Decentralized adaptive control

Decentralized Adaptive Control:
a simulation study.

Hellemons A.P.C.

WFW-rapport 90.054

Afstudeerhoogleraar: Prof. dr. ir. J.J. Kok

Stagebegeleider: Bram de Jager

Oktober 1990

Technische Universiteit Eindhoven
Faculteit der Werktuigbouwkunde
Vakgroep Fundamentele Werktuigbouwkunde

Summary

The article "Decentralized Adaptive Controllers based on the direct Method of Lyapunov" written by Lin Shi and Sunil K. Singh [1] shows the application of the direct method of Lyapunov for the design of decentralized adaptive controllers for tracking in nonlinear systems. It shows how the assumptions made about the nature of the nonlinear interactions influence the adaptation laws. They examine the role of an auxiliary adaptive signal and show how it can be used to improve the convergence rate and the ultimate bound of the tracking error. The control scheme is also computationally simple and therefore practical for real-time implementation. The proposed control scheme is simulated on a two-link robot and the simulations validate their conclusions. Future work will deal with the optimal choice of various parameters.
Chapter 1 gives a summary of the publication of Shi and Singh. I had to make research into the results of Shi and Singh, to see if they are correct and useful. Therefore I reproduced the simulation results. The results from Lemma are as good as the "old results" from Shi and Singh. But Theorem 1 and Theorem 2 show some differences. Especially for larger ω the difference for Theorem 1 en Theorem 2 is clear. It was in my intention to use Theorem 2 in a simulation of an imaginary RT-robot (one Rotation, one Translation). See Chapter 2. After that I would have looked at the influence of a few of the 10 different parameters in Theorem 2 on tracking a trajectory. However I didn't succeed to simulate the tracking of the RT-robot (Chapter 3). I decided to simulate the tracking with changed parameter values on the RR-robot from the publication of Shi and Singh. The results are shown in Chapter 4. It seems further experiments are necessary to examine the use of the new Theorem.

SYMBOLS

x	scalar
$\mathbf{x}$	column
$\mathbf{x}_i$	element i of column $\mathbf{x}$
X	matrix
X^T	transposed of matrix X
$x(t)$	x as function of t
$\dot{x}$	$\frac{dx}{dt}$

CONTENTS

1 THE PUBLICATION OF SHI AND SINGH

1.1 Summary of the publication.

For the control of nonlinear interconnected systems, an effective approach is to apply decentralized control strategies such that each subsystem is controlled independently based on local information. Many decentralized schemes have been investigated for interconnected systems. In [2] Gavel and Siljak designed a decentralized adaptive controller by imposing certain structural restrictions on the nonlinear interactions. The resulting controller ensured global stability. In their work, the local adaptation laws were designed using the direct method of Lyapunov which guaranteed the ultimate boundedness of the tracking error to a residual set. The study of Shi and Singh draws much of its inspiration from their work. They attempt to point out that most of the techniques being employed for decentralized adaptive control which are based on Lyapunov functions have an underlying similarity. Shi and Singh make several assumptions about the structure and behavior of the interactions. They look how the assumptions influence the adaptation law and the actual behavior. They examine the role of an auxiliary adaptive signal and show how it can be used to improve the convergence rate and the ultimate bound of the tracking error. The control scheme is also computationally simple and therefore practical for real-time implementation. The proposed control scheme is simulated on a two-link robot and the simulations validate their conclusion.

1.2 System description.

Shi and Singh adhere to the notation from [2]
The model:

$$\dot{\mathbf{x}}_i = \mathbf{A}_i\mathbf{x}_i + \mathbf{b}_i u_i + \mathbf{b}_i \mathbf{z}_i(t,x), \qquad i = 1,2,..,N \tag{1}$$

where $\mathbf{z}_i(t,x)$ describes the strength of interactions from other subsystems. the parameters $\mathbf{A}_i$ and $\mathbf{b}_i$ may be unknown.
The adaptive controller attempts to guide the plant along reference trajectories $\mathbf{x}_{mi}$,generated by a linear reference model.

$$\dot{\mathbf{x}}_{mi} = \mathbf{A}_{mi}\mathbf{x}_i + \mathbf{b}_{mi}\mathbf{r}_i \qquad i = 1,2,..,N \tag{2}$$

$\mathbf{A}_{mi}$ is a stable matrix. Therefore it satisfies the Lyapunov matrix equation, i.e.,for any positive definite matrix $\mathbf{Q}_i$, there exists an unique positive matrix $\mathbf{P}_i$ such that

$$\mathbf{A}_{mi}^T\mathbf{P}_i + \mathbf{P}_i\mathbf{A}_{mi} = -\mathbf{Q}_i \tag{3}$$

Define the position error $\mathbf{e}$ (t) for each subsystem as

$$\mathbf{e}_i(t) = \mathbf{x}_i(t) - \mathbf{x}_{mi}(t) \tag{4}$$

then

$$\dot{\mathbf{e}}_i = \dot{\mathbf{x}}_i - \dot{\mathbf{x}}_{mi} = \mathbf{A}_i\mathbf{x}_i + \mathbf{b}_i\mathbf{u}_i + \mathbf{b}_i\mathbf{z}_i - \mathbf{A}_{mi}\mathbf{x}_{mi} - \mathbf{b}_{mi}\mathbf{r}_i \tag{5}$$

Assuming that the pairs $(\mathbf{A}_i, \mathbf{b}_i)$ and $(\mathbf{A}_{mi}, \mathbf{b}_{mi})$ are in the companion controllable canonical form, there exist a constant vector $\mathbf{k}_i^*$ and the constant k_{0i}^* such that

$$\mathbf{A}_{mi} = \mathbf{A}_i + \mathbf{b}_i\mathbf{k}_i^{*T}, \quad \mathbf{b}_{mi} = \mathbf{b}_i k_{0i}^* \tag{6}$$

They assume the sign of $\mathbf{b}_i$ is known. Let $k_{0i}^* > 0$. The adaptation law in the adaptive controller will attempt to find the constant vector $\mathbf{k}_i^*$ and the constant k_{0i}^* .

We construct the decentralized adaptive control law

$$u_i(t)=\vartheta_i^T v_i \ , \qquad i = 1,2,..,N \tag{7}$$

where $\vartheta_i^T = (\mathbf{k}_i^T, k_{0i})$ is the time-varying adaptation gain vector. v_i is a regressor vector.

1.3 The three implementations.

<u>Theorem 1:</u> Choose (7) with adding an auxiliary signal f (t)

$$u_i(t) = \vartheta_i^T \mathbf{v}_i + f_i(t) \ , \qquad i = 1,2,...,N \tag{8}$$

$$\text{where } \mathbf{v}_i = (\mathbf{x}_i^T, r_i)^T \tag{9}$$

Choose the adaptation laws

$$\vartheta_i = \dot{\phi}_i^* - \Gamma_i(\mathbf{b}_{mi}^T \mathbf{P}_i \mathbf{e}_i)\mathbf{v}_i \tag{10}$$

where

$$\phi_i^* = -\Pi_i(\mathbf{b}_{mi}^T \mathbf{P}_i \mathbf{e}_i)\mathbf{v}_i \tag{11}$$

$$\phi_i = \vartheta_i - \vartheta_i^* \ , \ \vartheta_i^* = (\mathbf{k}_i^{*T}, k_{0i}^*)^T \tag{12}$$

$$\dot{f}_i = \dot{f}_i^* - \alpha_i \mathbf{b}_{mi}^T \mathbf{P}_i \mathbf{e}_i \tag{13}$$

where

$$\dot{f}_i^* = -\pi_i \mathbf{b}_{mi}^T \mathbf{P}_i \dot{\mathbf{e}}_i \tag{14}$$

π_i and α_i are positive constants, Γ_i and Π_i are positive definite matrices.

If the nonlinear interactions are "slowly time-varying"

$$\left\| \dot{z}_i(t,\mathbf{x}) \right\| \approx 0$$

then, the system trajectories track the model trajectories with an error that approaches zero asymptotically. To proof this, they choose a Lyapunov function $V(\mathbf{e},\phi,\mathbf{f}) \geq 0$ with $\dot{V}(\mathbf{e},\phi,\mathbf{f}) \leq 0$. The proof they give is no longer valid if the interactions are rapidly time varying.

Lemma: impose the restriction that the interactions are bounded linearly in states.

$$z_i(t,\mathbf{x}) \le \sum_{j=1}^{N} \xi_{ij}\left\|\mathbf{x}_j\right\| \qquad \xi_{ij} \ge 0$$

from [2]:

$$\mathbf{u}_i(t) = \vartheta_i^T \mathbf{v}_i \tag{15}$$

where $\mathbf{v}_i^T = [\mathbf{e}_i^T, \mathbf{r}_i]$ (16)

choose the adaptation laws

$$\dot{\vartheta}_i = -\Gamma_i(\mathbf{b}_{mi}^T \mathbf{P}_i \mathbf{e}_i)\mathbf{v}_i - \sigma\Gamma_i\vartheta_i \tag{17}$$

where Γ_i is a positive definite matrix, and $\sigma>0$ is a "σ-modification" term used to improve the robustness of the overall system.

The solutions of the error dynamics are globally ultimately bounded with respect to $V_f(\mathbf{e},\phi)$

Theorem 2: choose the control law

$$u_i(t) = \vartheta_i^T \mathbf{v}_i + f_i(t) \tag{18}$$

$$\mathbf{v}_i^T = [\mathbf{e}_i^T, \mathbf{r}_i] \tag{19}$$

$$\dot{\mathbf{f}}_i(t) = -\pi_i \mathbf{b}_{mi}^T P_i \dot{\mathbf{e}}_i - \alpha_i \mathbf{b}_{mi}^T P_i \mathbf{e}_i \tag{20}$$

The solutions of the error dynamics are again globally ultimately bounded. Shi and Singh state that the system response is faster, and the ultimate bound of the error dynamics $\tilde{V}_f(\mathbf{e},\phi,\mathbf{f})$ is less by using the auxiliary signal.

Finally Shi and Singh suppose the magnitude of the interactions has an upper bound

$$\left\|z_i(t,\mathbf{x})\right\| \le \beta$$

Then, Shi and Singh state the closed loop error dynamics is again globally ultimately bounded.

1.4 Remarks.

A Shi and Singh:"For Lemma the convergence rate of the error dynamics can be evaluated by the ratio $\chi=-\dot{V}(e,\phi)/V(e,\phi)$. Obviously, if we can find another Lyapunov function candidate $\tilde{V}(e,\phi)$ such that $\tilde{\chi}=-\dot{V}(e,\phi)/V(e,\phi) \geq \chi$, then we can conclude that new closed-loop error dynamics have better transient performance. We now show how the performance of this system can be improved by using an auxiliary signal."

For Theorem 2 they use the ratio $\tilde{\chi}=-\dot{\tilde{V}}(e,\phi,f)/\tilde{V}(e,\phi,f)$.
After a few computations they got:

$$\tilde{V} \geq V \tag{21}$$

and

$$\dot{\tilde{V}} \geq \dot{V} \tag{22}$$

Then

$$\tilde{\chi}=-\tilde{V}(e,\phi,f)/\tilde{V}(e,\phi,f) \geq \chi=-\dot{V}(e,\phi)/V(e,\phi) \tag{23}$$

"So the error dynamics of Theorem 2 has a faster convergence rate compared to Lemma."

Remark 1: They make a mistake by saying $\tilde{\chi} \geq \chi$. If we look better, we see this conclusion can't be drawn from (21) to (23). So there is no proof in the publication of Shi and Singh the error dynamics of Theorem 2 has a faster convergence rate compared to Lemma.

B For a choice of initial conditions $\dot{\theta}_2$ at t=0 we look at figure 4:

Lemma	$\dot{\theta}_2(0) = 3$
Theorem 1	$\dot{\theta}_2(0) = 2$
Theorem 2	$\dot{\theta}_2(0) = 2$
reference trajectory	$x_{m2} = 0$

This in contradiction to figure 8, where for all three plants the velocity error $\dot{\theta}_2 - x_{m2}$ is 3 at t=0.

REPRODUCTION

2.1 The simulations.

It is in my effort to create the same simulation results as Shi and Singh. There for I use the same decentralized adaptive control schemes in simulations of the two-link robot manipulator for tracking time-varying trajectories proposed in [1].
I want to know the position, velocity and acceleration of the two links as a function of time. There for the dynamic equations have to be solved:

$$u_1=[2.25 + 1.22\cos(\theta_2)]\ddot{\theta}_1+ [0.59 + 0.61\cos(\theta_2)]\ddot{\theta}_2- 1.22\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 -0.61\sin(\theta_2)(\dot{\theta}_2)^2+ 6.75\cos(\theta_1) + 2.35\cos(\theta_1+ \theta_2)$$

$$u_2=[0.59 + 0.61\cos(\theta_2)]\ddot{\theta}_1 + 0.59\ddot{\theta}_2 +0.61\sin(\theta_2)(\dot{\theta}_1)^2+ 2.35\cos(\theta_1+ \theta_2)$$

Showing the state vectors $\mathbf{x}_1 = (\theta_1,\dot{\theta}_1)^T$ and $\mathbf{x}_2 = (\theta_2,\dot{\theta}_2)^T$, the motion of the robot can be described as:

$$\dot{\mathbf{x}}_i= \begin{pmatrix} 0 & 1 \\ a_1 & a_2 \end{pmatrix} \mathbf{x}_i + \begin{pmatrix} 0 \\ b \end{pmatrix} u_i + \begin{pmatrix} 0 \\ b \end{pmatrix} z_i(t,\mathbf{x}), \qquad i=1,2 \qquad (24)$$

Where $z_i(t,\mathbf{x})$ contains all nonlinearities of the robot dynamics. The reference model in (2) for each joint is chosen as:

$$\mathbf{A}_{mi}= \begin{pmatrix} 0 & 1 \\ -1 & -2 \end{pmatrix}, \qquad \mathbf{b}_{mi}= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad (25)$$

with $\mathbf{x}_{mi}= (\theta_i,\dot{\theta}_i)^T$ and substitution of these matrices in (2) gives:

$$r_i= \ddot{\theta}_i+ 2\dot{\theta}_i+ \theta_i \qquad (26)$$

The reference trajectories used in the simulations are chosen as:

$$\theta_1(t)= 1 + \sin(t) + \sin(\omega t) \qquad (27)$$

$$\theta_2(t)= 1 + \cos(t) +\cos(2.0t) \qquad (28)$$

substitution of these equations in (26) gives the forces $r_i(t)$, which are substituted for $u_i(t)$ in the dynamic equations.

With $Q_i = I_i$, the matrix Lyapunov equation is solved to yield:

$$P_i = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, \qquad \mathbf{b}_{mi}^T P_i = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \tag{29}$$

For all three implementations applies:

*the time-varying adaptation gain vector $\vartheta_i^T = (\mathbf{k}_i^T, k_{0i})$ (30)

*the position error $\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}_{mi}$ (31)

For the adaptation laws in [1] is chosen $\Gamma_1 = 100 I_3$, $\Gamma_2 = 50 I_3$ and $\sigma = 0.01$. For the auxiliary signal $\pi_1 = 100, \alpha_1 = 50, \pi_2 = 100, \alpha_2 = 50$

Theorem_1:

$$u_i(t) = \vartheta_i^T \mathbf{v}_i + f_i(t), \quad i = 1,2 \tag{32}$$

where

$$\mathbf{v}_i = (\mathbf{x}_i^T, r_i)^T$$

$$\dot{f}_i = -\pi_i \mathbf{b}_{mi}^T P_i \dot{\mathbf{e}}_i - \alpha_i \mathbf{b}_{mi}^T P_i \mathbf{e}_i$$

$$\dot{\vartheta}_i = -\sigma \Gamma_i \vartheta_i - \Gamma_i (\mathbf{b}_{mi}^T P_i \mathbf{e}_i) \mathbf{v}_i$$

Lemma:

$$u_i(t) = \vartheta_i^T \mathbf{v}_i, \quad i = 1,2 \tag{33}$$

where

$$\mathbf{v}_i = (\mathbf{e}_i^T, r_i)^T$$

$$\dot{\vartheta}_i = -\sigma \Gamma_i \vartheta_i - \Gamma_i (\mathbf{b}_{mi}^T P_i \mathbf{e}_i) \mathbf{v}_i$$

Theorem_2:

$$u_i(t) = \vartheta_i^T \mathbf{v}_i + f_i(t), \quad i = 1,2 \tag{34}$$

where

$$\mathbf{v}_i = (\mathbf{e}_i^T, r_i)^T$$

$$\dot{f}_i = -\pi_i \mathbf{b}_{mi}^T P_i \dot{\mathbf{e}}_i - \alpha_i \mathbf{b}_{mi}^T P_i \mathbf{e}_i$$

$$\dot{\vartheta}_i = -\sigma \Gamma_i \vartheta_i - \Gamma_i (\mathbf{b}_{mi}^T P_i \mathbf{e}_i) \mathbf{v}_i$$

*In the publication for Theorem 1 is given:

$$\dot{\vartheta}_i = \dot{\phi}_i^* - \Gamma_i (\mathbf{b}_{mi}^T P_i \mathbf{e}_i) \mathbf{v}_i$$

with $$\phi_i^* = -\Pi_i [\mathbf{b}_{mi}^T P_i \mathbf{e}_i] \mathbf{v}_i$$

I do not choose for this notation, because in [1] is chosen for $\sigma\Gamma_i \mathbf{v}_i$ in Lemma and Theorem 2. In their chapter **simulation results** they didn't chose matrix Π_i, that is why I suppose Shi and Singh chose this notation too.

Finally Lemma gives 10 and Theorem 1 and 2 give 12 differential equations, which are solved by the program MATLAB. There for the subroutine "ODE 45" is used.
[T,X] = ODE45('**x**dot',t0,tf,**x**0) integrates a system of ordinary differential equations described by the M-file xdot.m over the interval t0 to tf and using initial conditions **x**0. It returns the state vector **x**(t) with tolerance = 1.e-6.
For the state vector $\mathbf{x}^T$ I choose

$$[\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, k_{11}, k_{12}, k_{01}, k_{21}, k_{22}, k_{02}] \qquad \text{Lemma}$$

$$[\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, k_{11} k_{12}, k_{01}, k_{21} k_{22}, k_{02}, f_1, f_2] \qquad \text{Theorem 1,2}$$

I choose tf=10 seconds because:

1. computation for tf=20 s asks too much memory space.
2. after 10 s no further details are shown by the graphics with respect to representation.
3. the computations take too much time.

About the initial conditions **x**0 : The starting conditions have to be gathered from the figures, because the were not given in the publication. Also the starting conditions for the adaptation gain vectors are not presented in [1]. I choose them $[0,0,0]^T$.
If we, for the choice of $\dot{\theta}_2$ at t=0, look in figure 4. We see

Lemma	$\dot{\theta}_2(0) = 3$
Theorem 1	$\dot{\theta}_2(0) = 2$
Theorem 2	$\dot{\theta}_2(0) = 2$
reference trajectory	$x_{m2} = 0$

This in contradiction to figure 8, where for all three plants the velocity error $\dot{\theta}_2 - x_{m2}$ is 3 at t=0.

For **x**0 I choose

$$x0^T=[0\ 1.5\ 0\ 3\ 0\ 0\ 0\ 0\ 0\ 0] \qquad \text{Lemma}$$
$$x0^T=[0\ 1.5\ 0\ 3\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] \qquad \text{Theorem1,2}$$

2.2 The results.

The simulation results of the decentralized adaptive controller using the results from the Lemma, Theorem 1 and Theorem 2 for ω=0.5 are shown in figures 1,2,4,5,6,7 and 8, and for ω=3 and ω=6 in figure 11,12 respectively 15. Figure 9 shows the control effort. I don't present all figures, because they do not show us extra information. All figures are in Appendix A.
In general one can say the figures resemble the old figures. But if, for example, we look at 5,12 and 15 we see some clear differences.
The differences can be caused while Shi and Singh:

1. used another integration method.
2. used a bigger tolerance in their computations.
3. evaluated f_i and the accelerations in a different way. There for, they used a trapezoidal integration rule [1], because the acceleration signal of each subsystem is directly available.
4. they chosed the initial conditions otherwise.
5. faults in their or my programs.

My simulations were computed with the program MATLAB on "cirp 640 XT" computer with a tolerance of 1.e-6. The used programs are shown in figure (32) and (33) Appendix A. For Theorem 2 only. From this I conclude my simulations are correct.

