ADAPTIVE CONTROL OF A FOUR-BAR LINKAGE

by

Stephen O. Carlson

Thesis submitted to the Faculty of

the Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

in

MECHANICAL ENGINEERING

APPROVED:

Arvid Myklebust, Ph.D. Chairman

Harry H. Robertshaw, Ph.D.

Robert G. Leonard, Ph.D.

April, 1985

Blacksburg, Viriginia

ACKNOWLEDGEMENTS

The author would like to thank Dr. Arvid Myklebust, Dr. Harry Robertshaw, and Dr. Robert Leonard for their guidance and patience throughout the preparation of this thesis. A word of thanks to my fellow M. E. graduate students and , among others, who have helped me recognize my accomplishments and have provided invaluable friendship here at Virginia Tech. A special note of thanks goes to my family whose support has been and continues to be an invaluable asset. Finally, thank you for converting my manuscript into a presentable form.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

Nomenclature*

Character	Definition	Units
a	Kinematic link length	inch (cm)
D	Disturbance torque	in-oz (cm-dyn)
f	Algebraic function	
G	Gravational acceleration	in/s^2 cm/s^2
g	Rotational velocity influence coefficient	
$\bar{g}$	Translational velocity influence coefficient	
h	Rotational acceleration influence coefficient	
$\bar{h}$	Translational acceleration influence coefficient	
H	Height above reference line	inch (cm)
I	Effective mass moment of inertia	in-oz s^2 (cm-dyn-s^2)
$\bar{I}$	Mass moment of inertia above the center of gravity	in-oz s^2 (cm-dyn-s^2)
k	Current time sample	
$\underline{K}$	Kalman gain matrix	
$\bar{M}$	Mass	oz s^2/in (grams)
M	Effective mass	oz s^2/in (grams)
N	Number of links in the mechanism	
$\underline{P}$	Covariance matrix	
r	Distance to link center of gravity	inch (cm)

*Underscore capital denotes a matrix quantity.
Underscore lower case denotes a vector quantity.

S	Denotes translational motion	
T	Applied torque	in-oz (cm-dyn)
t	Time	
u	Control output	
V	Translational velocity	in/s (cm/s)
$\underline{v}$	Basis function vector	
x	Gravitational influence coefficient	
$\underline{y}$	Linear estimation coefficients	
•	Denotes the time derivative d/dt	

Greek Symbols

α	Rotational acceleration	$rads/s^2$
Δ	Incremental change	
δ	Sample interval	sec
ε	Random errors	
θ	Angular position center of gravity	rad
λ	Positive weight	
ρ	Exponential data weighting factor	
ϕ	Link angular position	rad
ω	Rotational velocity	rad/s

Superscripts

d	A desired quantity
p	A predicted value
*	An equivalent or global quantity

Subscripts

c	Centrifugal acceleration disturbance

Nomenclature (continued)

G	Gravitational disturbance
i	Mechanism input link
ℓ	Mechanism link
n	Nominal value
CG	Link center of gravity

1.0 Introduction

This study concerns the reduction of input link rotational velocity fluctuations of a four-bar linkage using input torque control. The control torques are calculated using discrete-time adaptive techniques. The main objective is to reduce the velocity fluctuations caused by the mechanism nonlinear dynamics. This guarantees the kinematic conditions assumed when the four-bar mechanism is designed for kinematic motion specifications. The solution uses the available driving motor torque to reduce the nonlinear dynamic effects without compromising the response to a reference velocity input signal. Three self-tuning adaptive controls were developed. They are mostly derived from proposed robot position controllers which have shown some success.

The control performance is evaluated using a digital simulation of the four-bar dynamics. This simulation is verified using experimental velocity and torque data from an industry standard servomotor control system. This standard system is simulated and used as a primary reference for performance evaluation. Control performance was evaluated using a percentage fluctuation around the desired velocity and the response to a step change in the reference input velocity.

1.1 Uses of the Four-Bar Linkage

The four-bar linkage is a common mechanism used in the manufacturing environment. The mechanism's ability to perform many basic tasks such as moving piece parts between transport conveyors, generating a desired input/output rotation function, and moving a point through a trajectory finds applications in automatic machines. Typically, the

mechanism is driven from an internal drive shaft, however, in small machines or single mechanism machines, the four-bar is directly driven using a motor or servo-motor.

There are many uses for the four-bar, however, they are classified as three basic mechanism types, line-of-motion generation, function generation, and body guidance. These basic mechanism types can be easily designed by computer to perform these fixed motion tasks or only slightly variable tasks at low cost [1].

For example, consider a simple line-of-motion mechanism used in a semi-flexible inspection station. The mechanism is to provide a known motion to transport a film under an optical sensor. The inspection station, thus the mechanism, is required to operate at different speeds due to required inspection response. The kinematic motion can be readily designed using established computer techniques and the film motion specifications. However, these design techniques usually assume the input motion (e.g. input velocity) is a fixed constant or is constant over a revolution. This assumption is valid for mechanisms driven with a large inertia driver. However, for a mechanism driven with a low inertia servomotor, the assumption is not true. Myklebust [1] has observed 25% velocity fluctuation around a mean velocity over an input link revolution. This velocity fluctuation will disrupt the desired path motion causing, here, poor inspection. The problem then is to control the input velocity fluctuations and maintain a rapid transient response without using an oversize driving motor.

1.2 Problem and Solutions

Dynamically, the problem is to control the mechanism velocity over an input revolution where the effective mass moment of inertia changes as a known function of the input position. These changes are caused by the relative motion between the links. The inertia and work functions due to the film motion are small, therefore, no step discontinuities are added to the inertia variation. The problem solution then involves applying the proper correction torques to maintain the constant velocity and provide the required response to the desired velocity input function.

Previous solutions have been to reduce the nonlinear dynamic effects for the mechanism shape changes. The simplest solution is to add flywheels to the input. This solution simply makes the inertia changes small in comparison to the average inertia. Therefore, disturbance torque due to the inertia variation cannot accelerate the average inertia as much thus reducing the velocity fluctuations. Unfortunately, the flywheel also retards the response to the driving motor torque. Other solutions typically revolve around torque-balancing techniques using external springs and dampers [1,2]. These solutions are velocity dependent operating at maximum effectiveness in a narrow velocity region. In particular, Myklebust [2,3] has considered using a harmonic oscillator attached to the four-bar coupler link. His conclusion is that the oscillator was to complex to implement as a standard practice. Again, this solution is velocity dependent; therefore, restraining the mechanism operating range.

The velocity control solution studied here was suggested by

Myklebust [2,3]. This solution uses the available motor torque to reduce the velocity fluctuations over a broad operating range. A low average inertia and the required peak motor torque combine to provide a rapid system transient response. The solution objective is to design a mechanism input torque control to produce the kinematic design input conditions for a broad catagory of mechanisms.

Self-tuning discrete time adaptive techniques were studied as possible solutions. These adaptive techniques have shown respectable improvement in robot position control over standard linear control techniques, which was the only information source on mechanism control. The techniques allow the controller to use the apriori knowledge of the mechanism dynamics based on the input position to aid velocity control. The resulting control algorithms are checked for application feasibility and performance using a computer simulation of the proposed mechanism control system (Fig. 1.1). The system consist of an experimental four-bar linkage, a PMI D.C. servomotor and amplifier, and a PMI 1000 line optical-encoder. The control computer is based on the Intel 8086/8087 microprocessors. Application feasibility requires the estimated microprocessor calculation time of the control algorithm be less than the controller sample interval. Overall performance is based on the velocity fluctuations and the ability to track a velocity input function.

1.3 Study Development

The study develops as follows. A nonlinear dynamic model, based on an experimental four-bar mechanism is developed using Kinematic

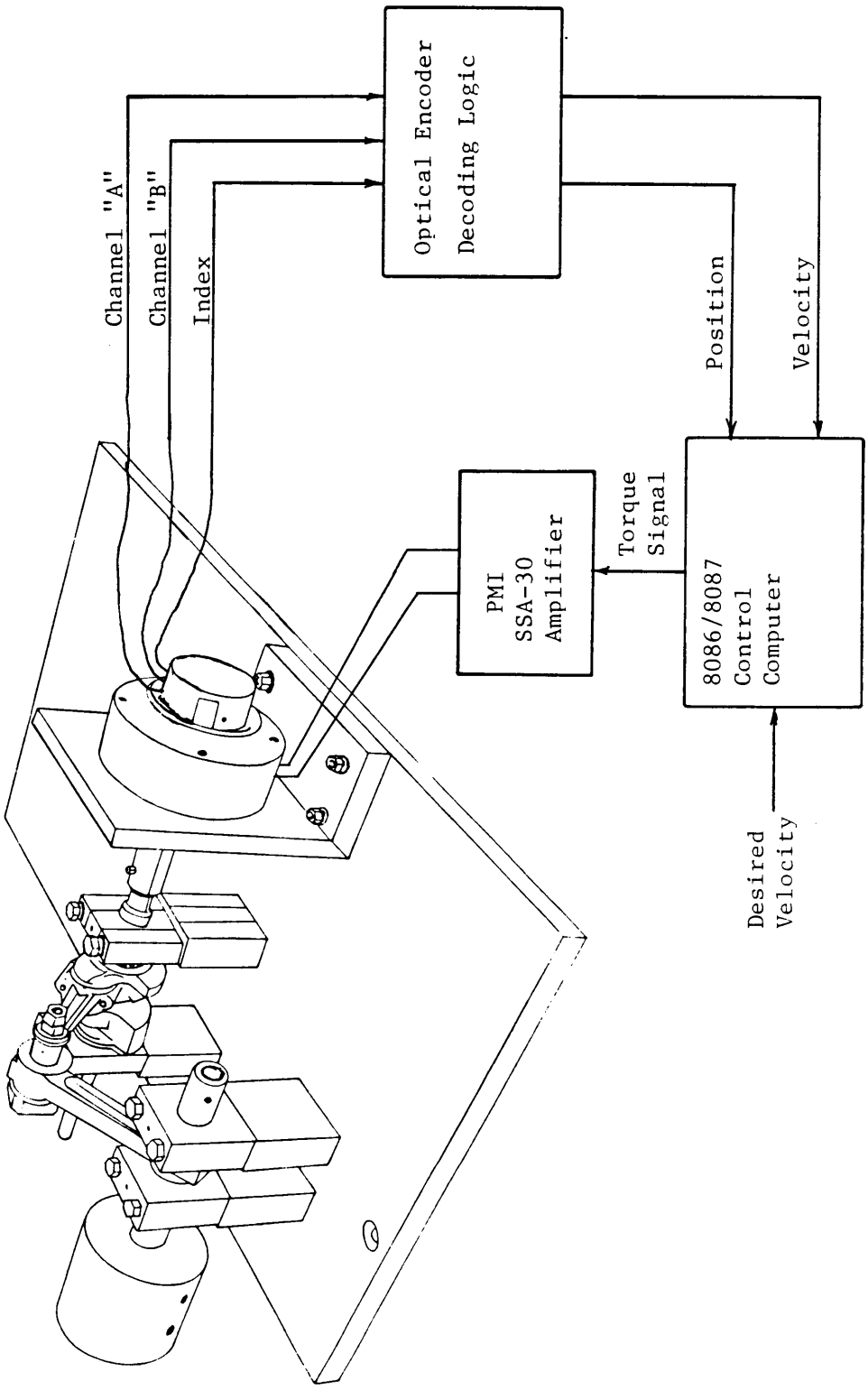


Fig. 1.1. Computer Controlled Four-bar Mechanism

Influence Coefficients. This technique yields a nonlinear dynamic equation with algebraic coefficients which vary as functions of the mechanism input position. The nonlinearities are discussed as a means to better define the overall problem. The dynamic model is used to describe an open-loop control system using an industry standard D.C. servomotor/amplifier, the experimental four-bar mechanism, and an optical encoder feedback instrument. The open-loop system is simulated using the Advanced Continuous Simulation Language. The simulation is used to evaluate the proposed adaptive control performance.

Four control methods, one a standard analog control, and three adaptive are developed as possible input torque control methods. In each case, the performance is evaluated using the simulated open-loop control system. Comparison between the controls is based on the ability to maintain the kinematic input condition (constant velocity) and track both step and ramp desired velocity functions. The results show a computed-torque technique yields the best performance. Several suggestions for continuing research are briefly discussed.

2.0 Influence Coefficient Formulation

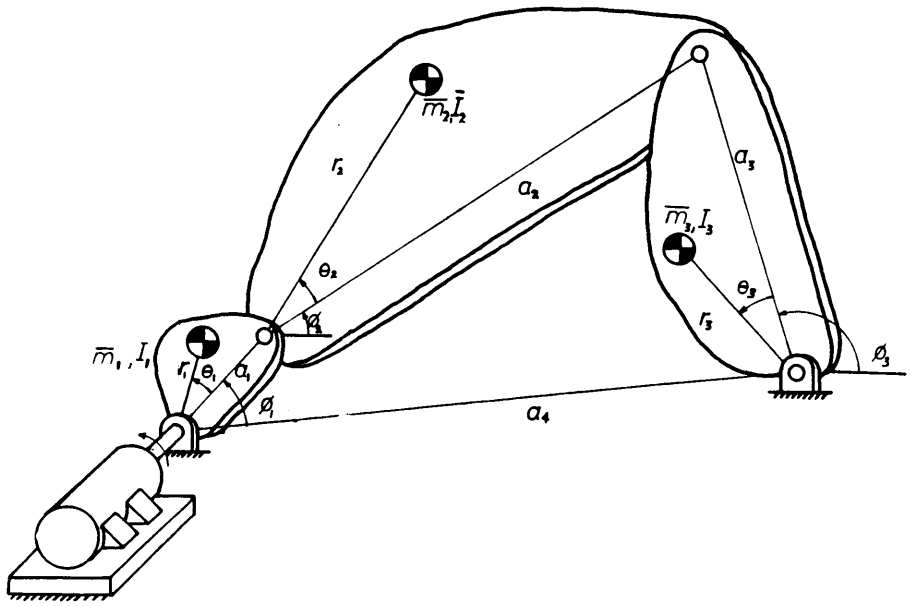
The four-bar dynamics are typically described as highly nonlinear. However, since the mechanism has only one kinematic degree of freedom, the dynamics can be completely described using the input link position, velocity, and acceleration. Consider the general motor driven four-bar, Fig. (2.1a), as a deformable rotor, Fig. (2.1b), that has the ability to vary in time its mass moment of inertia and center of gravity. Kinematic Influence Coefficients, [4], provide time-invariant closed form solutions for these changes as functions of the input link position only. This yields two dynamic state space equations, two, versus the eight Euler-Lagrange state equations used by Myklebust [2,3] but, yields the same velocity response, and is easier to use to describe the nonlinear dynamics and for designing the adaptive controllers.

In this section, the Kinematic Influence Coefficient method from [4] is discussed for solving the nonlinear four-bar dynamics. The required nonlinear four-bar influence coefficients are derived for a general four-bar. The nonlinear characteristics are discussed as a means to explain the nature of the linkage control problem for use in later sections.

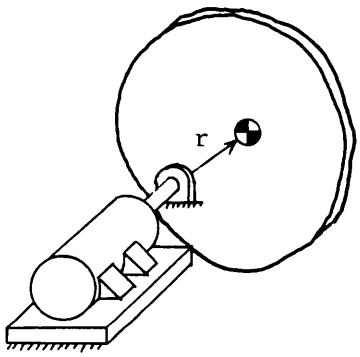
2.1 Kinematic Influence Coefficient Method

Kinematic influence coefficients describe the four-bar dynamics in closed form as a nonlinear differential equation with algebraic coefficients. Benedict [4] provides a complete development of this technique.

The objective is to describe the effect of each link on the input



(a)



(b)

Fig. 2.1 Motor Driven Time Varying Inertial Systems

link using the rigid link kinematic motion contraints. In this discussion, the subscript, i, refers to input link, and the subscript, ℓ, refers to any other link. Consider the mechanism input link, i, and another link, ℓ, as Fig. (2.2a). The rotational motion of link ℓ is completely described by the function $f_\ell(\phi_i)$ in terms of the position ϕ_i,

$$\phi_\ell = f_\ell(\phi_i(t)) \tag{2.1-1}$$

The argument, $\phi_i(t)$, of the function f_ℓ is now dropped except where ambiguity may result. The time derivative of (2.1-1) is

$$\dot{\phi}_\ell = \frac{df_\ell}{d\phi_i}\,\omega_i \ . \tag{2.1-2}$$

Here

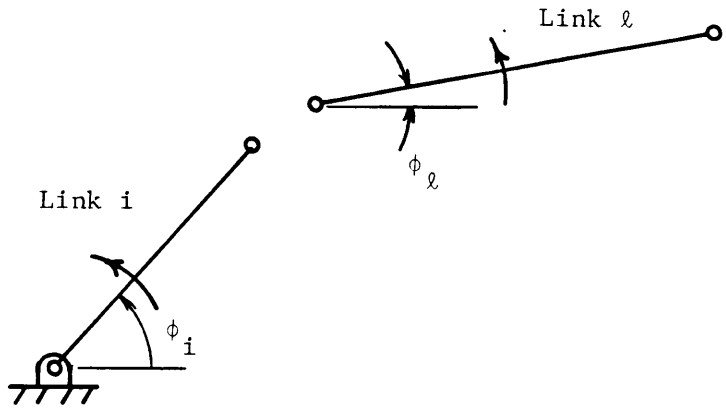
$$\frac{d\phi_i}{dt} = \omega_i \tag{2.1-3}$$

is the input link velocity. Define

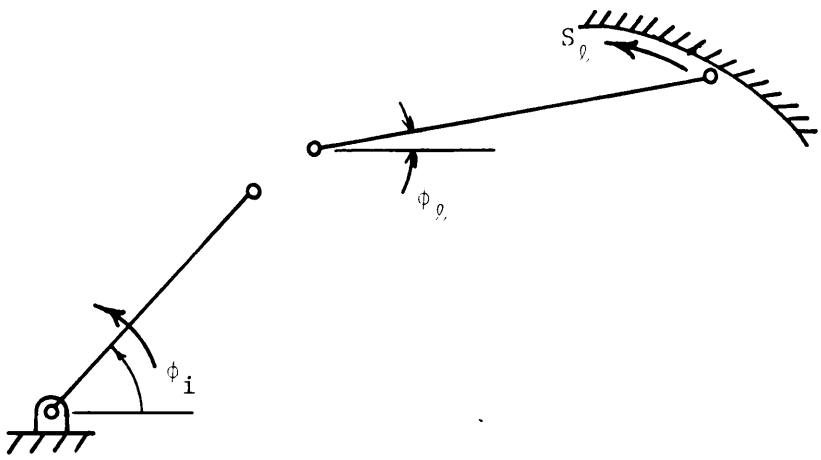
$$\frac{df_\ell}{d\phi_i} = g_\ell(\phi_i) \tag{2.1-4}$$

as the rotational velocity influence coefficient of link ℓ on link i. Differentiating equation 2.1-2 with respect to time again yields

$$\ddot{\phi}_\ell = \frac{df_\ell}{d\phi_i}\left(\frac{d\phi_i}{dt}^2\right) + \frac{df_\ell}{d\phi_i}^2\ \frac{d\phi_i}{dt} \ . \tag{2.1-5}$$



(a)



(b)

Fig. 2.2 General Relative Link Motion

Here define

$$\frac{d^2 f_\ell}{d\phi_i^2} = \frac{dg_\ell}{d\phi_i} = h_\ell \ , \tag{2.1-6}$$

where h_ℓ is the rotational acceleration influence coefficient. Equation 2.1-5 is rewritten as

$$\ddot{\phi}_\ell = g_\ell \ \alpha_i + h_\ell \ \omega_i^2 \tag{2.1-7}$$

using

$$\alpha_i = \frac{d\phi_i^2}{dt^2} \tag{2.1-8}$$

and equation 2.1-6.

Equations 2.1-1, 2, 7 completely describe the rotational motion of link ℓ with respect to the motion of link i using the Kinematic Influence Coefficients g_ℓ and h_ℓ. Where the coefficients g_ℓ and h_ℓ are algebraic functions of the input position, ϕ_i, only.

The same derivation is used for translational motion. Let the translational motion, S_ℓ, of link ℓ, Fig. (2.2b) be described as

$$S_\ell = \bar{f}_\ell(\phi_i(t)) \ . \tag{2.1-9}$$

Again the argument $\phi_i(t)$ of $\bar{f}_\ell$ is dropped. Taking two derivatives with respect to time yields

$$\dot{S}_{\ell} = \frac{d\bar{f}_{\ell}}{d\phi_i} \frac{d\phi}{dt} \tag{2.1-10}$$

$$\ddot{S}_{\ell} = \frac{d\bar{f}_{\ell}}{d\phi_i} \frac{d\phi_i^2}{dt^2} + \frac{d^2\bar{f}_{\ell}}{d\phi_i^2} \left(\frac{d\phi_i}{dt}\right)^2 . \tag{2.1-11}$$

Define the linear velocity and acceleration influence coefficients as

$$\frac{d\bar{f}_{\ell}}{d\phi_i} = \bar{g}\ (\phi_i) \tag{2.1-12}$$

and

$$\frac{d\bar{f}_{\ell}^2}{d\phi_i^2} = \frac{d\bar{g}_{\ell}}{d\phi_i} = \bar{h}_{\ell}(\phi_i) \tag{2.1-13}$$

respectively. Using 2.1-3 and 8, equation 2.1-10 is written as

$$\ddot{S}_{\ell} = \bar{g}_{\ell}(\phi_i)\ \alpha_i + \bar{h}_i(\phi_i)\ \omega_i^2 . \tag{2.1-14}$$

The influence coefficients equations 2.1-4, 6, 12, 13 are the basic coefficients required to analyze the four-bar dynamics.

