

1994021779

N94- 26282

**Direct Model Reference Adaptive
Control of Robotic Arms**

442521

**Howard Kaufman, David C. Swift,
Steven T. Cummings, and Jeffrey R. Shankey
Rensselaer Polytechnic Institute
Troy, New York**

Direct Model Reference Adaptive Control of Robotic Arms*

Howard Kaufman, David C. Swift, Steven T. Cummings and Jeffrey R. Sankey
Electrical, Computer and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, NY 12180-3590
Tel: 518-276-6081
Fax: 518-276-6261
email: kaufman@ecse.rpi.edu

Abstract

This paper presents the results of controlling A PUMA 560 Robotic Manipulator and the NASA shuttle Remote Manipulator System (RMS) using a Command Generator Tracker (CGT) based Model Reference Adaptive Controller (DMRAC). Initially, the DMRAC algorithm was run in simulation using a detailed dynamic model of the PUMA 560. The algorithm was tuned on the simulation and then used to control the manipulator using minimum jerk trajectories as the desired reference inputs. The ability to track a trajectory in the presence of load changes was also investigated in the simulation. Satisfactory performance was achieved in both simulation and on the actual robot. The obtained responses showed that the algorithm was robust in the presence of sudden load changes. Because these results indicate that the DMRAC algorithm can indeed be successfully applied to the control of robotic manipulators, additional testing was performed to validate the applicability of DMRAC to simulated dynamics of the shuttle RMS.

1 Introduction

Direct adaptive control offers the potential for uniform control of robotic manipulators in the presence of uncertain flexibilities, changing dynamics due to unknown and varying payloads, and nonlinear joint interactions without explicit parameter identification.

One such direct adaptive algorithm that is especially attractive for robotic control is the direct model reference adaptive controller (DMRAC) discussed in [1-3]. This adaptive algorithm is very appealing for robotic control because of the following features:

- asymptotically zero output error with all states bounded,
- lack of dependence on plant parameter estimates,
- direct applicability to multiple input-multiple output plants,

*This paper is based upon research performed under NASA grant NAGW-1333 and under NSF grant ECS-9111565.

- sufficiency conditions which are independent of plant dimension,
- control calculation which does not require adaptive observers or the need for full state feedback,
- ease of implementation, and
- successful experimental validation.

This procedure has been previously used to control a single link flexible robotic joint and a nonlinear model of a two link Puma [4,5]. In view of the excellent tracking results demonstrated in these papers, it was concluded that this adaptive algorithm should be used to control an actual Puma arm. This effort has consisted of two main thrusts: namely, control of the representative simulation model developed in [6], and the transition of the tuned algorithm to the actual robotic arm.

Because results indicated that the performance of the DMRAC algorithm was robust with respect to representative load variations, additional applicational studies were initiated using the NASA shuttle Remote Manipulator System (RMS).

2 Direct MRAC Development

2.1 Basic algorithm

The linear time invariant model reference adaptive control problem is considered for a plant which is described by

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) \quad (1)$$

$$y_p(t) = C_p x_p(t) \quad (2)$$

where $x_p(t)$ is the $(n \times 1)$ plant state vector, $u_p(t)$ is the $(m \times 1)$ control vector, $y_p(t)$ is the $(q \times 1)$ plant output vector, and A_p, B_p are matrices with appropriate dimensions.

The objective is to find, without explicit knowledge of A_p , and B_p , the control $u_p(t)$ such that the plant output vector $y_p(t)$ approximates "reasonably well" the output of the following (and usually lower order) reference model:

$$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t) \quad (3)$$

$$y_m(t) = C_m x_m(t) \quad (4)$$

The MRAC algorithm is given by [1]:

$$u_p(t) = K_e(t)[y_m(t) - y_p(t)] + K_z(t)x_m(t) + K_u(t)u_m(t) \quad (5)$$

with the gains $K_e(t)$, $K_z(t)$, and $K_u(t)$ being adaptive. The adaptive gains are concatenated into the matrix $K_r(t)$ which is given by

$$K_r(t) = [K_e(t), K_z(t), K_u(t)] \quad (6)$$

and the vector $r(t)$ is defined by

$$r(t) = [y_m(t) - y_p(t), x_m(t), u_m(t)]^T \quad (7)$$

Then

$$u_p(t) = K_r(t)r(t) \quad (8)$$

The adaptive gains are obtained as a combination of an integral gain and a proportional gain as shown below:

$$K_r(t) = K_p(t) + K_I(t) \quad (9)$$

$$K_p(t) = [y_m(t) - y_p(t)]r^T(t)\bar{T} \quad (10)$$

$$\dot{K}_I(t) = [y_m(t) - y_p(t)]r^T(t)T \quad (11)$$

Sufficient conditions for stability derived for a constant model command in [2]. These conditions require that the matrices T and \bar{T} be respectively chosen as positive definite and positive semidefinite, and that the plant be almost strictly positive real (ASPR), that is, for the plant represented by the triple (A_p, B_p, C_p) there exists a matrix K_e (not needed for implementation) such that the fictitious stabilized plant described by the triple $(A_p - B_p K_e C_p, B_p, C_p)$ is positive real. If these sufficient conditions hold, then all states and gains are bounded and the output error vanishes asymptotically.

The adaptive control of plants that are not ASPR is a more difficult problem when utilizing the CGT based MRAC laws. BarKana [3] suggested augmenting the plant with parallel dynamics such that the augmented plant is ASPR in which case the previously described adaptive controller may be utilized. To illustrate this concept, consider the non-ASPR plant described by the transfer matrix

$$G_p(s) = C_p(sI - A_p)^{-1}B_p \quad (12)$$

Then, choose a matrix $H(s)$ such that the augmented plant transfer matrix

$$G_a(s) = H^{-1}(s) + G_p(s) \quad (13)$$

is ASPR. In [3] it is shown that $G_a(s)$ will be ASPR provided that

- $H(s)$ itself is ASPR
- $H(s)$ stabilizes the closed loop output feedback system with transfer function.
 $[I + G_p(s)H(s)]^{-1}G_p(s)$.

An easily implementable version of $H(s)$ which has had extensive use is

$$H(s) = K(1 + s/s_0)$$

resulting in:

$$G_a(s) = \frac{D}{1 + s/s_0} + G_p(s) \quad (14)$$

where $D = K^{-1}$.

Unfortunately, the error which is ensured to be stable is not the true difference between the plant and the model. Rather, the error is now the difference between the outputs of the augmented plant and the model. However, in [3] it is shown that if the plant is output stabilizable via high gain output feedback, then $\|D\|$ may be chosen to be small. Thus, the augmented plant error will be approximately equal to the original plant error.

One procedure for eliminating this output error is to also incorporate the supplementary feedforward of (14) into the reference model output as shown in [2]. To illustrate this, denote the augmented plant output as z_p where

$$z_p = y_p + Hu_p \quad (15)$$

$$= y_p + H[K_z x_m + K_u u_m + K_e e_y] \quad (16)$$

and

$$H = D/(1 + sr). \quad (17)$$

In a similar manner, define an augmented model output as

$$z_m = y_m + H[K_z x_m + K_u u_m] \quad (18)$$

Now, for adaptive control of the augmented plant, consider the error between the augmented plant and model outputs. Thus,

$$\begin{aligned} (z_m - z_p) &= y_m - y_p - HK_e e_x \\ &= y_m - y_p - HK_e(z_m - z_p) \\ \text{or } z_m - z_p &= (I + HK_e)^{-1}(y_m - y_p) \end{aligned} \quad (19)$$

Consequently if as in [2] z_p is forced to follow z_m , then y_p will follow y_m .