Shi and Singh state in the figures one can see the transient response and the steady-state tracking error are improved significantly by using the auxiliary signals.
Figure 6,7 and 8 show this is true for ω=0.5, although fig.5

doesn't make this very clear.
If we look at the new figures 11 and 15 , we see this is not true for $\omega=3$.
For $\omega=6$ Shi and Singh state the system is not stable for this case. In figure 15 we can see the difference again, and conclude this is not true again.

3 RT-ROBOT.

The schemes of Theorem 2 are used in simulations of a RT-robot(one rotation, one translation) . If this succeeds, the motions will be studied after changing a few parameters one by one. Imagine the robot holds a torch. The torch will cut out a hole in a plate with constant velocity. See figure (30) and (31).

For the robot:

dynamic equations:

$$H(\mathbf{q})\ddot{\mathbf{q}} + B(\mathbf{q},\dot{\mathbf{q}}) = T \tag{35}$$

where

$$\mathbf{q} = \begin{pmatrix} r \\ \varphi \end{pmatrix} \qquad T = \begin{pmatrix} F \\ M \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{36}$$

$$H = \begin{pmatrix} m+m_1 & 0 \\ 0 & I + ml^2/3 - mlr + (m+m_1)r^2 \end{pmatrix} \tag{37}$$

$$B = \begin{pmatrix} -\{(m+m_1)r - ml/2\}\dot{\varphi}^2 \\ 2\{(m+m\ )r_1 - ml/2\}\dot{r}\dot{\varphi} \end{pmatrix} \tag{38}$$

choose

$m = 10$ [kg]
$m_1 = 5$ [kg]
$I = 5$ [kgm^2]
$l = 1$ [m]

then

$$H = \begin{pmatrix} 15 & 0 \\ 0 & 8^1/3 - 10r + 15r^2 \end{pmatrix} \tag{39}$$

$$B = \begin{pmatrix} (5-15r)\dot{\varphi}^2 \\ (30r-10)\dot{r}\dot{\varphi} \end{pmatrix} \tag{40}$$

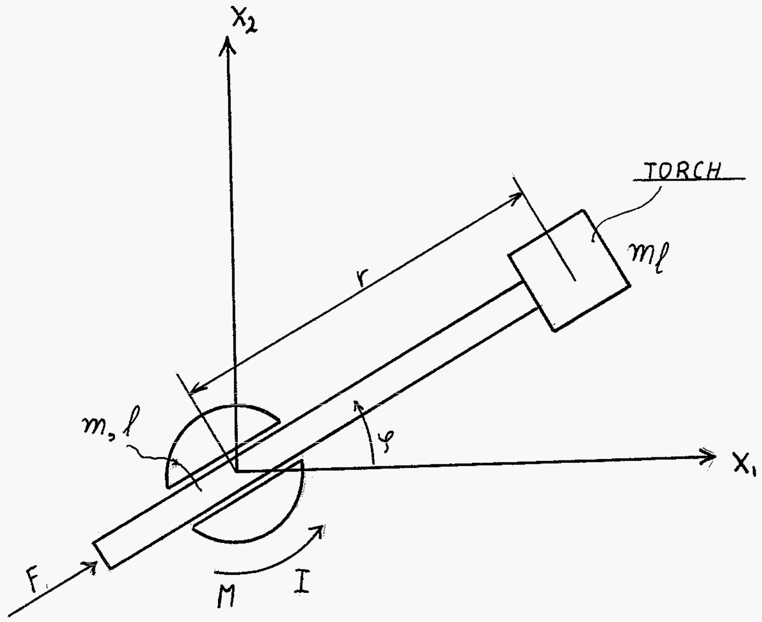


FIGURE 30 : THE MODEL

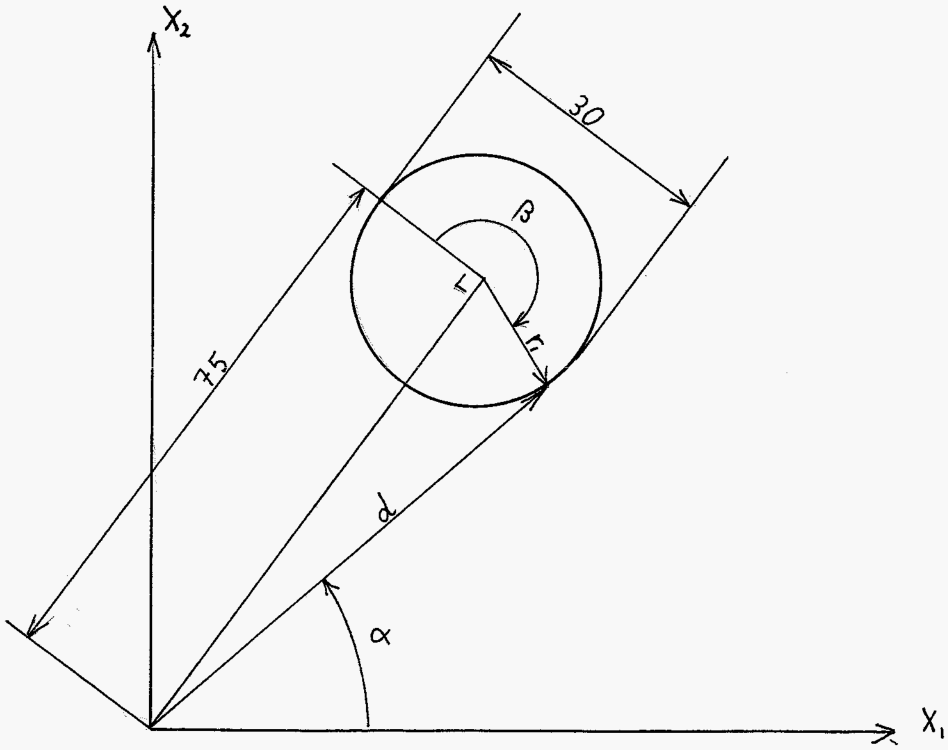


FIGURE 31 : REFERENCE TRAJECT

The reference trajectory:

The reference trajectory the robot has to follow, follows from figure (1b).
If we choose

$$\mathbf{x}_{m1} = \begin{pmatrix} \alpha \\ \dot{\alpha} \end{pmatrix} \quad \text{and} \quad \mathbf{x}_{m2} = \begin{pmatrix} d \\ \dot{d} \end{pmatrix} \tag{41}$$

and

$$\mathbf{A}_{mi} = \begin{pmatrix} 0 & 1 \\ -1 & -2 \end{pmatrix}, \quad \mathbf{b}_{mi} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{42}$$

with $Q_i = I_2$, we solve the matrix Lyapunov equation
$\mathbf{A}_{mi}^T \mathbf{P}_i + \mathbf{P}_i \mathbf{A}_{mi} = -\mathbf{Q}_i$ to yield

$$P_i = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, \quad \mathbf{b}_{mi} P_i = [0.5 \ 0.5] \tag{43}$$

substitution in the linear reference model,

$$\dot{\mathbf{x}}_{mi} = \mathbf{A}_{mi} \mathbf{x}_i + \mathbf{b}_{mi} \mathbf{s}_i \qquad i = 1,2,..,N \tag{44}$$

gives the required forces $s_i(t)$

$$\mathbf{s}_1(t) = \mathbf{u}_1(t) = \ddot{\alpha}(t) + 2\dot{\alpha}(t) + \alpha(t) \tag{45}$$

$$s_2(t) = u_2(t) = \ddot{d}(t) + 2\dot{d}(t) + d(t) \tag{46}$$

where from figure (1b), after a few computations

$$d(t) = \sqrt{75^2 + 150 r_1 \sin(\beta) + r_1^2} \tag{47}$$

$$\alpha(t) = \arctan\left(\frac{r_1 \cos(\beta)}{r_1 \sin(\beta) + 75} \right) \tag{48}$$

and β is function of time $\beta(t)$, with $\dot{\beta}$ = constant
further

$$\dot{d}(t) = \frac{75 r_1 \dot{\beta} \cos(\beta)}{r} \tag{49}$$

$$\ddot{d}(t) = \frac{-75 r_1 \dot{\beta}^2 \sin(\beta) - \dot{d}^2}{d} \tag{50}$$

$$\dot{\alpha}(t) = \cos^2(\alpha) - \frac{r_1\dot{\beta}(r_1 + 75\sin(\beta))}{(r_i\sin(\beta) + 75)^2} \tag{51}$$

$$\ddot{\alpha}(t) = \frac{2\dot{\alpha}\cos(\alpha)\sin(\alpha)r_1\dot{\beta}(r_1 + 75\sin(\beta))}{(r_1\sin(\beta) + 75)^2} - \tag{52}$$

$$\cos^2(\alpha)\frac{r_1\dot{\beta}^2\cos(\beta)[\ 75(r_1\sin(\beta) + 75) - 2r_1(r_1 + 75\sin(\beta))\ ]}{(r_1\sin(\beta) + 75)^3}$$

since $s_1(t)$ and $s_2(t)$ are known, we can substitute them in the dynamic equations.
For the adaptation laws (34), we select $\Gamma_1 = 100I_3$, $\Gamma_2 = 50I_3$ and $\sigma = 0.01$. For the auxiliary signal, we choose $\pi_1 = \pi_2 = 100$ and $\alpha_1 = \alpha_2 = 50$. The computer will solve the dynamic equations with the program MATLAB and subroutine ODE45 and give us **q**(t). See Appendix A figure (34).There for ODE45 uses **'xdot'**, **x0**, t0, tf. [T,X] = ODE45('**xdot**',t0,tf,**x0**)

I choose

$$\mathbf{x}^T = [r, \dot{r}, \varphi, \dot{\varphi}, k_{11}, k_{12}, k_{01}, k_{21}, k_{22}, k_{02}]$$

So

$$\text{xdot} = \dot{\mathbf{x}}^T = [\dot{r}, \ddot{r}, \dot{\varphi}, \ddot{\varphi}, \dot{k}_{11}, \dot{k}_{12}, \dot{k}_{01}, \dot{k}_{21}, \dot{k}_{22}, \dot{k}_{02}]$$

here, from (35) we got

$$\ddot{r} = {}^1/_{15}\{\ (-5+15r)\dot{\varphi}^2 + u_1\} \tag{53}$$

$$\ddot{\varphi} = \{\ (-30r+10)\dot{r}\dot{\varphi} + u_2\} \ / (8^1/_3 - 10r + 15r^2) \tag{54}$$

and for $\mathbf{x0}^T = [0\ 0.5\ 0\ 0.5\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$

The problem is the motion of the robot becomes unstable within 10 sample times. The computations are restarted after changing the parameters and initial conditions ±15 times. But with 10 different parameters this work looks endless. This because I have got no knowledge about the influence of the several parameters on the motion. I decide to study the influence of changed parameters on the motion of the RR-robot in the publication of Shi and Singh.

4 CHANGED PARAMETERS.

The same simulations are now computed after changing the parameters.The results are shown in figures (16) to (29) in Appendix A.

The results are discussed in **A** to **E**.

A. Change the matrix $A_{mi} = \begin{pmatrix} 1 & 0 \\ -1 & -2 \end{pmatrix}$ in $A_{mi} = \begin{pmatrix} 1 & 0 \\ -20 & -20 \end{pmatrix}$

this changes the signal r_i as follows

now

$$r_i(t) = 20\,\theta_i(t) + 20\,\dot{\theta}_i(t) + \ddot{\theta}_i(t) \tag{55}$$

and

$$u_i(t) = \mathbf{k}_i \mathbf{v}_i(t) + k_{0i} r_i(t) \tag{56}$$

while $\|r_i(t)\|$ becomes bigger, $\|u_i(t)\|$ becomes bigger. This could lead to oscillating.

In figure 17,19 and 21 we can see this. The oscillating behavior is less for Theorem 1 and 2.

B. Change the parameter σ in $\dot{\vartheta}_i = -\sigma\Gamma_i\vartheta_i - \Gamma_i(\mathbf{b}_{mi}^T P_i \mathbf{e}_i)\mathbf{v}_i$

See figure 22 and 23. Changing σ has influence on the robustness of the overall system. From the equation above we see that the term $-\sigma\Gamma\vartheta$ has a decreasing effect on ϑ. σ takes care $\|\vartheta\|$ will not become too big.

Figure 22 and 23 show increasing σ (σ=0.1) has the effect of achieving smoother control.

Shi and Singh state decreasing σ (σ=0.001) has the effect of achieving smaller tracking error but could also lead to oscillating.

In fig.22 and 23 we can see this doesn't have to be true.

C. Changing Γ (Figure 24 and 25 for Γ=10I and Γ=200I).

The effect of changing Γ is not very easy to see from the equations In fig. 24 and 25 we see changing Γ leads to oscillating for Lemma. The effect for Theorem 1 and 2 is less compared to Lemma.

Shi and Singh state increasing Γ has the effect of achieving smaller tracking error but could also lead to oscillating. In the figures this isn't very clear to see again.

D. Changing π_i in $\dot{f}_i = -\pi_i \mathbf{b}^T_{mi} P_i \dot{\mathbf{e}}_i - \alpha_i \mathbf{b}^T_{mi} P_i \mathbf{e}_i$ for Theorem 1 and 2.

→Increasing π_i will lead to an increasing first term in the equation. So the effect of a change in $\dot{\mathbf{e}}$ will be bigger in comparison to $\mathbf{e}$. This leads to a smaller error in the velocity and a bigger error in the position.

→Decreasing π_i will have the reverse effect. So a bigger error in the velocity and a smaller error in the position.

In figure 26 and 27 this effect is shown for $\pi_1=\pi_2=50$ and $\pi_1=\pi_2=200$.

Remark: For changing α_i one can follow the same reasoning. Increasing π_i leads to a smaller error in the position and a bigger error in the velocity.

E. Changing $\mathbf{b}^T_{mi}P$ has influence on $\dot{\vartheta}_i$ and $\mathbf{f}_i$ in $\mathbf{b}^T_{mi}P=[0.5\ 1]$ and $\mathbf{b}^T_{mi}P=[1\ 0.5]$. See figures 28 and 29.

$$u_i(t) = \vartheta^T_i \mathbf{v}_i + f_i(t) \text{ for Theorem 1 and 2.}$$

$$\dot{f}_i = -\pi_i \mathbf{b}^T_{mi} P_i \dot{\mathbf{e}}_i - \alpha_i \mathbf{b}^T_{mi} P_i \mathbf{e}_i$$

$$\dot{\vartheta}_i = -\sigma\Gamma_i \vartheta_i - \Gamma_i (\mathbf{b}^T_{mi} P_i \mathbf{e}_i) \mathbf{v}_i$$

→$\mathbf{b}^T_{mi}P = [0.5\ 1]$ the "weight" of the error in the velocity becomes bigger. This will lead to a smaller error in the velocity.

→$\mathbf{b}^T_{mi}P = [1\ 0.5]$ the "weight" of the error in the position becomes bigger. This will lead to a smaller error in the position.

5

CONCLUSIONS

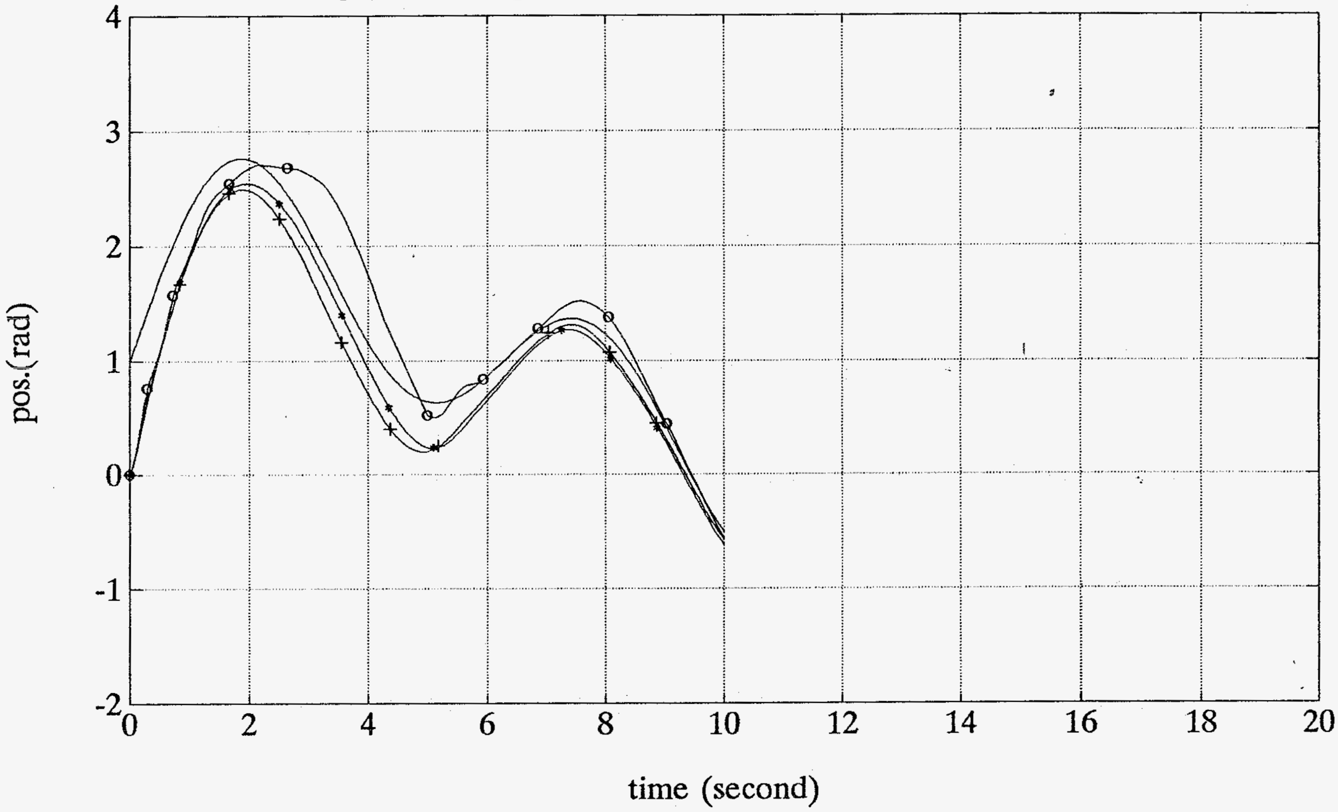
The influence on the system behavior after changing the parameters in the decentralized adaptive control schemes is well to forecast. The problems start when the 10 parameters in the control schemes have to be chosen to guide a new system, because changing the parameters can lead to a better system behavior but also to oscillating.

For $\omega=0.5$ the auxiliary signal leads to smoother control and smaller tracking error. But this is not true for bigger ω. At this point it is difficult to say if an auxiliary signal improves the performance for varying ω. I think further experiments are necessary to examine this.