Gravitational forces are external forces applied to the link ℓ. Their effects on input link are found using vitual work [4], or a work-energy approach. The method used is problem dependent, and selection depends on which method is easier to use. The requirement is to show the effects of gravity on the link ℓ as an equivalent torque applied to the input, i. The work-energy method is used for the four-bar problem.

The work-energy method states the work done by the equivalent torque, $T_{G\ell}$, at the input must be equal to the change in the potential

energy of link ℓ. Stated mathematically

$$T_{G\ell}\ \Delta\phi_i = -\bar{M}_\ell\ G\ \Delta H_\ell\ (\phi_i)\ ,$$

where $H_\ell(\phi_i)$ - height of link ℓ center of gravity

G - gravitational acceleration

$\bar{M}_\ell$ - mass of link ℓ .

Here, the height H_ℓ is a function of the input position. Dividing by the increment $\Delta\phi_i$, and taking the limit as $\Delta\phi_i \to 0$ yields

$$T_{G\ell} = -\bar{M}_\ell\ G \lim_{\Delta\phi_i \to 0} \frac{\Delta H_\ell(\phi_i)}{\Delta\phi_i} = -\bar{M}_\ell \frac{dH(\phi_i)}{d\phi_i} \quad . \tag{2.1-15}$$

We can then define the gravitational influence coefficient

$$x_\ell = \frac{dH_\ell(\phi_i)}{d\phi_i} \tag{2.1-16}$$

and the equivalent torque at the input link i is

$$T_{G\ell} = -\bar{M}_\ell\ G\ x_\ell\ . \tag{2.1-17}$$

The total effect for all the links is found using superposition and is written as the sum of the individual link contributions, $T_{G\ell}$

$$T^*_G = \sum_{\ell=1}^{N-1} -\bar{M}_\ell\ Gx_\ell\ . \tag{2.1-18}$$

The generalized system Intertia, I^*, is described using the influence coefficients equations 2.1-4,6,12,13. Here the kinetic energy of link ℓ is equivalent to the kinetic energy of an effective inertia, I_ℓ, at the input

$$KE_\ell = \frac{1}{2} I_\ell \omega_i^2 = \frac{1}{2} [\bar{I}_\ell \omega_\ell^2 + \bar{M}_\ell V_\ell^2] . \qquad (2.1\text{-}19)$$

KE_ℓ - kinetic energy of link ℓ

Substituting

$$\omega_\ell = g_\ell \omega_i$$

and

$$V_\ell = \bar{g}_\ell \omega_i$$

and factoring out ω_i^2 gives the effective kinetic energy of link ℓ at the input link as

$$\frac{1}{2} I_\ell \omega_i^2 = \frac{1}{2} [\bar{I}_\ell g_\ell^2 + \bar{M}_\ell \bar{g}^2] \omega_i^2 \qquad (2.1\text{-}20)$$

or the effective inertia, I_ℓ, of link ℓ at the input as

$$I_\ell = \bar{I}_\ell g_\ell^2 + \bar{M}_\ell \bar{g}^2 . \qquad (2.1\text{-}21)$$

Equation (2.1-20) is the inertial influence of link ℓ on link i. The total inertial influence, I^*, is the sum of the individual link contributions.

$$I^* = \sum_{\ell=1}^{N-1} (\bar{I}_\ell \, g_\ell^2 + \bar{M}_\ell \, \bar{g}_\ell^2) \qquad (2.1\text{-}22)$$

In equation (2.1-22), the factors are either constant or known functions of the input position ϕ_i. Therefore, I^* is a known function in ϕ_i. For the four-bar linkage, the coupler link, link 2, has both rotational and translational motion. The input and rocker links, 1 and 3, have only rotational motion. Therefore, the equivalent inertia, I^*, using (2.1-22) is

$$I^* = [\bar{I}_1 + \bar{I}_2 \, g_2^{\,2} + \bar{I}_3 \, g_3^{\,2} + \bar{M}_2 \, \bar{g}_2^{\,2}] \, . \qquad (2.1\text{-}22a)$$

The described influence coefficients are sufficient to analyze the Inertia dynamics and the gravitational effects for the four-bar linkage. Benedict, [4], describes in detail the complete method for analyzing joint friction and other nonlinearities. This detail was not required for this study.

The result of the kinematic influence technique is to reduce the mechanism to a generalized rotor. The dynamics can then be written in two state space equations with algebraic coefficients using an equivalent mass-and-force technique [4], [5].

The equivalent mass-and-force technique yields the dynamic equations for the generalized rotor. If we consider only the inertial

effects the input power must equal the rate of change in the system kinetic energy.

$$T^* \omega_i = \frac{d}{dt} \left(\frac{1}{2} I^*(\phi_i) \omega_i^2\right) \tag{2.1-23}$$

Since $I^*(\phi_i)$, is a function of ϕ_i which is a function of time, the time derivative yields

$$T^* \omega_i = \frac{1}{2} \frac{dI^*(\phi_i)}{d\phi_i} \frac{d\phi_i}{dt} \omega_i^2 + I^*(\phi_i) \alpha_i \omega_i , \tag{2.1-24}$$

and the torque required to overcome the inertial effects is

$$T^* = \frac{1}{2} \frac{dI^*}{d\phi_i} \omega_i^2 + I^*(\phi_i) \alpha_i . \tag{2.1-25}$$

The complete model must also include the external gravity disturbance torque T_G^*. Therefore the complete system model is then

$$T^* = I^*(\phi_i) \alpha_i + \frac{1}{2} \frac{dI^*(\phi_i)}{d\phi_i} \omega_i^2 - T_G^* . \tag{2.1-26}$$

The system model equation (2.1-26) is a single differential equation which is easily inverted to yield the nonlinear dynamic model

$$\alpha_i = \frac{T^* - \frac{1}{2} \frac{dI^*(\phi_i)}{d\phi_i} \omega_i^2 + T_G^*}{I^*(\phi_i)} . \tag{2.1-27}$$

The dynamic model is further reduced to state space form by introducing

the linear differential state equation

$$\dot{\phi}_i = \omega_i \, .$$

Let $$\frac{d\omega}{dt} = \dot{\omega} = \alpha$$

then the state space model is

$$\dot{\phi}_i = \omega_i \tag{2.1-28}$$

$$\dot{\omega}_i = \frac{T^* - \frac{1}{2}\frac{dI^*(\phi_i)}{d\phi_i}\omega_i^2 + T_G^*}{I^*(\phi_i)} \, . \tag{2.1-29}$$

Using the influence coefficient functions (2.1-22a), (2.1-17) and the relations (2.1-6, 12), the nonlinear dynamic model becomes

$$\dot{\phi}_i = \omega_i \tag{2.1-30}$$

$$\dot{\omega}_i = \frac{T^* - [\bar{I}_2 g_2 h_2 + \bar{I}_3 g_3 h_3 + \bar{M}_2 \bar{g}_2 \bar{h}_2]\ \omega_i^2 - G[\bar{M}_2 x_1 + \bar{M}_2 x_2 + \bar{M}_3 x_3]}{[\bar{I}_1 + \bar{I}_2 g_2 + \bar{I}_3 g_3^2 + \bar{M}_2 \bar{g}_2^2]} \tag{2.1-31}$$

This model is then integrated numerically to yield the four-bar response to the input torque T^*.

2.2 Derivation of the Four-Bar Influence Coefficients

The following discussion details the derivation of the influence coefficients required for the four-bar linkage dynamic analysis.

The influence coefficients are based strictly on the mechanism geometry. Using the notation of Fig. 2.3 with input link i = 1, the mechanism position solution for the branch shown is

$$\phi_2 = 2 \tan^{-1} \left\{ \frac{-A + \sqrt{A^2 + B^2 + C^2}}{(C - B)} \right\} \qquad (2.2\text{-}1)$$

where

$$A = \sin\phi_1$$

$$B = \cos\phi_1 - \frac{a_4}{a_1}$$

$$C = \frac{(a_1^2 + a_2^2 + a_4^2 - a_3^2)}{2\, a_1\, a_2} - \left\{ \frac{a_1}{a_2} \cos\phi_1 \right\}$$

and

$$\phi_3 = 2 \tan^{-1} \left\{ \frac{A - \sqrt{A^2 + B^2 - C^2}}{(B + C)} \right\} \qquad (2.2\text{-}2)$$

where

$$A = \sin\phi_1$$

$$B = \cos\phi_1 - \frac{a_4}{a_1}$$

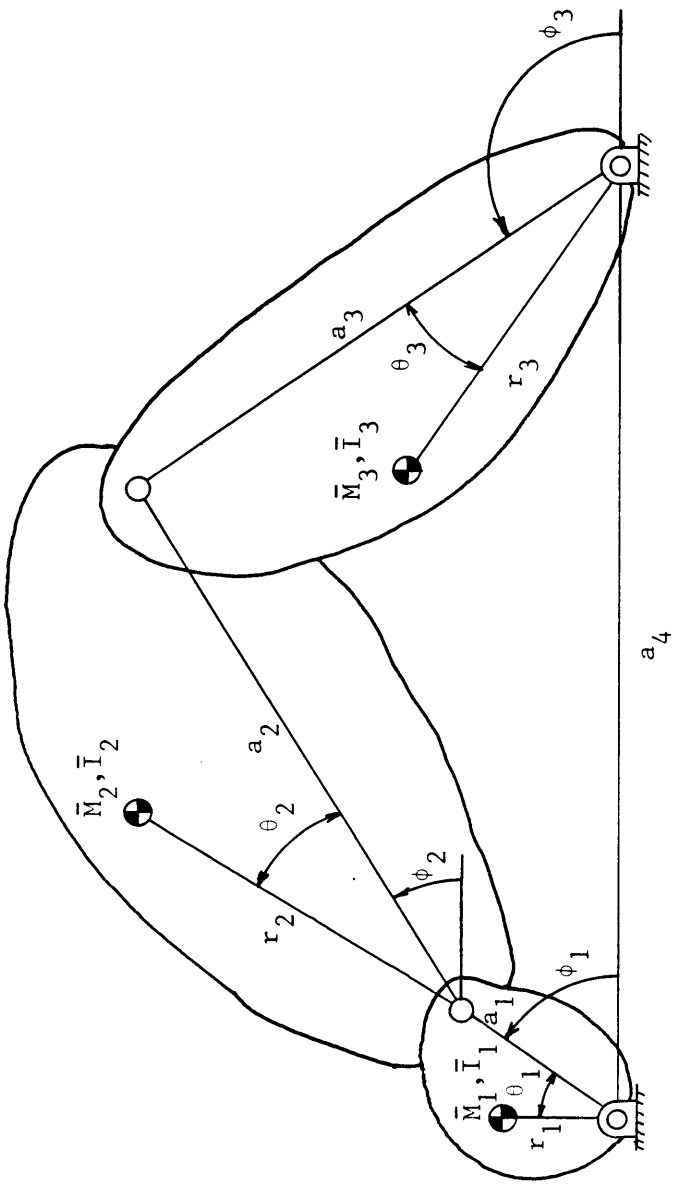


Fig. 2.3 Generalized Four-Bar Linkage

$$C = \frac{a_1^2 + a_3^2 + a_4^2 - a_2^2}{2\, a_1\, a_3} - \frac{a_4}{a_3} \cos\phi_1 \ .$$

The velocity coefficient, g_ℓ, is the velocity ratio between link ℓ and the input link i and are derived using the vector loop equations. From Fig. 2.3 , the position vector loop equation

$$\vec{a}_1 + \vec{a}_2 = \vec{a}_3 + \vec{a}_4$$

is separated into the $\vec{i}$ & $\vec{j}$ component equations

$$a_1 \cos\phi_1 + a_2 \cos\phi_2 = a_4 + a_3 \cos\phi_3 \qquad (\vec{i}) \qquad (2.2\text{-}3)$$

and

$$a_1 \sin\phi_1 + a_2 \sin\phi_2 = a_3 \sin\phi_3 \qquad (\vec{j}) \ . \qquad (2.2\text{-}4)$$

In equations 2.2-3,4 the links are assumed rigid. The time derivatives of equations 2.2-3,4 yield

$$-a_1 \sin\phi_1\, \dot{\phi}_1 - a_2 \sin\phi_2\, \dot{\phi}_2 = -a_3 \sin\phi_3\, \dot{\phi}_3 \qquad (2.2\text{-}5)$$

and

$$a_1 \cos\phi_1\, \dot{\phi}_1 + a_2 \cos\phi_2\, \dot{\phi}_2 = a_3 \cos\phi_3\, \dot{\phi}_3 \ . \qquad (2.2\text{-}6)$$

solving equations 2.2-5,6 alternately for $\dot{\phi}_2$ and $\dot{\phi}_3$ in terms of $\dot{\phi}_1$

yield the required velocity influence coefficients g_2 and g_3.

$$g_2 = \frac{\dot{\phi}_2}{\dot{\phi}_1} = \frac{a_1}{a_2} \frac{\sin(\phi_1 - \phi_3)}{\sin(\phi_3 - \phi_2)} \tag{2.2-7}$$

$$g_3 = \frac{\dot{\phi}_3}{\dot{\phi}_1} = \frac{a_1}{a_3} \frac{\sin(\phi_1 - \phi_2)}{\sin(\phi_3 - \phi_2)} \cdot \tag{2.2-8}$$

The acceleration influence coefficients h_2, h_3 from equation 2.1-8 are

$$h_2 = \frac{dg_2}{d\phi_1} = \frac{a_1}{a_2}\left[\frac{\sin(\phi_3-\phi_2)\cos(\phi_1-\phi_3)(1-g_3) - \sin(\phi_1-\phi_3)\cos(\phi_3-\phi_2)(g_3-g_2)}{\sin^2(\phi_3-\phi_2)}\right] \tag{2.2-9}$$

and

$$h_3 = \frac{dg_3}{d\phi_1} = \frac{a_1}{a_2}\left[\frac{\sin(\phi_3-\phi_2)\cos(\phi_1-\phi_2)(1-g_2) - \sin(\phi_1-\phi_2)\cos(\phi_3-\phi_2)(g_3-g_2)}{\sin^2(\phi_3-\phi_2)}\right] \tag{2.2-10}$$

Again, the derivatives

$$\frac{d\phi_\ell}{d\phi_1} = g_\ell$$

are the velocity influence coefficients (2.2-7, 8).

The other required influence coefficients, $\bar{g}_2$, $\bar{h}_2$ describe the coupler, link 2, linear motion. Observing equation 2.1-3.1, the translational coefficients appear as $\bar{g}_2^{\,2}$ and $\bar{g}_2\,\bar{h}_2$. This observation is used as a stopping point in the derivation.

Link 2 translational influence coefficient $\bar{g}_2$ is derived using the velocity of the center of gravity of link 2. From Fig. 2.3, the position of link 2 center of gravity, S_{CG_2}, is

$$S_{CG_2} = a_1 \cos\phi_1 \vec{i} + a_1 \sin\phi_1 \vec{j} + r_2 \cos(\phi_2 + \theta_2)\vec{i} + r_2 \sin(\phi_2 + \theta_2)\vec{j} . \tag{2.2-11}$$

The velocity of this point is the time derivative of (2.2-11)

$$V_{CG_2} = -a_1 \sin\phi_1 \dot{\phi}_1\vec{i} + a_1 \cos\phi_1 \dot{\phi}_1\vec{j} - r_2 \sin(\phi_2 + \theta_2) \dot{\phi}_2 \vec{i} + r_2 \cos(\phi_2 + \theta_2) \dot{\phi}_2\vec{j} . \tag{2.2-12}$$

Using the influence coefficient terms for $\dot{\phi}_2$, $\dot{\phi}_3$ and collecting the vector components yields

$$V_{CG_2} = [(-a_1 \sin\phi_1 - r_2 g_2 \sin(\phi_2 + \theta_2)\vec{i} + (a_1 \cos\phi_1 + r_2 g_2 \cos(\phi_2 + \theta_2)\vec{j}] \dot{\phi}_1 \tag{2.2-13}$$

We are required to find the term $\bar{g}_2^{\,2}$ which can be interpreted as

$$\bar{g}_2^{\,2} = \left(\frac{V_{CG_2}}{\dot{\phi}_1}\right)^2$$

and is the magnitude squared of the vector equation (2.2-13). This yields

$$\bar{g}_2^{\,2} = \left(\frac{V_{CG_2}}{\dot{\phi}_1}\right)^2 = a_1^2 + r_2^2\ g_2^2 + 2a_1\ r_2 g_2\ \cos(-\phi_1 + \phi_2 + \theta_2) \tag{2.2-14}$$

The $\bar{h}_2$ coefficient only appears as $\bar{g}_2\ \bar{h}_2$ in equation (2.1-31) and is defined as

$$\bar{h}_2 = \frac{d\bar{g}_2}{d\phi_1}\ .$$

It is determined by taking the derivative of equation (2.2-14) with respect to the input position ϕ_1. This manipulation yields

$$\frac{d(\bar{g}_2^{\,2})}{d\phi} = 2\ \bar{g}_2\ \frac{d(\bar{g}_2)}{d\phi_1} = \bar{g}_2\ \bar{h}_2 =$$

$$r_2^2\ g_2 h_2 + a_1 r_2[h_2\cos(-\phi_1 + \phi_2 + \theta_2) - g_2\sin(-\phi_1 + \phi_2 + \theta_2)(g_2 - 1)]. \tag{2.2-15}$$

We now have the required influence coefficients to describe the effective mechanism moment of inertia. However, the terms to describe the external gravity torques have not been determined.

The gravity disturbance influence coefficients are derived using a work-energy technique. Using equation (2.1-16), the gravity coefficient for link 2 is

$$x_2 = \frac{dH_2}{d\phi_1}\ .$$

From the mechanism geometry,

$$H_2 = a_1 \sin\phi_1 + r_2 \sin(\phi_2 + \theta_2)$$

and

$$x_2 = \frac{dH_2}{d\phi_1} = a_1 \cos\phi_1 + r_2 \cos(\phi_2 + \theta_2)\, g_2 \qquad (2.2\text{-}16)$$

Similarly, for x_3

$$H_3 = r_3 \sin(\phi_3 + \theta_3)$$

$$x_3 = \frac{dH_3}{d\phi_1} = r_3 \cos(\phi_3 + \theta_3)\, g_3 \,. \qquad (2.2\text{-}17)$$

Here, the velocity coefficients g_2, g_3 can be interpreted as the mechanical advantage of link ℓ on link i. For the input, the gravity disturbance is

$$x_1 = r_1 \cos(\phi_1 + \theta_2) \,. \qquad (2.2\text{-}18)$$

The total effect of gravity on the mechanism is then

$$T_G^{*} = -G[\bar{M}_1 x_1 + \bar{M}_2 x_2 + \bar{M}_3 x_3] \qquad (2.2\text{-}19)$$

using (2.1-18, 2.2-16, 17, 18).

These are the required kinematic influence coefficients for the

four-bar linkage. The coefficients are strictly algebraic functions of the input position and the known mechanism parameters.

2.3 Discussion of Nonlinear Characteristics

The influence coefficient formulation allows classification of the nonlinear four-bar dynamics. The system equation using the influence coefficient equivalents for I^* and T_G^* is

$$T^* = [\bar{I}_1 + \bar{I}_2 g_2^{\,2} + \bar{I}_3 g_3^{\,2} + \bar{M}_2 \bar{g}_2^{\,2}]\alpha_i - [\bar{I}_2 g_2 h_2 + \bar{I}_3 g_3 h_3 + \bar{M}_2 \bar{g}_2 \bar{h}_2]\ \omega_i^{\,2} - G[\bar{M}_1\ x_1 + M_2\ x_2 + \bar{M}_3\ x_3]\ . \qquad (2.3\text{-}1)$$

The position-varying centrifugal acceleration term, $\omega_i^{\,2}$ term, and the gravity torque are interpreted as disturbances, D_c, and, D_g, on the linear inertial system

$$T^* = I_r\ \alpha + D_c + D_g\ ,$$

where I_r is the average mechanism inertia. Fig. 2.4 is a block diagram for the linear system. The disturbance D_g has a constant magnitude for any input velocity, however the magnitude of disturbance D_c varies with the mechanism velocity squared. Above velocities of 20 rad/sec, this nonlinear disturbance dominates the mechanism dynamics. Under the linear assumption, four-bar linkage velocity control reduces to controlling an inertial load and rejecting these torque disturbances.

However, we know the terms of equation 2.3-1 are strictly algebraic functions of the input position and their forms are known. Thus we know

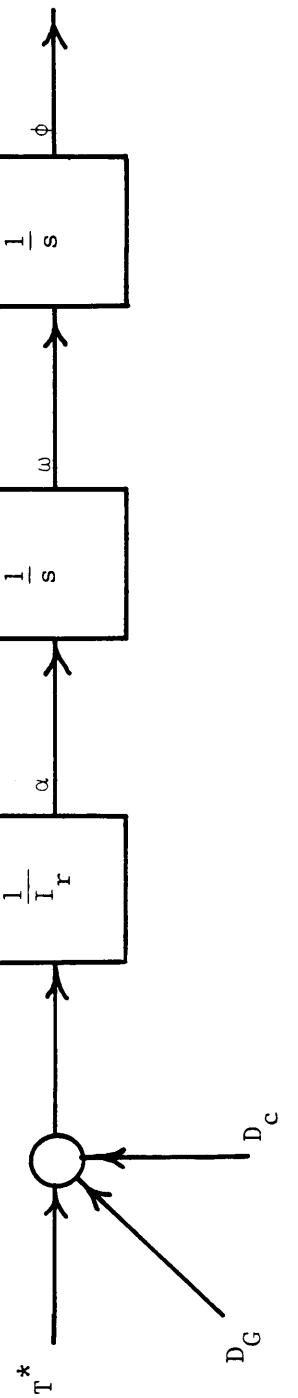


Fig. 2.4 Linear Four-Bar Dynamic Model Block Diagram

the forms of the disturbance torques exactly. The block diagram, Fig. 2.5, is a representation of the full nonlinear dynamics which depicts this knowledge. It also shows the nonlinear complexity disregarded in the linearized model. Therefore, the control objective should be to use the available information to reject the disturbances caused by the mechanism nonlinearities.