3 Puma Model Development

In order to test the performance of the Direct Model Reference Adaptive Controller (DM-RAC), an accurate non-linear coupled model of the PUMA manipulator was needed. A full explicit dynamic model of the PUMA 560 manipulator, derived by Armstrong, Khatib, and Burdick [6], was selected. The formulation of the PUMA model was computationally efficient using 25% fewer calculations than a 6 degree of freedom recursive Newton-Euler method. The algebraic formulation of the model also allowed for the easy addition of a load by modifying the link 6 mass, center of mass, and inertia parameters.

Figure 1 shows the six rotational joint axis, $\{z_1, \dots, z_6\}$, for the PUMA 560. Only the rotational, z_1 , axis are shown in the figure. Positive rotations follow the right hand rule - counter-clockwise looking down the z axis. The six joint of the PUMA 560 are as follows:

- **Joint 1.** A vertical rotation about the base, z_1 .

- *Joint 2.* A horizontal rotation about the shoulder, z_2 .
- *Joint 3.* A horizontal rotation about the elbow, z_3 .
- *Joint 4.* A twist of the wrist, z_4 .
- *Joint 5.* A inclination of the wrist, z_5 .
- *Joint 6.* A twist of the mounting flange, z_6 .

The position of the manipulator in Figure 1 illustrates the zero position. Note that when Joint 5 is at zero, axis z_4 and z_6 coincide.

The dynamic equations of motion used to model the PUMA are:

$$A(\theta)\ddot{\theta} + B(\theta)[\dot{\theta}\dot{\theta}] + C(\theta)[\dot{\theta}^2] + g(\theta) = \Gamma \quad (20)$$

where

$A(\theta)$ is the 6×6 positive definite kinetic energy matrix,

$B(\theta)$ is the 6×15 matrix of coriolis torques,

$C(\theta)$ is the 6×6 matrix of centrifugal torques,

$g(\theta)$ is the 6 vector of gravity torques,

$\ddot{\theta}$ is the 6 vector of joint accelerations,

$[\dot{\theta}\dot{\theta}]$ is the 15 vector of velocity products, where

$$[\dot{\theta}\dot{\theta}] = [\dot{\theta}_1\dot{\theta}_2, \dot{\theta}_1\dot{\theta}_3, \dots, \dot{\theta}_1\dot{\theta}_6, \dot{\theta}_2\dot{\theta}_3, \dots, \dot{\theta}_4\dot{\theta}_6, \dot{\theta}_5\dot{\theta}_6]^T$$

$[\dot{\theta}^2]$ is the 6 vector of squared velocities, where

$$[\dot{\theta}^2] = [\dot{\theta}_1^2, \dot{\theta}_2^2, \dots, \dot{\theta}_6^2]^T$$

and Γ is the 6 vector of joint torques.

The above model can be cast into state space form by solving Equation (20) for $\ddot{\theta}$ and choosing the following 12×1 state vector,

$$x^T = [\theta^T, v^T] \quad (21)$$

where

$$\begin{aligned} \theta &= [\theta_1, \dots, \theta_6]^T \\ v &= [\dot{\theta}_1, \dots, \dot{\theta}_6]^T \end{aligned}$$

Thus, the state space model is as follows,

$$\begin{aligned}\dot{\theta} &= v \\ \dot{v} &= A^{-1}(\theta)[\Gamma - B(\theta)[\dot{\theta}\dot{\theta}] - C(\theta)[\dot{\theta}^2] - g(\theta)]\end{aligned}$$

The controlled output vector for the plant was

$$y_{plant} = \theta + \alpha v \quad (22)$$

where, α is a diagonal 6×6 matrix of velocity weighting factors.

This velocity term is present to help remove high frequency oscillations caused by the controller. The maximum allowable torques (in n-m) were [97.6, 186.4, 89.4, 24.2, 20.1, 21.3]

4 Implementation Issues

4.1 Reference Model

The first decision to be made in implementing the DMRAC algorithm is the choice of reference model order. If one chooses the order too low, then excessively large gains may occur even in a well-tuned controller. This may produce greater than desired accelerations in the robot arm joints resulting in joint torque saturations leading to poor model following. If one chooses the order too high then excessive response delays may be incurred. For the PUMA 560, an independent second order reference model was selected for each of the six joints. This is not unreasonable since in a PUMA 560, as with many manipulators, the mass matrix is approximately diagonal for all θ making the system nearly decoupled.

Thus, for each joint, the reference model transfer function was:

$$y_{m_i}(s)/u_{m_i}(s) = \frac{w_{n_i}^2}{s^2 + 2\zeta_i w_{n_i} s + w_{n_i}^2}$$

where

$$w_{n_i} = 5$$

and

$$\zeta_i = 1 \quad i = \{1, 2, 3, 4, 5, 6\}.$$

Critical damping was selected so as to reduce the possibility of joint angle overshoots. This conforms to a standard safety feature of robot arm controllers which tends to avoid obstacle collisions. Of course, once the choice of critical damping is made, the choice of natural frequency governs the speed of model response to inputs. A choice of $w_n = 5$ yields a 90% rise time of about 0.8 sec.

4.2 Command Generation

For Testing purposes a minimum jerk trajectory was generated through the following positions at the noted times.

Joint Position (deg)						Time (sec)
1	2	3	4	5	6	
0	-45	180	0	45	90	0
90	-90	90	45	0	45	6
0	0	180	0	90	90	13
0	-45	180	0	45	90	18

The resulting angular paths for each joint were then used as the reference model commands $u_{m,i}(t)$.

4.3 Bias Introduction

For the PUMA 560 manipulator, the origin of the coordinate system should be such that the adaptation gains have a non-zero excitation throughout the range of interest. For example, assume that in order to maintain an output of $y_p = [0, \dots, 0]^T$, a non-zero input, u_p , is required. However a zero command to the reference model, $u_m = [0, \dots, 0]^T$, will result in a zero model output and a zero state vector. Thus in this case $e_y = y_m - y_p$ will also be zero, and the vector, $r(t)$, defined by (3.10) will be zero resulting in a zero control. Since the plant requires a non-zero control to maintain a zero output, the DMRAC algorithm requires a non-zero error signal in order to apply a non-zero control which will result in a steady-state error at the zero output position.

If the reference model coordinates are shifted by a constant bias term, then a zero command to the reference model, $u_m = [0, \dots, 0]^T$ will produce non-zero outputs for the model state and output vectors which, in turn, will produce a non-zero command to the plant. This bias term is subtracted from the model command, u_m , and the plant output, y_p , as follows,

$$u_m(t) = \hat{u}_m(t) - q_{bias} \quad (23)$$

$$y_p(t) = \hat{y}_p(t) - q_{bias} \quad (24)$$

where $\hat{u}_m(t)$ is the original model command in the original coordinate system, $u_m(t)$ is the new biased model command to be applied to the model dynamics, $\hat{y}_p(t)$ is the actual plant output, $y_p(t)$ is the new biased plant output to be used to form the error signal, and q_{bias} is a constant bias term. For robotic manipulators, q_{bias} has units of radians and should be selected such that a new plant output of $y_p = [0, \dots, 0]$ correspond to an equilibrium position. By examining the zero position of the robot, Figure 1, it is clear that $y_p = \{0, 0, 0, 0, 0, 0\}$ is not an equilibrium. However bias of,

$$q_{bias} = \{0, 90, 90, 0, 0, 0\} \text{ degrees} \quad (25)$$

will shift the zero position to that shown in Figure 2.

4.4 Feedforward Design

The feed-forward filter dynamics for joint i is given as,

$$D_i(s) = \frac{K_{d_i}}{1 + \tau_i s} \quad (26)$$

where K_{d_i} is the DC gain and τ_i is the time constant.