LITERATURE

1. Lin Shi, Sunil K. Singh, Decentralized adaptive controllers based on the direct method of Lyapunov. Thayer school of engineering Dartmouth College, Hanover.
2. Gavel, Donald T. , Siljak, D.D. Decentralized adaptive control: structural conditions for stability. In: IEEE Trans. on Automatic Control, Vol-34, No.4, pp413-426, 1989.

figure 1. The position response for the first joint (w=0.5)



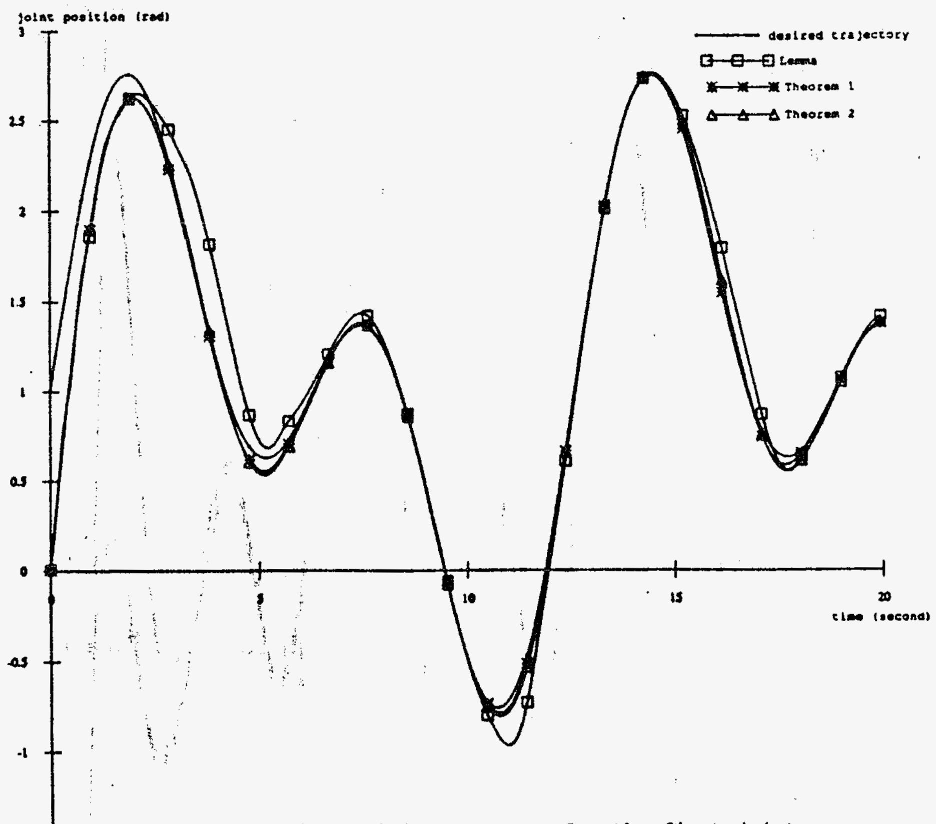
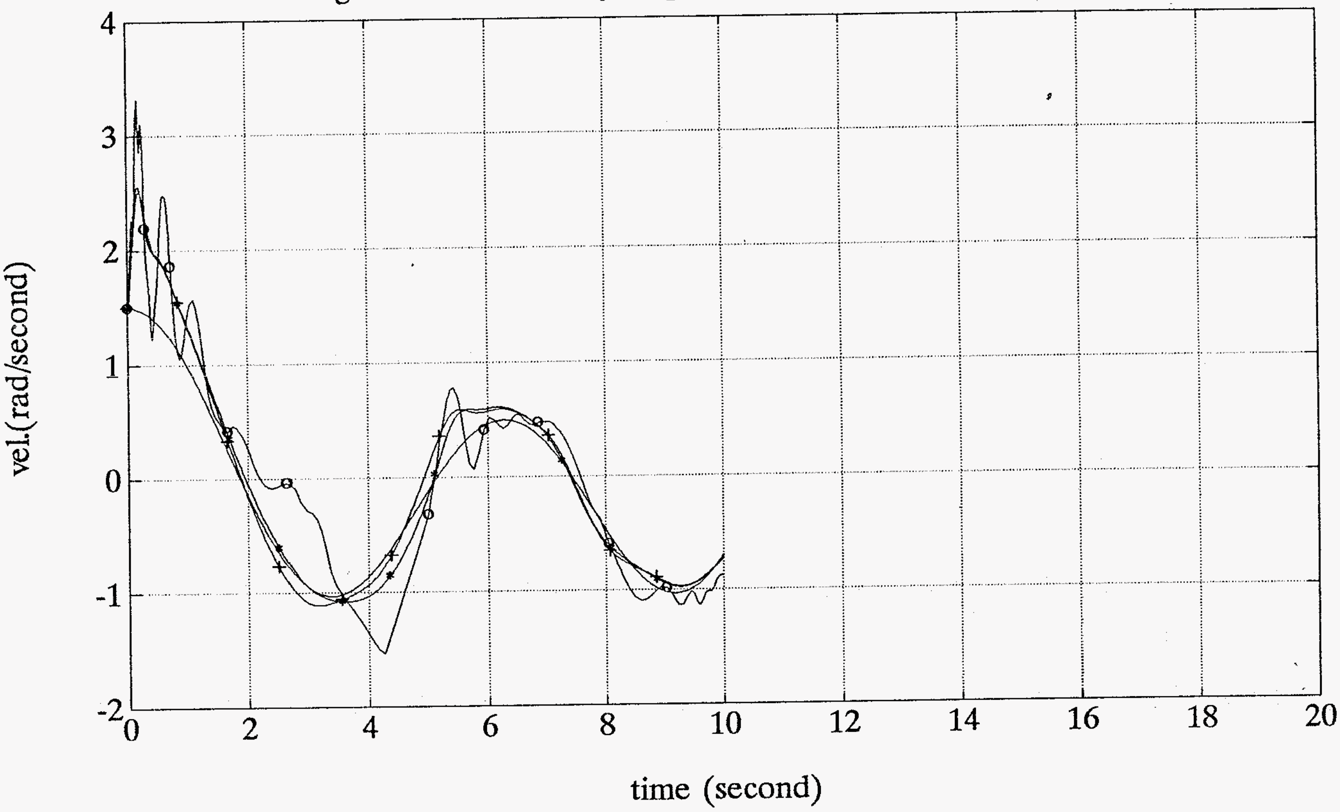


Figure 1. The position response for the first joint

figure 2. The velocity response for the first joint (w=0.5)



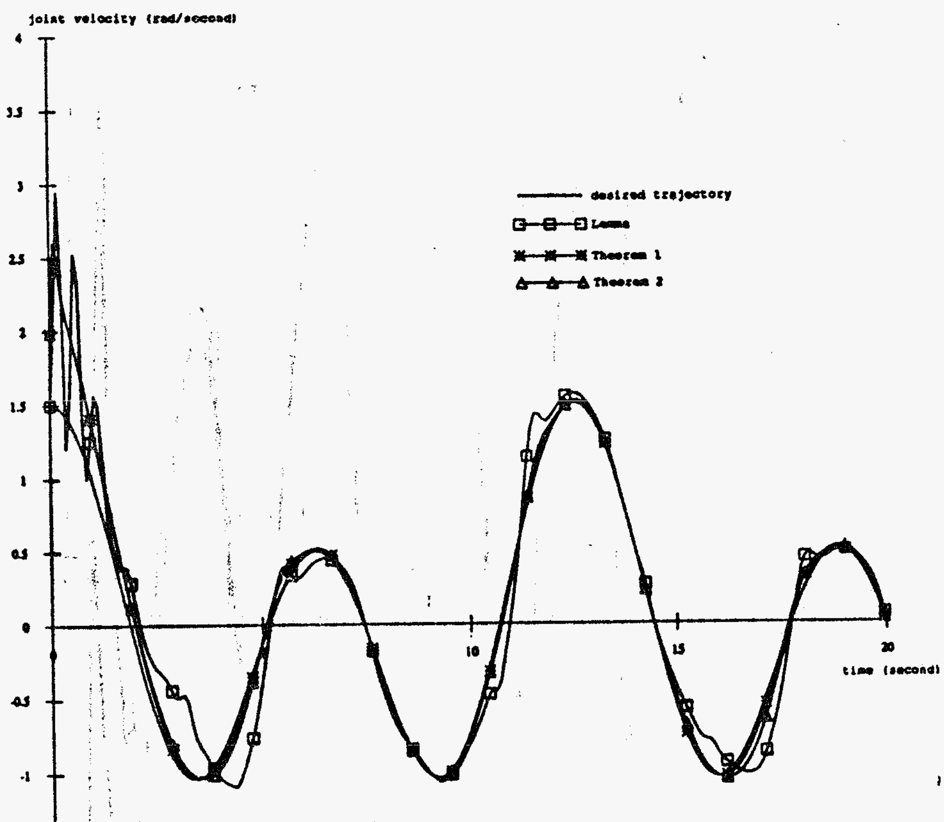
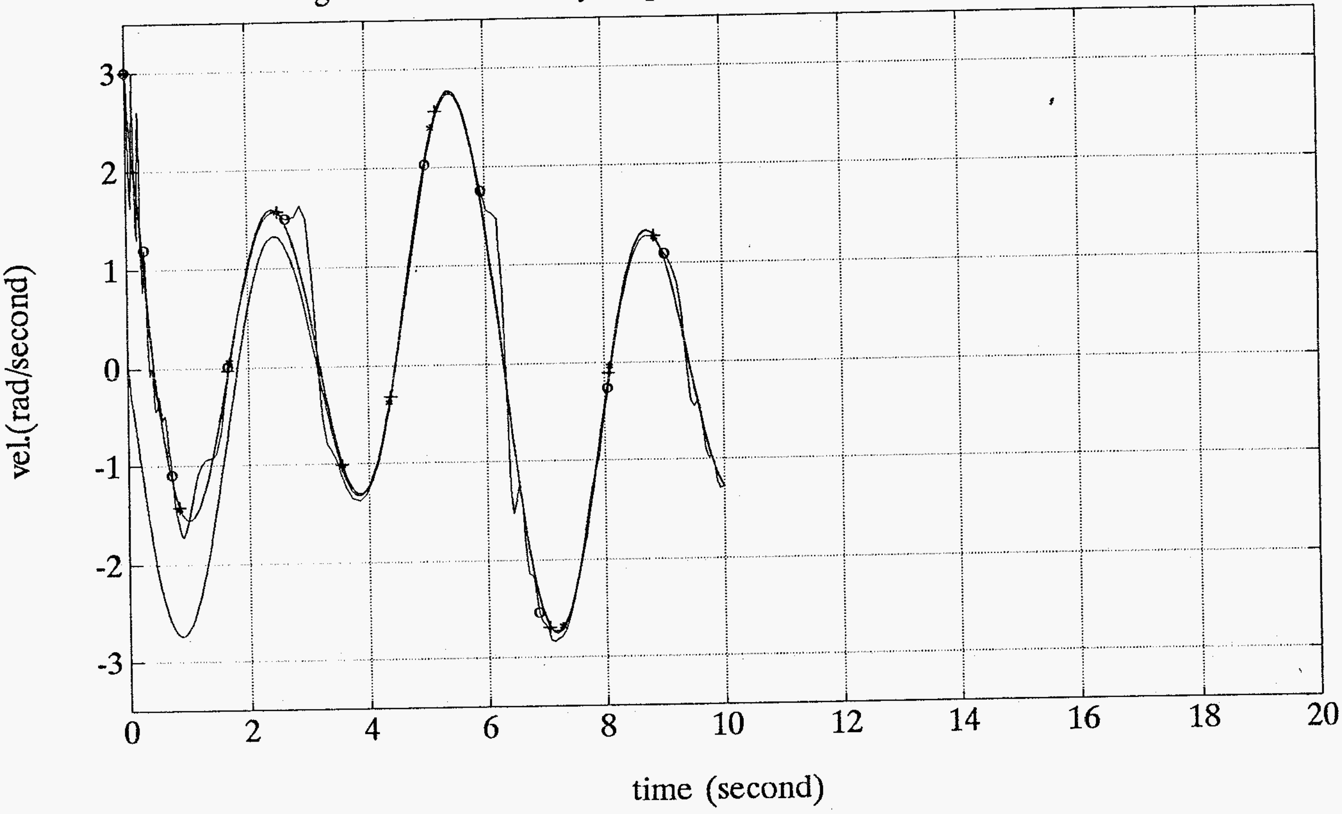


Figure 2. The velocity response for the first joint w = 0.5

figure 4. The velocity response for the second joint (w=0.5)



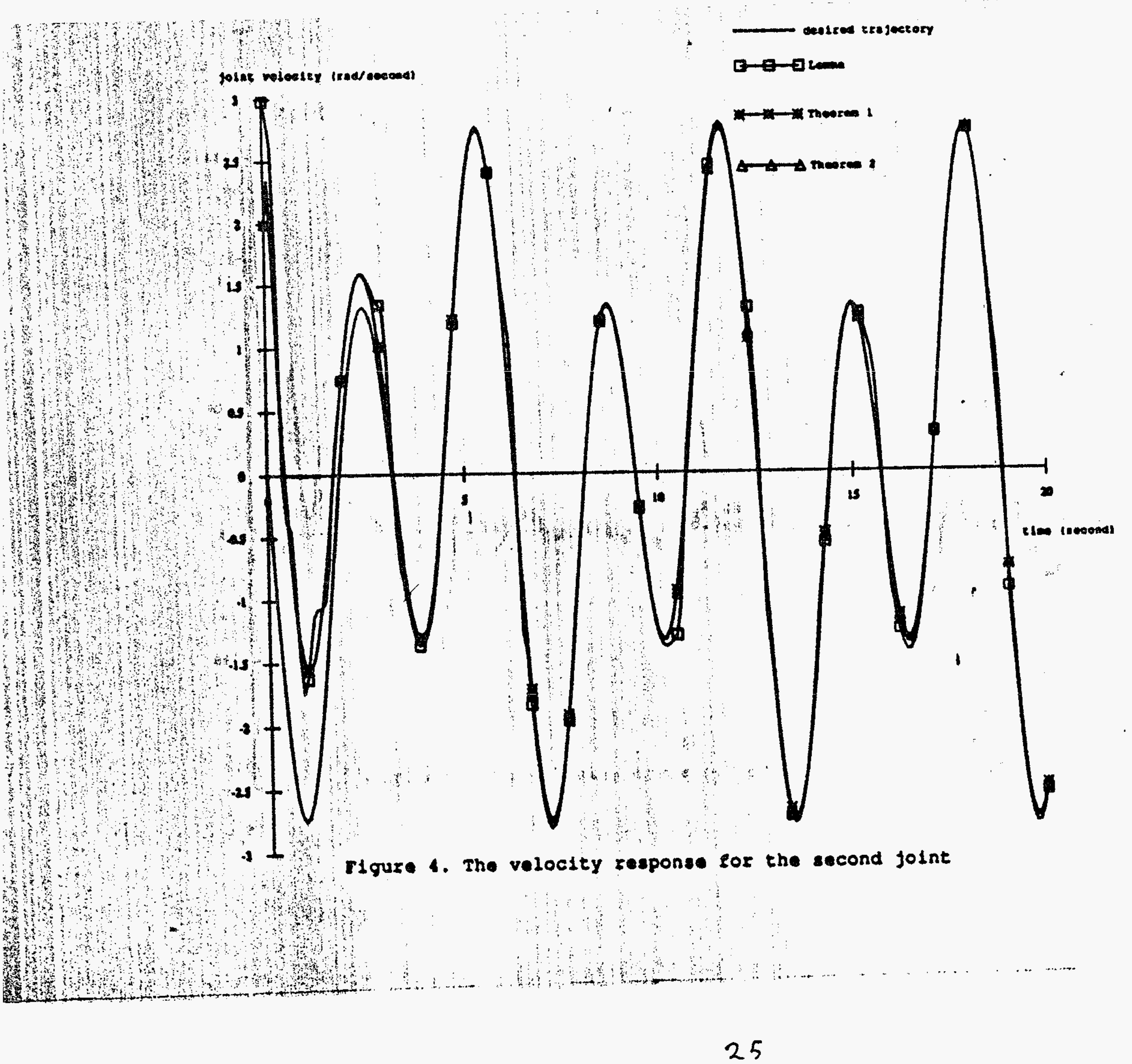
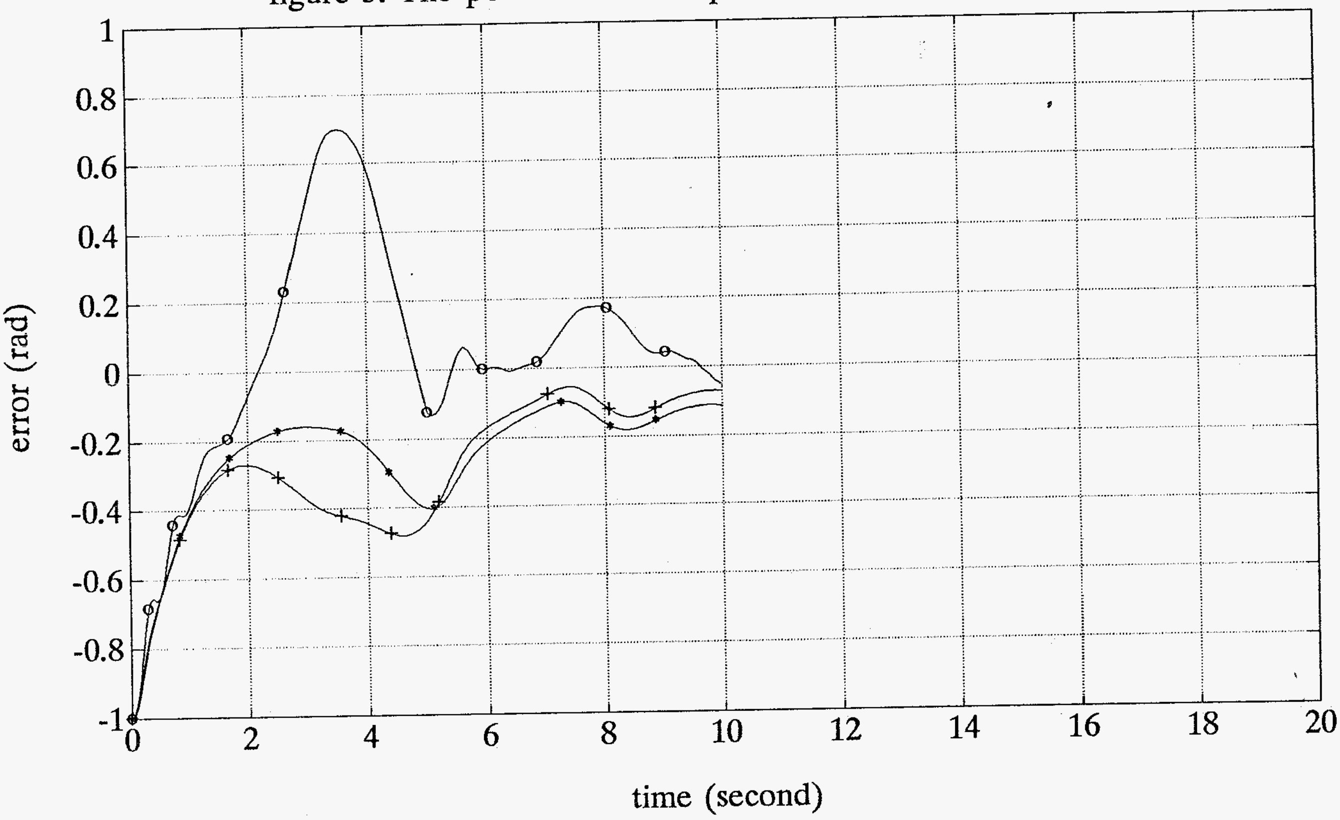


Figure 4. The velocity response for the second joint

figure 5. The position error response for the first joint (w=0.5)