Several of the previous solutions to provide constant velocity have been to cancel the effects of these disturbances. For the gravity torque mechanism link balancing is effective. However the nonlinear centrifugal disturbance is the major problem. We can also note that the inertial of link 1, $\bar{I}_1$ does not appear in the centrifugal term. Therefore, adding flywheels to the input will not cancel the nonlinearities but only reduce their effects. The position-varying centrifugal disturbances can only be cancelled using a torque.

Figs. 2-6,7,8 are plots of the inertia variation, the centrifugal acceleration coefficient, and the gravity disturbance over one revolution of the experimental mechanism. The plots were generated using the Kinematic Influence Coefficient equations. In general, these parameters vary as sinusoidal functions of the input position. The inertia and centrifugal parameters vary at twice the input rotational frequency, therefore create torque disturbances at twice the input frequency. This imposes considerations for the adaptive control requirement for "slowly time-varying parameters." It also introduces the highest frequency disturbances into the control loop which is used to simplify the control system component models.

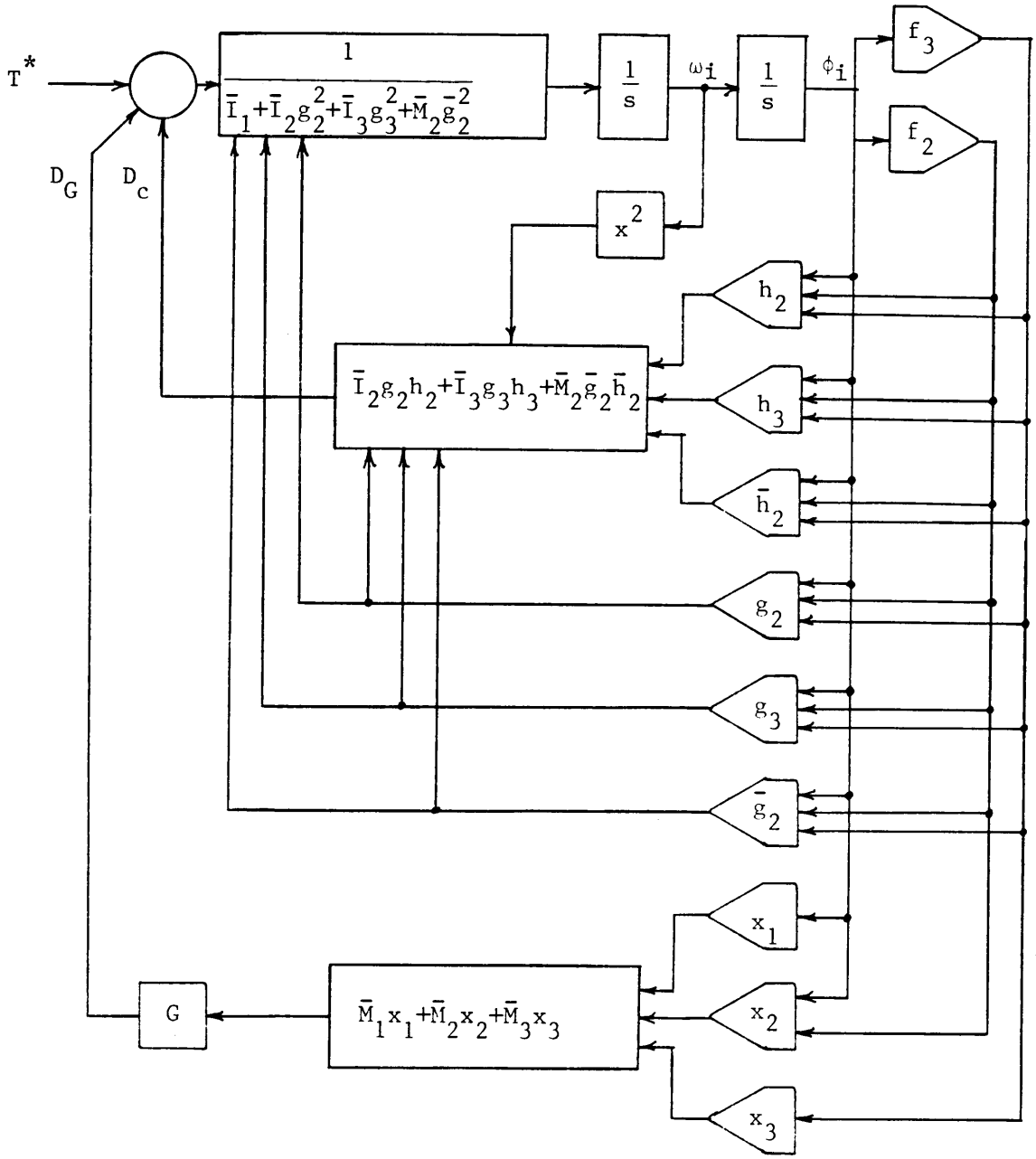


Fig. 2.5 Nonlinear Four-Bar Dynamic Model Block Diagram

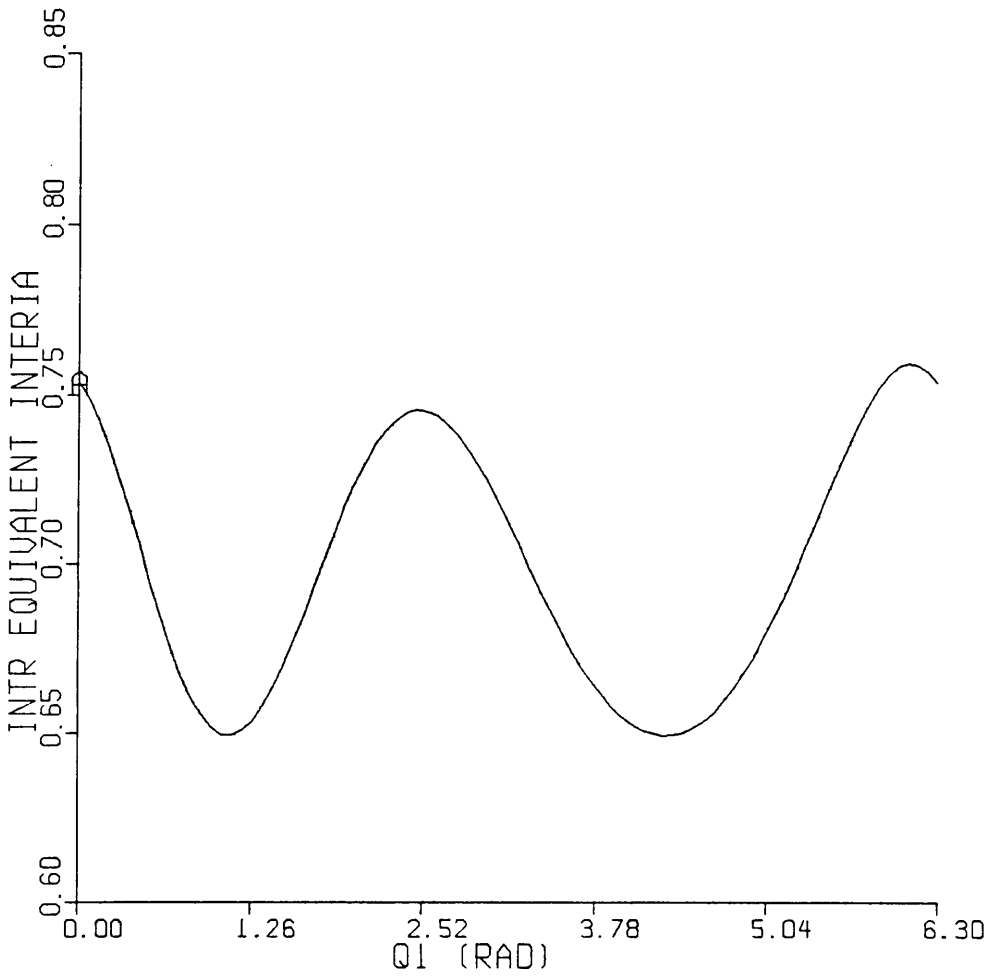


Fig. 2.6. Experimental Four-Bar Equivalent Inertia Variation

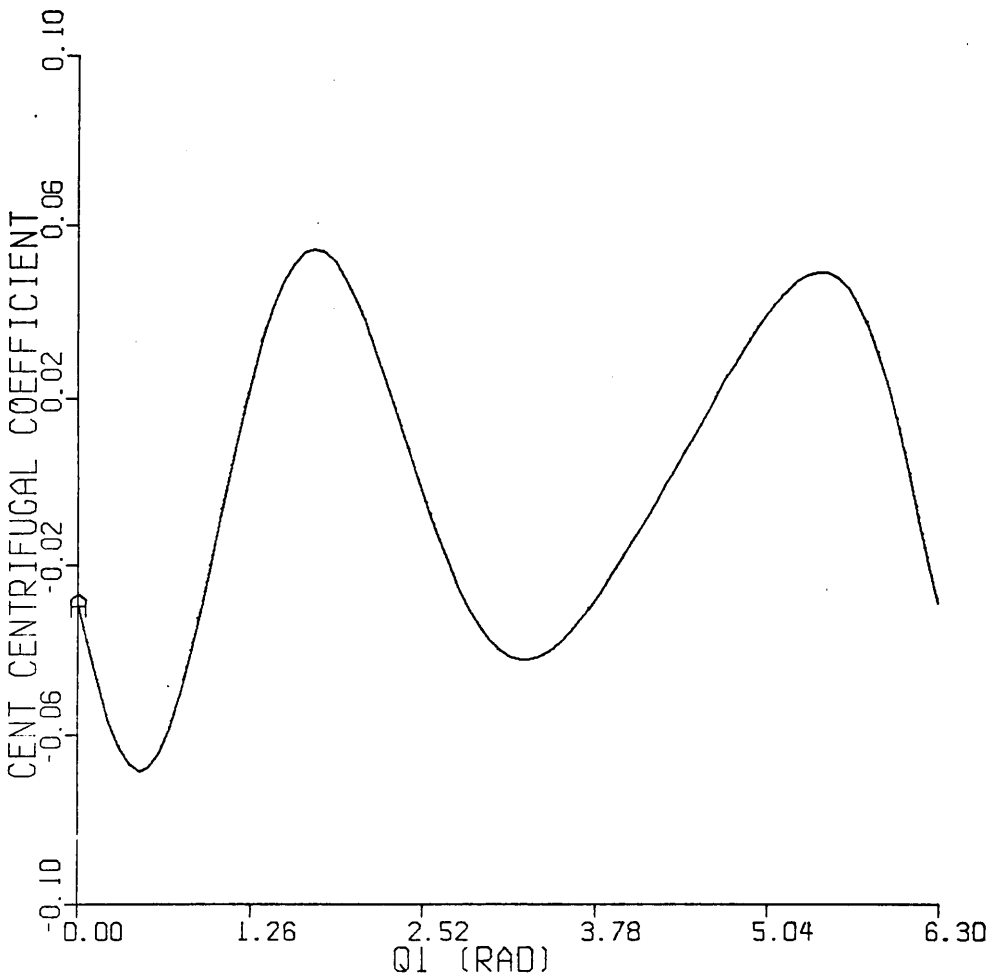


Fig. 2.7. Experimental Four-Bar Centrifugal Acceleration Coefficient

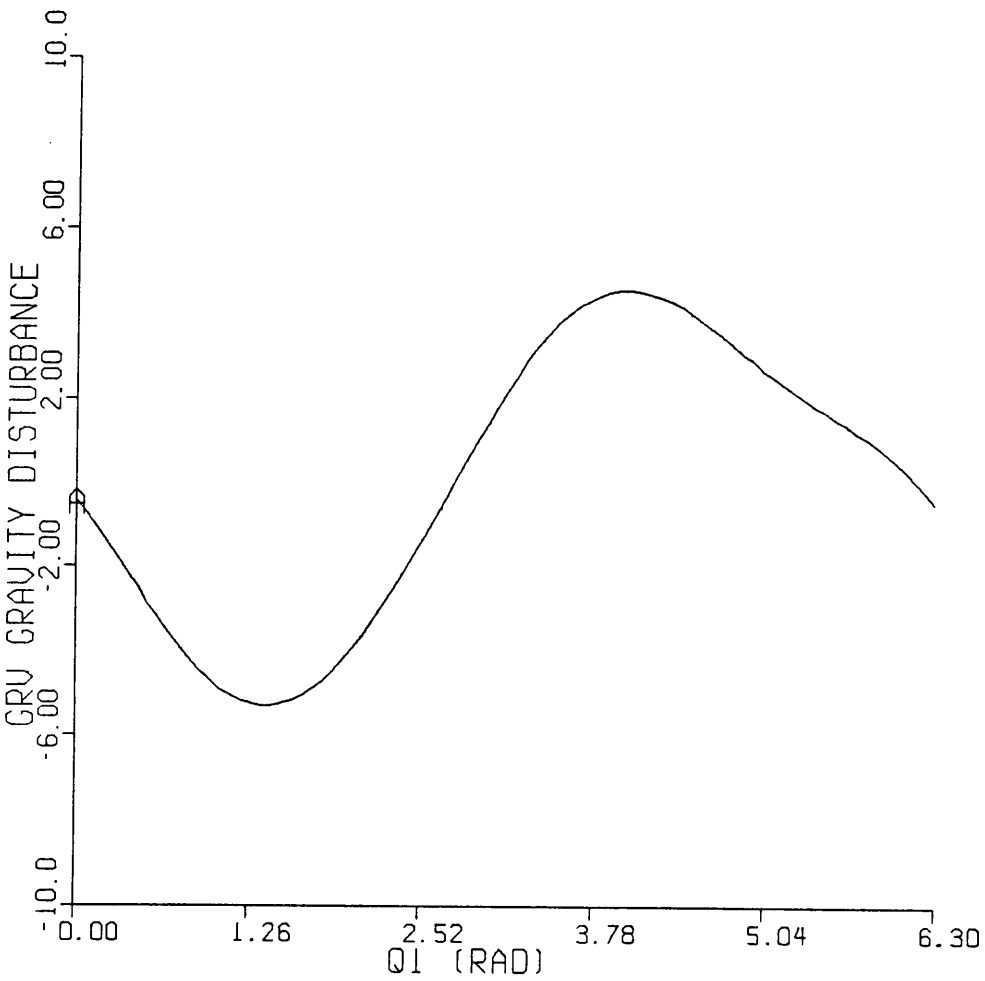


Fig. 2.8. Experimental Four-Bar Gravitational Disturbance Torque

3.0 Open Loop Dynamic Analysis

The open-loop mechanism control system is analyzed in three parts, the nonlinear four-bar linkage dynamics, the motor/amplifier actuator, and the optical encoder instrumentation. The four-bar dynamics are modeled after an experimental mechanism using kinematic influence coefficients from Section 2. The dynamic model is compared to an Euler-Lagrange formulation, and heuristically compared to experimental data. The motor/amplifier combination forms a closed-loop current control, and is subsequently modeled as a torque generator. The optical encoder serves as the feedback instrument and is modeled using digital considerations. These constituent parts are combined to form a computer simulation for testing the performance of the adaptive control solution.

3.1 Four-Bar Model

The four-bar linkage, Fig. 3.1, is the experimental mechanism used in this study. The link kinematic properties were easily determined, however the values of the link dynamic parameters ($\bar{I}$,$\bar{M}$) were estimated using link models. The nonlinear dynamics are modeled using the Kinematic Influence Coefficients and an equivalent mass-and-force method from Section 2.

The state space dynamic model equations 2.1-28,29 are repeated here for convenience.

$$\dot{\phi}_1 = \omega \qquad (3.1\text{-}1)$$

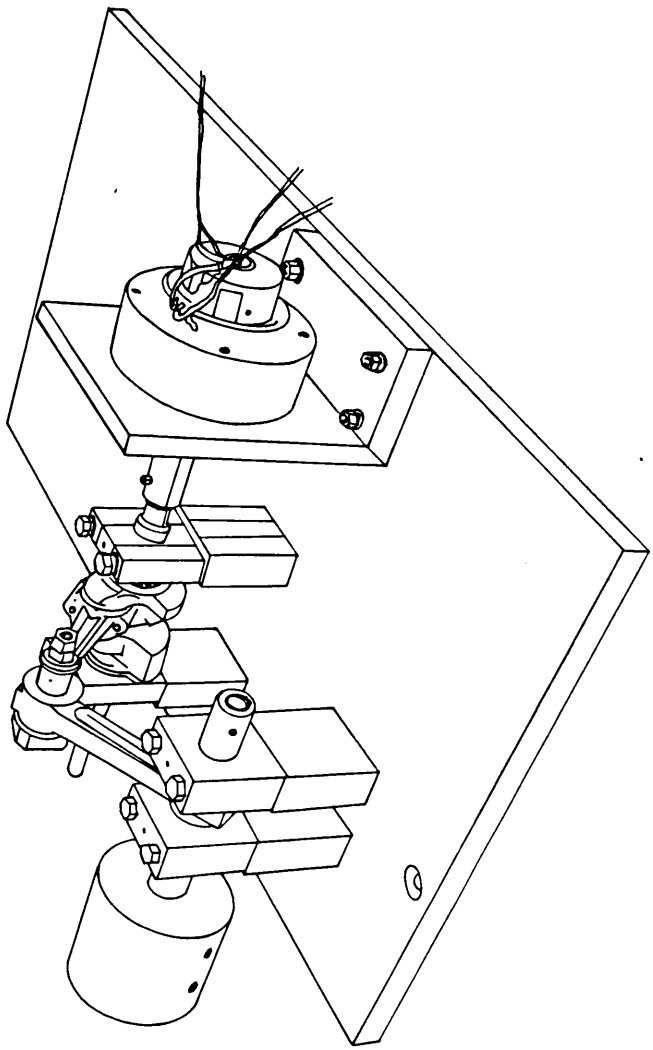


Fig. 3.1. Experimental Four-Bar Mechanism

$$\dot{\omega} = \frac{T^* - \frac{1}{2}\frac{dI^*}{d\phi_i}\omega^2 - T_G^*}{I^*} . \qquad (3.1\text{-}2)$$

In this section, the variable ω is the input rotational velocity and the argument of I^* is omitted. To complete the dynamic model, the Kinematic Influence Coefficients and functions for equivalent mechanism inertia, I^*, and gravity disturbances, T_G^* are given below.

$$I^* = (\bar{I}_1 + \bar{I}_2 g_2^2 + \bar{I}_3 g_3^2 + \bar{M}_2 \bar{g}_2^2) \qquad (3.1\text{-}3)$$

$$\frac{1}{2}\frac{dI^*}{d\phi_1} = \bar{I}_2\, g_2 h_2 + \bar{I}_3\, g_3 h_3 + \bar{M}_2\, \bar{g}_2 \bar{h}_2 \qquad (3.1\text{-}4)$$

$$T_G^* = -G(\bar{M}_1\, x_1 + \bar{M}_2\, x_2 + \bar{M}_3\, x_3) \qquad (3.1\text{-}5)$$

$$g_2 = \frac{a_1}{a_2}\frac{\sin(\phi_1 - \phi_3)}{\sin(\phi_3 - \phi_2)} \qquad (3.1\text{-}6)$$

$$g_3 = \frac{a_1}{a_3}\frac{\sin(\phi_1 - \phi_2)}{\sin(\phi_3 - \phi_2)} \qquad (3.1\text{-}7)$$

$$h_2 = \frac{a_1}{a_2}\left[\frac{\sin(\phi_3-\phi_2)\cos(\phi_1-\phi_3)(1-g_3)-\sin(\phi_1-\phi_3)\cos(\phi_3-\phi_2)(g_3-g_2)}{\sin^2(\phi_3-\phi_2)}\right] \qquad (3.1\text{-}8)$$

$$h_3 = \frac{a_1}{a_2}\left[\frac{\sin(\phi_3-\phi_2)\cos(\phi_1-\phi_2)(1-g_2)-\sin(\phi_1-\phi_2)\cos(\phi_3-\phi_2)(g_3-g_2)}{\sin^2(\phi_3-\phi_2)}\right] \qquad (3.1\text{-}9)$$

$$\bar{g}_2^2 = a_1^2 + r_2^2 g_2^2 + 2 a_1 r_2 g_2 \cos(-\phi_1 + \phi_2 + \theta_3) \qquad (3.1\text{-}10)$$

$$\bar{g}_2\bar{h}_2 = r_2^2 g_2 h_2 + a_1 r_2[h_2\cos(-\phi_1 + \phi_2 + \theta_3) - g_2\sin(-\phi_1 + \phi_2 + \theta_2)(g_2-1)] \qquad (3.1\text{-}11)$$

$$x_1 = r_1 \cos(\phi_1 + \theta_1) \tag{3.1-12}$$

$$x_2 = a_1 \cos\phi_1 + r_2 g_2 \cos(\phi_2 + \theta_2) \tag{3.1-13}$$

$$x_3 = r_3 g_3 \cos(\phi_3 + \theta_3) \tag{3.1-14}$$

Equations 3.1-1,14 are used to simulate the mechanism response to the input torque T^*.

The values for the simulated linkage parameters were determined from the experimental mechanism. The kinematic link lengths were measured directly however the dynamic parameters estimated using link models. These link models are made up of rods and counterweights. They are reviewed in detail in Appendix A. Fig. 3.2 is a schematic of the simulated mechanism. The link parameters used in the simulation are given in Table 3.1. Here, the coupling between the motor and the input link is assumed rigid. Therefore, the motor inertia, J, is added directly to the input link inertia.