5 PUMA Simulation Results

In this section, we briefly discuss the tuning process and present plots of a simultaneous, six joint response of the PUMA 560 under DMRAC control.

5.1 Tuning

Once the reference model has been chosen, one must choose values for the various DMRAC parameters. Specifically these are

\bar{T} \equiv proportional gain weighting matrix, eq. (10)

T \equiv integral gain weighting matrix, eq. (11)

D \equiv plant/model feedforward gain, eq. (17)

τ \equiv plant/model feedforward time constant, eq. (17)

$\underline{\alpha}$ \equiv 6 vector of plant rate feedforward gains, eq. (22)

For the fully centralized DMRAC algorithm with the plant derivative output term and the supplementary feed-forward in the reference model and plant, there are 1182 parameters to be selected as shown in Table 5. At first, this number seems very intimidating, but as will be shown, the number of tuning parameters can be greatly reduced by some simplifications and by adjusting the parameters in groups rather than individually.

The most drastic reduction in the number of tuning parameters can be achieved by forcing the integral and proportional adaptation weighting matrices, T and \bar{T} to be diagonal. This reduces the number of tuning parameters from 1182 to 78.

The reference model dynamics have 12 tuning parameters, six w_m 's and six ζ_i 's.

It is customary in robotic applications to tune controllers for critical damping so that there is no overshoot. Overshoot may cause a robot end effector to penetrate the surface of the work environment. The undamped natural frequency terms, w_{n_i} are chosen such that the reference model will have a specified step response. Typically, the reference model dynamics are chosen such that they are "reasonable" for the plant to follow since the DMRAC algorithm will try to force the plant to follow the model output. For the case

of a PUMA 560 Manipulator, all of the w_n , were initially set to 5.0. The model's dynamic parameters can be changed as needed if the robot is having problems tracking the model.

Initially, the plant output derivative weights, α , are set to zero. These weights are used to remove high frequency components from the plant control signal, u_m , and should only be used when needed as they will affect the transient response.

The feed-forward filter has 12 tuning parameters, six gains K_d , and six time constants τ_i . A good first choice for the τ_i is approximately one-tenth the model time constant.

Initially the value of τ_i were all set to 0.1 s, and the six DC filter gains K_d , were set to 1.0. Increasing the filter gain was seen to typically improve the tracking performance.

The diagonal components of \bar{T} and T were initially, all set to unity. A reasonable method of tuning a DMRAC controller is to start the plant at an equilibrium position and apply small step inputs. After a reasonable performance is achieved with the step inputs, the DMRAC should be fine tuned using typical plant trajectories.

If the closed loop system is very sensitive to initial conditions, start with small steps as described above, let the system reach steady-state, and then save all of the DMRAC controller state information (integral adaptation matrix, K_I ; reference model state vector, x_m ; and the filter state vector) to be used as initial conditions for the next run. This will significantly cut down the adaptation time required for the gains to reach their steady-state values.

In order to compare the tuning results, some criterion must be established. For this excessive, the goal was to kept the peak model following errors small and to keep the error trajectory as close to zero as possible. Small errors were tolerable during motion. It was also desired to achieve zero error in steady-state.

The step response with the initial tuning values was sluggish for Joints 1, 4, 5, and 6 with overshoot and oscillations. Joints 2 and 3 settled into their steady-state values quickly but with very large steady-state errors. The process used to complete the tuning was as follows:

1. Refine the tuning for a 10 degree step from the equilibrium position.
2. Using the refined parameter values, move the robot to the shutdown position of Figure 12 and save the DMRAC internal state values at that position for use as initial conditions.
3. Refine the tuning for a 10 degree step from the shutdown position using the initial conditions from Step 2.
4. Refine the tuning from typical min-jerk trajectories from the shutdown position.

The final tuning parameter values after Step 4 are shown in Table 6. The weighting matrix values for Joints 1, 2, and 3 differ from the weighting matrix values for the last three joints by a factor of about 100 which reflects the mass/inertia difference between the upper arm and the wrist. The weighting matrix values which are multiplied by the " x_m ," products are about a factor of seven lower than the values multiplying the " x_{m1} " products since the

second state variable of each decoupled reference model had a higher peak value in a transient. The Joint 1, 2, and 3 reference models have an undamped natural frequency of 4.0 rad/sec where the wrist model used 7.0 rad/sec which again reflected the inertia difference between the upper arm and the wrist. The feed-forward filter values were set to $K_d = 6.0$ and $\tau = 0.1$ for all joints. The alpha values were increased from the initial values of zero to damp out some high frequency oscillations.

5.2 Response

Initially the PUMA arm is in the [0 0 0 0 0 0] position. The final shutdown position was (0, -45, 180, 0, 45, 90) degrees as shown in Fig. 3. Simulation results of the PUMA 560 dynamics responding to the tuned DMRAC controller are displayed in Figure 4.

Note that the model following is excellent for all 6 joints. Furthermore it was observed that all joint torques were smooth and below their saturation limits. In addition, for this specific case, the use of the feedforward component did not significantly affect the response, although in other cases (eg. step response) use of the feedforward resulted in significant improvements.

6 PUMA Experimental Results

Because the simulator results of the previous section indicated that DMRAC should be useful for robot control, a set of experiments was performed on an actual PUMA 560 manipulator. The tuning process was similar to that described in the previous section. All parameters were initialized at those values from the simulation studies. Only minor variations were required. Final values are in Table 1.

Examples presented illustrate performance of the DMRAC for tracking various trajectories in the presence of static and dynamic load changes. In all cases the robot starts at the shutdown position and follows a trajectory which finishes at the shutdown position.

6.1 Three Joint Trajectory Tracking Study

The trajectory listed in Table 1 is very similar to the one used in the simulation (Section 5). The arm first moves to a straight up position, curls up, and then moves back to the safe position. The wrist joints remain locked in their shutdown positions of {0.0,45.0,90.0} degrees.

The response to the first trajectory is shown in Figure 5. The response is quite good. The effects of stiction can be seen on Joint 2 at $t = 15$ seconds in Figure 5. Figures 6-8 show the model following error and the link torques for Joints 1, 2, and 3 respectively. Figure 6b shows that the Joint 1 torque signal was quite noisy. This noise did not have a physically detectable effect on the actual arm motion. Typically one can feel or hear a noisy torque signal on the actual arm.

The stiction effect mentioned above for Joint 2 can also be seen in figure 4a at $t = 15$ sec near the 'X' at the peak error location. When stiction grabs a joint, the error ramps up as does the torque (Figure 4b).

6.2 Static Load Changes

This section describes the ability of the DMRAC algorithm to adjust to static load variations. The trajectory of Table 3 will be run with different loads in the gripper. The algorithm will first be allowed to adjust to the load, and then the trajectory will be started.

The wrist joints remained locked in their shutdown positions of {0.0, 45.0, 90.0} degrees. Five different loads were run for the trajectory - 0kg, 1kg, 2kg, 3kg, and 4kg.

Figures 9 and 10 show the response for Joints 2 and 3 respectively. The numbers on the plots are to help identify which curve represents which payload. For Joint 3, the peak errors vary from 2.4390 degrees for the no load case to 3.9972 degrees for the 4kg load case. The load changes make up only about 50% of the error. The other 50% is due to the adaptation to the changing arm dynamics. For Joint 2, the peak errors are around 0.8-1.0 degrees. As with Joint 3, the portion of the error due to the load change for Joint 2 is small compared to the no load case.

For Joint 1, the error signals did not vary by more than 0.1 degrees between the five different load cases.