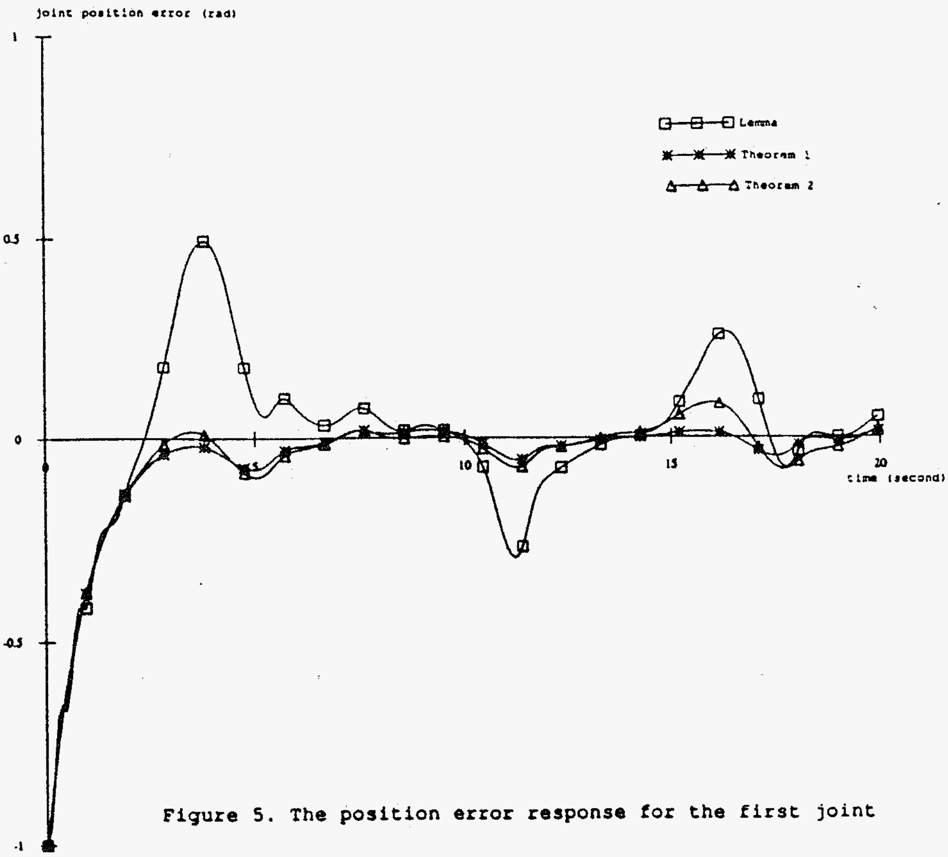
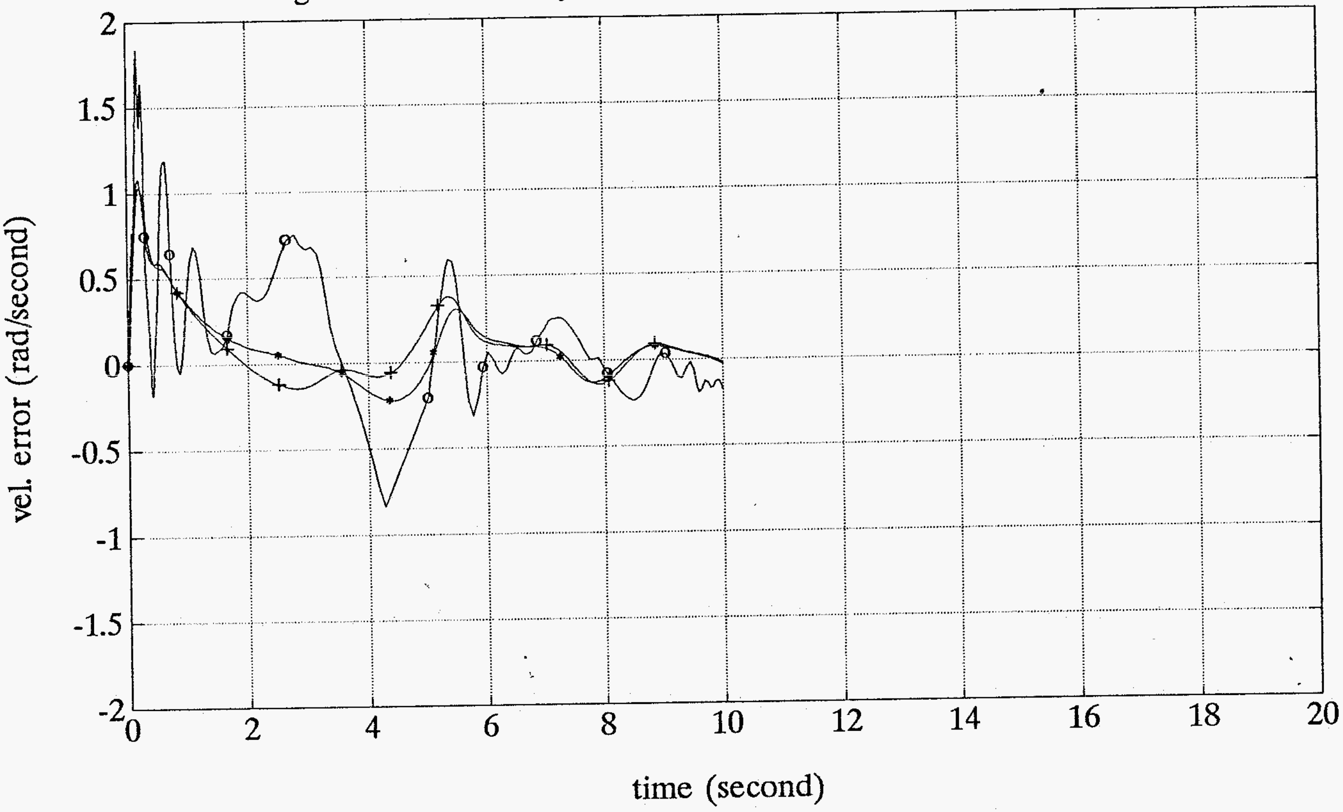


Figure 5. The position error response for the first joint

figure 6. The velocity error response for the first joint (w=0.5)



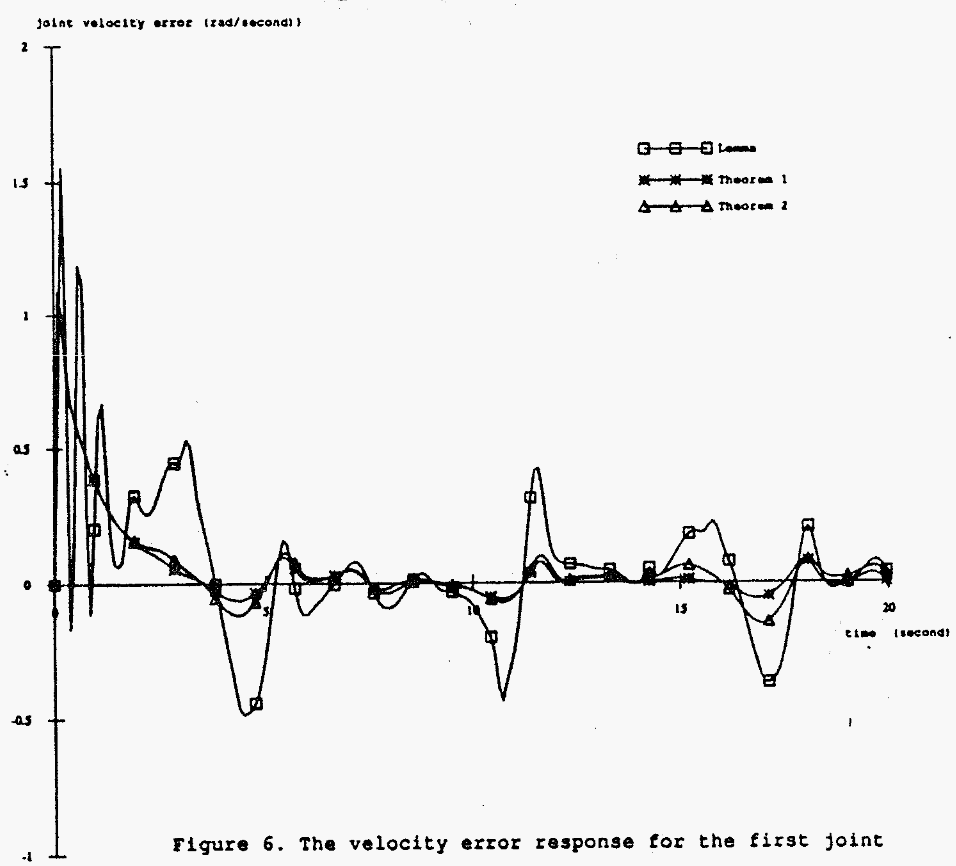
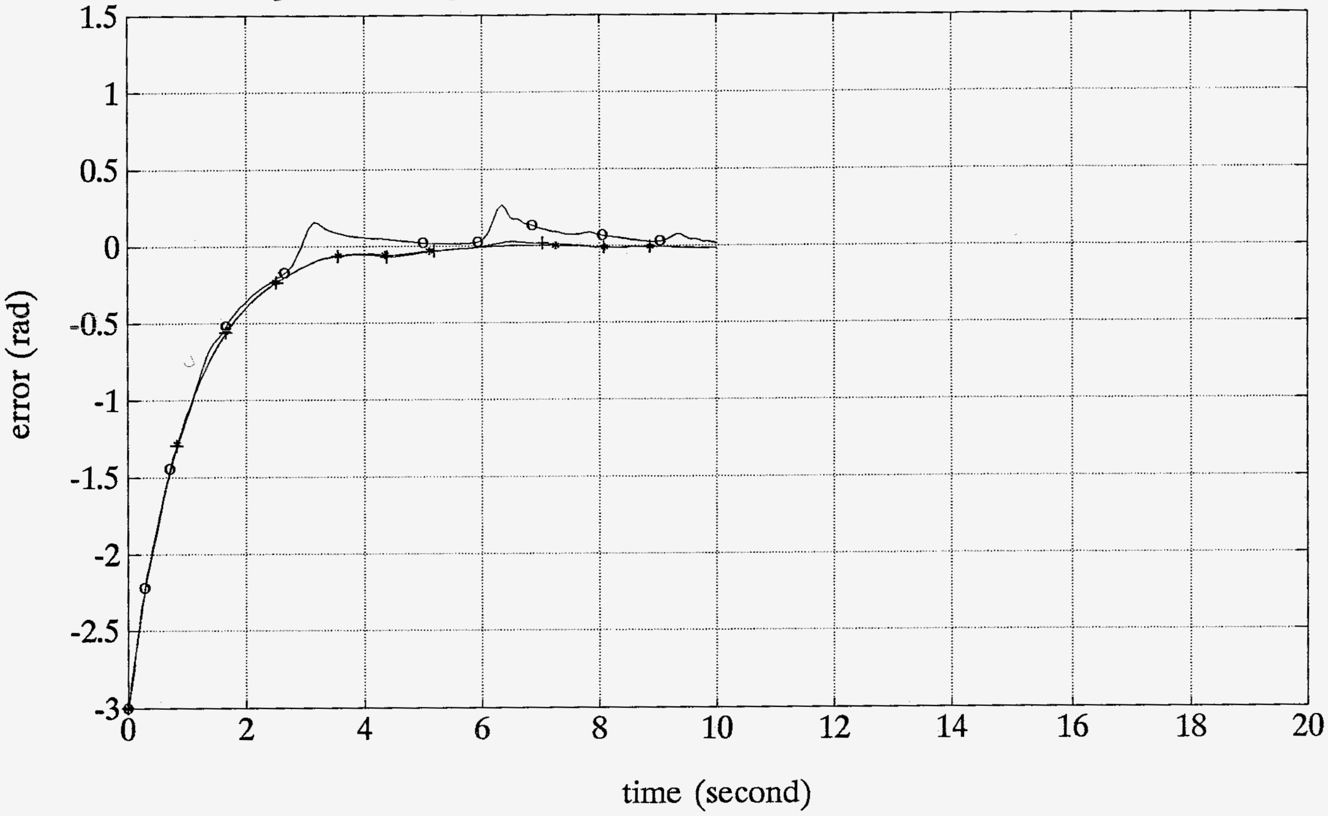


Figure 6. The velocity error response for the first joint

figure 7. The position error response for the second joint (w=0.5)



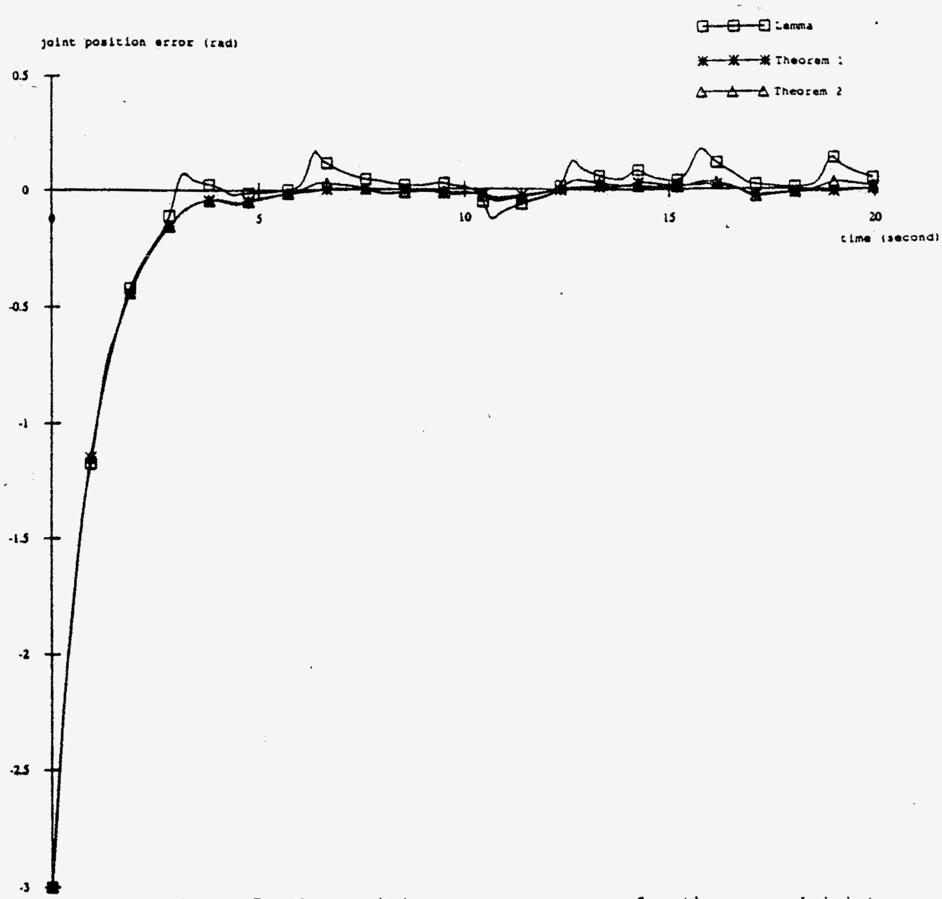
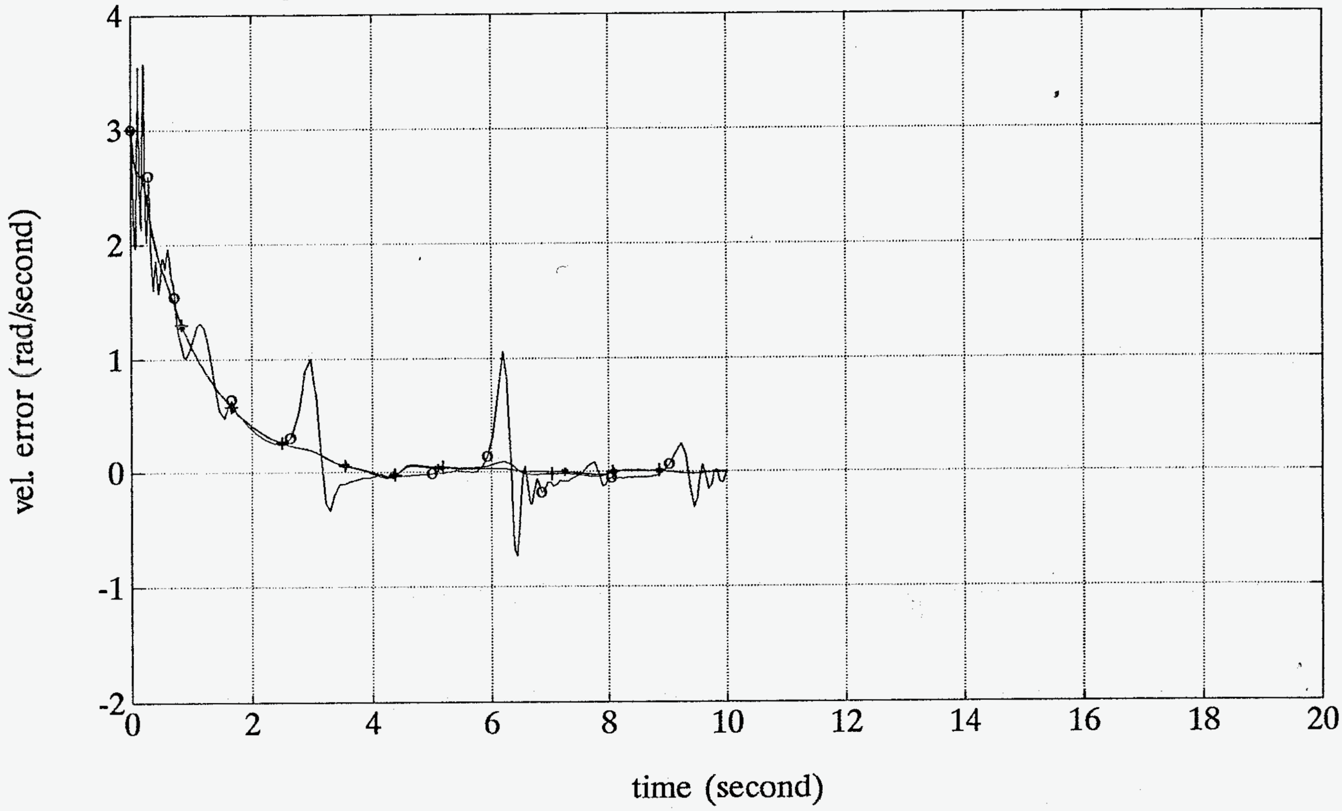


Figure 7. The position error response for the second joint

figure 8. The velocity error response for the second joint (w=0.5)



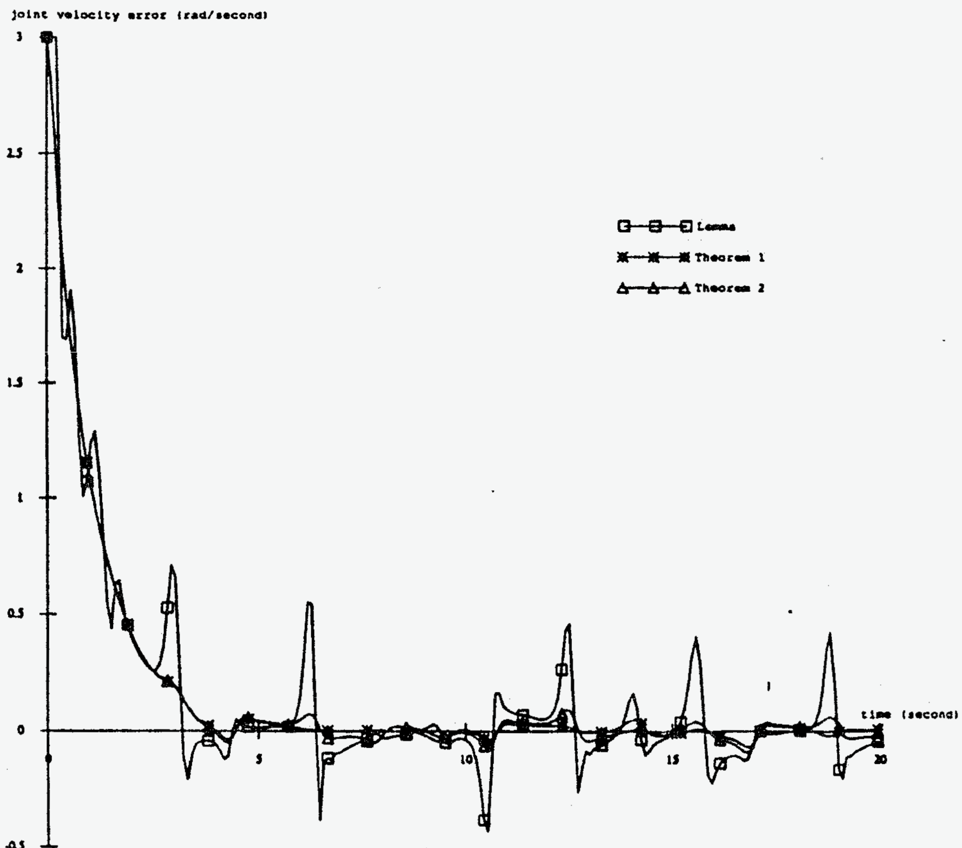
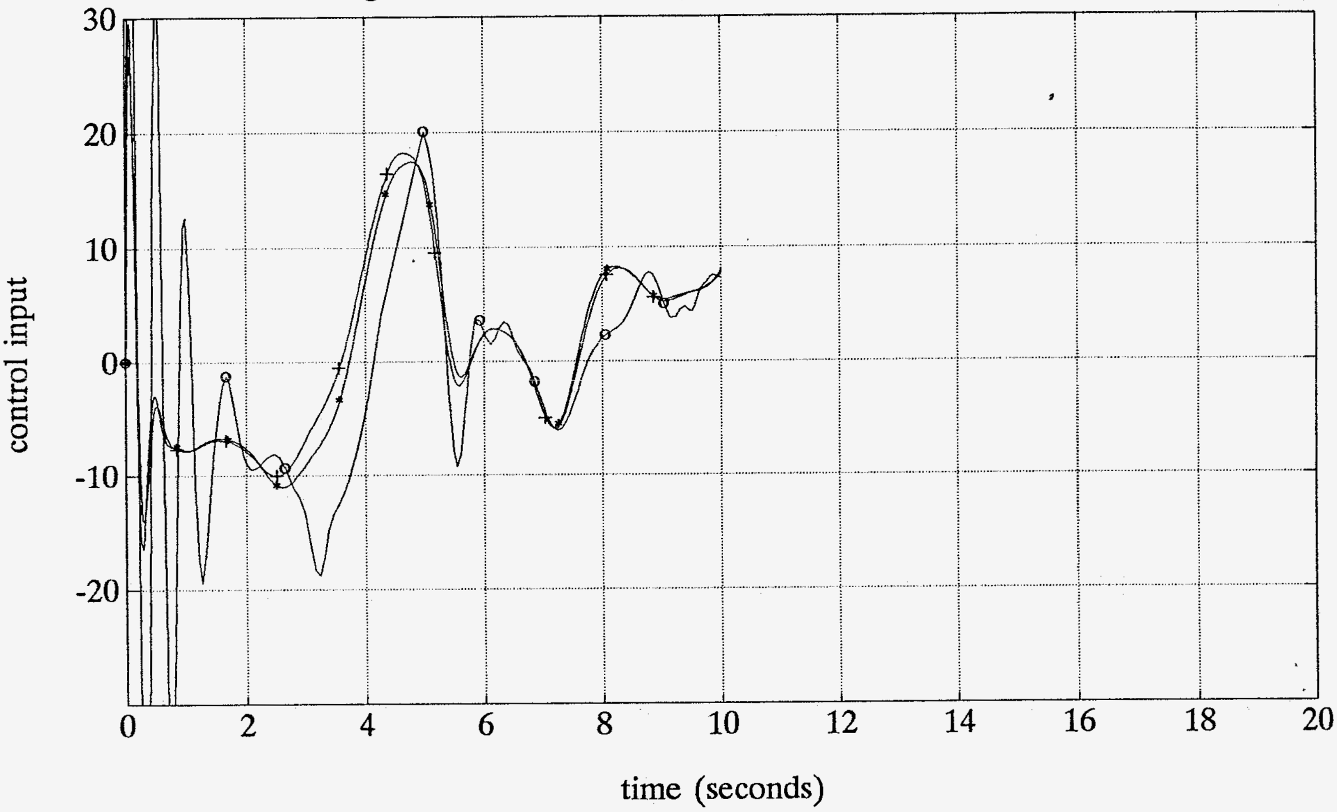