3.2 Verification of the Influence Coefficient Model

The complete Kinematic Influence Coefficient model was compared with the Myklebust Euler-Lagrange formulation and heuristically compared to experimental data. When using the motor/mechanism system described in [3], the two formulations yield the same velocity response. Figs. 3.3 and 3.4 are velocity plots from digital integration of the Influence Coefficient and Euler Lagrange formulations. No gross differences are noticed.

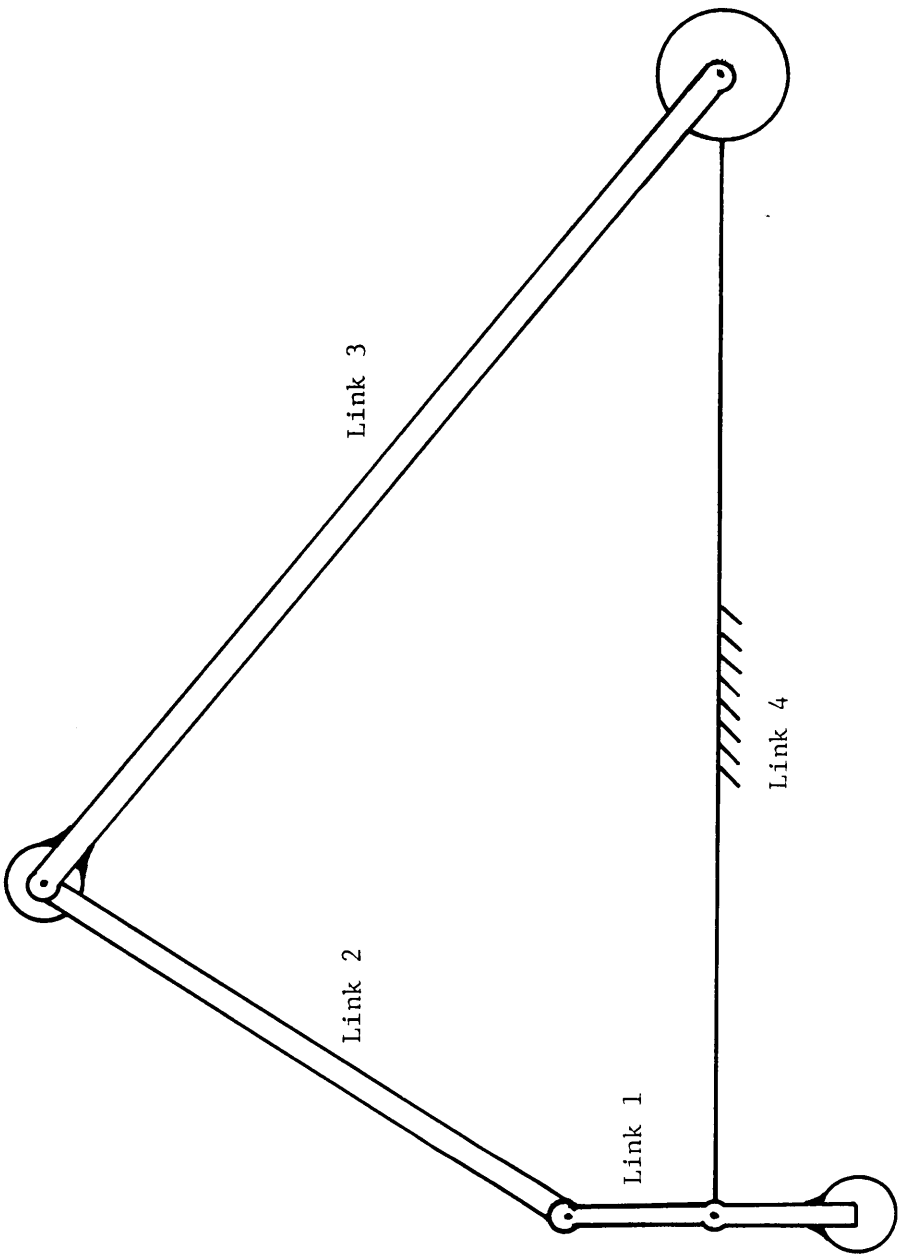


Fig. 3.2 Experimental Four-Bar Simulation Model

Table 3.1 Simulated Mechanism Parameters

Model Parameter	Customary	S.I.
a_1	0.75 in	1.9 cm
a_2	3.25 in	8.3 cm
a_3	5.5 in	14.0 cm
a_4	6.0 in	15.2 cm
* $\bar{I}_1$	0.648 oz-in-s^2	45.76 dyn-cm-s^2
$\bar{I}_2$	0.00635 oz-in-s^2	0.448 dyn-cm-s^2
$\bar{I}_3$	5.0 oz-in-s^2	353 dyn-cm-s^2
$\bar{M}_1$	0.0827 oz-s^2 in	905 gm
$\bar{M}_2$	0.007 oz-s^2 in	76.6 gm
$\bar{M}_3$	0.104 oz-s^2 in	1140 gm
r_1	0.375 in	0.95 cm
r_2	1.625 in	4.2 cm
r_3	2.06 in	5.2 cm
θ_1	3.1415 rads	
θ_2	0.0 rads	
θ_3	0.0 rads	

*Includes the motor inertia J

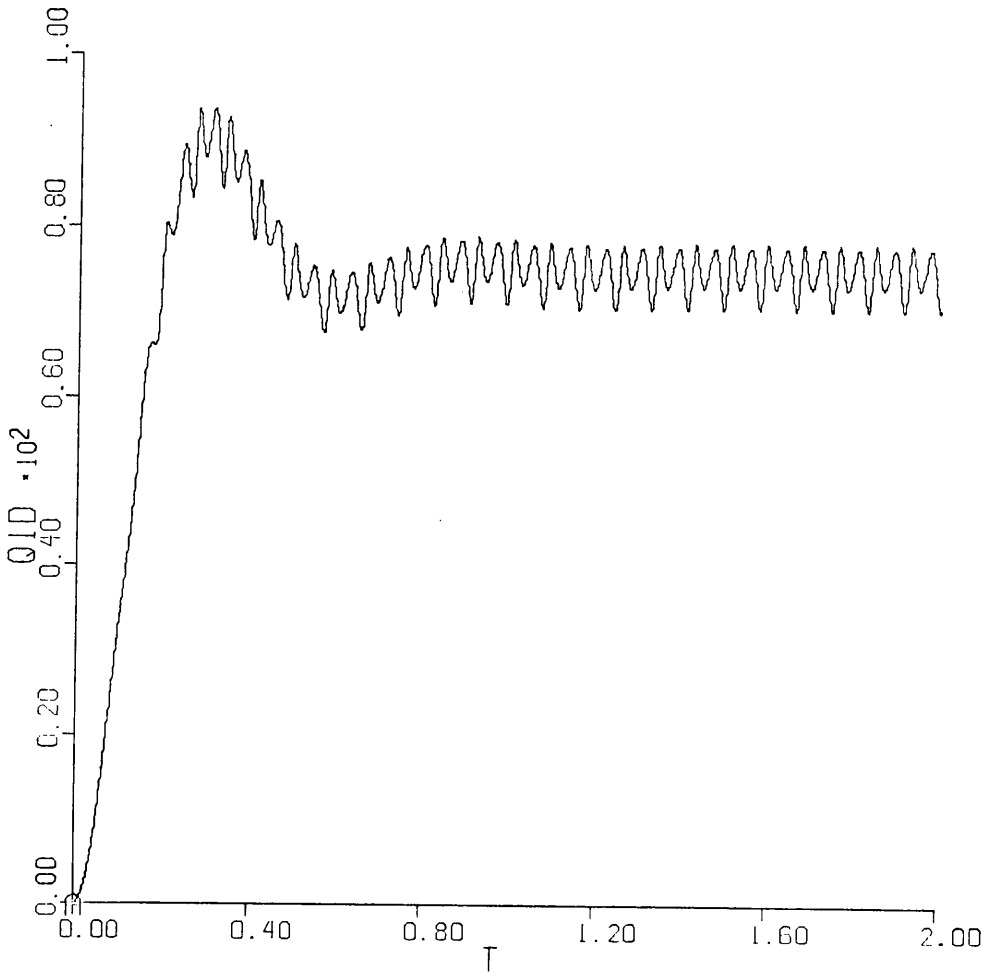


Fig. 3.3. Velocity Response, Influence Coefficient Model

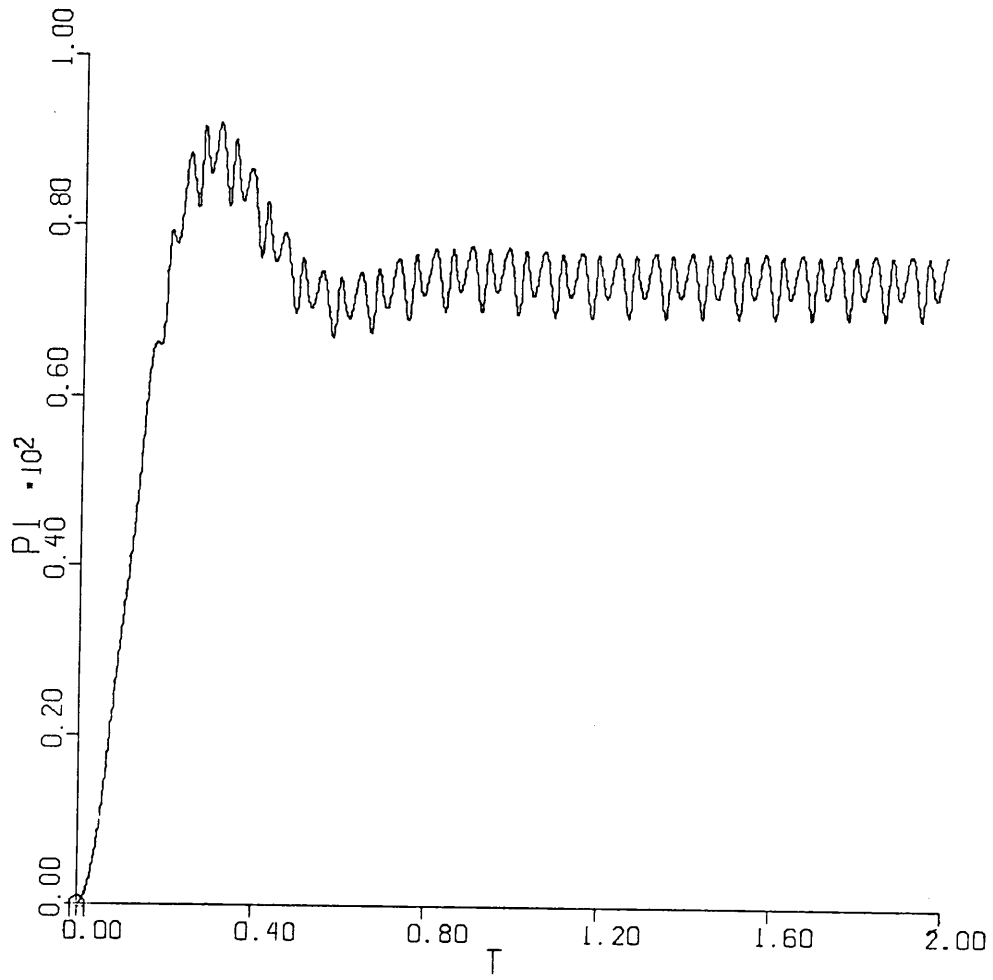


Fig. 3.4. Velocity Response, Euler-Lagrange Model

Data was collected from the experimental mechanism and PMI servo motor control system. The PMI controller has a built-in Lead-Lag controller which was used to form a closed velocity control loop. Real-time data for steady state, and start-up transient response was recorded using a ZONIC 5003 FFT, Figs. 3.5 and 3.7. Note that the velocity signals are inverted, therefore, the more negative the signal the faster the input rotation. This data was taken using the maximum feedback gain and the maximum cut-off frequency of the PMI Lead-Lag Controller. With this tuning, the Lead-Lag control behaves as a proportional control [6].

The author can only use heuristic arguments to compare the computer simulation to the experimental data. Observations at the time of the experiment revealed flaws in the mechanism bearings and joint connections. These flaws introduce varying frictional forces and impact noises into the experimental data. These are not accounted for in the simulation. The best comparisons are the basic shapes of the torque and velocity waveforms.

Figs. 3.5 and 3.6 are the velocity plots for the experimental mechanism and the simulation, respectively. These plots were made for the startup and steady-state response with a controller setpoint velocity of 105 rad/sec. Both plots show the characteristic uneven oscillation in the steady-state velocity with a frequency twice the rotational frequency. The torque plots, Figs. 3.7 and 3.8, also reveal similar waveforms. Using the correlation between the simulation and data waveforms, it can be concluded that the mechanism dynamic model is at least similar to the real experimental mechanism.

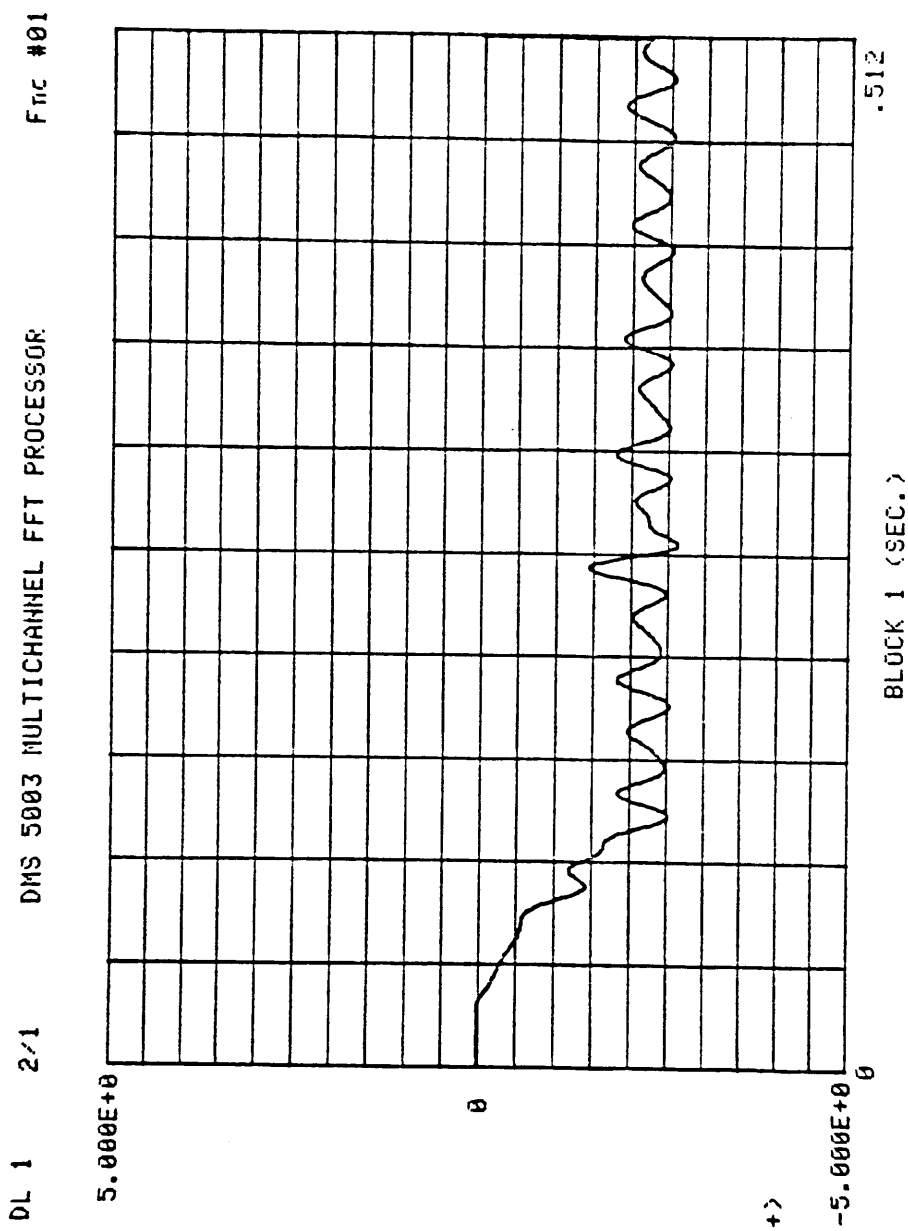


Fig. 3.5. Experimental Mechanism Velocity: PMI Control, Maximum Gain, 0 - Compensation

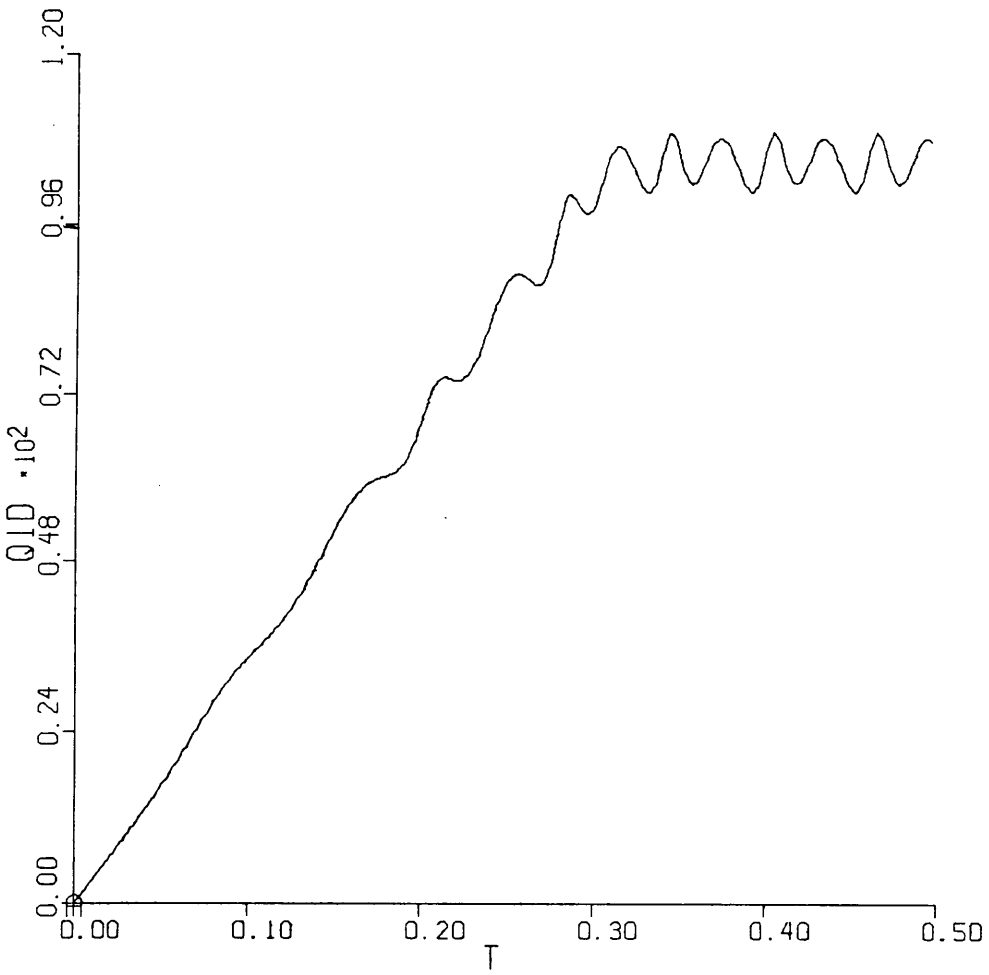


Fig. 3.6. Simulated Mechanism Velocity: Proportional Control, Forward Gain = 10

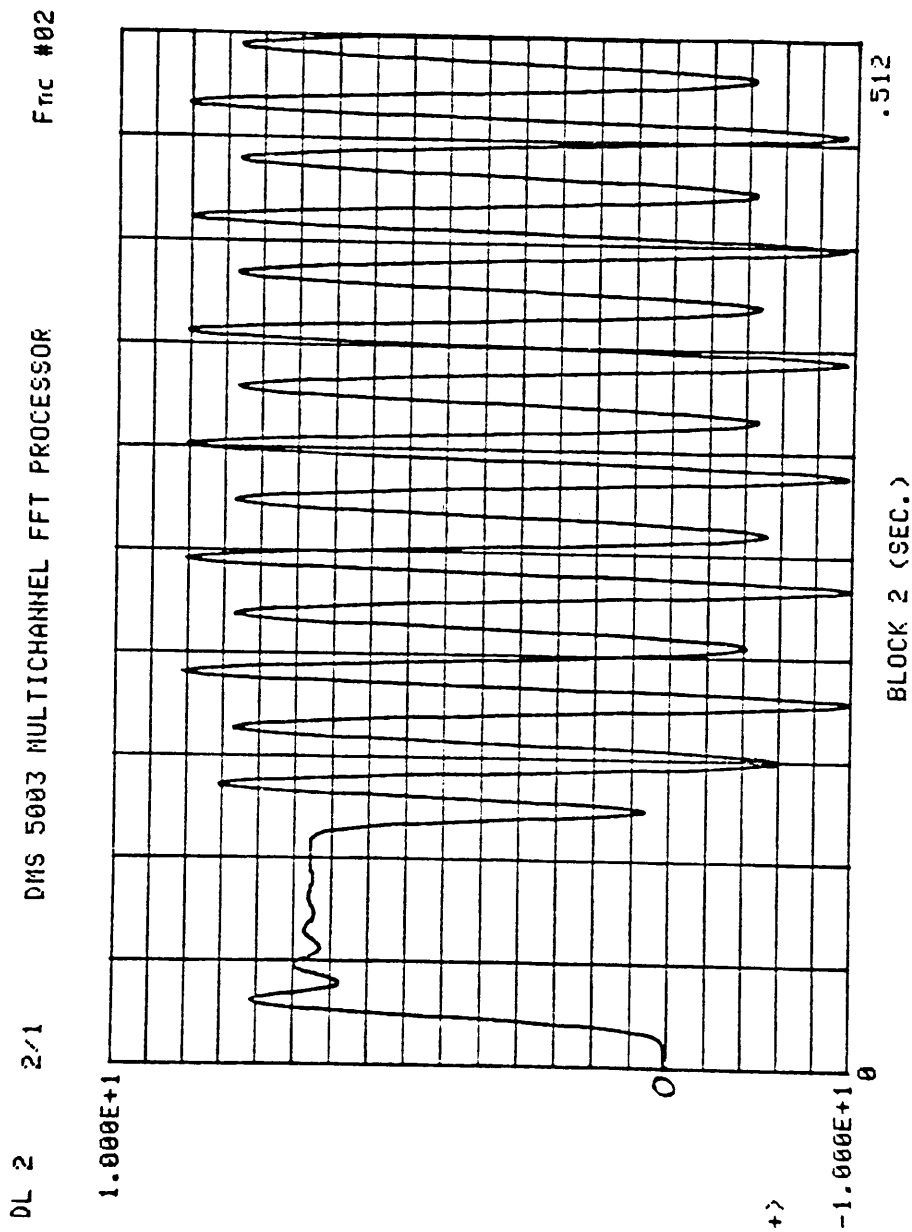


Fig. 3.7. Experimental Mechanism Applied Motor Torque: PMI Control, Maximum Gain, 0 - Compensation