6.3 Dynamic Load Changes

To illustrate the effects of dynamic load change, the trajectory of Table 4 was considered. While running the same trajectory, various loads were added to the gripper while the robot were in motion. The same loads used in the previous section were employed. The wrist joints remained locked in their shutdown positions of {0.0, 45.0, 90.0} degrees. Note: The 1kg and 4kg loads were added at about $t = 6.76$ seconds and the 2kg and 3kg loads were added at about $t = 7.34$ sec.

Figure 11 shows the model following error for Joint 2 for all loads. The numbers on the graphs indicate which peaks in the error plots match up with the various loads. This figure shows that the DMRAC algorithm has a good load disturbance rejection. The transient period only lasts about 2 seconds.

Figure 12 shows the error for Joint 3 for the various loads. Joint 3 suffers more with a load disturbance having a peak error of almost 5 degrees when the 4kg load is added. Again, the transient period is roughly 2 seconds. After the transient, good tracking performance was achieved with the additional loads.

As with the static load case, the model following errors for Joint 1 did not vary by more than 0.1 degrees.

7 Adaptive Control of the Shuttle RMS

7.1 Introduction

Because of the previous demonstrated capabilities of DMRAC, consideration was given to its application to the NASA shuttle Remote Manipulator System (RMS). This system experiences damped oscillations of the end effector after the motion control input by the shuttle operator has been removed [7]. It is desired to design a controller that will take control of the RMS, after the operator releases the motion control joystick, and increase the damping of the oscillations. Linear models have been developed for three manipulator orientations expected to be encountered during normal payload handling [7]. This section discusses work on a direct model reference adaptive controller design for the RMS based on the adaptive algorithm discussed in Section 2.

7.2 Linear Plant and Feedforward

The three linear plants are 3-input, 3-output with 6 states. The plants all have a feedforward compensator $H(s)$, since they are not ASPR. Three types of algorithm feedforward were examined in the course of this work.

static:

$$H(s) = \text{diag3}\{d_{11}, d_{22}, d_{33}\}$$

1st order:

$$H(s) = \text{diag3}\{d_{11}/(\tau_{11}s + 1)\}$$

2nd order:

$$H(s) = \text{diag3}\{d_{11}/[(\tau_{11}s + 1) * (\tau_{21}s + 1)]\}$$

The scalar feedforward provided the best results (based on work to date) and was used for all presented simulations. It was found that, for the scalar feedforward, the combined plants (plant 1,2 or 3 in parallel with feedforward) were all ASPR for:

$$0.125 < d < 1.0$$

where $H(s)$ was $\text{diag3}\{d, d, d\}$. That is, the closed loop system formed from the inverse of $H(s)$ in negative feedback with the respective plant, had all the characteristic roots in the left half plane for:

$$1.0 < d^{-1} < 8.0$$

After simulations were performed with many of the possible combinations of values within this range, it was found that $d = 0.25$ provided the best results for all three orientations.

No ASPR analysis was performed for the 1st and 2nd order feedforwards and the position 1 plant. The stability of the adaptive algorithm for the dynamic feedforwards was somewhat a function of the adaptive gains, T and \bar{T} , for given feedforward time constants and gains. Tuning for the dynamic feedforwards was difficult, and very little to

no improvement over the uncontrolled system could be achieved. The dynamic feedforward compensator might possibly provide improved results with further work on the time constants and gains.

7.3 Reference Model

Originally, a reference model were developed for each position using a LQR design based on the uncontrolled plant at the respective orientations. The model used the uncontrolled plant's B, C, and D matrices with the model A matrix formed as follows:

$$A_{m_i} = (A_{p_i} + B_{p_i} * K_i)$$

where K_i is from the LQR design for the i -th orientation.

Satisfactory control of the plants could not be achieved by the adaptive algorithm using these models. A new reference model was then developed using two dominant eigenvalues from the original LQR model for position 1. The new model has a damping ratio of ζ and a natural frequency of 1.0 r/sec so that:

$$h_m(s) = 1/(s^2 + 2\zeta s + 1)$$

This new model was utilized as the reference for each plant output, that is:

$$\begin{aligned} H_m(s) &= \text{diag3}\{h_m(s), h_m(s), h_m(s)\} \\ Y_m(s) &= H_m(s) * u_m(s) \end{aligned}$$

7.4 Simulation Sequence

The sequence for simulation represented a 3.0 second perturbation followed by use of the controller to dampen out oscillations.

The three plant outputs were:

$$\begin{aligned} Y_1 &= \text{shoulder yaw} \\ Y_2 &= \text{shoulder pitch} \\ Y_3 &= \text{elbow pitch} \end{aligned}$$

The plant inputs were limited to 0.7 deg/sec..

In order to simulate the perturbation, the following control sequence was input to the uncontrolled plant:

$$\begin{aligned} u_1 &= 0.7 \quad 0 < t < 1.5 \\ u_1 &= -0.7 \quad 1.5 < t < 3.0 \\ u_2 &\equiv u_3 \equiv 0. \end{aligned}$$

The resulting plant states at the end of this perturbation were the plant initial conditions for all controller simulations. The plant outputs at the end of the perturbation were the model initial condition (s). The model rate initial condition (s) were set to zero.

7.5 Simulation Results

Figures 13 a,b,c show the position 1 outputs for the following parameters:

$$T = \text{diag9}(6000, 10, 6000, 1, 1, 1, 1, 1, 1)$$
$$\bar{T} = \text{diag9}(1, .000001, 1, 100, 100, 100, 100, 100, 100)$$
$$D = \text{diag3}(.25, .25, .25)$$

model damping = 0.15

Note that all controlled outputs decay faster than do the outputs with no control.

Figures 14 a,b,c show the position 2 outputs using the above position 2 controller tuning parameters, and Figures 15 a,b,c show the position 1 outputs using controller parameters, tuned for position 2. For these cases the differences between the controlled and uncontrolled responses were not remarkable.

These results and other experiments show that a satisfactory level of control can be achieved by the MRAC with tuning tailored for the individual positions. Attempts to develop one set of controller tuning parameters that would provide satisfactory control for all three positions were not successful.

8 Conclusions and Recommendations

In summary, the DMRAC algorithm has been found to be an effective robotic control algorithm in both simulation and on the actual robotic manipulator. Its performance was robust with respect to static and dynamic load variations and also disturbances.

At present the DMRAC is being considered for all six joints of the actual PUMA and further tuning with dynamic feedforward is begin considered for the shuttle RMS.

References

- [1] K. Sobel and H. Kaufman, "Direct Model Reference Adaptive Control for a Class of MIMO Systems", *Advances in Control and Dynamics Systems*, C.T. Leondes, (ed.), Vol. XXIV, Academic Press, 1986, pp. 245-314.
- [2] H. Kaufman, G. Neat and R. Steinvorth, "Asymptotically Stable MIMO Direct Model Reference Adaptive Controller for Processes not Necessarily Satisfying a Positive Real Constraint," European Cont. Conf., Grenoble, July 1991, pp. 1872-1877.
- [3] I. BarKana, "Adaptive Control - A Simplified Approach," in Academic Press Advances in Control and Dynamic Systems, C.T. Leondes (ed.), Vol. 25, 1987.
- [4] R. Steinvorth, H. Kaufman, and G. Neat, "Direct Model Reference Adaptive Control with Application to Flexible Robotics," Proc. of IASTED Conf. on Adaptive Control and Signal Proc., NY, Oct. 1990, pp. 9-12.