Figure 8. The velocity error response for the second joint

figure 9. The control input for the first joint (w=0.5)



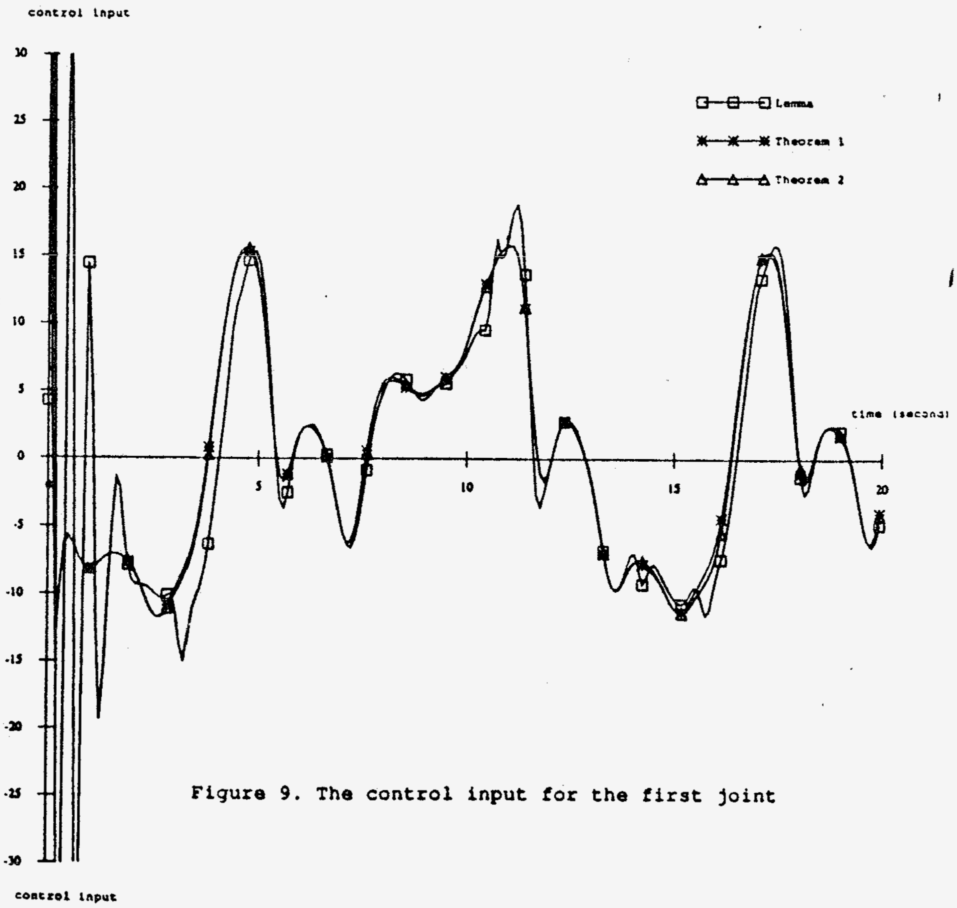
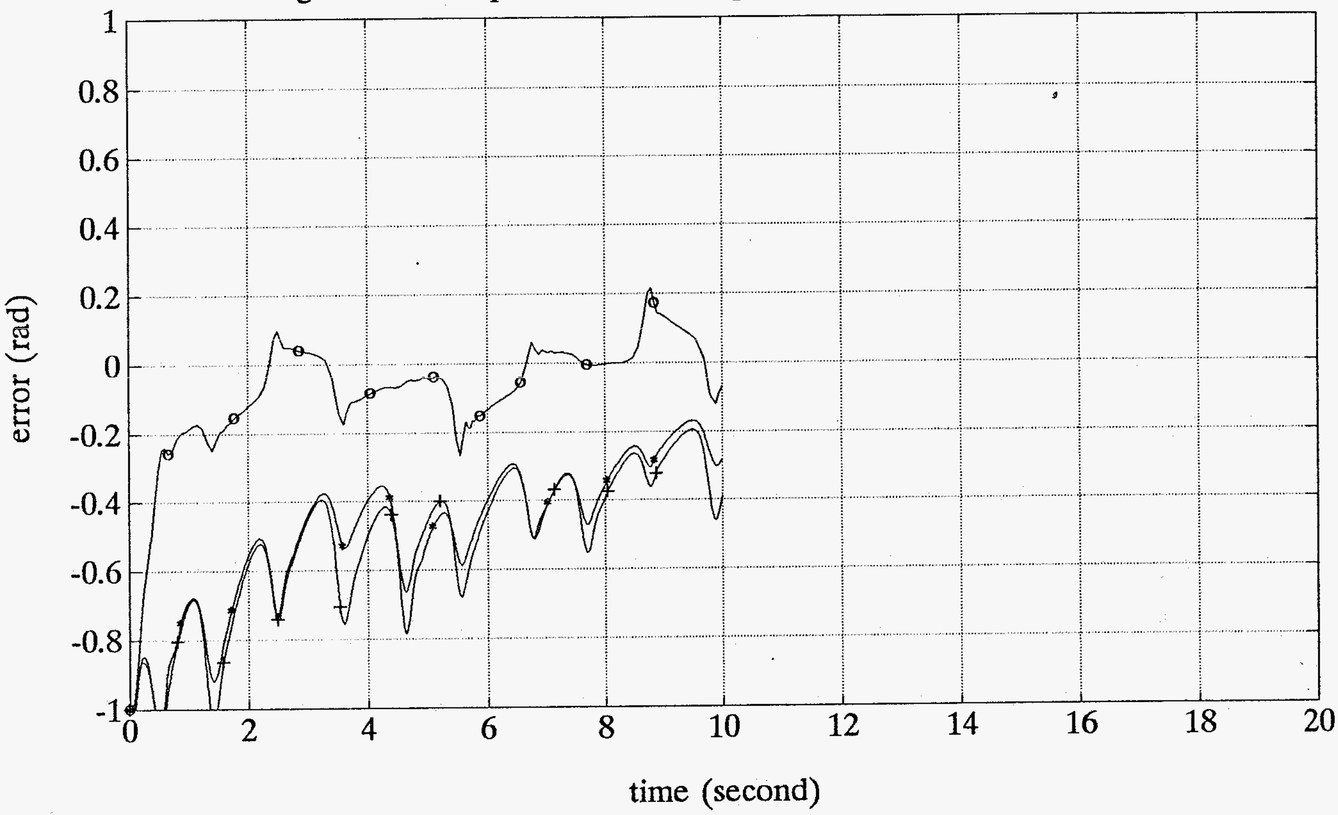


Figure 9. The control input for the first joint

figure 11. The position error response for the first joint (w=3)



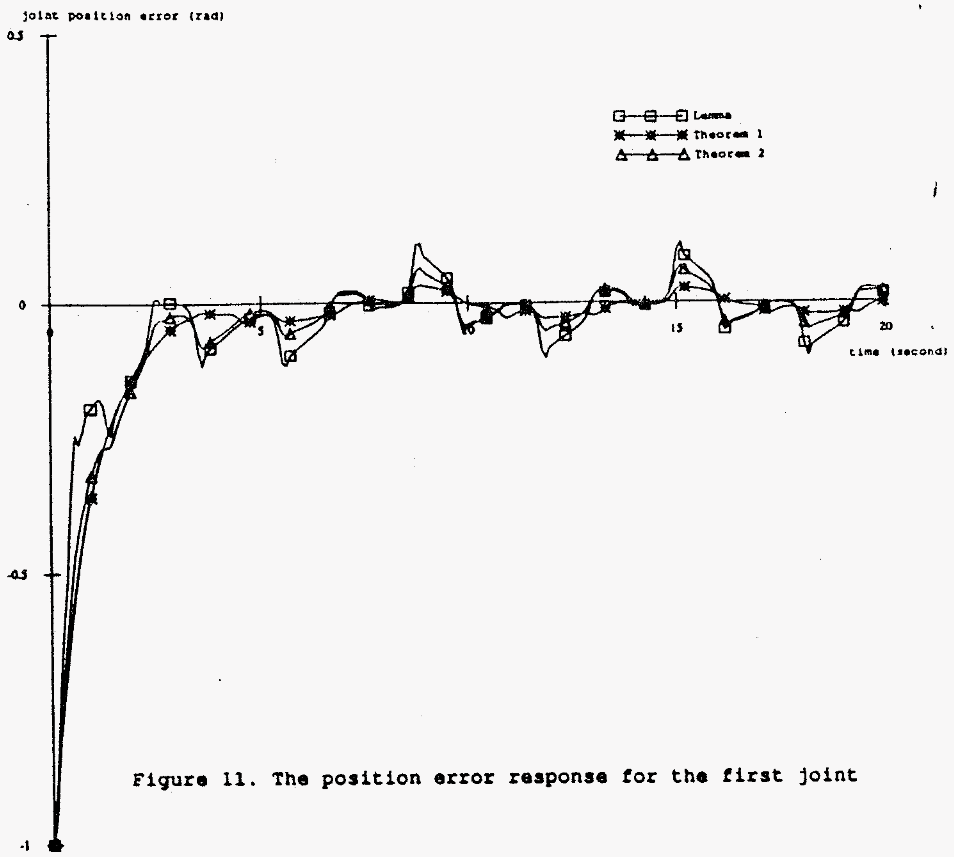
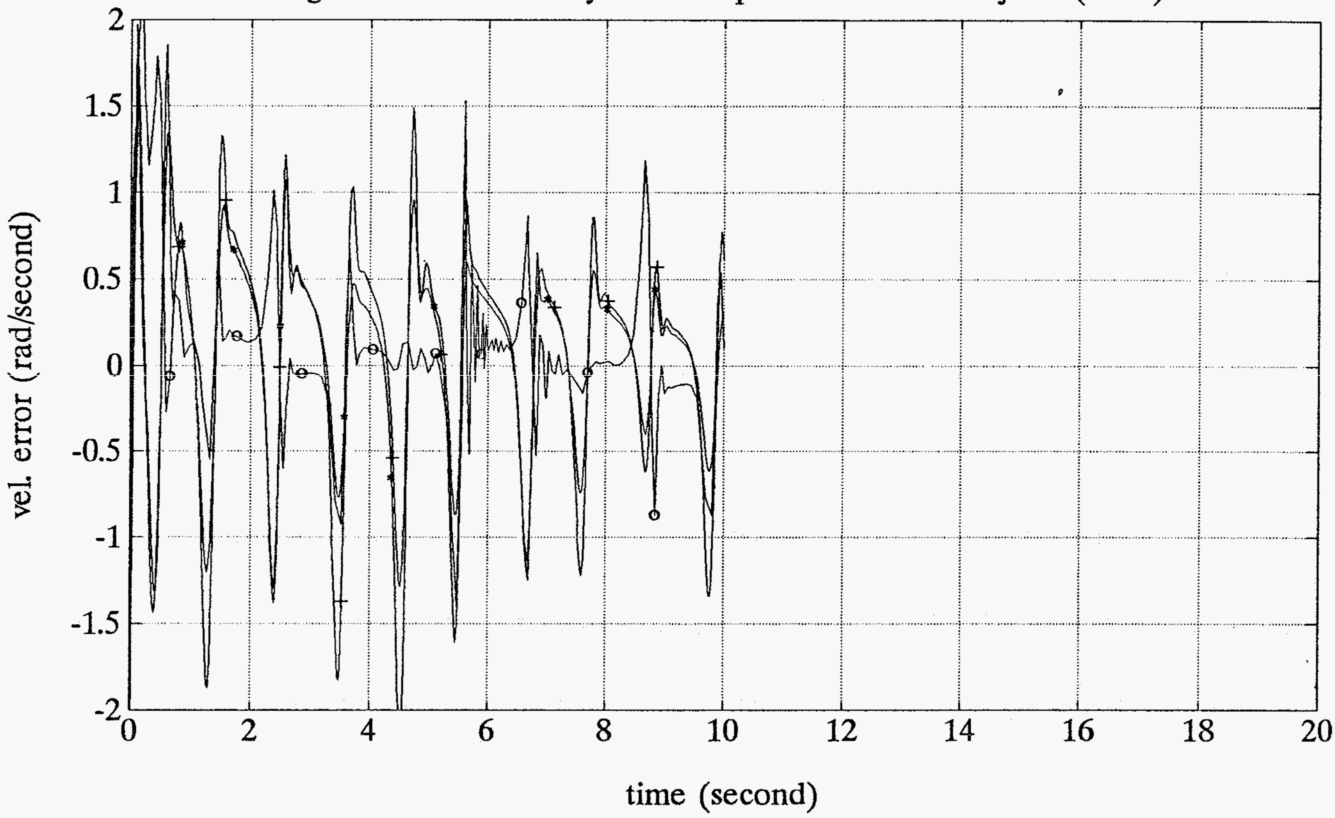


Figure 11. The position error response for the first joint

figure 12. The velocity error response for the first joint (w=3)



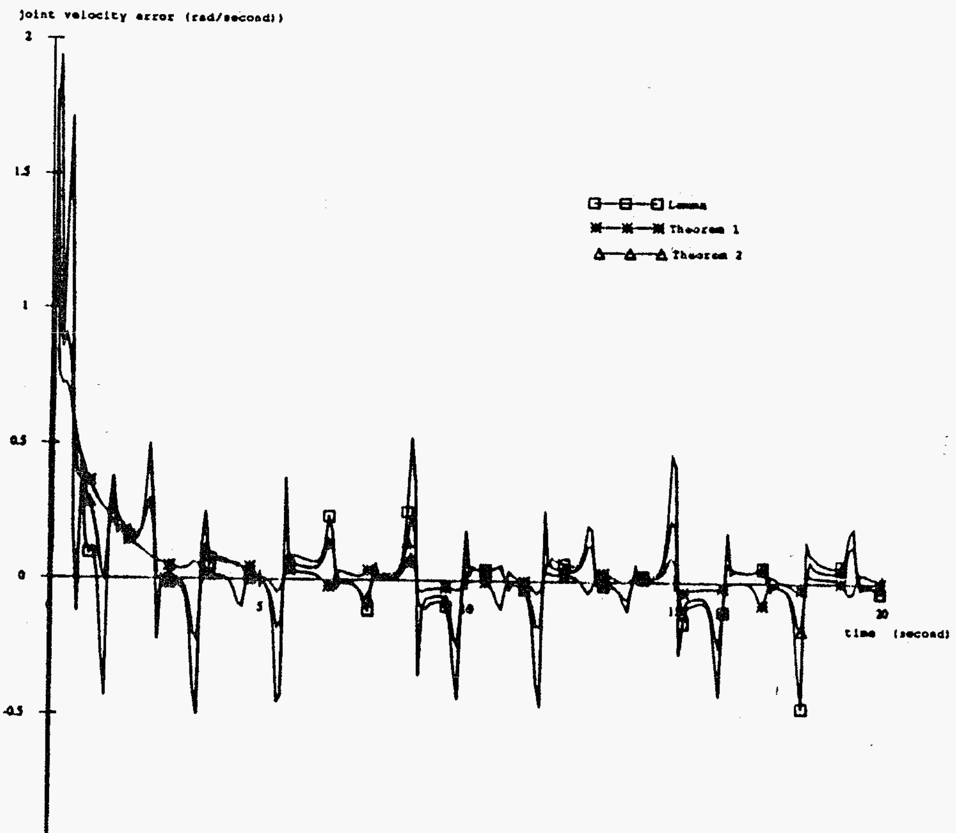
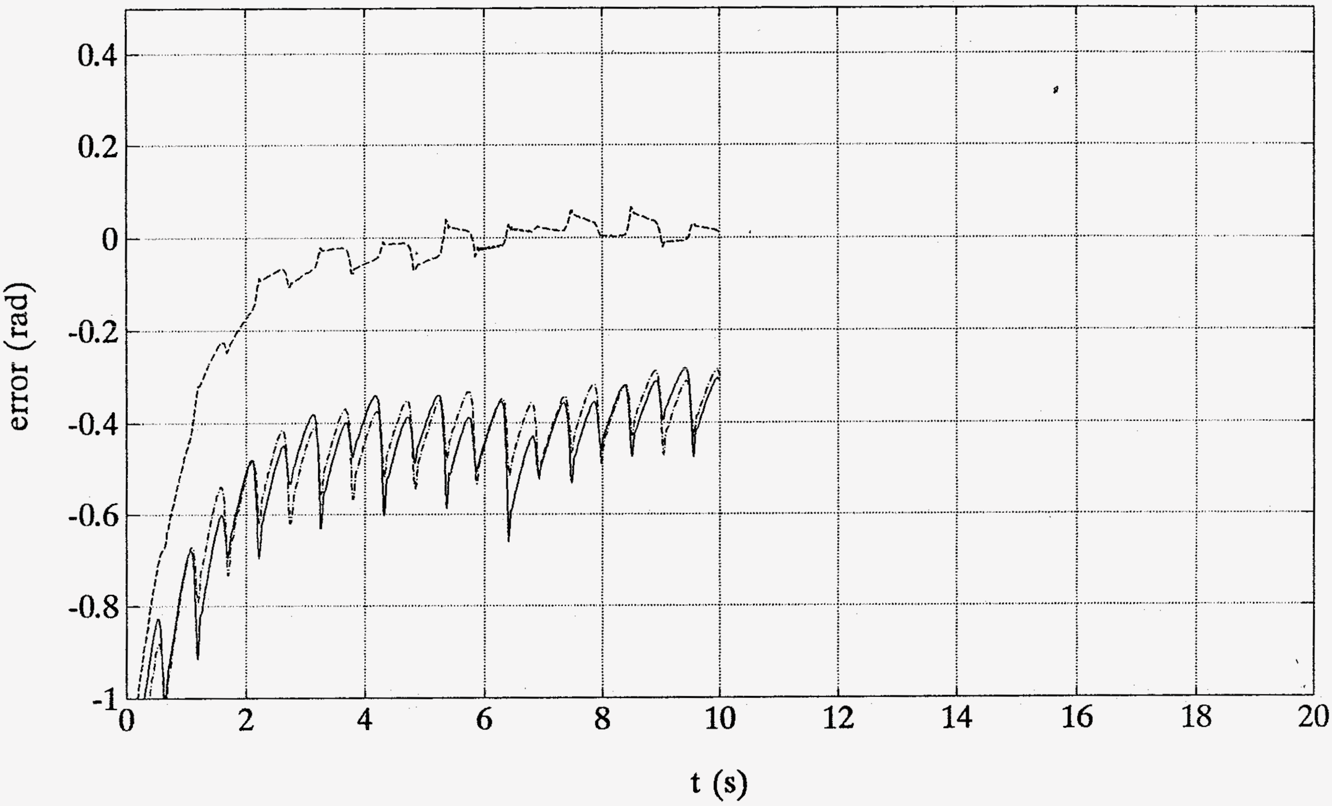


