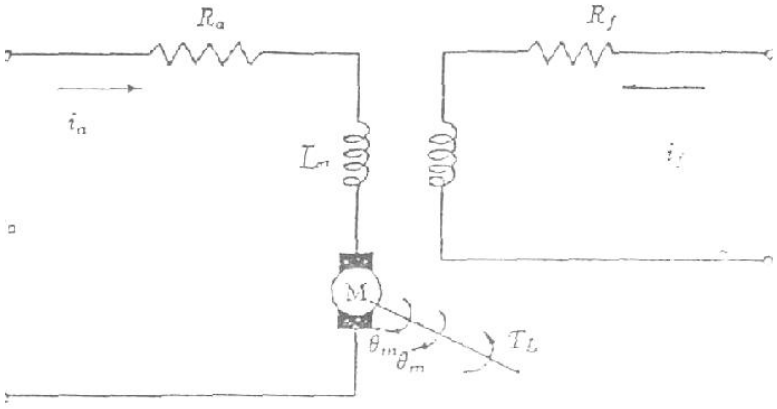

Fig. 1. Schematic diagram of a DC motor

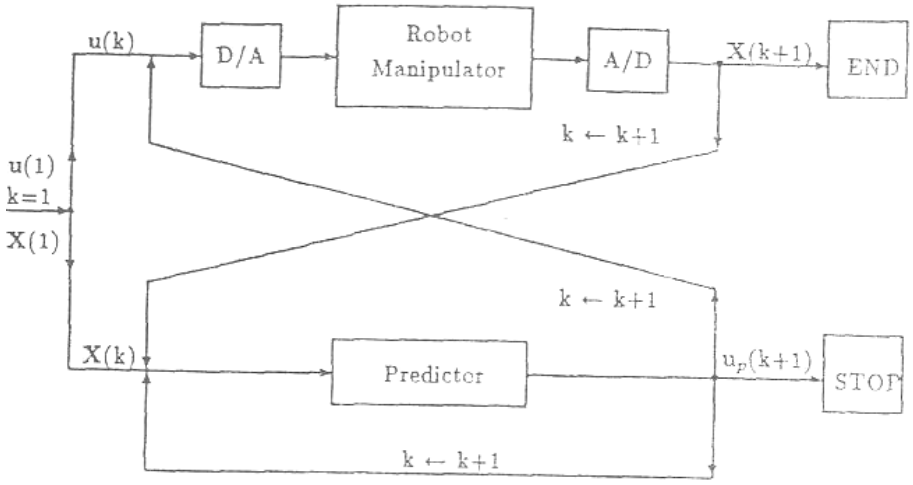

Fig. 2. State and control prediction diagram

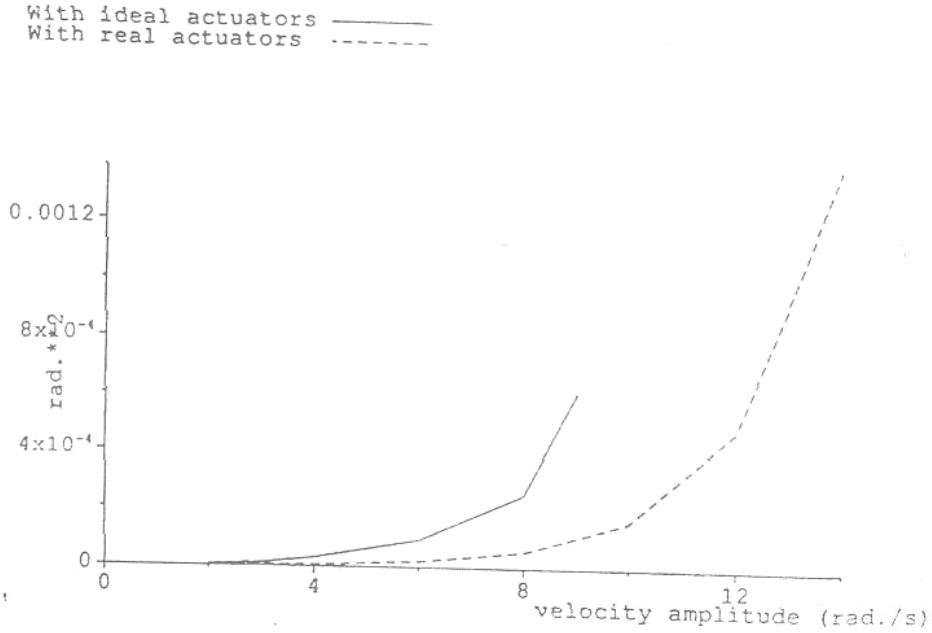

Fig. 3.a MSQERR1 vs velocity amplitude (with ideal and real actuator)