- [5] R. Steinvorth and H. Kaufman, "Direct Model Reference Adaptive Control of Robots," CISS, Baltimore, March 1991, pp. 667-672.
- [6] B. Armstrong, O. Khatib, and J. Bardick, "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", Proc. 1986 IEEE Robotics and Automation Conf., San Francisco, pp. 510-518.
- [7] M.A. Scott, M.G. Gilbert, and M.E. Demeo, "Active Vibration Damping of the Space Shuttle Remote Manipulator System", NASA Report TM- 104149, August 1991.

Table 1: Parameter Values for 3 Joint Trajectory Tracking Runs

\bar{T} (diag component)	e_z	20	40	40
	x_m	140	20	200
		30	200	30
	u_m	140	200	200
T (diag component)	e_z	30	30	40
	x_m	200	30	400
		60	400	60
	u_m	200	400	400
joint		1	2	3
Model	w_n	10	10	10
	ζ	1	1	1
Feed Forward	K_d	6	6	6
	r	0.05	0.05	0.05
alpha	α	0.02	0.02	0.02

Table 2: First Three Joint Tracking Test Trajectory

Knot Point	Joint Positions (deg)			Time (sec)
	1	2	3	
0	0	-45	180	-
1	-90	-90	90	6
2	0	0	180	8
3	0	-45	180	6

Table 3: Static Load Change Trajectory

Knot Point	Joint Positions (deg)			Time (sec)
	1	2	3	
0	0	-45	180	-
1	0	-45	180	3
2	45	0	0	10
3	0	-45	180	10

Table 4: Dynamic Load Change Trajectory

Knot Point	Joint Positions (deg)			Time (sec)
	1	2	3	
0	0	-45	180	-
1	0	-45	180	3
2	45	-90	90	10
3	0	-45	180	10

Table 5: Tunable Parameters

Parameters	Description	Values
T	24 x 24 integral weighting matrix	576
\bar{T}	24 x 24 proportional weighting matrix	576
w_n	Undamped natural frequency for Joint i model	6
ζ_i	Damping ratio for Joint i model	6
α	6 x 6 diagonal plant derivative weighting matrix	6
K_d	DC gain of Joint i supplementary feed-forward block	6
τ_i	Time constant of Joint i supplementary feed-forward block	6
	Total	1182

Table 6: Final Parameter Values

\bar{T} (diag component)	" e_x "	20	40	22	0.2	0.2	0.2
	" x_m "	140	20	140	35	100	22
	" x_m "	1.4	0.2	1.4	0.2	1.4	0.2
	" u_m "	140	160	110	1.4	1.4	1.4
T (diag component)	" e_x "	20	60	25	0.2	0.2	0.2
	" x_m "	140	20	150	35	140	25
	" x_m "	1.4	0.2	1.4	0.2	1.4	0.2
	" u_m "	140	160	130	1.4	1.4	1.4
Joint		1	2	3	4	5	6
Model	w_n	4	4	4	7	7	7
	ζ	1	1	1	1	1	1
Feed Forward	K_d	6	6	6	6	6	6
	τ	0.1	0.1	0.1	0.1	0.1	0.1
alpha	α	0.035	0.02	0.02	0.01	0.01	0.01