Figure 12. The velocity error response for the first joint

figure 15. The position error response for the first joint (w=6)



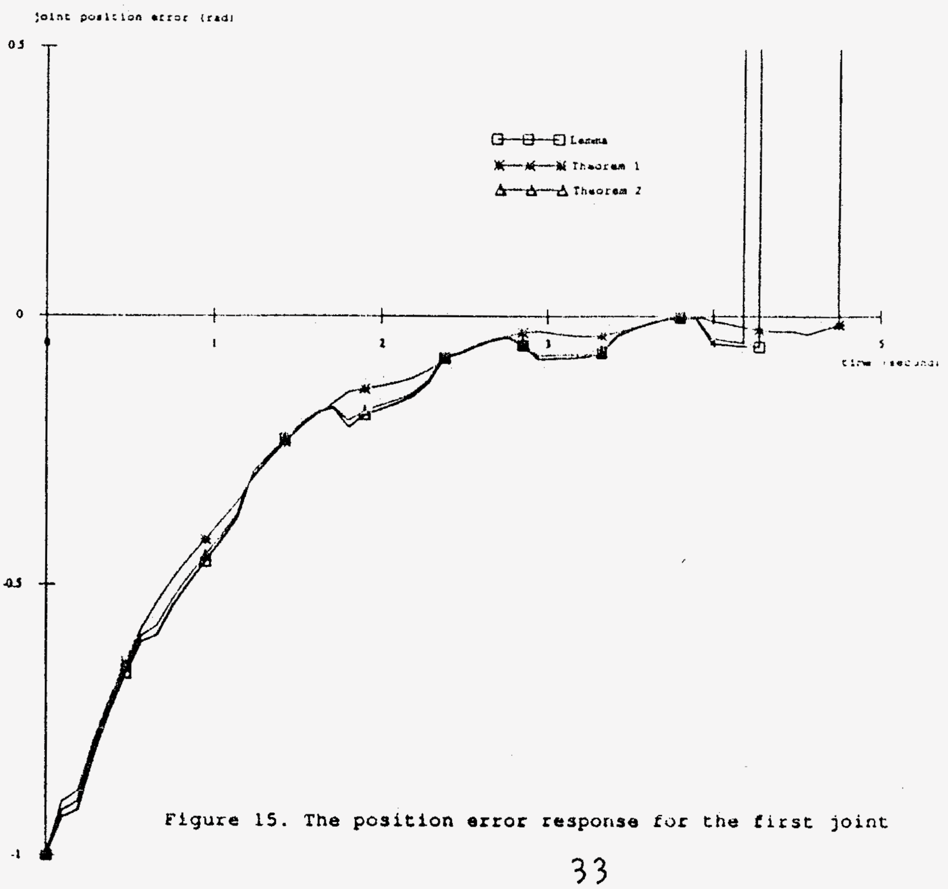


Figure 15. The position error response for the first joint

figure 16. The position error response for the first joint (w=0.5)

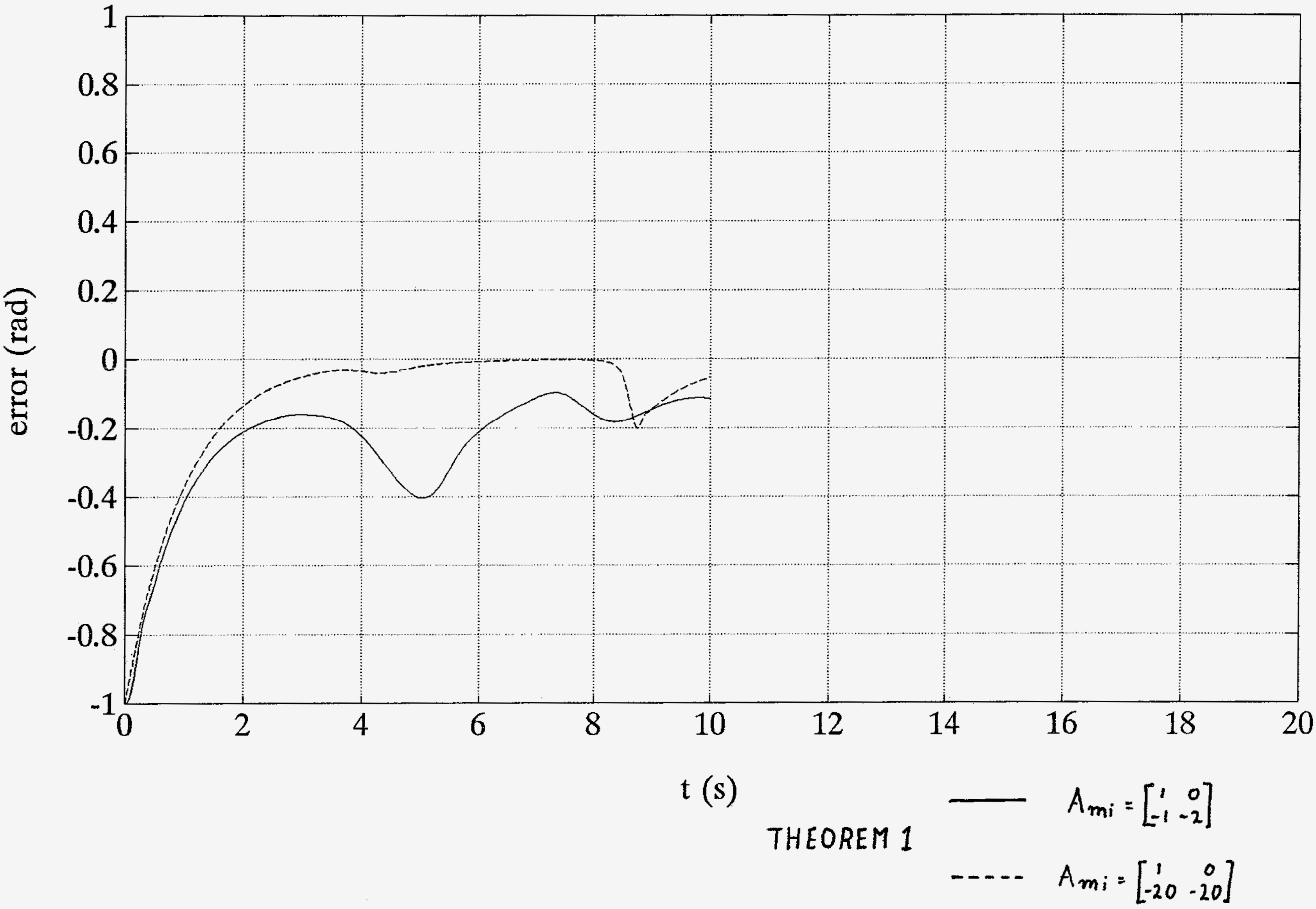


figure 17. The velocity error response for the first joint (w=0.5)

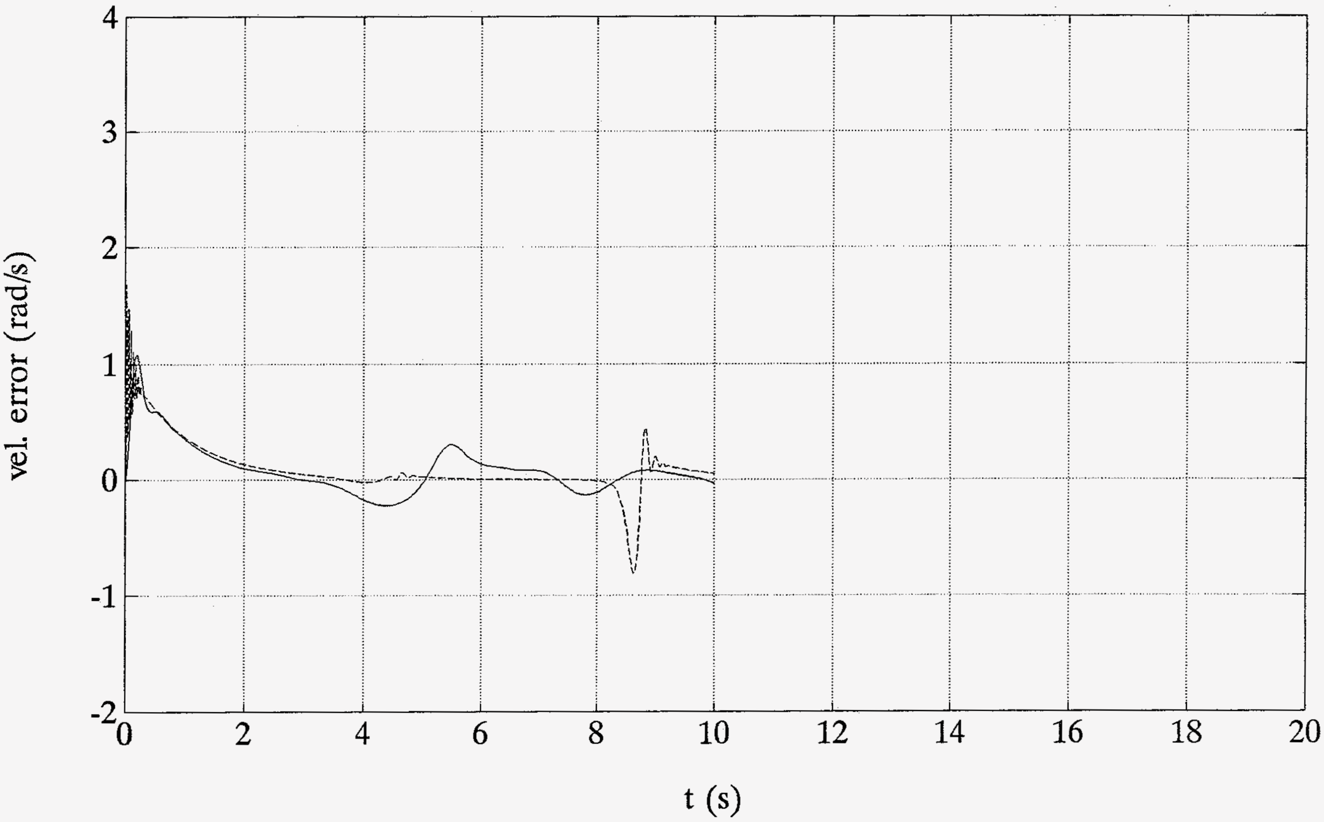
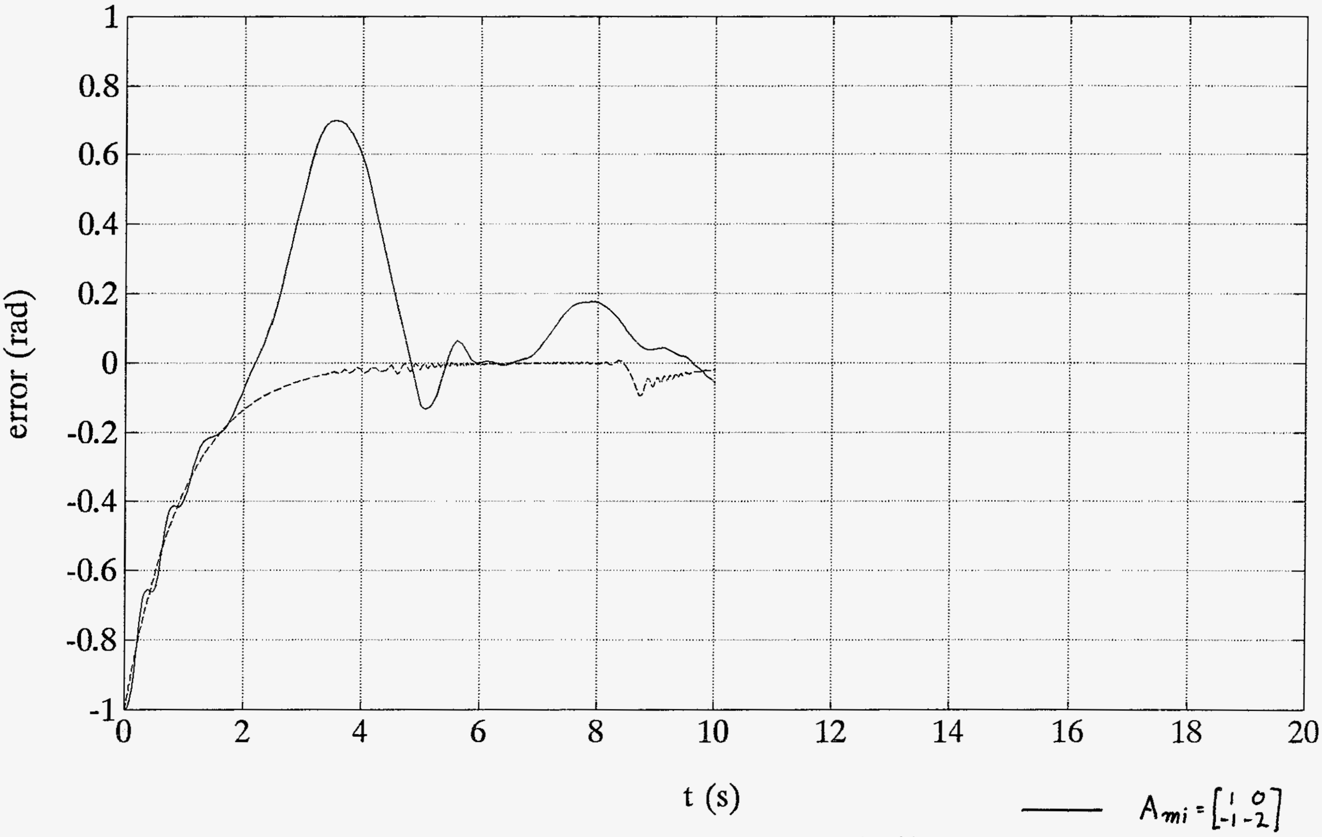


figure 18. The position error response for the first joint (w=0.5)



—— $A_{mi} = \begin{bmatrix} 1 & 0 \\ -1 & -2 \end{bmatrix}$

LEMMA

------ $A_{mi} = \begin{bmatrix} -1 & 0 \\ -20 & -20 \end{bmatrix}$

figure 19. The velocity error response for the first joint (w=0.5)

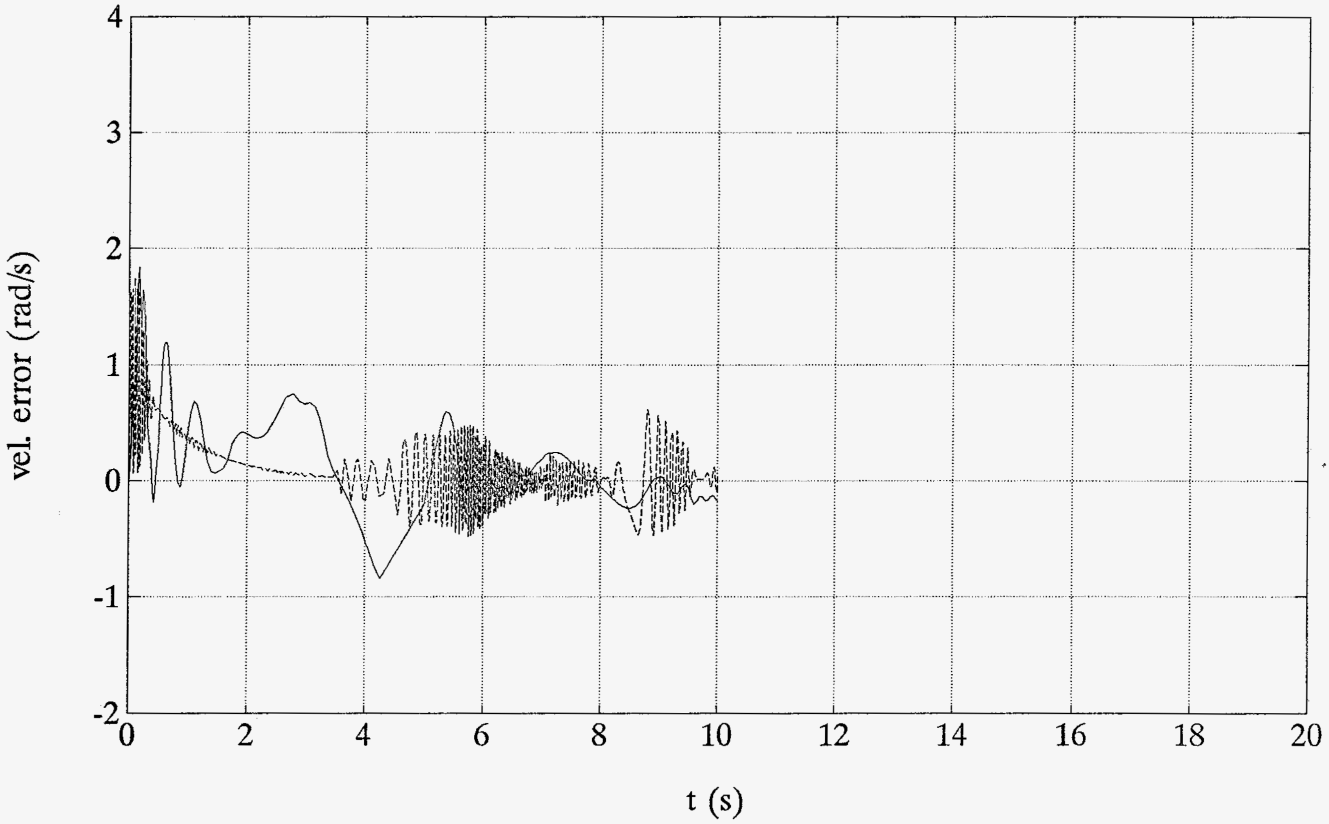


figure 20. The position error response for the first joint (w=0.5)

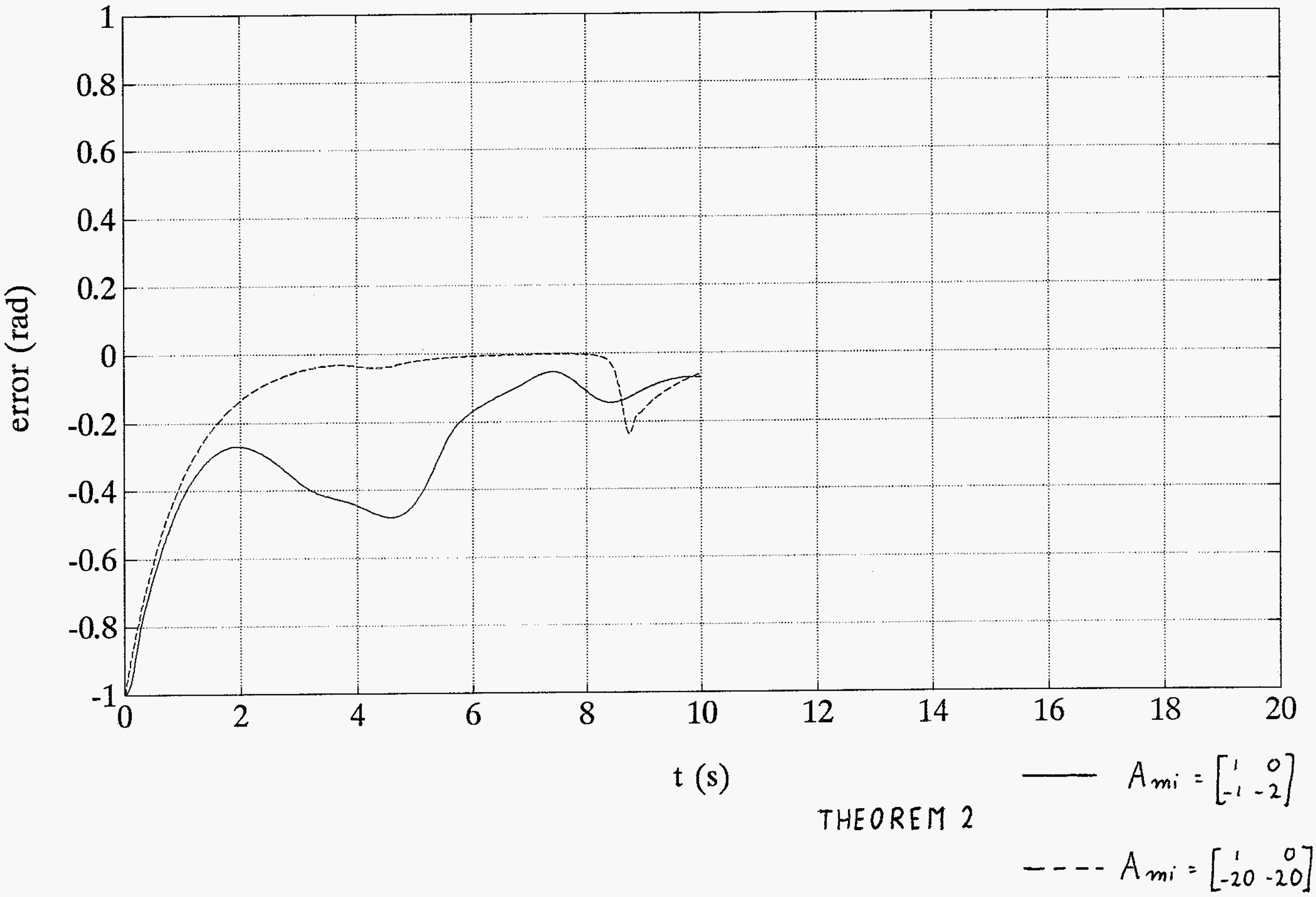


figure 21. The velocity error response for the first joint (w=0.5)