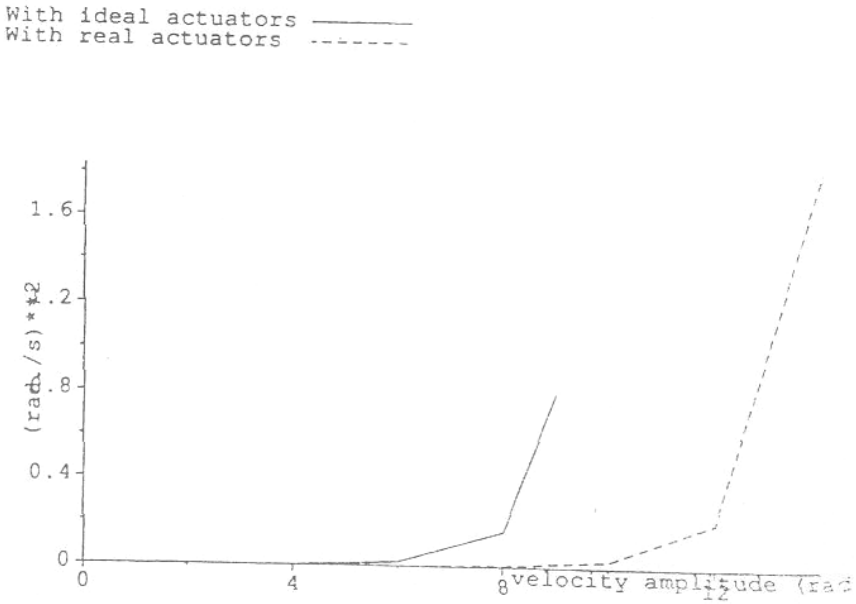

Fig. 3.b MSQERR1 vs velocity amplitude (with ideal and real actuator)

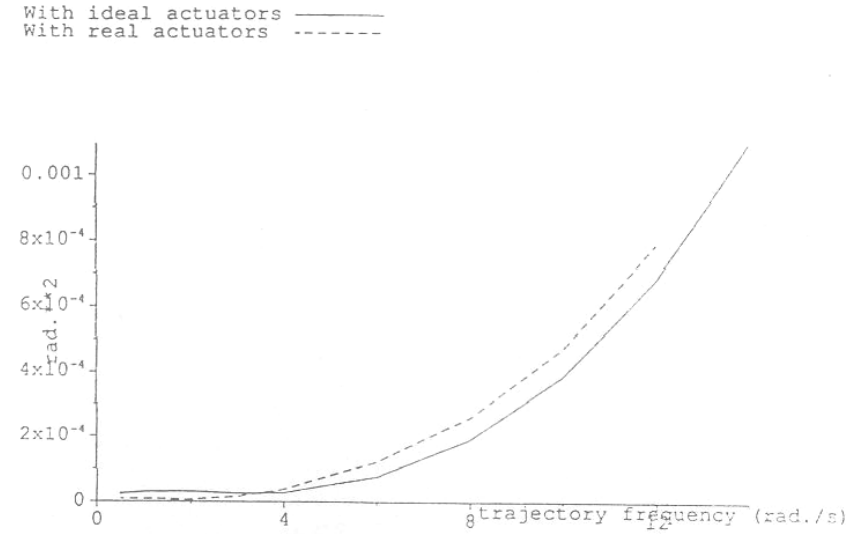

Fig. 4.a MSQERR1 vs trajectory frequency (with ideal and real actuator)