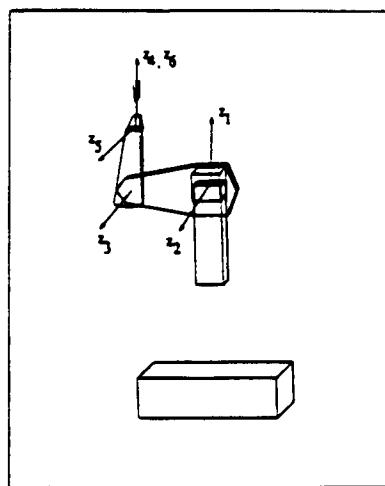


Figure 1: PUMA 560 Coordinate Frame

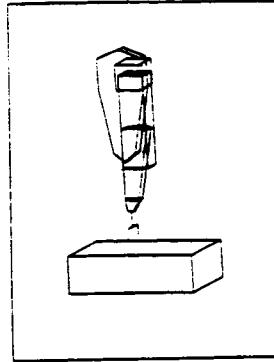


Figure 2: PUMA 560 in Stable Equilibrium

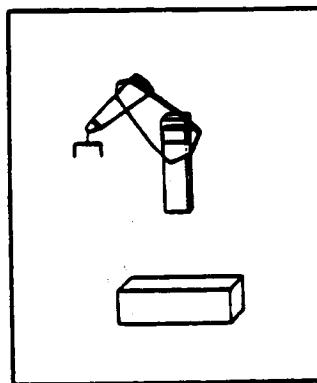


Figure 3: Shutdown Position

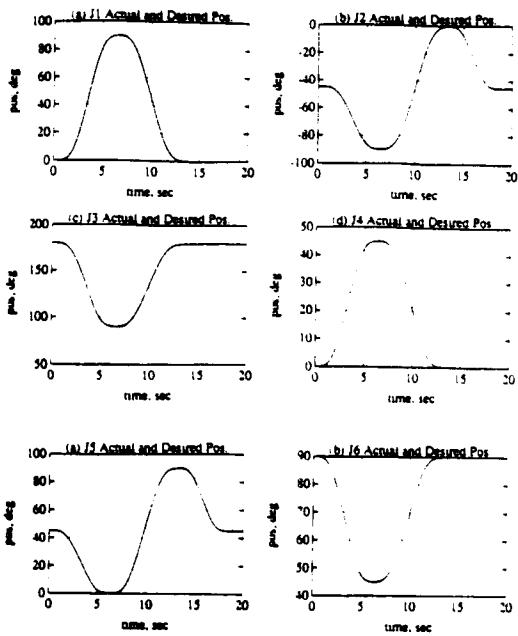


Figure 4: Simulation results for all 6 joints

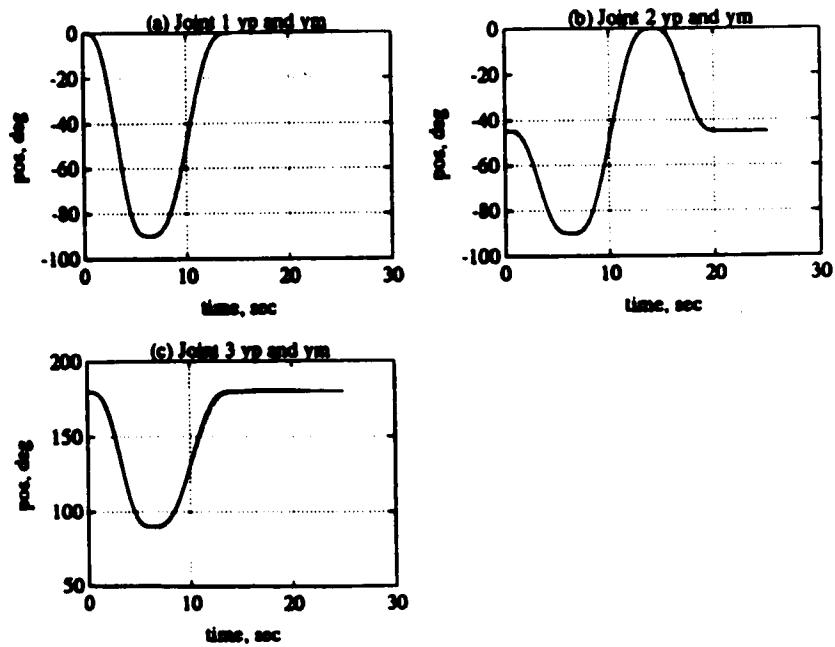
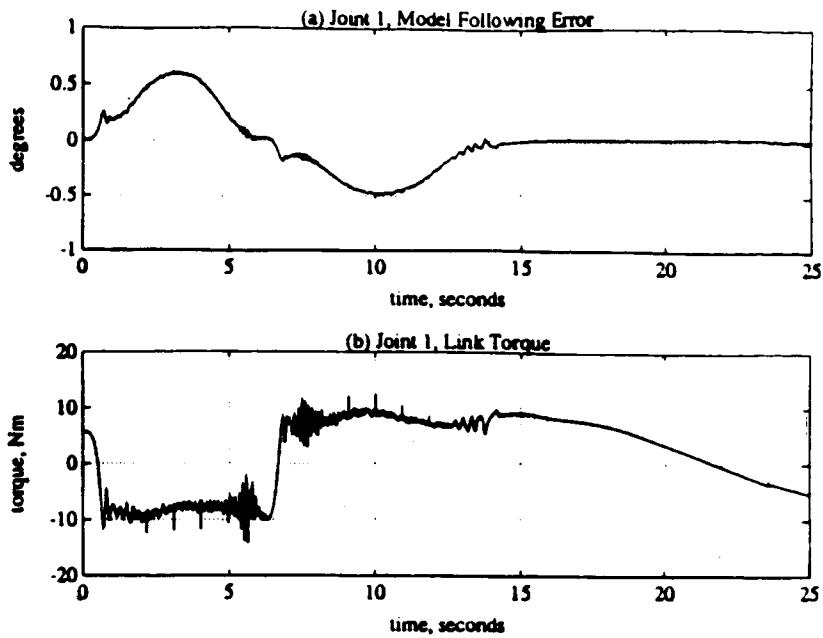
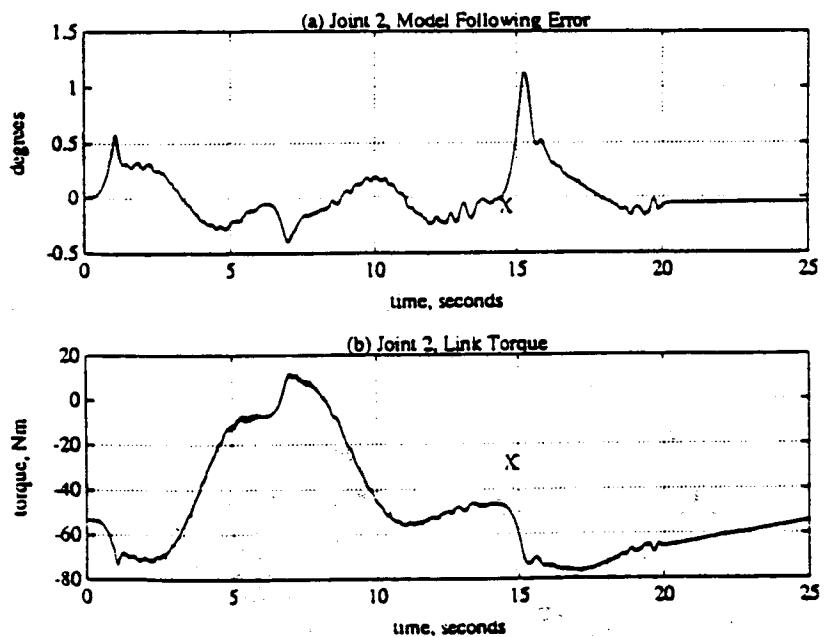


Figure 5: Plant and Model Output for First Trajectory.
 (a) Joint 1. (b) Joint 2. (c) Joint 3.



**Figure 6: Joint 1 Data for First Trajectory. (a) Model following error.
(b) Joint Troque.**



**Figure 7: Joint 2 Data for First Trajectory. (a) Model Following error. (b)
Joint Torque.**

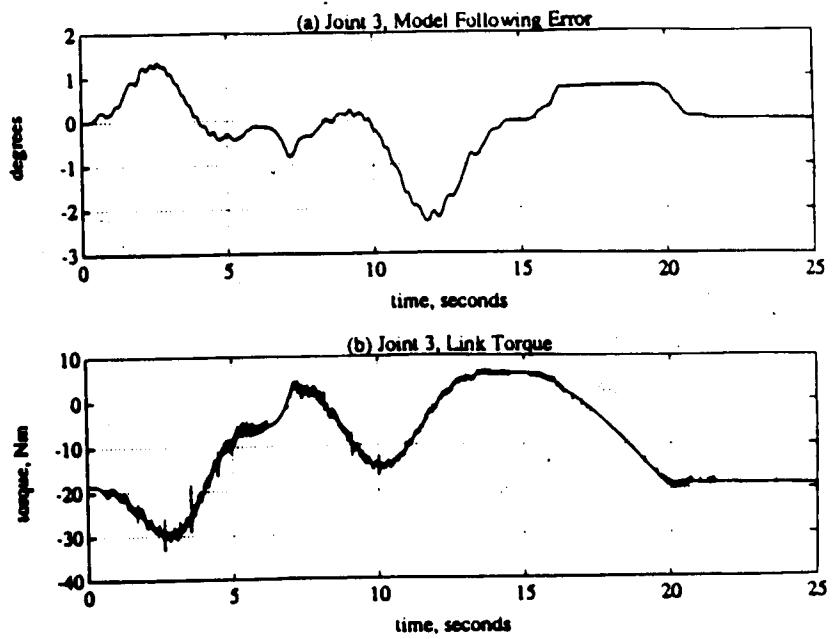


Figure 8: Joint 3 Data for First Trajectory. (a) Model following error. (b) Joint Torque.