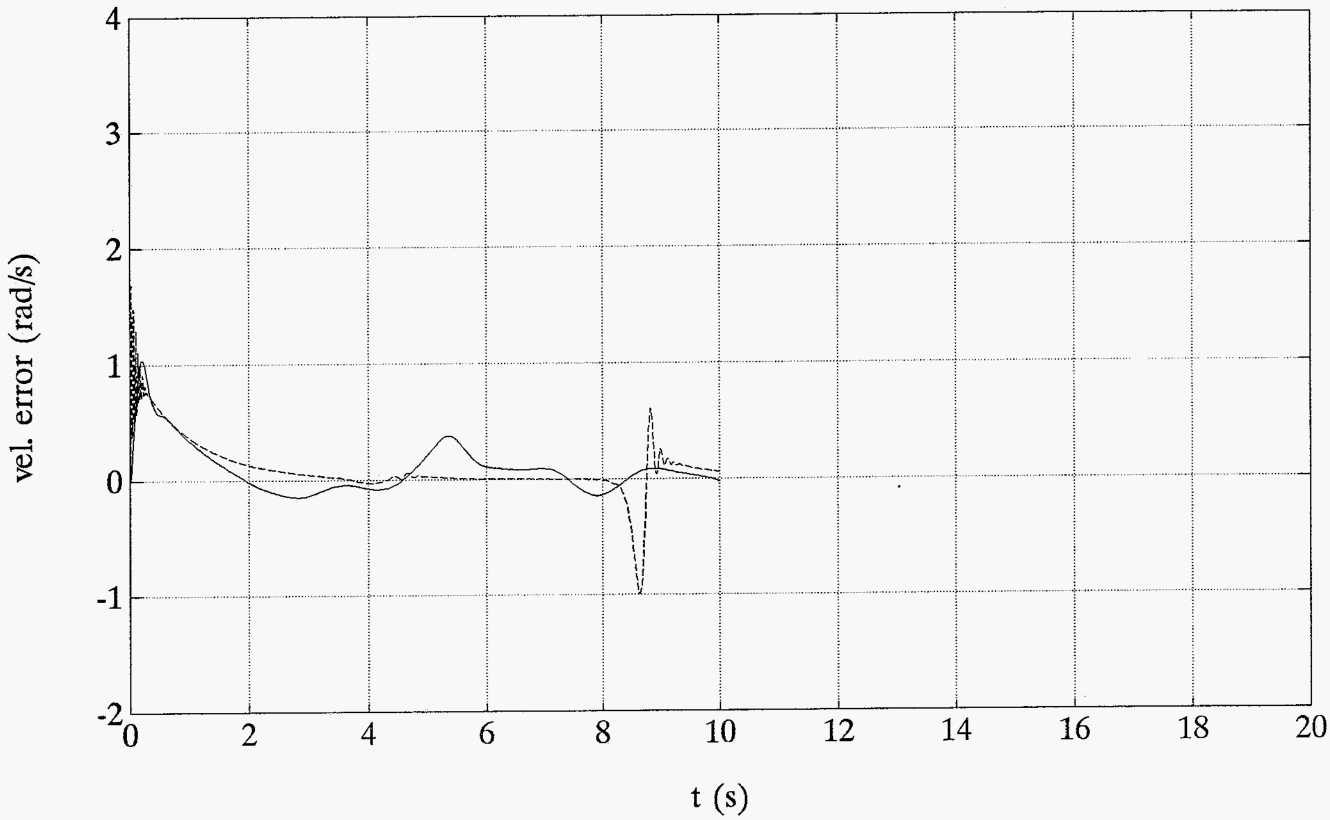


figure 22. The position error response for the first joint (w=0.5)

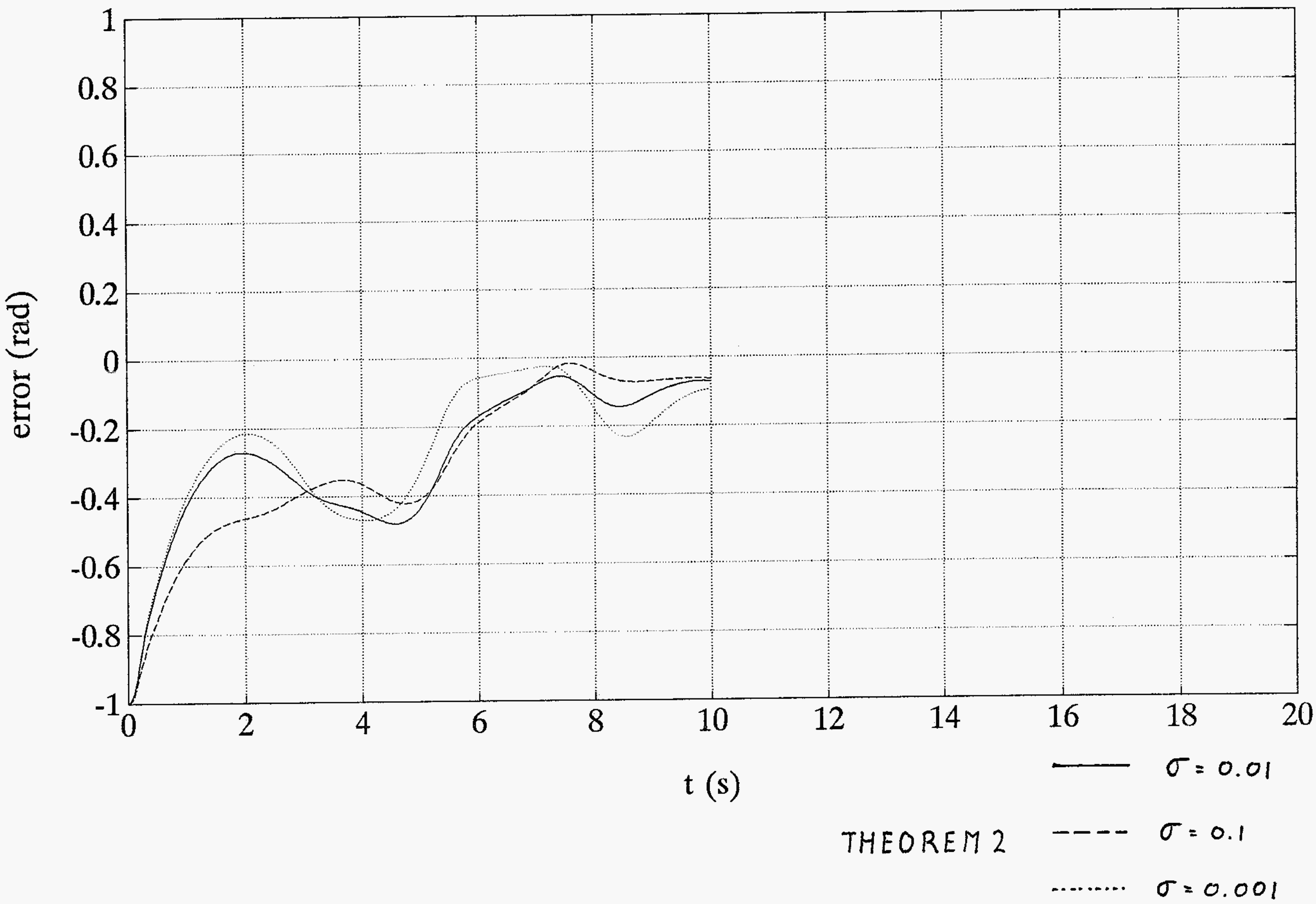


figure 23. The velocity error response for the first joint (w=0.5)

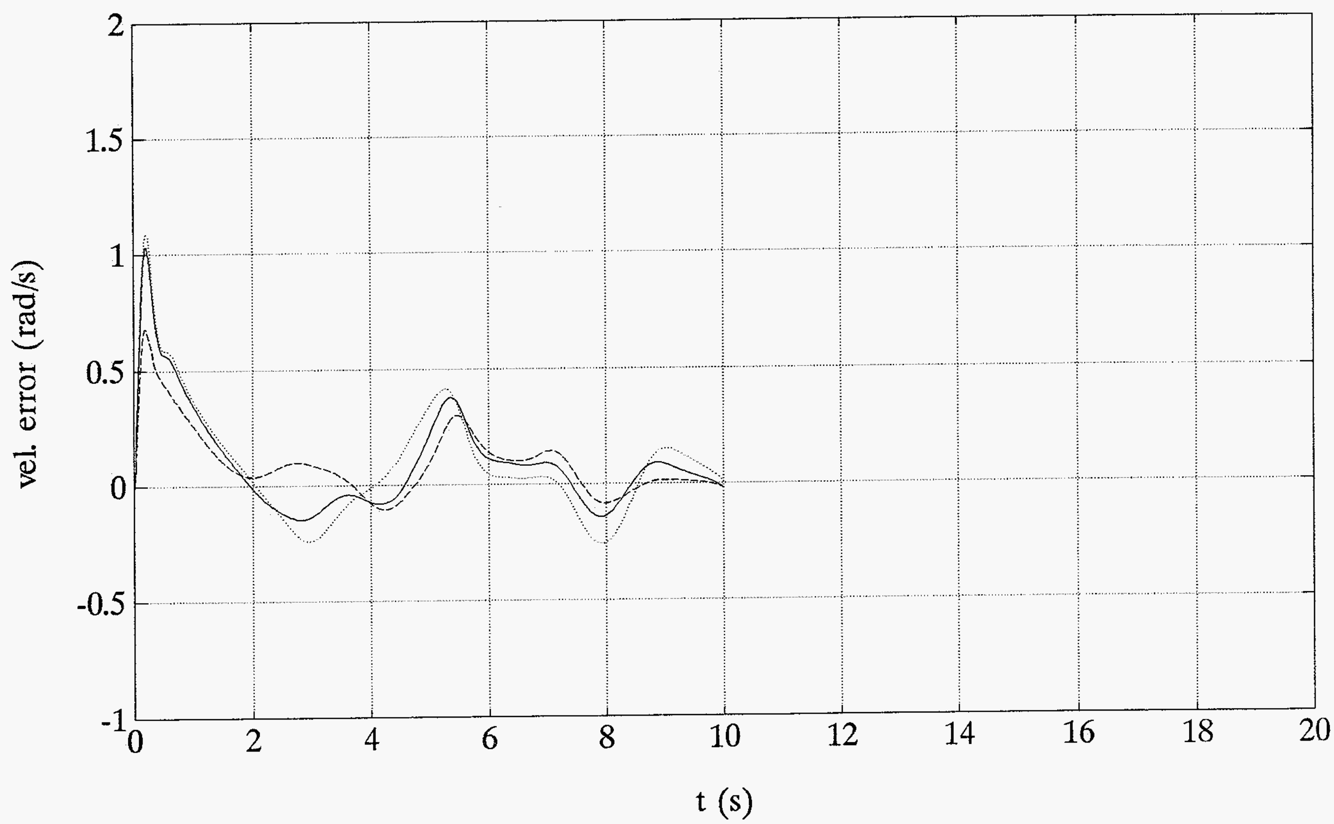


figure 24. The position error response for the first joint (w=0.5)

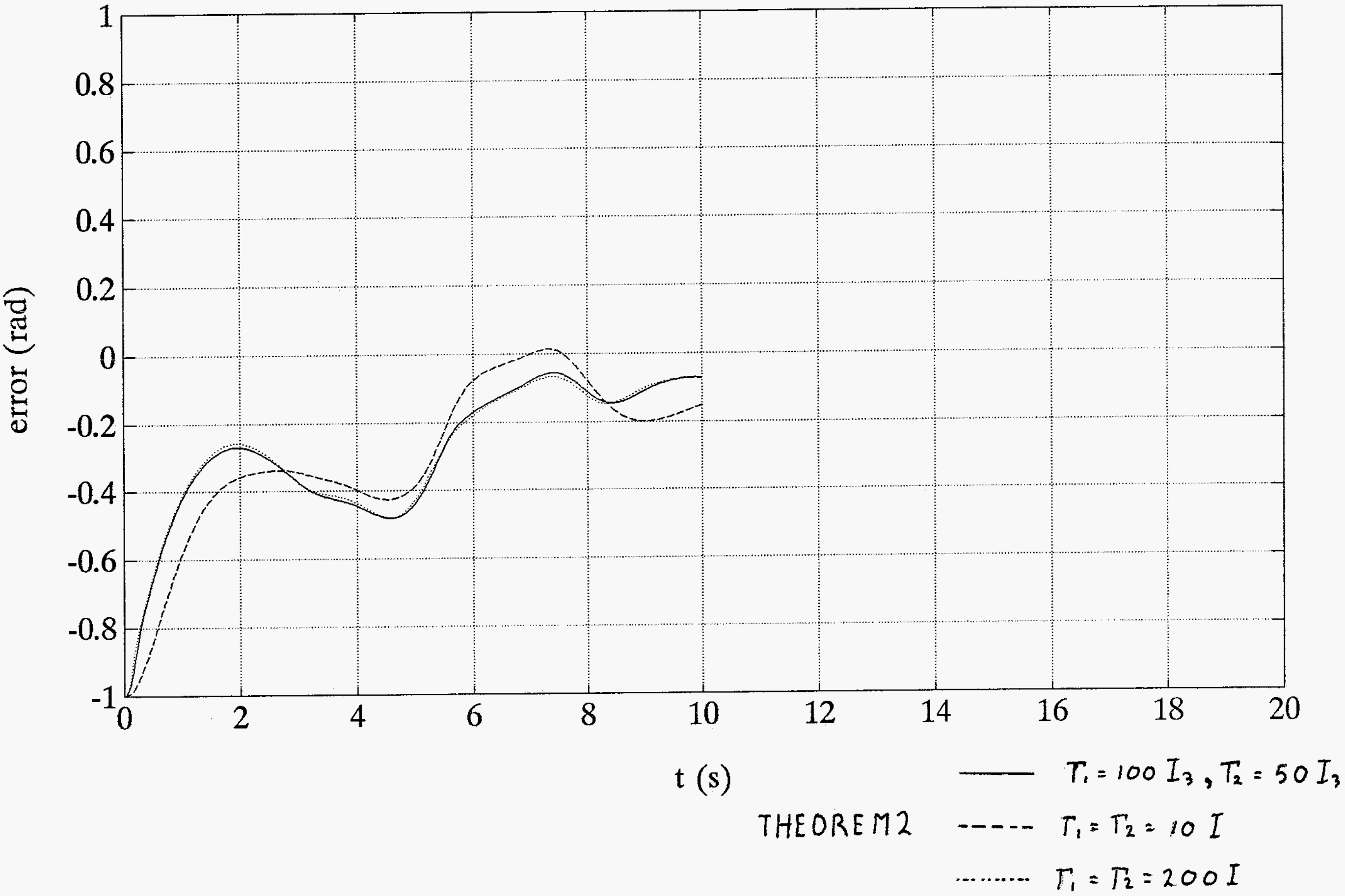


figure 25. The velocity error response for the first joint (w=0.5)

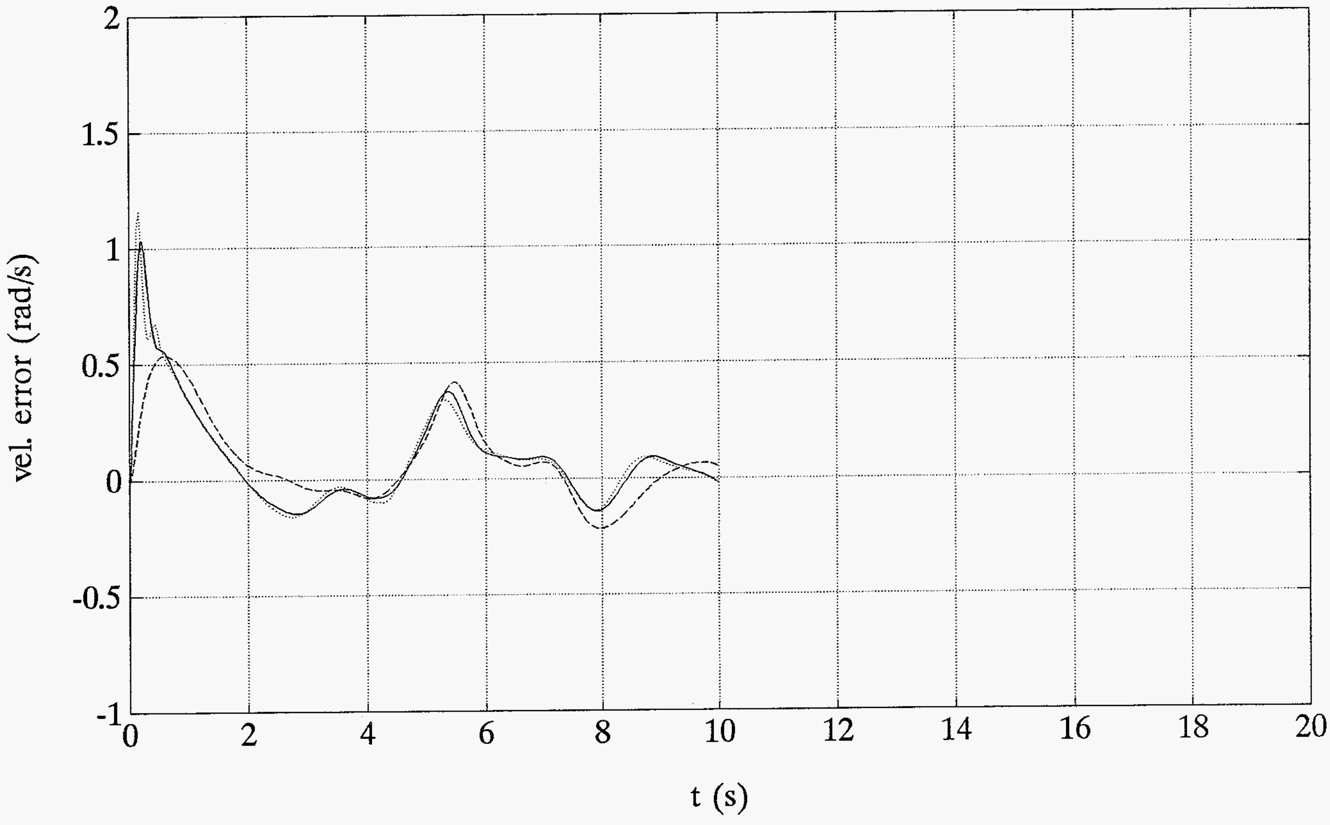


figure 26. The position error response for the first joint (w=0.5)