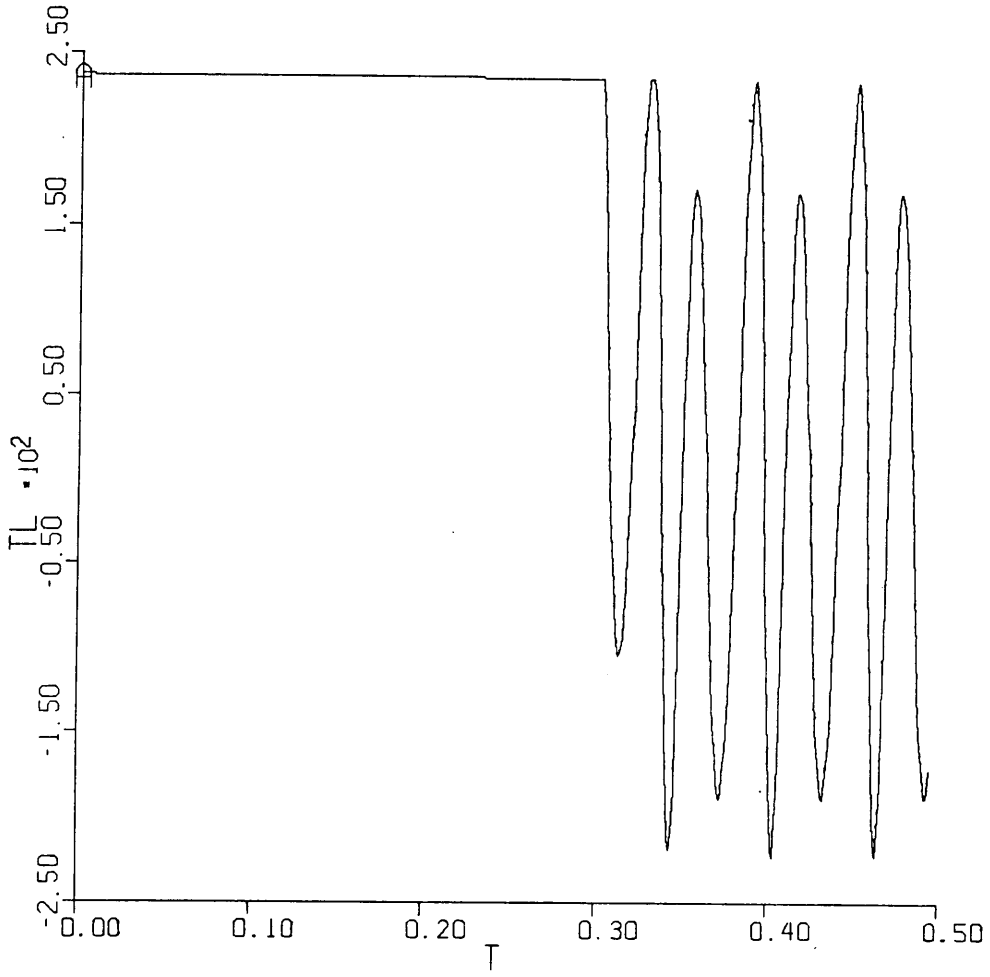


Fig. 3.8. Simulated Mechanism Applied Torque: Proportional Control, Forward Gain = 10

3.3 Actuator Model

The linkage actuator is a U94T permanent magnet D.C. servo-motor and companion switching servo amplifer produced by the PMI motor division of Kollmorgen Corporation [6]. The amplifier provides a motor current that directly follows an input signal. The permanent magnet motor acts as a linear convertor which produces a shaft torque proportional to the motor current.

The switching servo amplifier is described by PMI as a transductance amplifier which provides an output motor current porportional to an input voltage. The amplifier operates by switching the motor terminal voltage at a frequency of 5 kHz. Motor current is developed by varying the motor terminal voltage duty cycle. The terminal voltage waveform is feedback controlled to produce the desired motor current. Thus the motor electrical dynamics do not effect the input signal to output torque response of the motor amplifier system. PMI advertises a frequency response for this system to 1000 Hertz.

The motor amplifier is modelled as a torque-producing element with no internal dynamics. This model is sufficent considering a maximum Control Loop frequency of approximately 50 Hz (section 2.3) which is an order of magnitude slower than the motor/amp response. For completeness, the available torque is bounded to model the amplifier $\pm$ 30 amp current limits. The motor is constructed with internal friction bearings. These provide the only source of rotational dampening in the motor-mechanism system. Table 3.2 summarizes the motor specifications [7].

The simulated experimental mechanism and the system model equation

Table 3.2 PMI Motor Specifications

Torque Constant	6.1	oz in/amp	431 dyn-cm/amp
Rotor Inertia	0.008	oz in s^2	0.565 dyn-cm-s^2
Damping Coefficient	0.01146	oz in/s	0.809 dyn-cm/s
Back emf Constant	0.04297	volts/s	
Average Friction Torque	6.0	oz in	424 dyn-cm
Armature Inductance	100	μ-henries	
Average Terminal Resistance	1.025	ohms	
Tachometer Constant	0.02149	volt-s	

2.3-1, were used to calculate the required torque to exactly follow two known velocity functions. The two functions are the test input functions for the closed control loops. The objective is to anticipate the required actuator torque to control the mechanism.

The first function is a step increase in velocity from 50 rad/sec to 75 rad/sec. Fig. 3.9 is the required torque plot. Theoretically, an infinite amount of torque is required to accelerate the mechanism for the step change at .5 sec. Since, this amount is never available, the requirement is omitted. However, the plot shows the required torque to maintain the constant 50 rad/sec and 75 rad/sec velocities. The peak torque fluctuations for 50 rad/sec are +140, -175 in-oz (9900, -12.4 x 10^3 cm-dyn). The peak fluctuations for 75 rad/sec are +310, -390 in-oz (22 x 10^3, -27.5 x 10^3 cm-dyn).

The second function is a ramp function beginning at 50 rad/sec and ramping to 75 rad/sec at an acceleration of 75 rad/s^2 between time 0.1 sec and 0.4333 sec. Fig. 3.10 is the required torque plot. The peak required torque is approximately the same as the 75 rad/sec constant velocity peak torque.

Using the ± 30 ampere amplifier current limit, the actuator model is only capable of producing ± 183 in-oz torque (12.9 x 10^3 cm-dyn). This is sufficient for the 50 rad/sec velocity, but only half the required torque for the 75 rad/sec velocity. Therefore, the simulated actuator is modeled using two torque limits. The first is the straight PMI specifications for ± 183 in-oz (12.9 x 10^3 cm-dyn). The second is an arbitrary limit of ± 396 in-oz (28.0 x 10^3 cm-dyn) torque. These limits are used to demonstrate the effect of the limiting actuator on

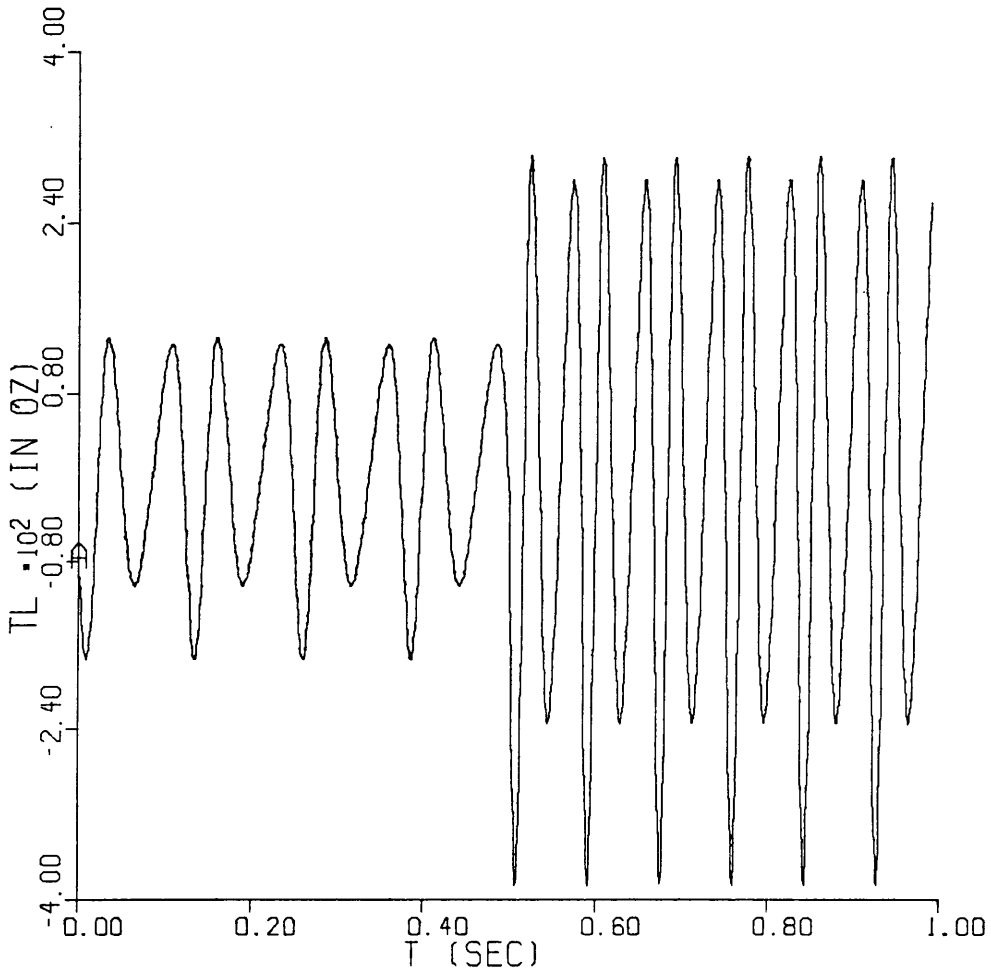


Fig. 3.9. Ideal Torque Requirement for Step Velocity Function

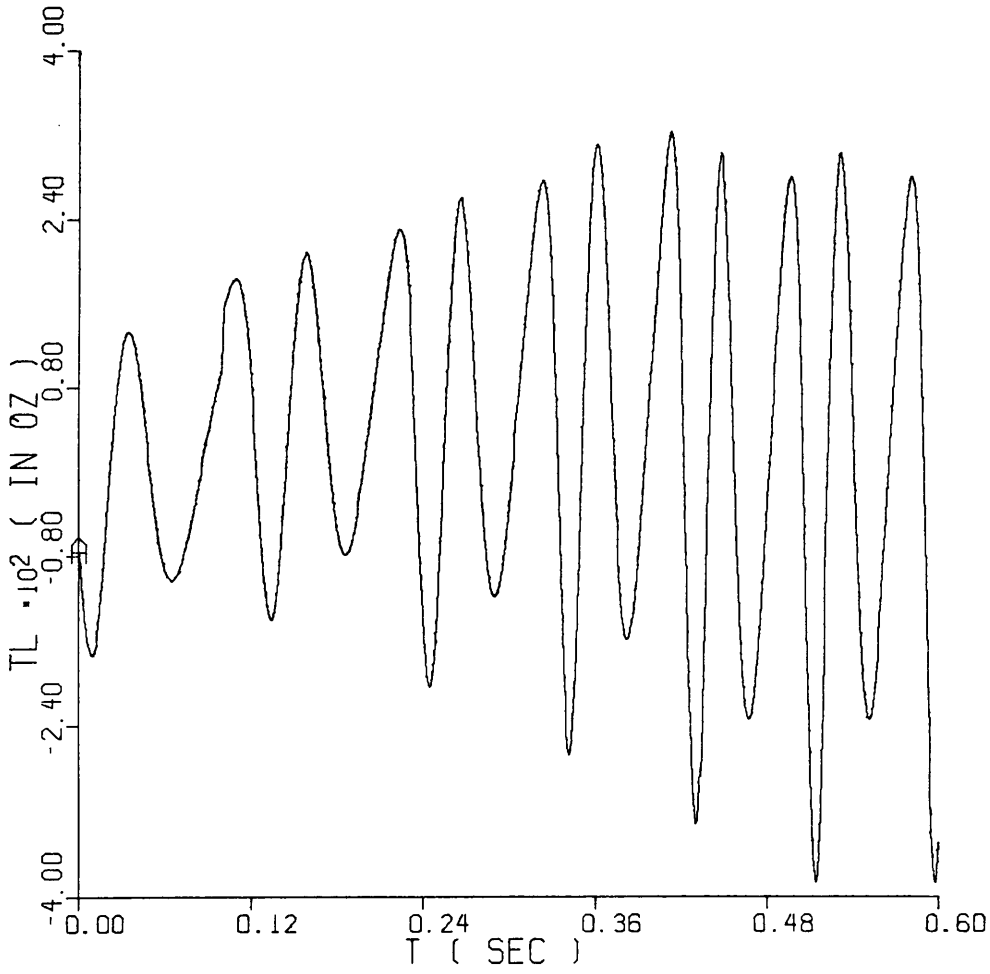


Fig. 3.10. Ideal Torque Requirement for Ramp Velocity Function

the overall closed loop control response.

3.4 Feedback Instrument Models

The measured feedback quantities for the discrete time controls are motor current (from the PMI amplifier), input position and velocity. The motor current, thus torque, is measured using an 12 bit analog-to-digital (A/D) converter. The converter is modelled by quantizing the applied torque with a resolution of 1 part in 4096 with sign.

The optical encoder with the interface logic [8] provides an all digital means for measuring the input position and velocity. The resolution of these measurements are directly related to the number of lines on the optical encoder. For a 1000 line encoder, the position is measured to 1 part in 4000 or .0016 radians. Therefore, the mechanism position is considered to be exactly measured, and no quantization effects are used to simulate the measurement. However, the method used to measure the velocity is to measure the time required to rotate over a known arc. This is accomplished using one data channel from the optical encoder. The data is present as square waves where one cycle is the arc of one encoder line. This signal is divided by 2 to yield the most repeatable measurement of an encoder interval. For a 1000 line encoder, this yields 500 velocity measurements per revolution.

The velocity logic measures the elapsed time for an interval to pass. This time is measured by counting a known clock and reading the count. The measured velocity, V, is then

$$V = \frac{D}{NT'} \, . \qquad (3.4\text{-}1)$$

where

D = .00638 rads

N = Number of clock periods

T' = The period of the clock (sec)

The counter is rising edge sensitive, therefore, the greatest error is minus one count, a missed count of the clock.

The simulated velocity measurement is calculated using equation 3.4-1 where the clock period is 1×10^{-7} seconds, the maximum counter limit is 4095, (12 bits) and the encoder arc is 0.00638 radians. These specifications reflect the 1000 line PMI optical encoder read with a 12 bit counter and a 10 megahertz clock. The counter value is calculated using

$$N = \text{INTEGER}\left(\frac{D}{\omega\, T'}\right) . \qquad (3.4\text{-}2)$$

where

ω = Actual velocity (rad/sec)

If the count, N exceeds 4095, the velocity logic assumes an infinite time has elapsed, and the measured velocity is zero. Otherwise, the measured velocity is

$$V = \frac{D}{N\, T'} , \qquad (3.4\text{-}3)$$

which is used by the discrete-time controller. For the velocity logic modeled, the minimum measured velocity is 15.3 rad/sec.

3.5 Computer Simulation

The models for the mechanism, motor, and instruments are combined to form a computer simulation of the open-loop control response. This simulation was implemented using Advanced Continuous Simulation Language (ACSL) [9].

ACSL provides a readily available method for integrating continuous nonlinear differential equations and generating graphic output. ACSL relieves the burden of developing and testing custom integration routines. It also provides built-in functions for discrete-time sampling and quantization. The ability to call FORTRAN subroutines gives ACSL a distinct advantage when working with discrete-time control simulations. Here, the control calculation can be implemented in a separate subroutine and called from a DISCRETE function block.

The ASCL model is divided into two sections. The continuous section integrates the four-bar dynamic model to yield the response to the control torque TL. The discrete section models the sampling function of the digital controller and calculates the modeled instrument measurements. The actual control law is calculated in a separate FORTRAN subroutine. This subroutine accepts the instrument measurements and produces the control torque which is then delayed one sample interval. The time delay is representative of the calculation delay present in the actual system. For the case of the continuous proportional control, the discrete function is not used and the control law is contained in the continuous section.

4.0 Standard Linear Control

The standard velocity control system is a model of the PMI SSA-30 servo control amplifier [6]. The control loop block diagram, Fig. 4.1, shows the built in lead-lag filter and adjustable forward gain, K_{FG}. The lead-lag filter accepts a reference input voltage and the U94MT tachometer voltage to close the control loop. The lead-lag network has an adjustable pole between .1 Hz and 28 Hz, a fixed zero at 28 Hz and a fixed pole at 1500 Hz which is an internal filter and omitted here. An analysis for this system shows the best performance is for a proportional control. The proportional control response is used as a comparison for the adaptive control strategies. Simulation cases for two input signals are used to establish this baseline performance.

4.1 Linear Control Analysis

The linear control analysis was performed using the linear four-bar model from Section 2.3 and the lead-lag control equation

$$\frac{V}{R} = \frac{P\ s + 1}{Q\ s + 1}$$

from [6]. the value for P is a fixed zero at 28 Hz. The value of Q can be varied to place a pole between .1 and 28 Hz. The control loop block diagram, Fig. 4.1, portrays the system dynamics for the velocity input ω^d and the disturbance input D. The velocity input is a step or ramp function. The disturbance, D, is classified in section 2.4 as a sinusoid with frequency 2 ω_1.

The requirement for the best performance, Section 2.3, is to reject

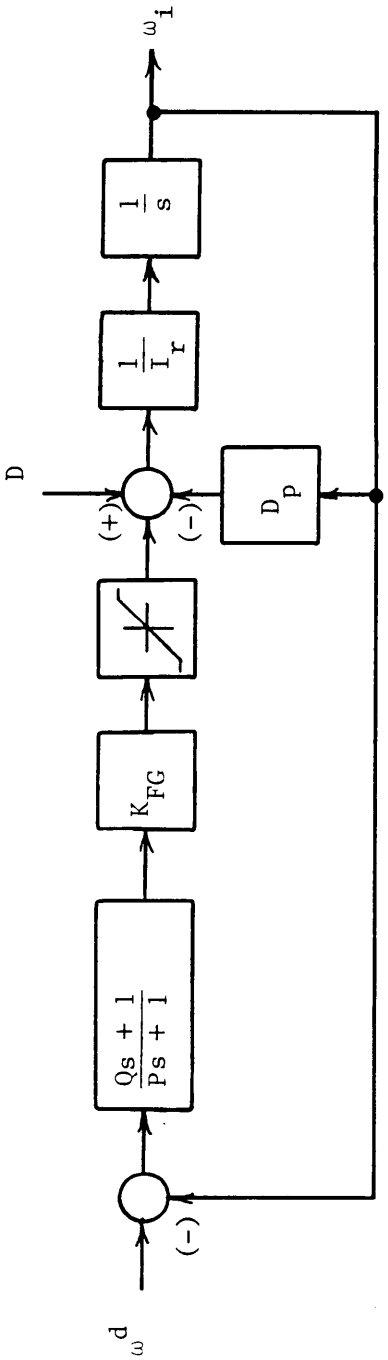


Fig. 4.1 Standard Control Block Diagram

the disturbance torque. Using Mason's gain formula, the disturbance transfer function is

$$\frac{\omega_i(s)}{D(s)} = \frac{1/I_r}{s + \frac{D_P}{I_r} + \frac{K_{FG}}{I}\left(\frac{P\,s + 1}{Q\,s + 1}\right)} .$$

The final value theorem yields the steady-state response to a step test disturbance signal as

$$\lim_{s \to o} \; s\left(\frac{1/I_r}{s + \frac{D_P}{I_r} + \frac{K_{FG}}{I_r}\left(\frac{P\,s + 1}{Q\,s + 1}\right)}\right)\frac{1}{s} = \frac{1}{D_P + K_{FG}}$$

Therefore, to reject the nonlinear disturbance the forward gain or friction damping should be increased. Since friction requires more input torque, increasing D_P is ill-advised.