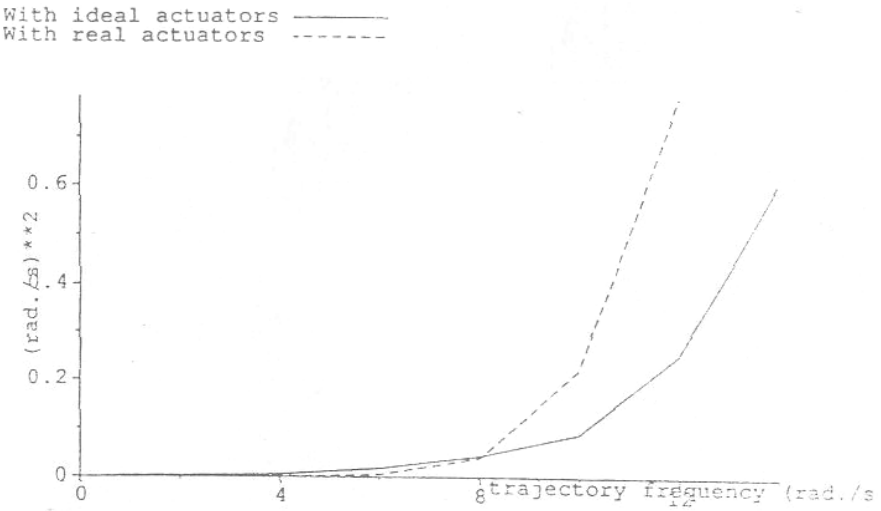

Fig. 4.b MSQERR1 vs trajectory frequency (with ideal and real actuator)

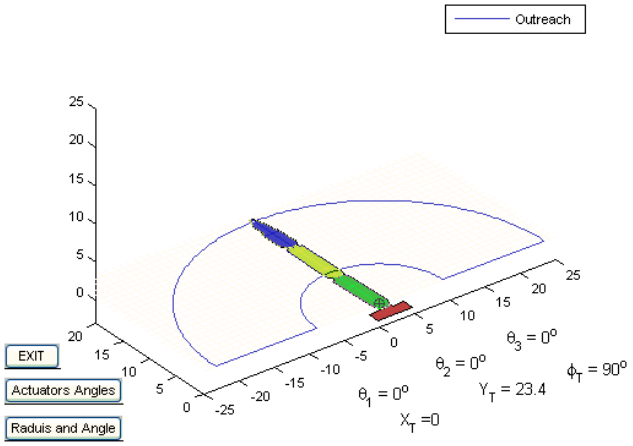

Fig. 5. Planar manipulator developed GUI

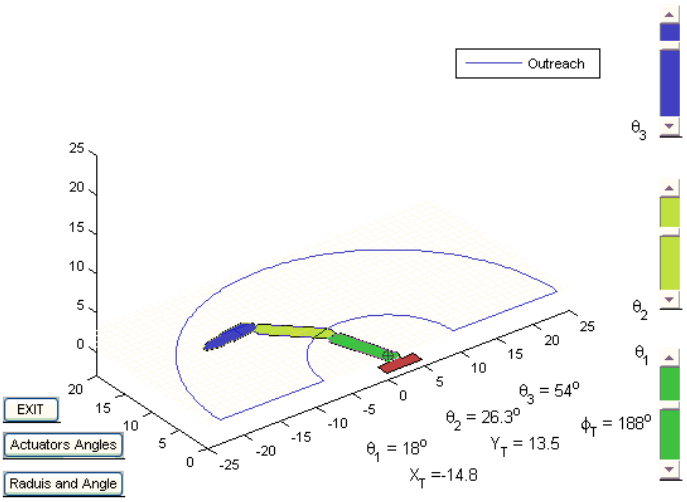

Fig. 6.a Planar manipulator GUI first mode (control the three actuator angles)