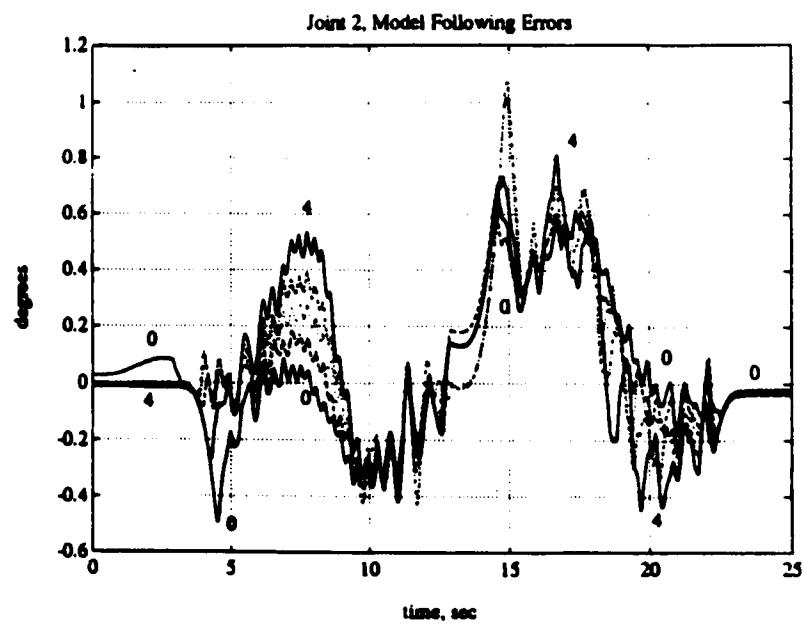


Figure 9: Joint 2 Static Load Model Following Error

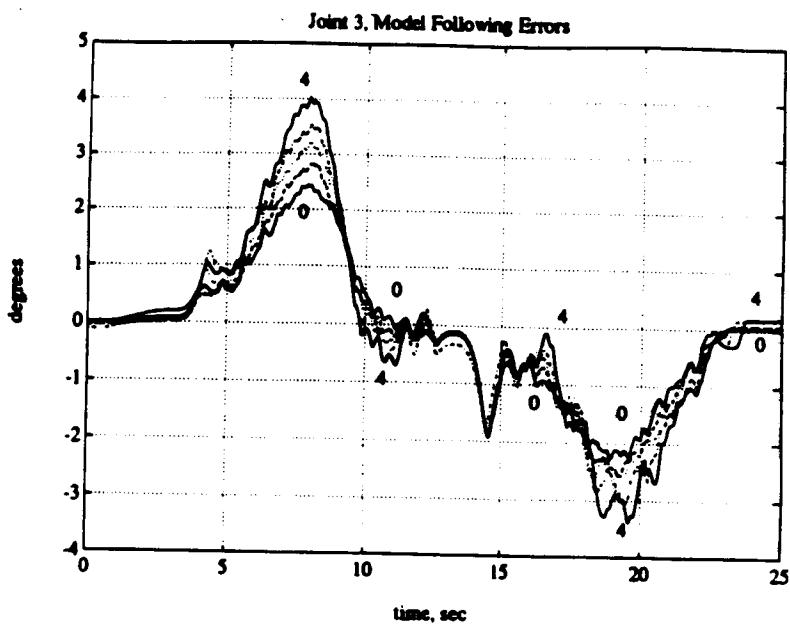


Figure 10: Joint 3 Static Load Model Following Error

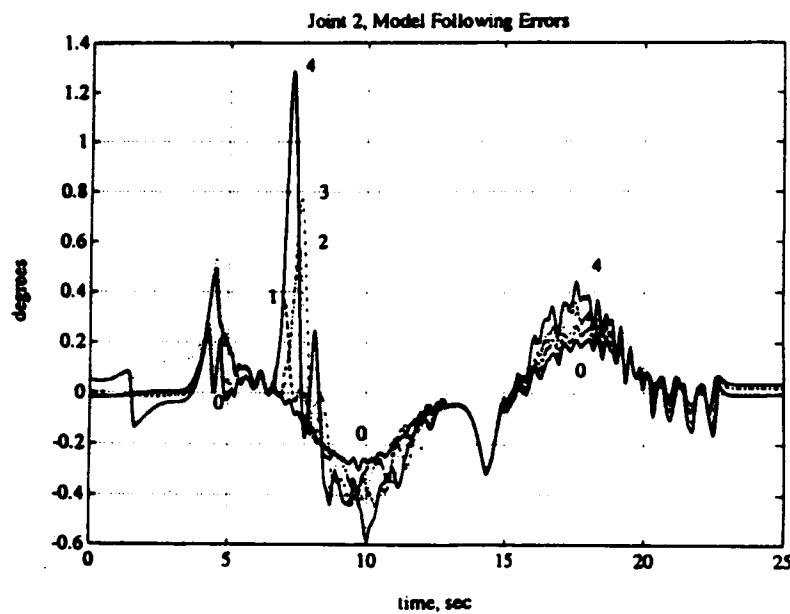


Figure 11: Joint 2 Dynamic Load Model Following Errors

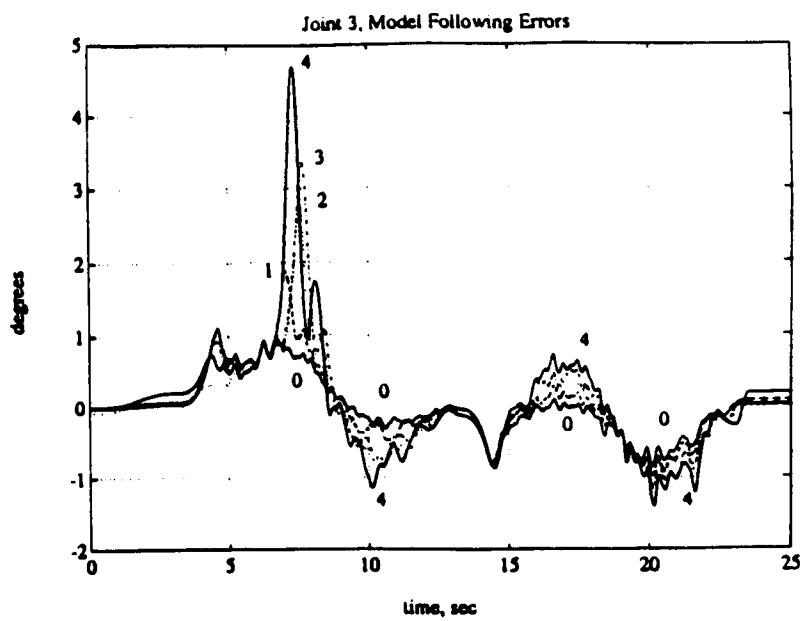


Figure 12: Joint 3 Dynamic Load Model Following Errors