From heuristic arguments, the adjustable pole is tuned to cancel the fixed zero to form the proportional control. The argument is that the 2 ω_i velocity fluctuations will be present in the error signal. If the controller filters these fluctuations out, then no opposing motor torque will be applied to correct the mechanism velocity. Therefore, poor performance will result.

Using the preceding analysis, the best standard velocity control, for the four-bar, is a proportional control using the maximum gain the motor current limits will allow. This conclusion was used to setup the linear control simulation. Using the PMI specifications, the lead-lag pole zero coefficients are $P = 5.684 \times 10^{-3}$, $Q = 6.6314 \times 10^{-3}$, which

are nearly cancelling.

The proportional control input response is used to estimate the lead-lag response. The proportional transfer function

$$\frac{\omega}{\omega^d} = \frac{K_{FG}/I_r}{S + \frac{K_{FG} + D_P}{I_r}}$$

which is a first-order system. Again, the best performance (for best transient response) is for a large forward gain.

4.2 Performance Measurement

The linear control performance is used as the reference performance for comparing the adaptive control performance. Each mechanism control system response is simulated for a step and a ramp input function. This combination yields the dynamic response for a discontinuous input and a tracking response. It also shows the ability of the control to perform at different input velocities.

Performance is measured using a percentage fluctuation measure and rise time. General comments are used to briefly describe the tracking behavior. Percentage fluctuation is defined as the peak-to-peak velocity fluctuation divided by the average or set point velocity,

$$\% F = \frac{F_{p-p}}{\omega_{AVG}} \times 100 \ . \qquad (4.2\text{-}1)$$

This measurement was chosen to represent the control scheme ability to maintain the linkage constant speed input condition. Rise time to a

step input is used to measure the control response speed. Here, rise time is defined as the time for the input velocity to accelerate from 50 rad/sec to 75 rad/sec. The time is measured from the input step to the instant the input velocity reaches 75 rad/sec. The controllers are also tested for tracking a 75 rad/sec^2 acceleration. The input waveform begins at 50 rad/sec velocity and ramps to 75 rad/sec at the required 75 rad/sec^2 acceleration.

The controller/mechanism responses are simulated for the PMI specification 183 in-oz (12.9×10^3 cm-dyn) motor torque limits, and for the 396 in-oz (28×10^3 cm-dyn) torque limit from the torque requirement of section 2.4. These different torque limits are used to investigate control under limited actuator power.

4.3 Simulation Studies

The ACSL simulation STANDARD, Appendix B, was used to evaluate the linear control response. The first case is for the lower torque limits and a forward gain $K_{FG} = 10$. The mechanism velocity and control setpoint are plotted together, Fig. 4.2. From this plot, the system has the following performance.

% F50 = 6%
% F75 = 8%
Rise Time = 0.1 sec

Where, % F50, % F75 are the percentage fluctuations at 50 rad/sec and 75 rad/sec, respectively. The torque plot (Fig. 4.3) shows that most of the available torque is used; therefore, the system is tuned for maximum

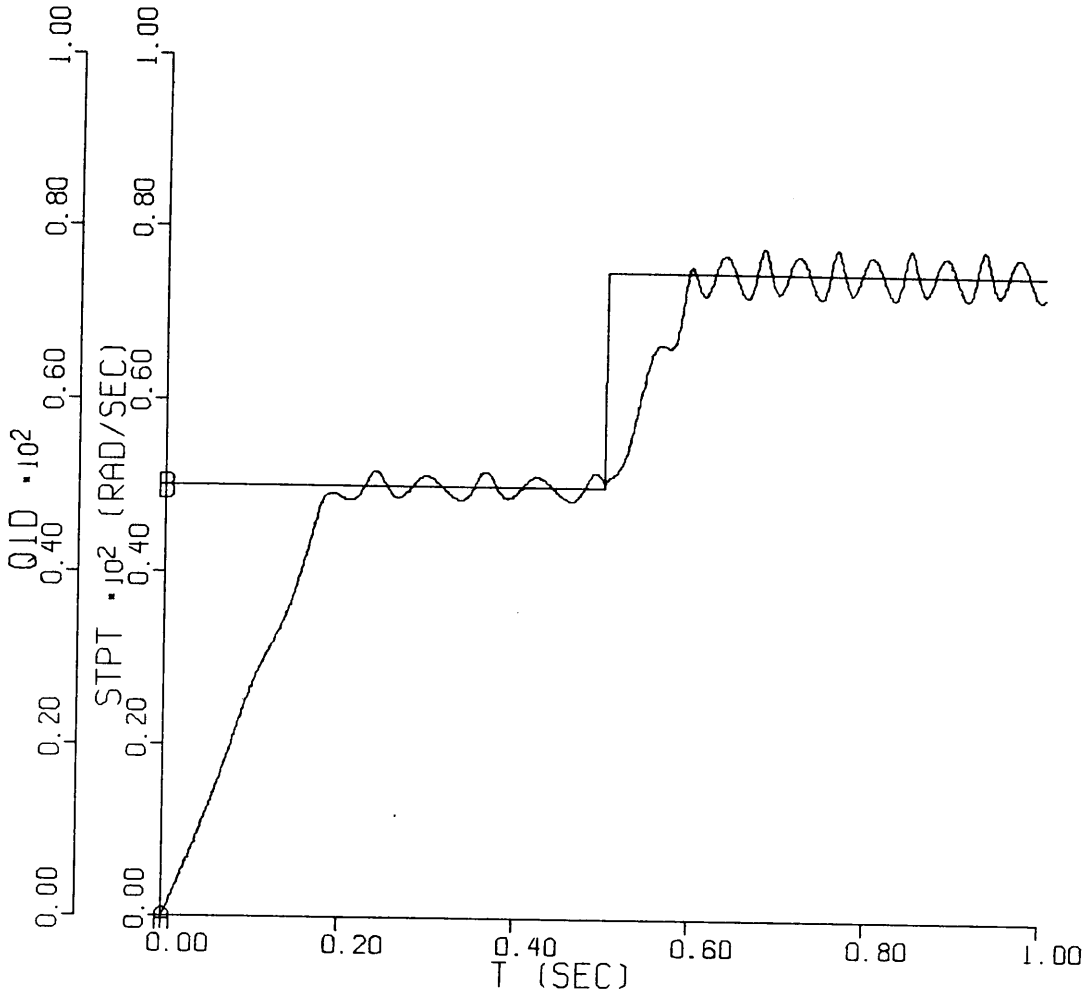


Fig. 4.2. Standard Control Velocity: K_{FG} = 10, Torque Limit = 133, Step

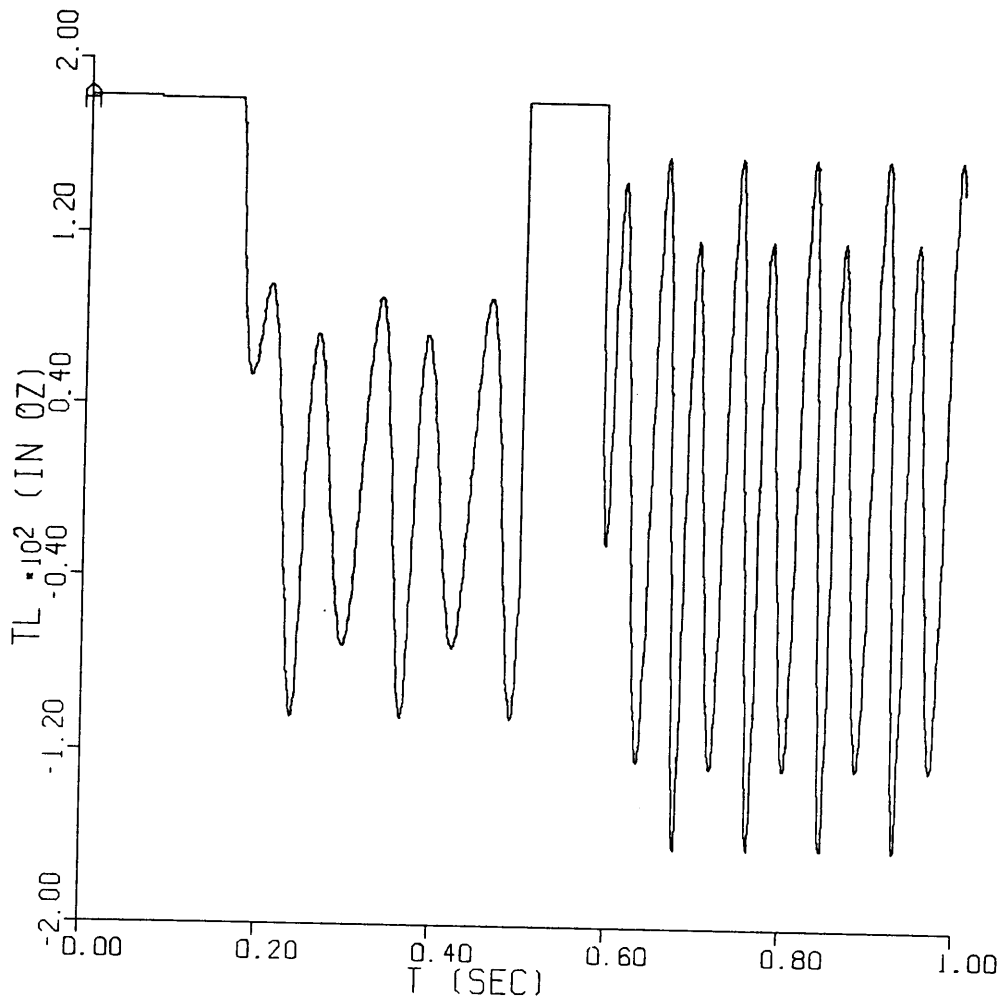


Fig. 4.3. Standard Control Torque: K_{FG} = 10, Torque Limit = 183, Step

performance. The torque saturation during start-up and at the step change are typical characteristics of the limited actuator. The tracking ability for this control is typical for a first-order system. Some steady-state error is present (Fig. 4.4), and there are no overshoots for either input signal. The ramp input torque plot (Fig. 4.5), is given for reference.

The second case is for the 396 in-oz (28×10^3 cm-dyn) torque limits, and a forward gain $K_{FG} = 35$. As expected, the fluctuations (Fig. 4.6),

$$\% \text{ F50} = 4\%$$
$$\% \text{ F75} = 5\%$$
$$\text{Rise Time} = .04 \text{ sec}$$

and the rise time are smaller. Again as expected the tracking ability (Fig. 4.7) is enhanced but still shows the first order character. The torque plot, Fig. (4.8), show the maximum torque was not used to correct the velocity fluctuations. Other values for the forward gain were evaluated with little change in the overall results. The torque plot (Fig. 4.9) for the ramp input shows similar results.

The linear simulation cases were used to establish the reference to compare the adaptive controllers. The implication is, to be worthwhile the adaptive controls need to improve the currently available performance.