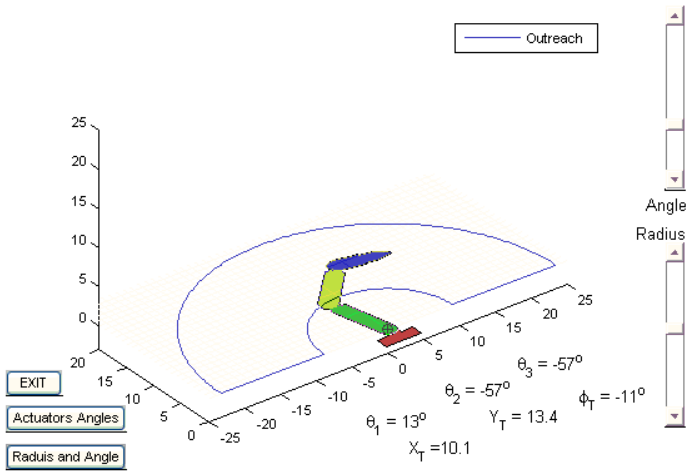

Fig. 6.b Planar manipulator GUI second mode
(control the radius and the angle of rotation of the manipulator tip)

NOMENCLATURES

A	2n x 2n matrix
B	2n x n matrix
B_m	viscous factional coefficient of motor rotor
C	that is C(q,q), n x 1 vector representing the combined effects of Coriolis, centrifugal and gravitational forces
C''	modified C, including actuator dynamics
e_a or e_{ai}	control voltage for each motor
e_a	control voltage vector
e_f	magnetic field voltage
i_a	armature current
i_f or I_f	field current
J	quadratic objective unction
J_m	rotor inertia of motor
K,K_e	optimal control gain matrices
K_b	back electro magnetic force constant
K_q	joint i gear ratio
K_i	torque constant of a motor
L_a	armature inductance
M	that is M(q), n x n generalized inertia matrix
M''	modified M, including actuator dynamics
MSQERR1	the sum of mean square joint position errors over all the joints
MSQERR2	the sum of mean square joint velocity errors over all the joints
n	number of degrees of freedom or number of joints
N_f	$N_f = t_f / T$, number of total control steps along a trajectory.
N_p	number of integration steps taken for prediction in one control period
N_t	$N_t = t_f / T_I$, total steps taken for numerical integration along a trajectory
q, q, q	n x 1 vectors of generalized joint displacements, velocities and act respectively
q_o, q_o, q_o	a nominal point on a specified trajectory
Q_1	2n x 2n positive definite weighting matrix
Q_2	n x n positive definite weighting matrix
R_a	armature resistance
T, T_1, T_2	control sampling period, step length for integration and step length for prediction respectively
u, u_e	n x 1 control vectors
u^*, u_e^*	optimal values of u and u_e respectively
u_p^*, u_{ep}^*	predicted optimal values of u and u_e respectively
X, X^*	2n x 1 state vector and its derivative
X_1, X_2	two n x 1 vectors representing the two parts of X, corresponding to the generalized joint displacements and velocities respectively
X_d	2n x 1 desired state vector
X_e	2n x 1 state error vector
X_{1e}, X_{2e}	error vectors of Xi and XT
X_p	predicted state vector
X_T	X coordinate of the manipulator tip in the case study
Y_T	Y coordinate of the manipulator tip in the case study
Γ	2n x n control gain matrix
θ_m, θ_{mi}	motor angular displacement
θ_m	motor angular displacement vector
θ_{m0}, θ_{mo} θ_{mo}	nominal point on a specified trajectory measured in motor shaft space

$\theta 1$	angle of rotation of actuator (1) in the case study
$\theta 2$	angle of rotation of actuator (2) in the case study
$\theta 3$	angle of rotation of actuator (3) in the case study
Φ	2n x 2n transition matrix
Φ_f	air-gap flux
τ	n x 1 generalized force vector
τ_L	motor output torque
τ_L	motor output torque vector
τ_m	torque developed by a motor
ψ	2n x 2n matrix
[.]T	transpose of [.]