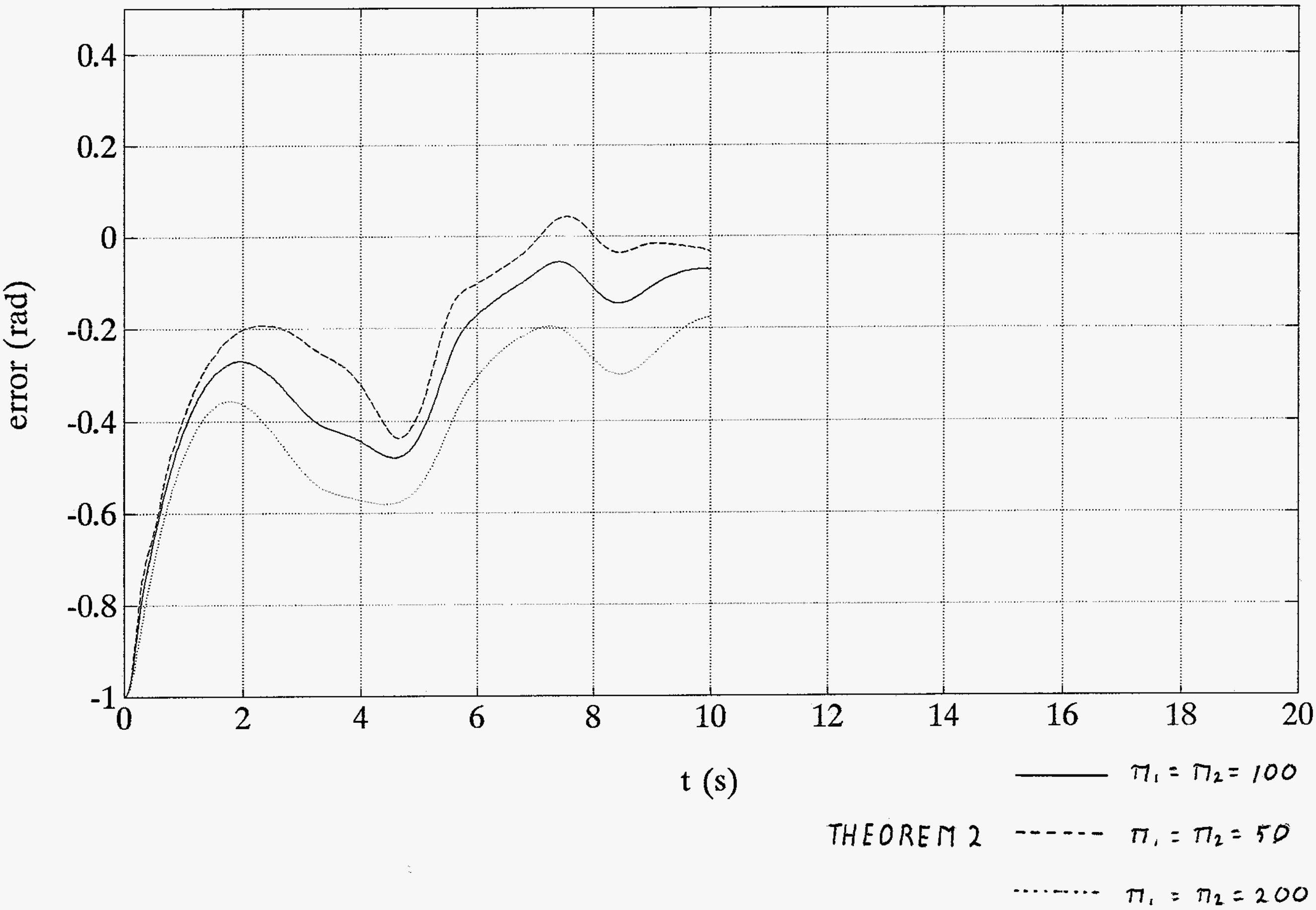


figure 27. The velocity error response for the first joint (w=0.5)

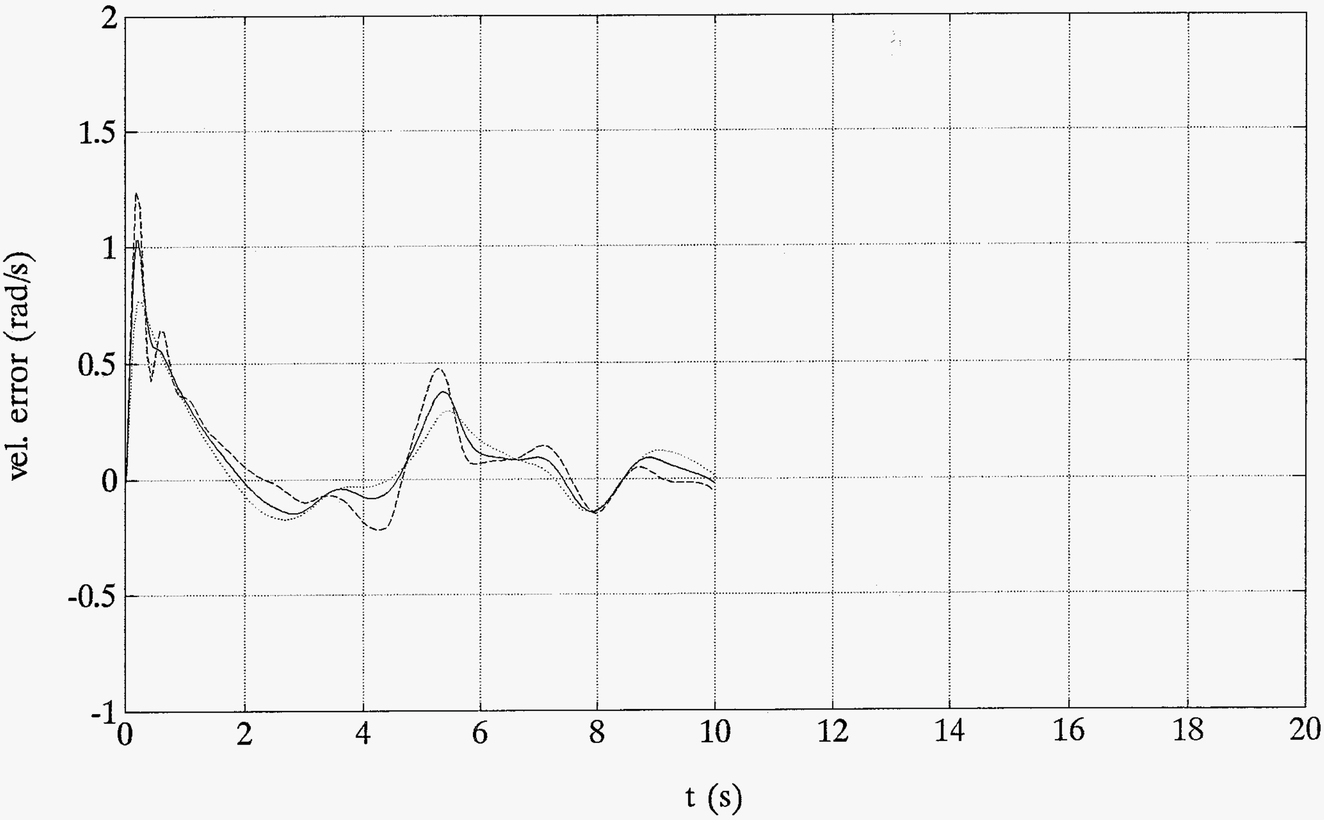


figure 28. The position error response for the first joint (w=0.5)

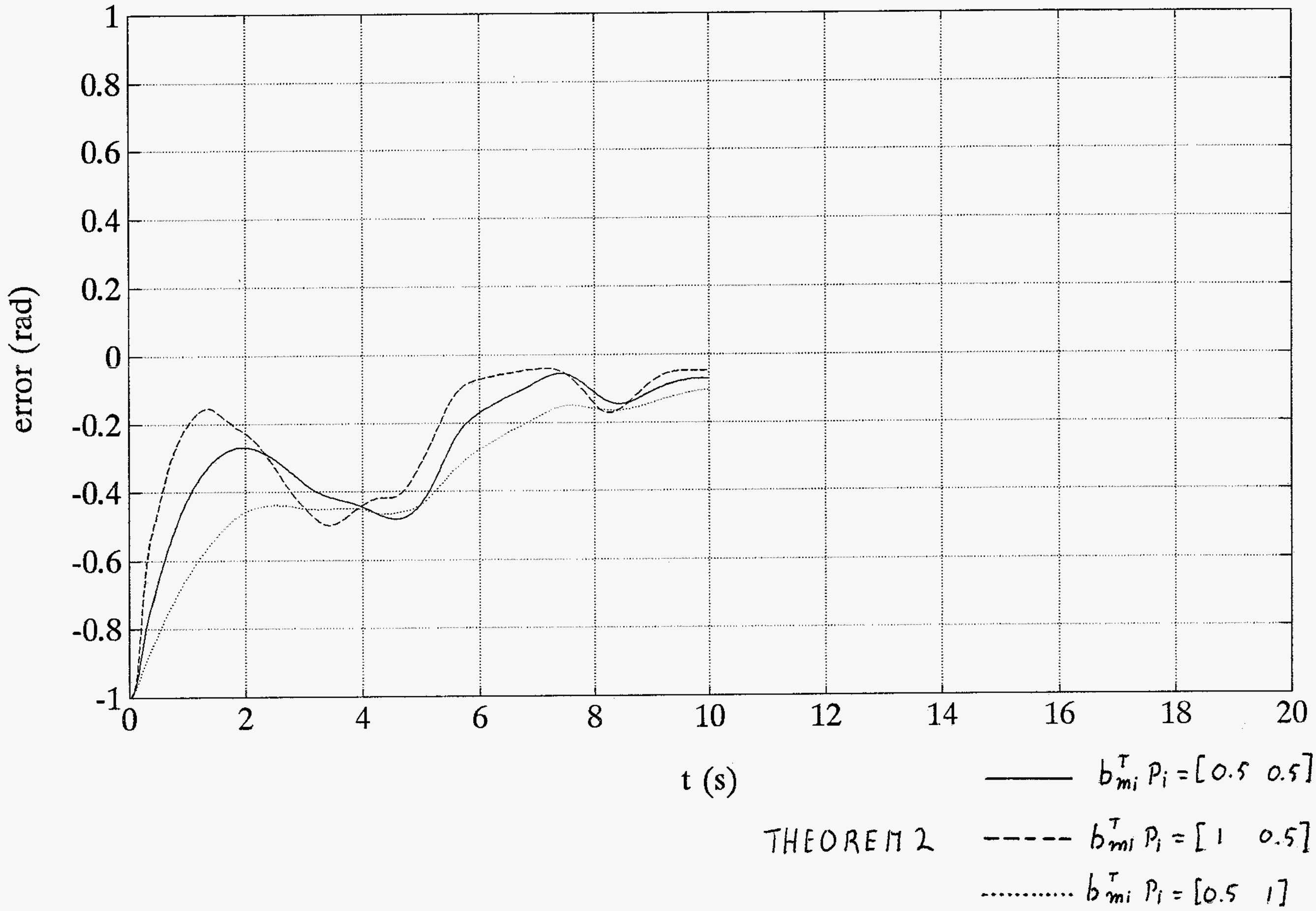
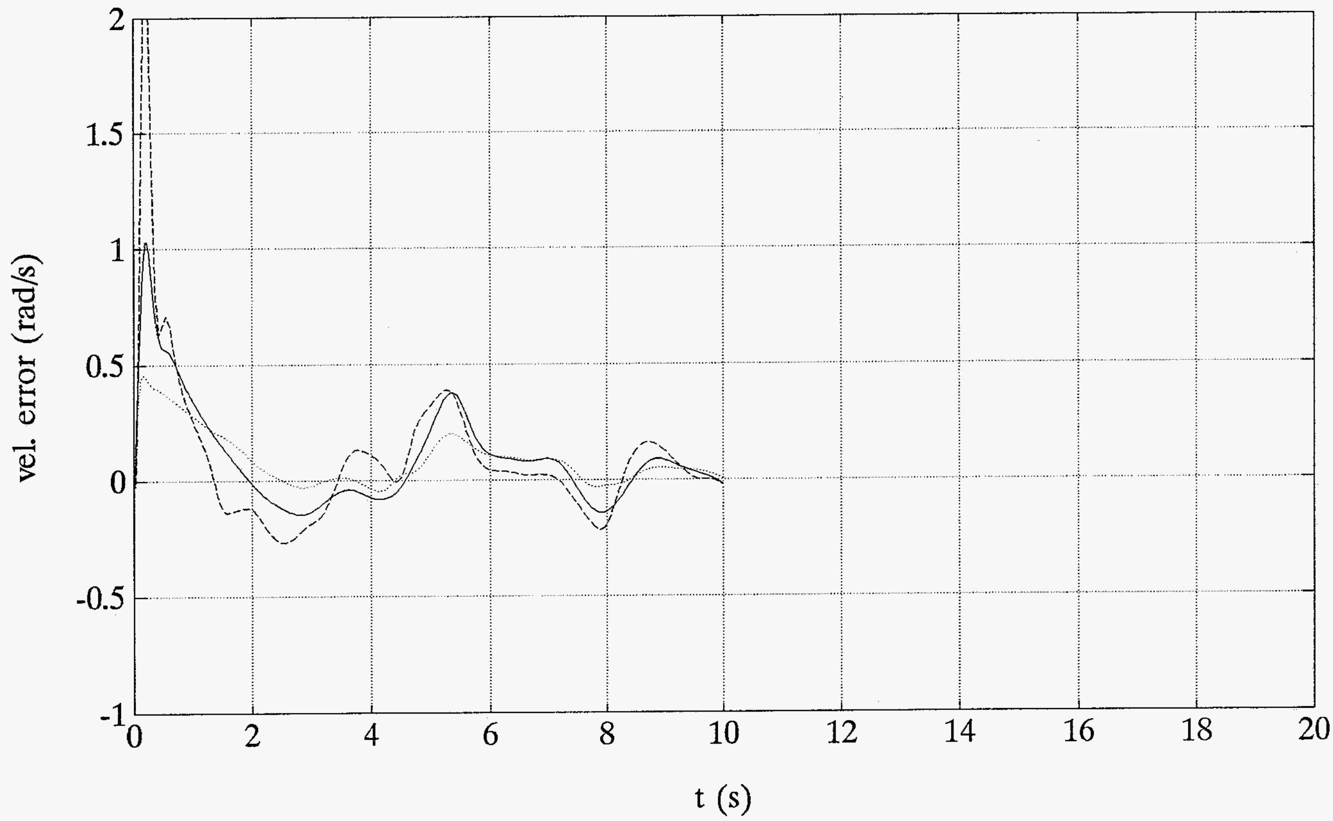


figure 29. The velocity error response for the first joint (w=0.5)



```
% theorie 2
clear
t0=0;
l(1)=0;l(2)=0;l(3)=0;
x0=[0;1.5;0;3;0;0;0;0;0;0;0;0];
for k=1:4
tf=k*2.5;
w=6;
[t,x]=ode45('wim',t0,tf,x0);
save ron3.m
h=l(1);
p=size(t);
n(k)=(p(1)+1)/2;
i=1:n(k);
j=i+h+l(2)+l(3);
q=size(i);
l(k)=q(2);
m=1:2:p(1);
c(j,:)=x(m,:);
t3(j,:)=t(m,:);
r=size(t3);
x0=c(r(1),:);
x0=x0';
t0=t3(r(1));
save ron2.m
end
plot(t3,c(:,1))
```

figure 32: Program for Theorem 2 (ω=6).

```
% theorie 2
function xdot=wim(t,x)
zigma=0.01;
tau1=100;
tau2=50;
alfa1=50;
alfa2=50;
pi1=100;
pi2=100;
w=.5;
r1=1+(1-w*w)*sin(w*t)+2*cos(t)+2*w*cos(w*t);
r2=1-3*cos(2*t)-2*sin(t)-4*sin(2*t);
xm1=1+sin(t)+sin(w*t);
xm1dot=cos(t)+w*cos(w*t);
xm1dd=-sin(t)-w*w*sin(w*t);
xm2=1+cos(t)+cos(2*t);
xm2dot=-sin(t)-2*sin(2*t);
xm2dd=-cos(t)-4*cos(2*t);
a=2.25+1.22*cos(x(3));
b=0.59+0.61*cos(x(3));
c=1.22*sin(x(3))*x(2)*x(4);
d=0.61*sin(x(3));
e=6.75*cos(x(1));
f=2.35*cos(x(1)+x(3));
g=0.59;
e1=x(1)-xm1+x(2)-xm1dot;
e2=x(3)-xm2+x(4)-xm2dot;
u1=x(5)*(x(1)-xm1)+x(6)*(x(2)-xm1dot)+x(7)*r1+x(11);
u2=x(8)*(x(3)-xm2)+x(9)*(x(4)-xm2dot)+x(10)*r2+x(12);
xdot(1)=x(2);
xdot(2)=1/(b*b-a*g)*(g*(-c+e+f)-d*g*x(4)*x(4)-b*d*x(2)*x(2)-b*f-g*u1+b*u
xdot(3)=x(4);
xdot(4)=1/(-b*b+a*g)*(b*(-c+e+f)-b*d*x(4)*x(4)-a*d*x(2)*x(2)-a*f-b*u1+a*
xv1=xdot(2);
xv2=xdot(4);
xdot(5)=-zigma*tau1*x(5)-0.5*tau1*e1*(x(1)-xm1);
xdot(6)=-zigma*tau1*x(6)-0.5*tau1*e1*(x(2)-xm1dot);
xdot(7)=-zigma*tau1*x(7)-0.5*tau1*e1*r1;
xdot(8)=-zigma*tau2*x(8)-0.5*tau2*e2*(x(3)-xm2);
xdot(9)=-zigma*tau2*x(9)-0.5*tau2*e2*(x(4)-xm2dot);
xdot(10)=-zigma*tau2*x(10)-0.5*tau2*e2*r2;
                                                        xm1dd
xdot(11)=-0.5*alfa1*(x(1)-xm1)-0.5*(alfa1+pi1)*(x(2)-xm1dot)-0.5*pi1*(xv1-
xdot(12)=-0.5*alfa2*(x(3)-xm2)-0.5*(alfa2+pi2)*(x(4)-xm2dot)-0.5*pi2*(xv2-
                                                        xm2dd
end
```

figure 33: Program which declares function 'wim' (Theorem 2 and $\omega=0.5$).

```
% theorie 2 , RT-robot
% w=0.5
function xdot=wim(t,x)
zigma=0.01;
tau1=100;
tau2=50;
alfa1=50;
alfa2=50;
pi1=100;
pi2=100;
w=.5;
r1=1+(1-w*w)*sin(w*t)+2*cos(t)+2*w*cos(w*t);
r2=1-3*cos(2*t)-2*sin(t)-4*sin(2*t);
xm1=1+sin(t)+sin(w*t);
xm1dot=cos(t)+w*cos(w*t);
xm1dd=-sin(t)-w*w*sin(w*t);
xm2=1+cos(t)+cos(2*t);
xm2dot=-sin(t)-2*sin(2*t);
xm2dd=-cos(t)-4*cos(2*t);
e1=x(1)-xm1+x(2)-xm1dot;
e2=x(3)-xm2+x(4)-xm2dot;
u1=x(5)*(x(1)-xm1)+x(6)*(x(2)-xm1dot)+x(7)*r1+x(11);
u2=x(8)*(x(3)-xm2)+x(9)*(x(4)-xm2dot)+x(10)*r2+x(12);
xdot(1)=x(2);
xdot(2)=1/(15)*(15*x(1)-5)*x(4)*x(4)+u1;
xdot(3)=x(4);
xdot(4)=1/(8+1/3-10*x(1)+15*x(1)*x(1))*(10-30*x(1))*x(1)*x(4)+u2;
xv1=xdot(2);
xv2=xdot(4);
xdot(5)=-zigma*tau1*x(5)-0.5*tau1*e1*(x(1)-xm1);
xdot(6)=-zigma*tau1*x(6)-0.5*tau1*e1*(x(2)-xm1dot);
xdot(7)=-zigma*tau1*x(7)-0.5*tau1*e1*r1;
xdot(8)=-zigma*tau2*x(8)-0.5*tau2*e2*(x(3)-xm2);
xdot(9)=-zigma*tau2*x(9)-0.5*tau2*e2*(x(4)-xm2dot);
xdot(10)=-zigma*tau2*x(10)-0.5*tau2*e2*r2;
xdot(11)=-0.5*alfa1*(x(1)-xm1)-0.5*(alfa1+pi1)*(x(2)-xm1dot)-0.5*pi1*(xv1-xm1dd
xdot(12)=-0.5*alfa2*(x(3)-xm2)-0.5*(alfa2+pi2)*(x(4)-xm2dot)-0.5*pi2*(xv2-xm2dd
end
```

figure 34: Program for RT-robot (Theorem 2 and ω =0.5).