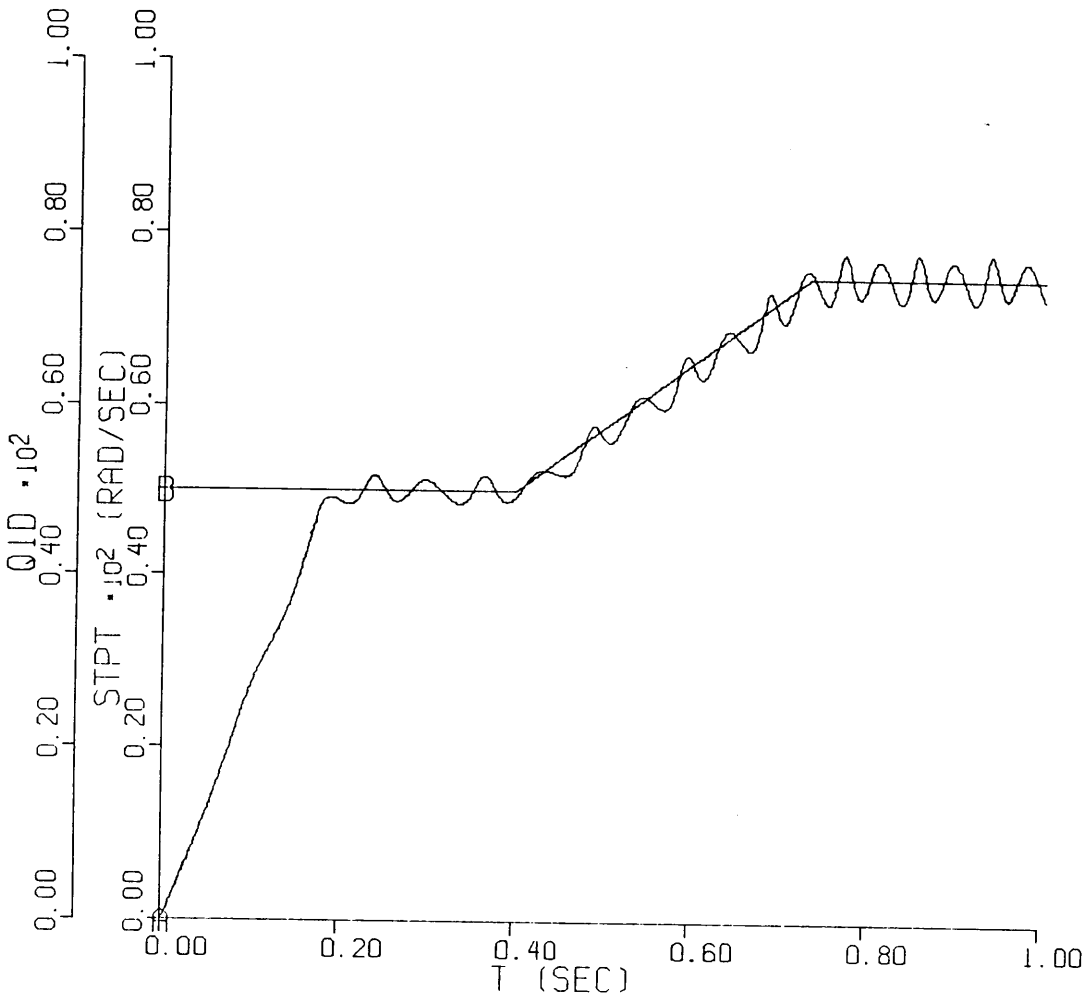


Fig. 4.4. Standard Control Velocity: K_{FG} = 10, Torque Limit = 183, Ramp

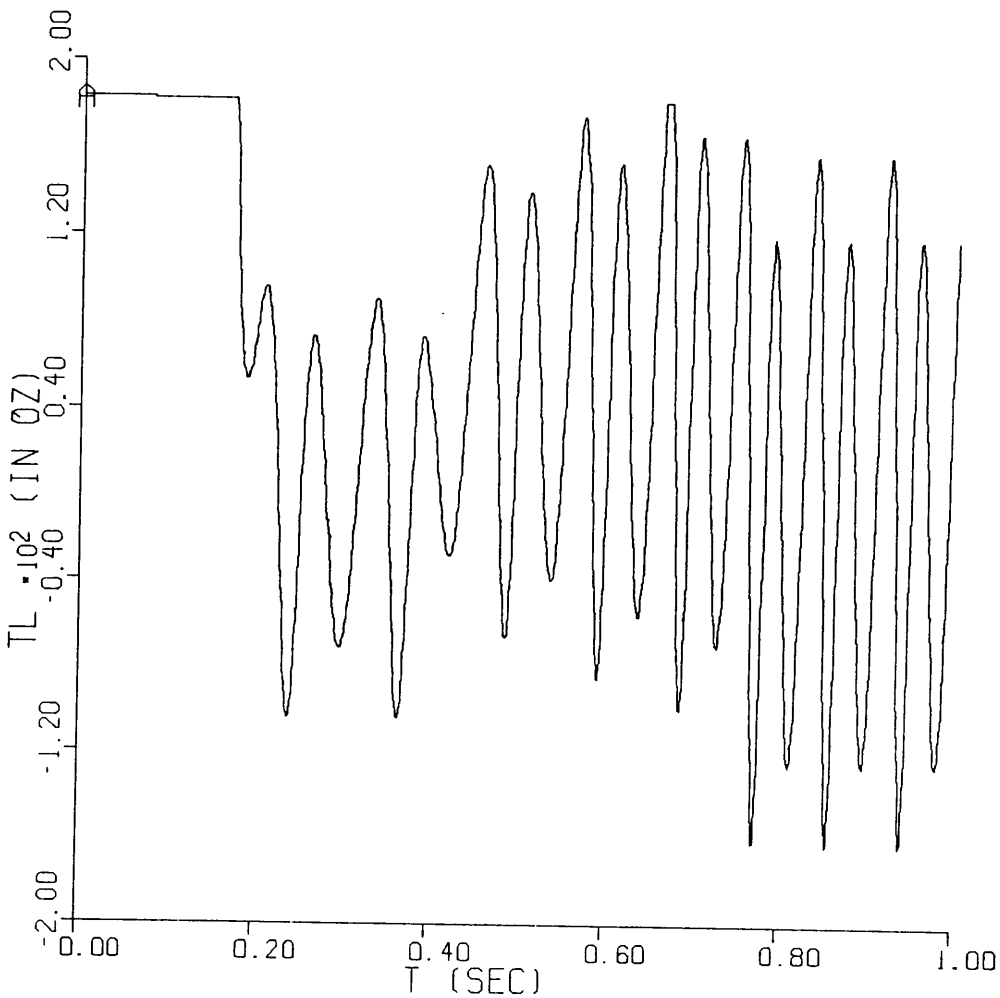


Fig. 4.5. Standard Control Torque: K_{FG} = 10, Torque Limit = 183, Ramp

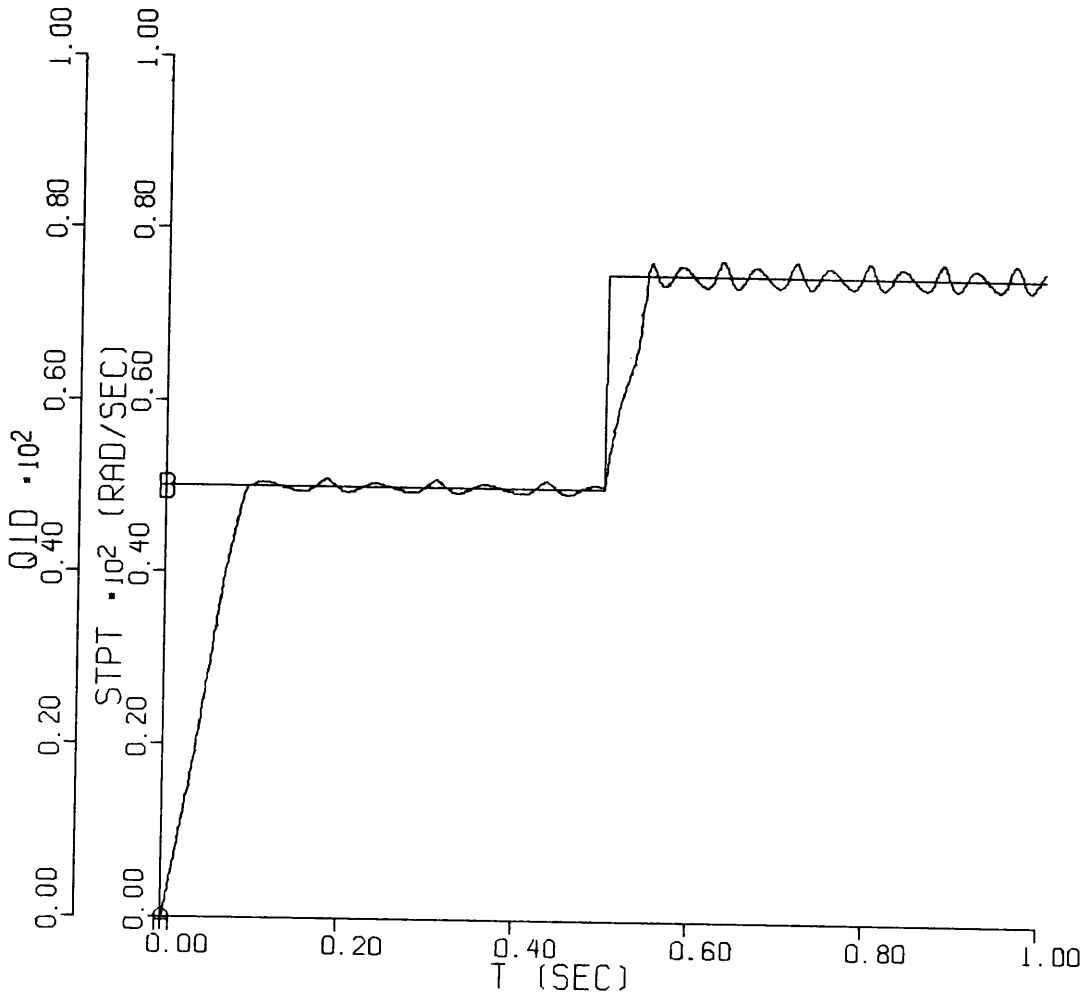


Fig. 4.6. Standard Control Velocity: K_{FG} = 35, Torque Limit = 396, Step

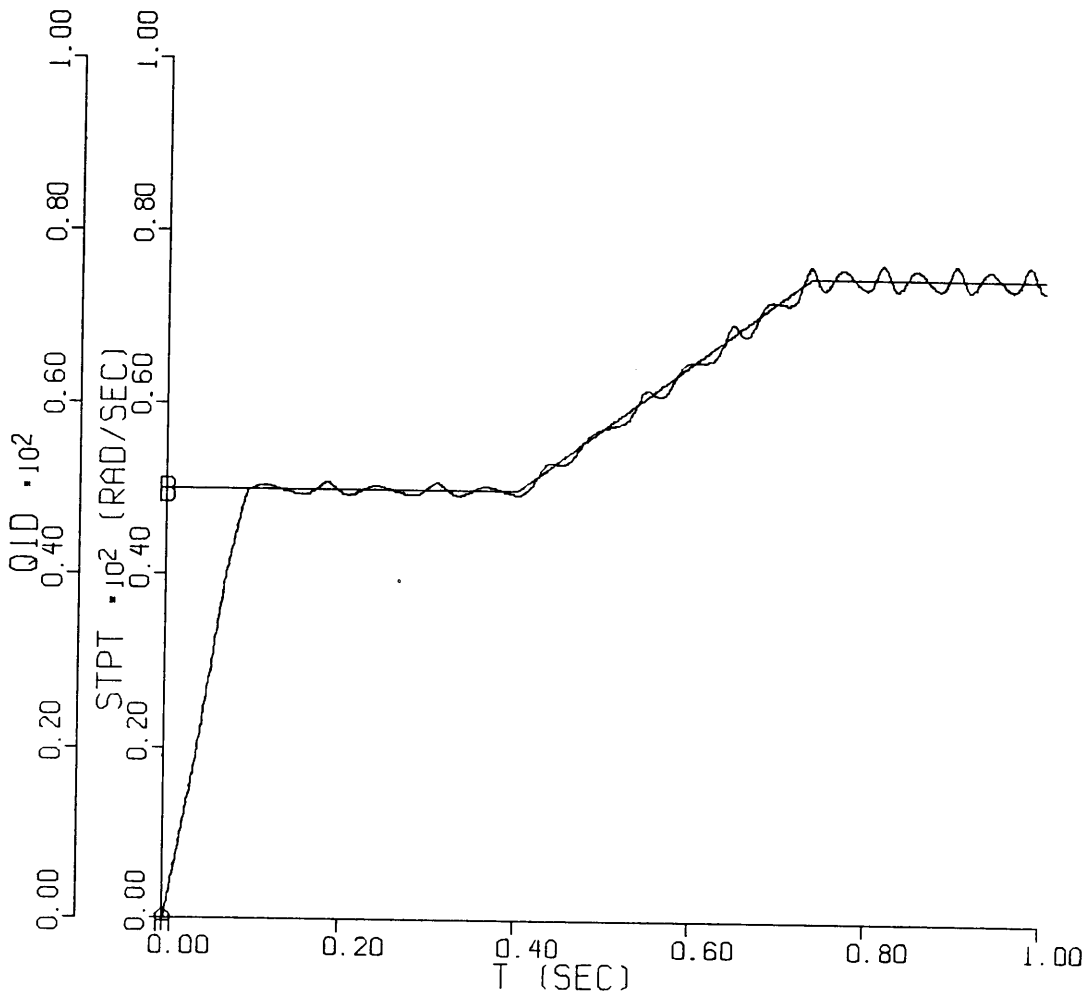


Fig. 4.7. Standard Control Velocity: K_{FG} = 35, Torque Limit = 396, Ramp

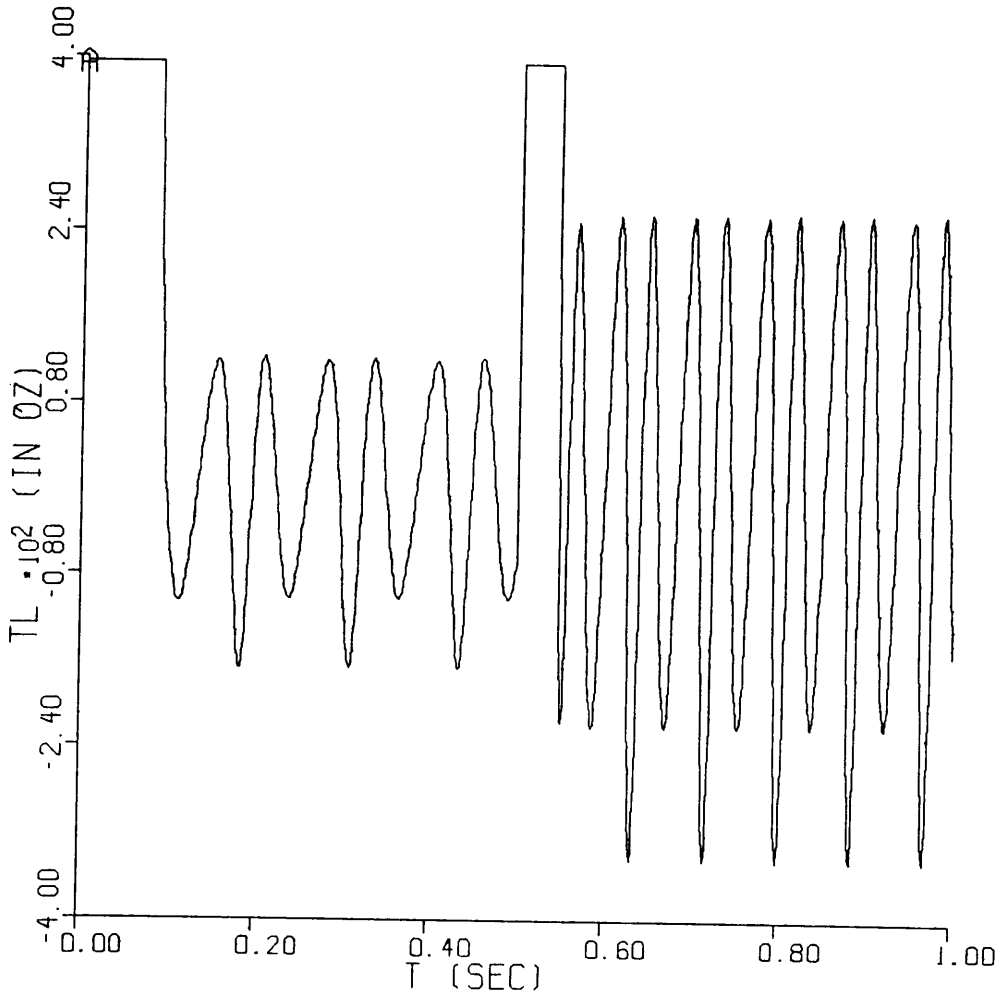


Fig. 4.8. Standard Control Torque: K_{FG} = 35, Torque Limit = 396, Step

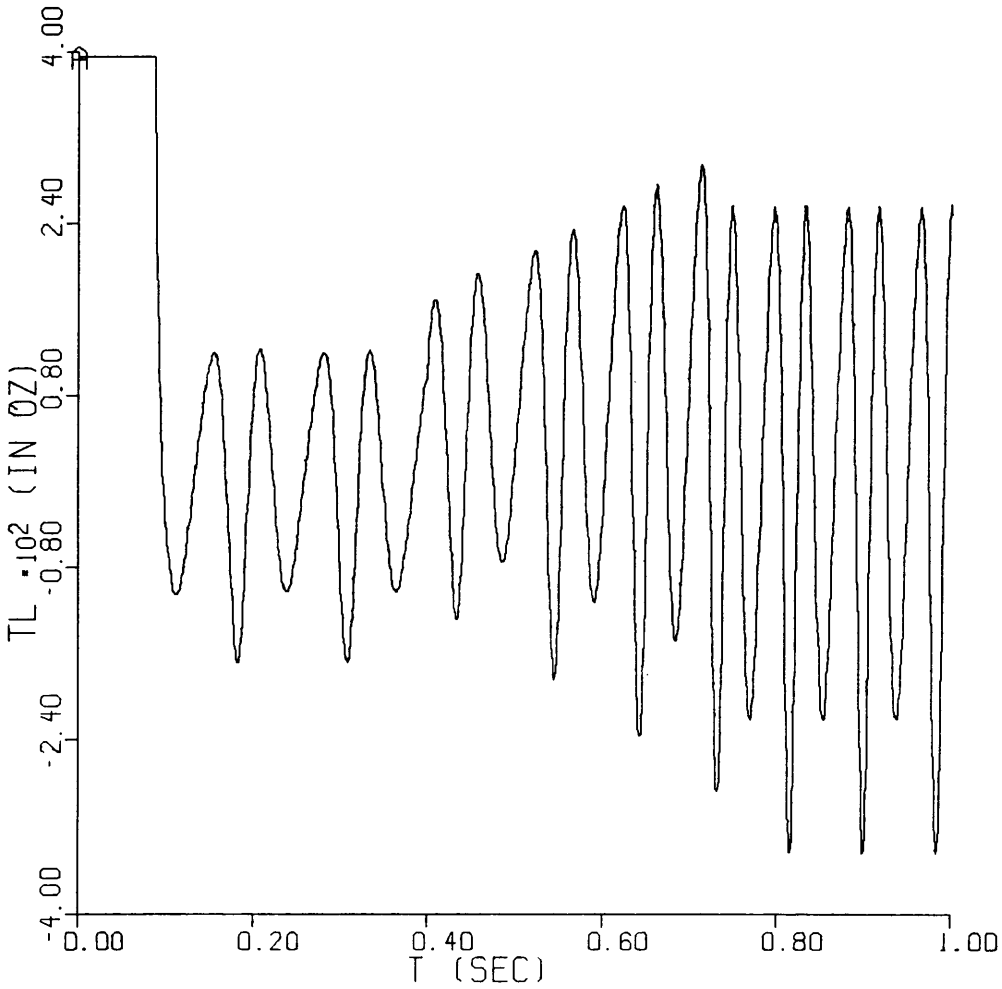


Fig. 4.9. Standard Control Torque: K_{FG} = 35, Torque Limit = 396, Ramp

5.0 Adaptive Control

Adaptive control techniques are useful where the system plant parameters are unknown or are "slowly time-varying". The objective is to recognize the unknown plant and automatically adjust the control to optimize a performance index. For the case of steady (but unknown) parameters, the control adapts to a steady-state solution. However, for "slowly time-varying" parameters the adaptation must be continuous to follow the parameter motion.

The primary aspect is the definition of "slowly time-varying" parameters. For this four-bar, the effective driven inertial changes with the input position at twice the input velocity. At a velocity of 75 rad/sec, this yields a "slowly time-varying" parameter that changes at a frequency of 150 rad/sec. However, given the inertia is a function of position, this can be used to redefine the "slowly time-varying" parameters to be coefficients of functions of the input position. These coefficients should then tend to be constant and the rotational variations resolved by the input position dependent functions.

This section covers the general formulation for the self-tuning adaptive control technique applied to the four-bar mechanism. The differences between the two branches of adaptive control: Model Reference adaptive and Self-tuning adaptive are highlighted. The required control calculation, optimal step-ahead control, and the required sequential least-squares parameter estimator are developed.

5.1 Model Reference Vs. Self-Tuning Adaptive Control

Adaptive Control is approached from two areas, Model Reference Adaptive Control, (MRAC), and Self-tuning Adaptive Control. Both methods yield similar results.

A block diagram for a typical MRAC is shown in Fig. 5.1 . For this method, the performance index, or performance requirement is characterized in the auxilary reference model. The goal of the control is to adjust the primary system or generate a control input to match the primary and reference model outputs. This is accomplished using an adaptation mechanism which drives the error, ε, to zero. This adaptation mechanism is typically derived from Lypunov or Popov stability criteria. Other formulations use a "steepest decent" optimization technique to minimize the error. Landau [10] offers a comprehensive review of Model Reference Adaptive Control. MRAC is not often used in mechanism control; however, Dubowsky [11] has developed a unique MRAC for a robotic manipulator. The Self-tuning adaptive technique was developed for four-bar control.

Self-tuning adaptive control relies on estimation of the unknown plant parameters using some inverse plant model. The adjustable control is then tuned using the estimated parameters to optimize the performance index. The performance index is typically used to derive the tuning algorithm. Fig. 5.2 is a general block diagram for the self-tuning adaptive system. Here, the estimation algorithm must be able to track the slowing time-varying plant parameters. However, the definition of the parameter variations can be changed using different forms of the inverse model.

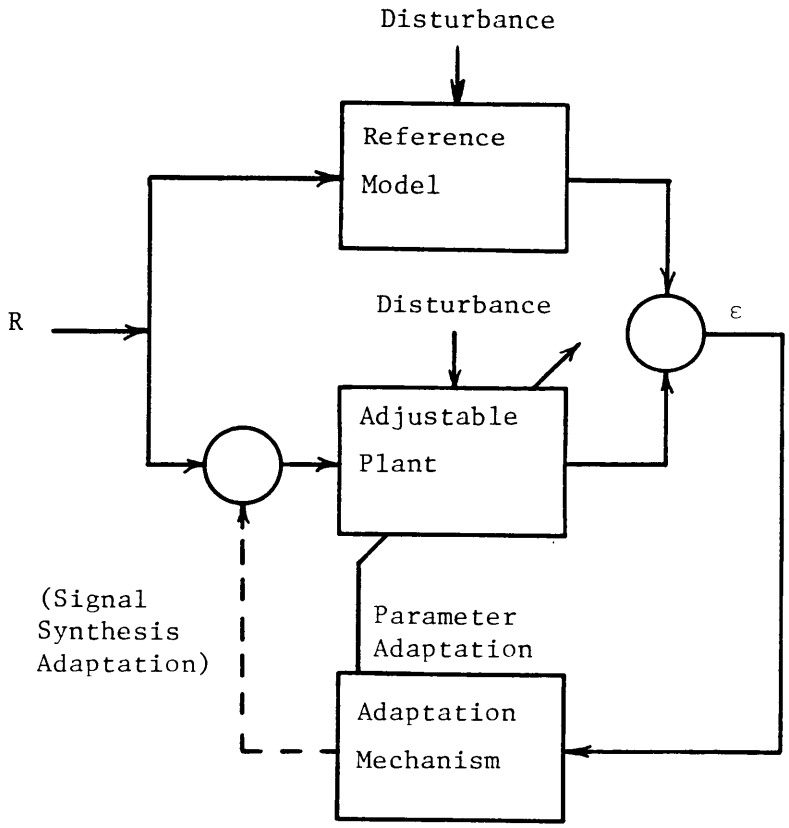


Fig. 5.1 General Model Reference Adaptive Control Configuration

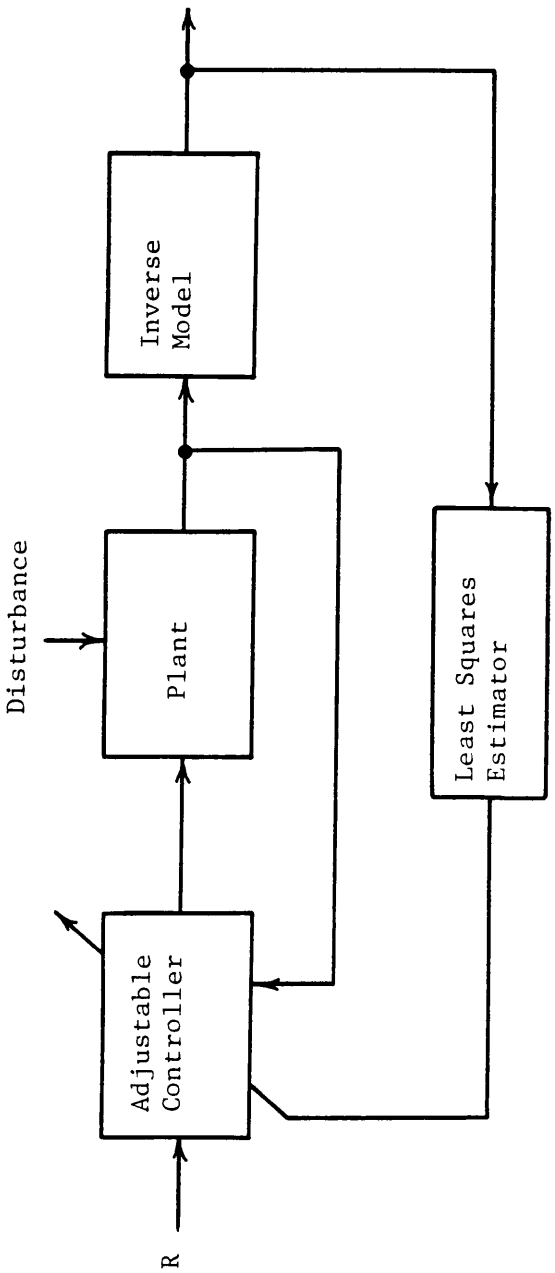


Fig. 5.2 General Self-Tuning Adaptive Control Configuration

Self-tuning adaptive control is loosely divided into direct and indirect (explicit, implicit) forms. In the direct form, the parameter estimates are used to directly calculate the required control adjustment. Therefore, the actual estimated parameter values become important. A self-tuning pole-placement regulator [12] is an example of a direct self-tuner. Here, the required control gains are directly calculated using the estimated plant pole locations. The indirect method does not have this dependency on the actual estimated values. In this formulation, the adaptive mechanism uses the output of an estimated parameter prediction model. Thus the prediction model output, not the parameter values, becomes important [13]. Distinction between the two is not always clear. The optimal step ahead control developed for the four-bar control uses both the prediction model output and the estimated parameters to calculate the applied control torque. Further discussion on self-tuning control can be found in [11] - [18]. Goodwin and Sin [18] provide an introductory discussion of self-tuning techniques.

The self-tuning adaptive control was adopted for the four-bar linkage controls developed in this study. Specifically, self-tuning optimal step-ahead control is used to formulate the adaptive control laws. The slowly time-varying parameters are estimated using sequential least squares with exponential data weighting. These constituent components are developed in section 5.2.

5.2 Self-Tuning Optimal Step-Ahead Control

Self-tuning adaptive control techniques were adopted for the four-bar mechanism control. This method can be used if the mechanism geometric changes are considered as the time-varying parameters. The adaptation then adjusts to account for these changes. The technique has had some success in the robot control which was the only located source for information on the control of mechanisms. This was the greatest motivation to choose the self-tuning technique.

The adaptive four-bar control uses an optimal step-ahead control law and sequential least-squares parameter estimation. The resulting self-tuning control incorporates both direct and indirect self-tuning approaches. The overall scheme is based on the ability to predict the next (k+1) mechanism velocity using the inverse model as a predictor. The control torque u(k), mechanism input torque T^*, is calculated using the desired velocity at time (k+1) and the predicted velocity at (k+1); an indirect approach. However, this difference is modified by a gain calculated using the estimated inverse model parameters; a direct approach.

This self-tuning control is similar to a parallel reference model MRAC. This similarity extends from the two-part role required of the inverse model. Typically, the self-tuning control inverse model is only used as a parameter estimation basis to model the real plant dynamics. However, the optimal step-ahead control law requires a predicted value for the plant (k+1) velocity. Therefore, the inverse model at time (k) is used to predict the required (k+1) velocity using the mechanism input torque T^* applied at time (k). This two-part role is illustrated in

block diagram form (Fig. 5.3). The inverse model and the prediction model are shown separately for clarity, but these are in essence the same model. The direct and indirect ideas are clearly seen. The control torque u(k) is calculated using the desired velocity $\omega^d(k+1)$ (velocity command input) and the predicted velocity $\omega^p(k+1)$ (the indirect approach) which is modified by the adjustable loop gain (the direct approach).

The MRAC similarity extends from the prediction model. The difference is instead of using the difference between a reference model and plant outputs to adjust the control gain, MRAC, we are predicting the mechanism velocity to make control corrections. This has several advantages. The prediction model provides a means to eliminate, or account for, the finite calculation time delay thus reducing the delay instability effects. Also, the mechanism disturbances excite the inverse model and estimation algorithm. Therefore, the combination inverse/prediction model can adapt to "model" the mechanism disturbance.

Optimal step-ahead control is derived by minimizing the performance index

$$J = E[\frac{1}{2} (\omega^d(k + 1) - \omega^p(k + 1))^2 + \frac{1}{2} \lambda u^2(k)] \qquad (5.2\text{-}1)$$

where

$E[\cdot]$	is the expectation operator.
$\omega^d (k + 1)$	is the desired velocity at time $(k + 1)$.
$\omega^p (k + 1)$	is the predicted velocity at time $(k + 1)$.

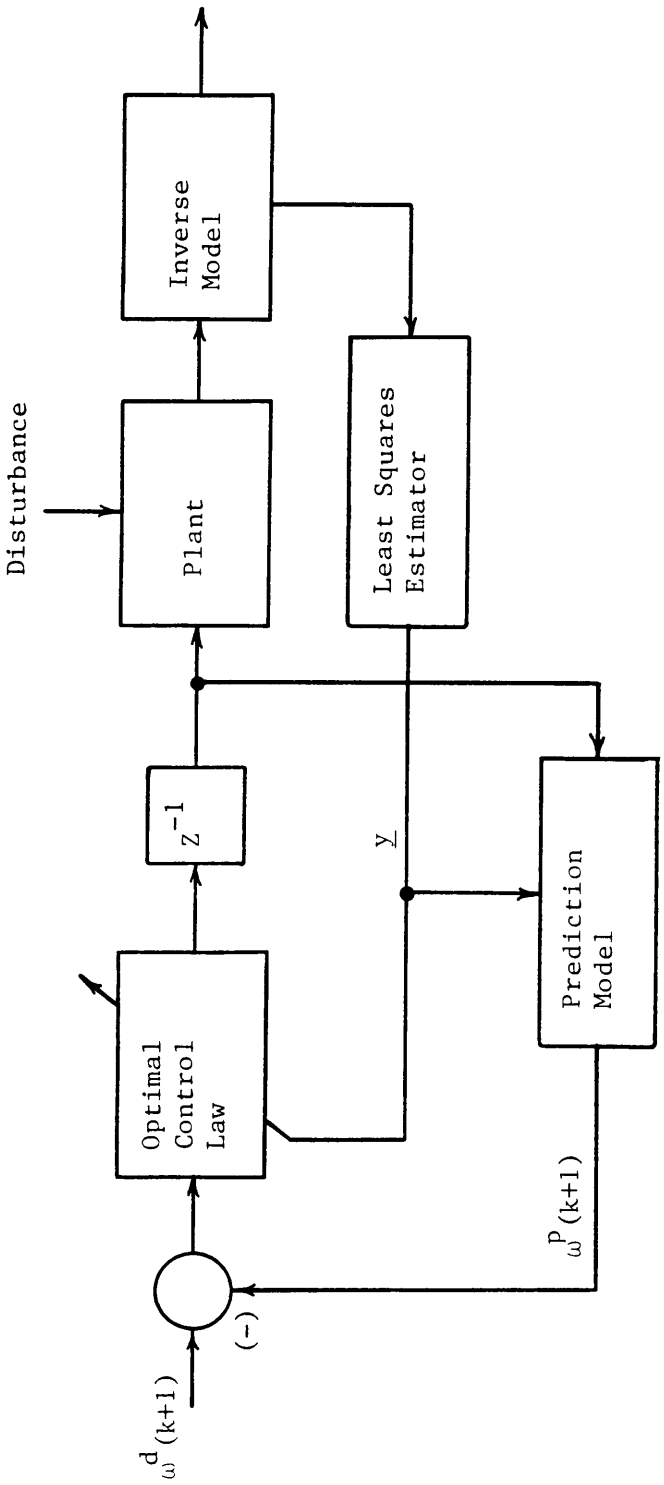


Fig. 5.3 Optimal Step-Ahead Adaptive Control Configuration

$u(k)$ is the applied control at time (k).