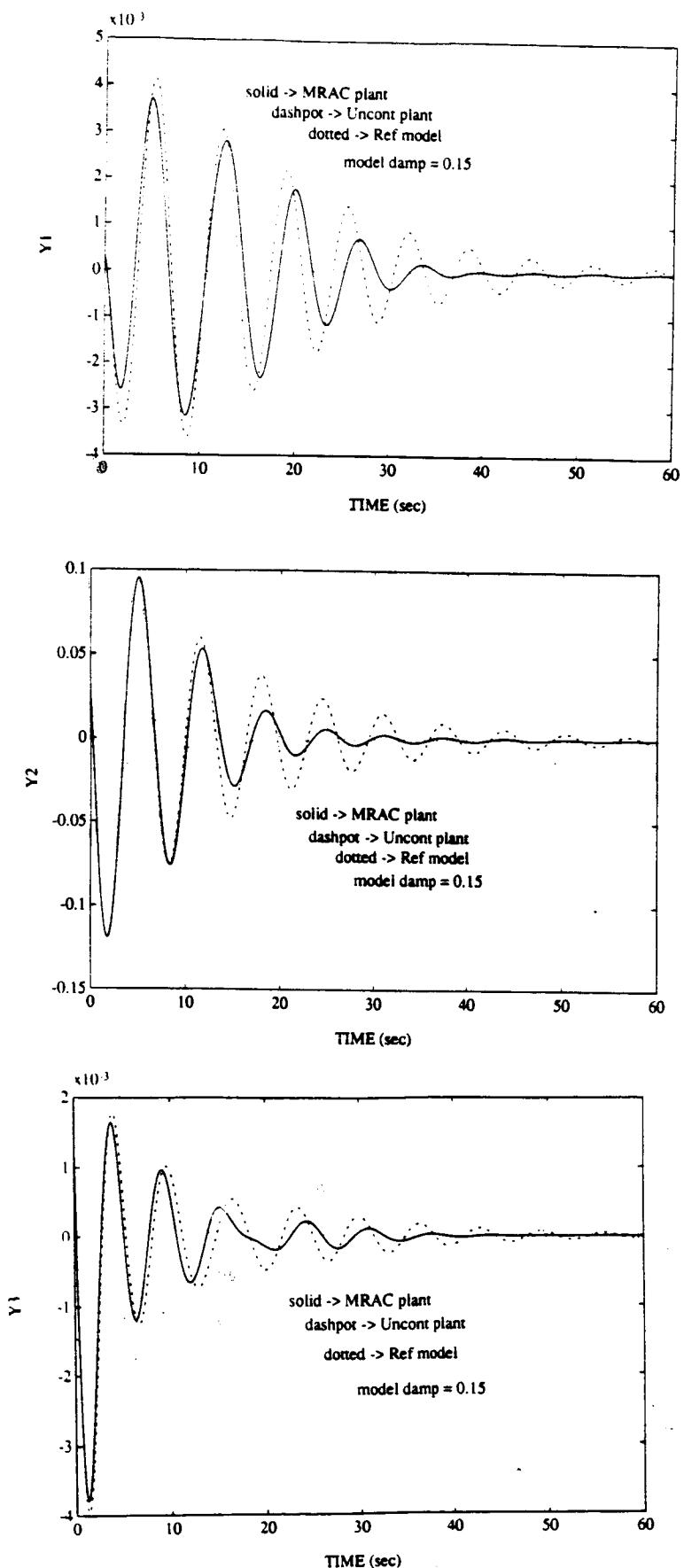


Figure 13: Shuttle RMS - Orientation 1 outputs for MRAC tuned to this position

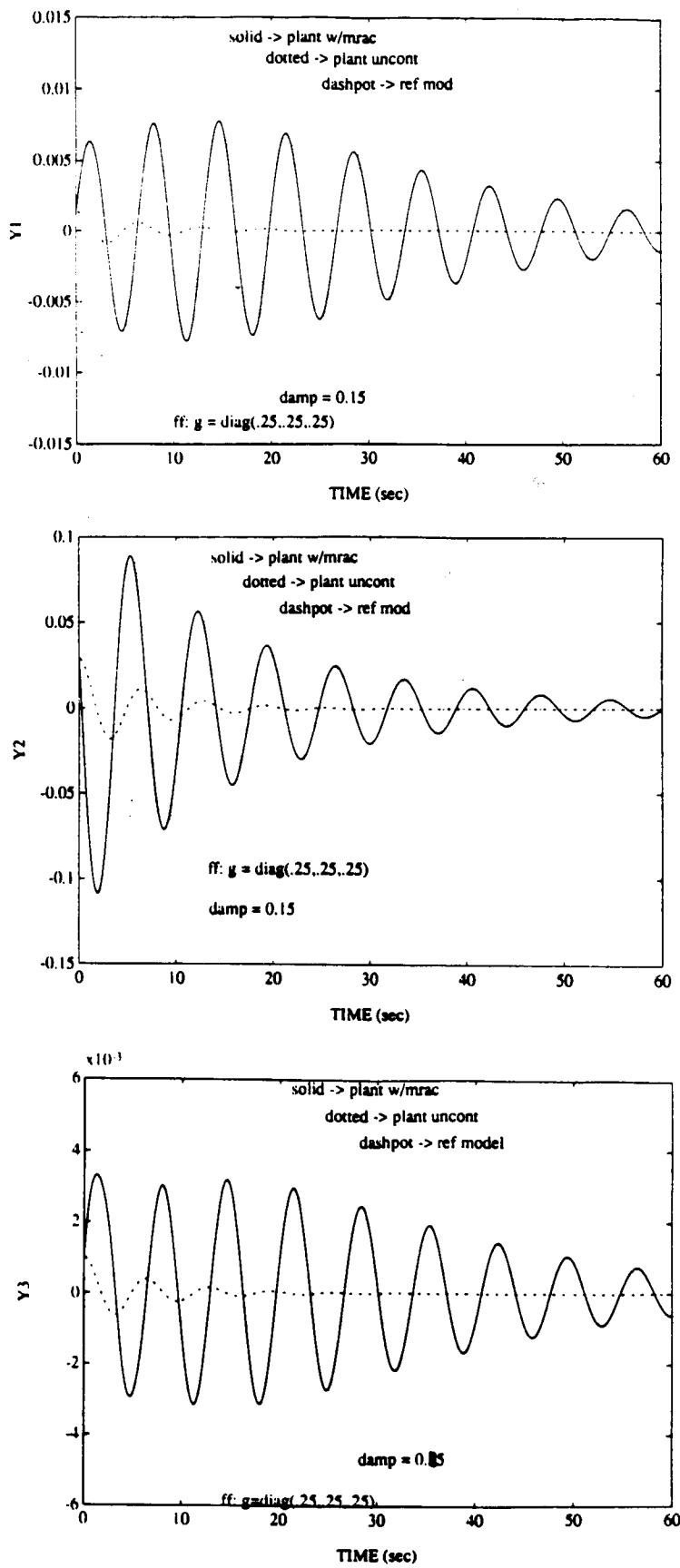


Figure 14: Shuttle RMS - Orientation 2 outputs using the MRAC tuned for orientation 1

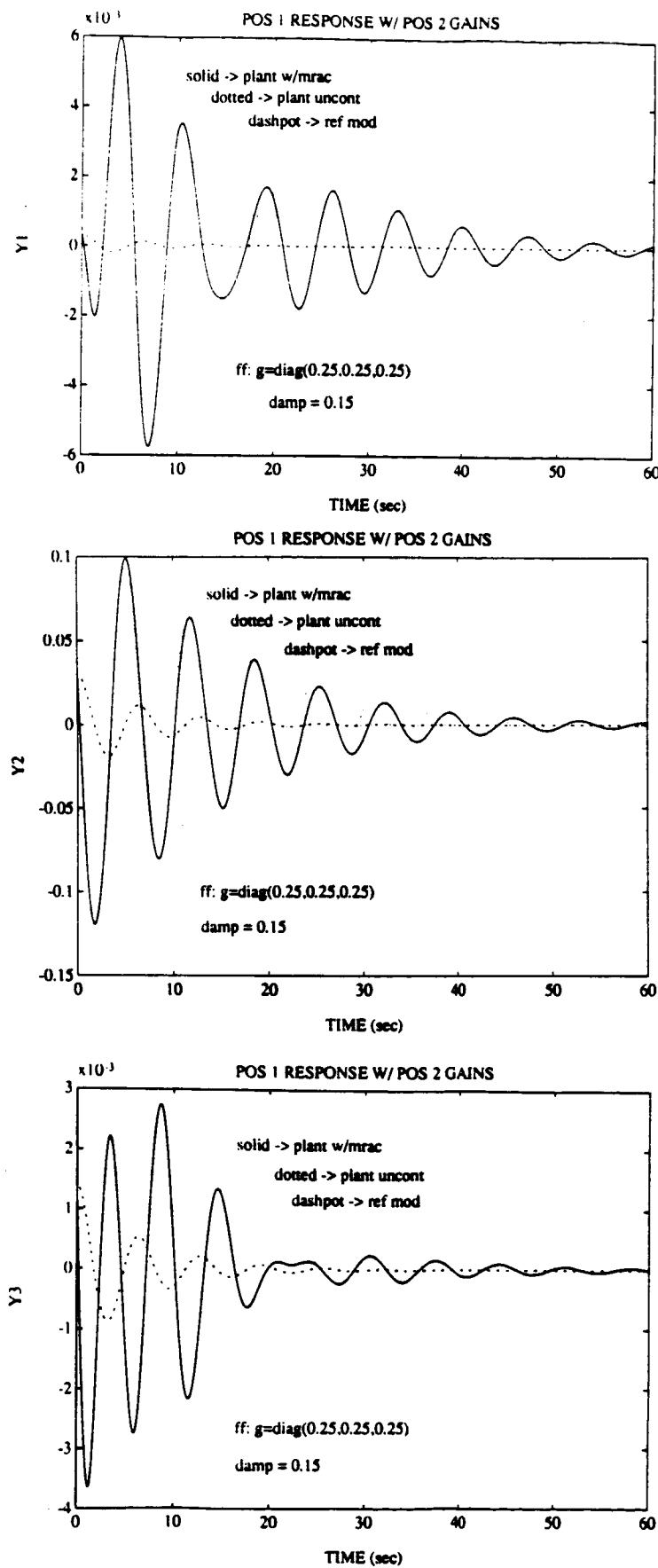


Figure 15: Shuttle RMS - Orientation 1 outputs using the MRAC tuned for orientation 2