λ is a positive control weight.

This amounts to minimizing the squared difference between the desired velocity path and the actual velocity path and the control energy required to move the mechanism over one sample instant.

The inverse/prediction model can take many forms, however to use the linear least square estimator it must be linear combinations of some basis function vector $\underline{v}(u(k), \omega(k))$ where

$$\underline{v}\ (u(k), \omega(k)) = [V_1\ (u(k), \omega(k)), \ldots V_n\ (u(k), \omega(k))] \tag{5.2-2}$$

The components V_1, V_2,... may or may not be functions of $u(k)$, $\omega(k)$, and $u(k)$ appears only in linear combinations. These basis functions are selected in some manner such that, a set of coefficients $\underline{y}$ will yield the desired predictor model

$$\omega(k+1) = = \underline{v}(k)\ (u(k), \omega(k))\ \underline{y}(k) + \varepsilon\ , \tag{5.2-3}$$

where ε is zero mean random modeling error. The arguments $(u(k), \omega(k))$ of $\underline{v}(k)$ are dropped unless ambiguity results. The coefficients $\underline{y}(k)$ are estimated over the basis functions $\underline{v}(k)$, which constitute the self-tuning inverse model, using sequential linear least squares. The prediction model, equation (5.2-3), using the estimated parameters $\hat{\underline{y}}(k)$ becomes

$$\omega^p(k+1) = \underline{v}(k)\ \hat{\underline{y}}(k) \tag{5.2-4}$$

The four-bar adaptive control uses two forms of the basis vector $\underline{v}(k)$. The first is simply the past velocities and input torque. The second is based on the position-varying nonlinear dynamics.

The control law is now derived from the minimization of the performance index

$$J = E[\frac{1}{2}\ (\omega^d(k+1) - \underline{v}(k)\ \underline{y}(k))^2 + \frac{1}{2}\ \lambda\ u^2(k)]\ . \tag{5.2-5}$$

Using the definition of the expectation operator, and the Euler equation necessary condition for a minimum, (note we are not considering time derivatives of the input) the optimal control is

$$\frac{\partial J}{\partial u(k)} = 0 = \left(\omega^d(k+1) - \underline{v}(k)\ \hat{\underline{y}}(k)\right) \frac{\partial \underline{v}(k)}{\partial u(k)}\ \hat{\underline{y}}(k) + \lambda\ u(k) \tag{5.2-6}$$

[18], [19]. Since $\underline{v}(k)$ is a function of the current control, $u(k)$, the solution is not solved for the general case and only solved for the cases involved for each four-bar control. A discussion for the selection of the basis function vector $\underline{v}$ and a complete statement for the optimal control problem is found in [16]. The optimal control problem is discussed by Bryson in [19]. The self-tuning name for this control arises because the control approaches the optimal control as the parameter estimates $\hat{\underline{y}}$ approach their true values $\underline{y}$.

The previous explanation should not be considered in any way a proof for the solution of the general optimal step ahead control.

Several substantial arguments have been omitted among other formal requirements. The main objective of this discussion was to expose the method used to derive the four-bar controls in a general format.

The parameters $\hat{\underline{y}}(k)$ of equation (5.2-4) are estimated using sequential linear least squares. The method is inherently applicable to digital control by allowing the parameters $\hat{\underline{y}}(k)$ to be updated after each data sample. The complete derivation is discussed in [18] [20] and the results are presented here.

Given the linear form of the predictor, equation (5.2-3), with only the time arguments given for clarity, define the model error as

$$\varepsilon(k) = \omega(k) - \underline{v}(k-1)\,\hat{\underline{y}}\,(k-1)\ . \qquad (5.2\text{-}7)$$

This is the difference between the measured velocity at time (k) and the predicted velocity from time (k - 1). We wish to select the values for $\hat{\underline{y}}(k-1)$ to minimize the expected error at time (k - 1)

$$E[\varepsilon\,(k-1)] = \frac{1}{k}\sum_{i=0}^{k-1} \varepsilon^2(i)\ . \qquad (5.2\text{-}8)$$

The solution to the single equation problem is the sequential linear least-squares algorithm

$$\hat{\underline{y}}(k) = \hat{\underline{y}}(k-1) + \underline{K}(k-1)\,[\omega(k) - \underline{v}(k-1)\,\hat{\underline{y}}(k-1)] \qquad (5.2\text{-}9)$$

where

$$\underline{K}(k-1) = \frac{\underline{P}\ (k-2)\ \underline{v}(k-1)}{\omega^{-1}(k-1) + \underline{v}^T(k-1)\ \underline{P}(k-2)\ \underline{v}(k-1)} \qquad (5.2\text{-}10)$$

and

$$\underline{P}(k-1) = \underline{P}(k-2) - \underline{K}(k-1)\ \ \underline{v}^T(k-1)\ \underline{P}(k-2) \qquad (5.2\text{-}11)$$

Here equation 5.2-9 updates the estimated parameter values $\hat{\underline{y}}(k)$ based on the model error, equation 5.2-7, and the Kalman gain matrix, equation 5.2-10. Equation 5.2-11 updates the covariance matrix $\underline{P}$ for the next sample. This algorithm treats all incoming data and the initial conditions matrices $\hat{\underline{y}}(0)$, and $\underline{P}(0)$ with the same importance. For time-varying systems and poor initial conditions, the most recent data has the most pertinent information. Therefore, we would like to disregard the past data and consider only the more recent data. This can be accomplished several different ways [18]. However, only one method, exponential data-weighting, was used for the four-bar control.

Exponential data weighting, or forgetting least-squares applies exponentially decaying weights to the past data. Therefore, the most recent data is considered to contain the best information on the present condition of the plant, and the past data is "forgotten". The algorithm is a slight modification to equation 5.2-10, 11 and written as

$$\hat{\underline{y}}(k) = \hat{\underline{y}}(k-1) + \underline{K}(k-1)\ [\omega(k) - \underline{v}(k-1)\ \hat{\underline{y}}(k-1)] \qquad (5.2\text{-}12)$$

$$\underline{K}(k-1) = \frac{\underline{P}(k-2)\ \underline{v}(k-1)}{\rho(k-1) + \underline{v}^T(k-1)\ \underline{P}(k-2)\ \underline{v}(k-1)} \qquad (5.2\text{-}13)$$

$$\underline{P}(k-1) = \frac{1}{\rho(k-1)} \left[\underline{P}(k-2) - \underline{K}(k-1)\, \underline{v}^T(k-1)\, \underline{P}(k-2)\right] \qquad (5.2\text{-}14)$$

with $0 < \rho(k) \leq 1$.

Here, $\rho(k)$ is the decay time constant for the exponentially decaying data weight. The value for ρ is typically a constant chosen between 0.9 and 1. Note here that $\rho = 1$ yields the straight least-squares estimation.

Several problems arise from exponential least-squares. These problems are related to the time varying properties of $\hat{\underline{y}}$ and the "richness" of the input data. The resultant problem is known as the burst effect [18]. The element values of $\underline{P}$ grow large forcing the estimator to be unstable. This causes a "burst" of poor estimates, $\hat{\underline{y}}$ to result and the elements of $\underline{P}$ to decrease. The estimator then re-estimates the parameters $\hat{\underline{y}}$ and the process repeats. Several methods have been used to overcome the burst problem. Although the burst effect was not observed in the four-bar controls, a variable forgetting factor

$$\rho(K) = 1 - \rho' \frac{\varepsilon^2(k)}{1 + \underline{v}^T(k-1)\, \underline{P}(k-2)\, \underline{v}(k-1)} \qquad (5.2\text{-}15)$$

was used in the ARMA control to observe its effect.

5.3 Adaptive Four-Bar Control

Three adaptive controllers using the self-tuning from section 5.2 where developed as possible four-bar linkage controls. The basic control block diagram (Fig. 5.3) is the same for each control. The main differences are the definitions of the inverse/prediction model, there-

fore the definition of slowly varying parameters. Note however, the perturbation control scheme uses an open-ended computed torque idea.

The four-bar adaptive controls were developed using proposed robot joint control schemes. The first is a linear autoregressive optimal step-ahead control from Koivo [21]. This control is based on a stochastic minimum variance control where the inverse model is constructed from the past plant input/output data. This is a linear adaptive control where the plant is assumed to be linear but operating under stochastic disturbances. This is clearly not the case for the four-bar linkage (section 2.3).

The second control derived is a perturbation control from Lee and Chung [23]. This control uses the well known four-bar dynamics, the mechanism position and the desired velocity trajectory to compute a nominal torque which is applied to the four-bar. Any departures from the velocity path, or computed torque modeling errors are controlled using an autoregressive adaptive control. The perturbation control is extremely mechanism dependent, but yields the best overall performance.

The third controller is a variation of the perturbation control using some ideas from Depkovich [24]. The idea is to adapt the computed-torque model to "best fit" the mechanism, and use full velocity feedback to the computed-torque model. This implies, for the optimal step-ahead formulation, an adaptation in the inverse model to account for the mechanism position, velocity, and acceleration. Briefly, the inverse model becomes a function of the position which changes the estimated slowly varying parameters to be the coefficients for a "computed-torque" model. The major problems with this control are the measurement of the

instantaneous acceleration to yield the proper estimates using the least-squares estimator, and the selection of the inverse model basis functions.

The application feasibility for each control was estimated from aspects of the required real calculation time and control computer memory requirements. This estimation was determined using the Intel 8086/8087 micro-processor pair execution times [22] and the simulated adaptive control sample rate. To be feasible, the control algorithm must have a real calculation time shorter than the controller sample interval. In the cases where look-up tables are used, the memory must not exceed 32 kilobytes. Although the memory requirements are easily fulfilled, the real calculation time required to compute the least-square parameter estimates is excessive.

6.0 ARMA Four-Bar Adaptive Control

The Autoregressive Moving Average (ARMA) adaptive control was the first adaptive technique applied to the four-bar. This is a linear adaptive control in that the predictor/inverse model components are linear in the input/output variables u, the applied control torque and ω, the velocity. The ARMA control originates from stochastic control principle and is covered in [18], [21].

The "slowly time-varying" parameter definition and the required real calculation time cause problems for the ARMA control. This difficulty is caused by the required controller sample rate to make the ARMA adaption rate fast with respect to the inertia fluctuations (Section 2.3). For this control, the only means to adjust the adaptation rate is to adjust the sample rate. Therefore, to control the "slowly varying" inertia, the controller must sample such that the inertia change between samples is a few percent or a "slow" change. This requires approximately 20 samples per input revolution. This rate will also reduce the sampled data effects in the ideal torque waveform (Fig. 3.9). The parameter variation rate is dependent on the input velocity. The combined requirement for slowly varying parameters and the maximum 75 rad/sec velocity yield a sample interval on the order of a few milliseconds. Unfortunately, the control law cannot be computed within this interval using the 8086/8087 microprocessors.

The control is not feasible to implement at the required sample rate, therefore consider the effects of reducing the sample rate. At the fast sample rate, the estimated ARMA coefficients would be expected to follow the inertia variations (I^*). As the sample rate decreases,

the estimates should tend to "vary" too fast for proper control until a nyquist sample rate is reached. Here, the nyquist would be 4 samples per revolution. Below the nyquist rate, the ARMA control should begin to estimate an average "inertia" value and control the four-bar as a constant inertia load without regard to the per revolution variations. The simulation runs show some of these effects, usually as a degradation of the velocity performance. No simulations were made below the nyquist sample rate to test that line of reasoning.

6.1 ARMA Derivation

The ARMA control is based on the prediction model

$$\omega(k+1) = \sum_{i=0}^{N-1} [a_i\ \omega(k-i) + b_i\ u(k-i)] + d + \varepsilon(k) \qquad (6.1\text{-}1)$$

where u(k-i), $\omega(k-i)$ are the input/output pairs for the past mechanism torque and velocity, d is an offset term and $\varepsilon(k)$ is the model error. The values of a_i, b_i, d are then estimated using least-squares to minimize the modeling error, thus yielding the best, in the least squares sense, prediction of $\omega(k+1)$.

N, an integer, is the predictor model order. The definition is from the discrete-time series approximation to a continuous system. Where an N^{th} order discrete-time series would model an N^{th} order continuous system. Goodwin [18] shows that the discrete model must be of equal or greater order than the continuous system. The ARMA four-bar adaptive control was simulated using 20^{th}, 10^{th}, and 3^{rd} order models. No degration was observed for the model reductions. The third order model

simulations are presented in Section 6.3.

The prediction model is rewritten using the notation of Section 5 as

$$\underline{v}(k) = [\omega(k),\ \omega(k-1),\ldots,\omega(k-N+1),\ u(k),\ u(k-1)\ldots u(k-N+1),\ 1] \tag{6.1-2}$$

$$\underline{y}(k) = [a^0,\ a^1,\ldots a^{N-1},\ b^0,\ b^1,\ldots b^{N-1},\ d]\ . \tag{6.1-3}$$

Sequential least-squares, equation 5.2-9,10,11, is used to estimate the values

$$\hat{\underline{y}}(k) = [\hat{a}^0,\ \hat{a}^1,\ldots \hat{a}^{N-1},\ \hat{b}^0\ ,\ \hat{b}^1\ ,\ldots \hat{b}^{N-1},\ \hat{d}] \tag{6.1-4}$$

to minimize the prediction model error $\varepsilon(k)$.

The optimal step-ahead control performance index (5.2-5) is written using the discrete desired velocity function, ω^d, at time $(k+1)$ and the estimated parameters $\hat{\underline{y}}$

$$J = E\{\frac{1}{2}(\omega^d(k+1) - \sum_{i=0}^{N-1}[\hat{a}_i\ \omega(k-i) + \hat{b}_i\ u(k-i)] + \hat{d})^2 + \frac{1}{2}\ \lambda\ u^2(k)\} \tag{6.1-5}$$

The minimization of the discrete function J, equation 6.1-5, is found by taking the derivative with respect to the current control u(k) equal to zero.

$$\frac{\partial J}{\partial(k)} = 0 = (\omega^d(k+1) - \sum_{i=0}^{N-1} (\hat{a}_i\ \omega(k-i) + \hat{b}_i\ u(k-i) + \hat{d})\ \hat{b}_o + \lambda\ u(k) \tag{6.1-6}$$

Solving for the current control u(k) yields the adaptive optimal control law

$$u(k) = \frac{\hat{b}^o}{(\hat{b}^o)^2 + \lambda} \{\omega^d(k+1) - [\sum_{i=0}^{N-1} \hat{a}_i\ \omega(k-i) + \sum_{i=1}^{N-1} \hat{b}_i\ u(k-i) + \hat{d}]\} . \tag{6.1-7}$$

The adaptive control law (equation 6.1-7) is represented in the block diagram (Fig. 6.1). The bracketed term $\{\cdot\}$, is the predicted error $e^p(k+1)$. The adjustable forward loop control gain, CGAIN is

$$\text{CGAIN} = \frac{\hat{b}^o}{(\hat{b}^o)^2 + \lambda}$$

the predicted error multiplier. The overall loop stability requires the CGAIN be positive, thus $\hat{b}^o$ must be positive. This characteristic is demonstrated by simulation in Section 6.3.

6.2 Control Algorithm

The ARMA adaptive control algorithm was programmed using equations 6.1-7, 5.2-9, 5.2-10,11. The control loop simulation ARMA appears in Appendix B for a third order prediction model. The ACSL simulation initializes all the required initial conditions

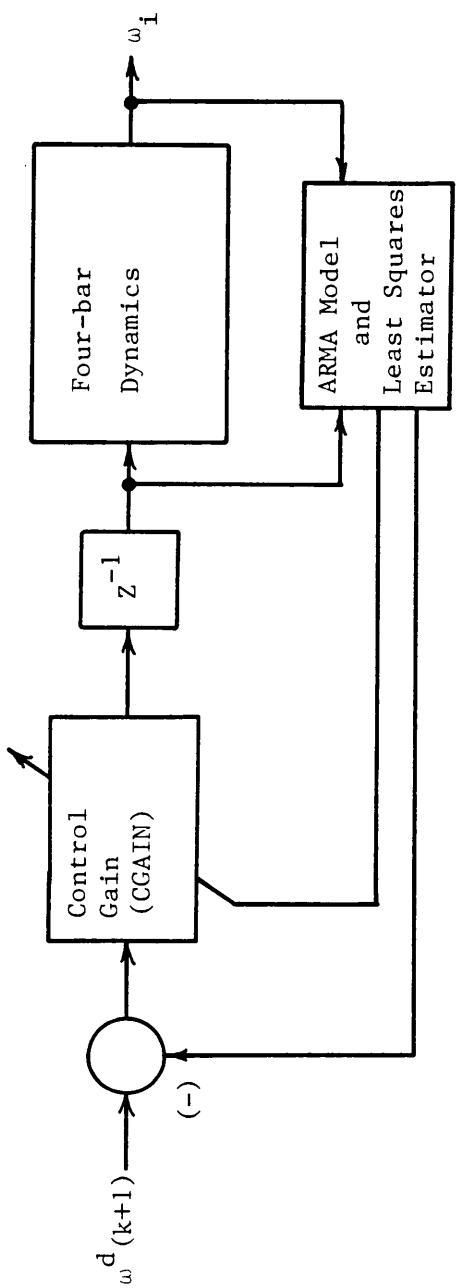


Fig. 6.1 ARMA Adaptive Control Block

$$\hat{\underline{y}} = [.1 \ldots\ldots .1]_{(1x7)}$$

$$\underline{v}(0) = [1, \ldots\ldots, 1]_{(1x7)}$$

$$\underline{P}(0) = \text{diag}\,[100, \ldots, 100]_{(7x7)}$$

Since the ARMA model depends on the velocity and torque data, some starting difficulties encountered. These problems are caused by the zero velocity optical encoder measurements below 15 rad/sec velocity. To overcome the start-up problem a constant 100 in-oz (7000 dyn-cm) torque is applied under input velocities of 15 rad/sec.

The control execution proceeds as follows. The previously calculated torque, $u(k - 1)$, is applied to the mechanism. The new measurement data $\omega(k)$, $u(k - 1)$ is collected. Using the measured velocity $\omega(k)$ the ARMA predictor parameters $\hat{\underline{y}}(k)$ are updated. The control $u(k)$ is calculated using the new $\hat{\underline{y}}$. At this point the control torque $u(k)$ could be applied to the mechanism, however, this would violate the simulated sample interval calculation time. Therefore, the torque is clipped to within the motor specification and held for one sample interval. The $\underline{P}$ matrix is updated and the current $\omega(k)$ and $u(k)$ are stored as past histories in $\underline{v}$ displacing the oldest measurements.

The control algorithm calculation time was estimated using the execution times for the assembly language subroutines INPROD and GINP from [22]. These routines are written for matrix calculations on the Intel 8086/8087 processor pair. For the third order ARMA control estimating seven parameters, the following calculation times are required.

Calculate	$\underline{K}$	(equation 5.2-13)	6.5×10^{-3} sec
Calculate	$\hat{\underline{y}}$	(equation 5.2-13)	2.0×10^{-3} sec
Calculate	u	(equation 6.1-7)	1.0×10^{-3} sec
Calculate	$\underline{P}$	(equation 5.2-14)	1.2×10^{-3} sec
Total time			10.7×10^{-3} sec

This is an optimistic estimate. Overhead for memory management is not completely accounted for. Therefore, a realistic sample interval may be 12.0×10^{-3} sec. This yields approximately 6 samples per input revolution at 75 rad/sec, which is one third the specified sample rate, and close to the four samples per revolution nyquist rate.

6.3 Simulation Runs

The ARMA four-bar adaptive control was simulated for the step and ramp input functions using control sample intervals of 12, 4, and 2 milliseconds (ms). These intervals correspond to the estimated calculation time which yields 6 samples per revolution at 75 rad/sec, 18 samples per revolution, and 36 samples per revolution at 75 rad/sec. The overall performance, from Section 4.2, is discussed for the decreasing sample rate trend, and for decreasing control torque weight λ at a constant 4 millisecond sample interval. This produces an increasing torque trend. Other observations which concern the least-squares estimation and the overall control loop stability are discussed as they arise.

The velocity plots (Fig. 6.2,3,4) for the step input function is a series of simulations for the decreasing sample interval. Here, the

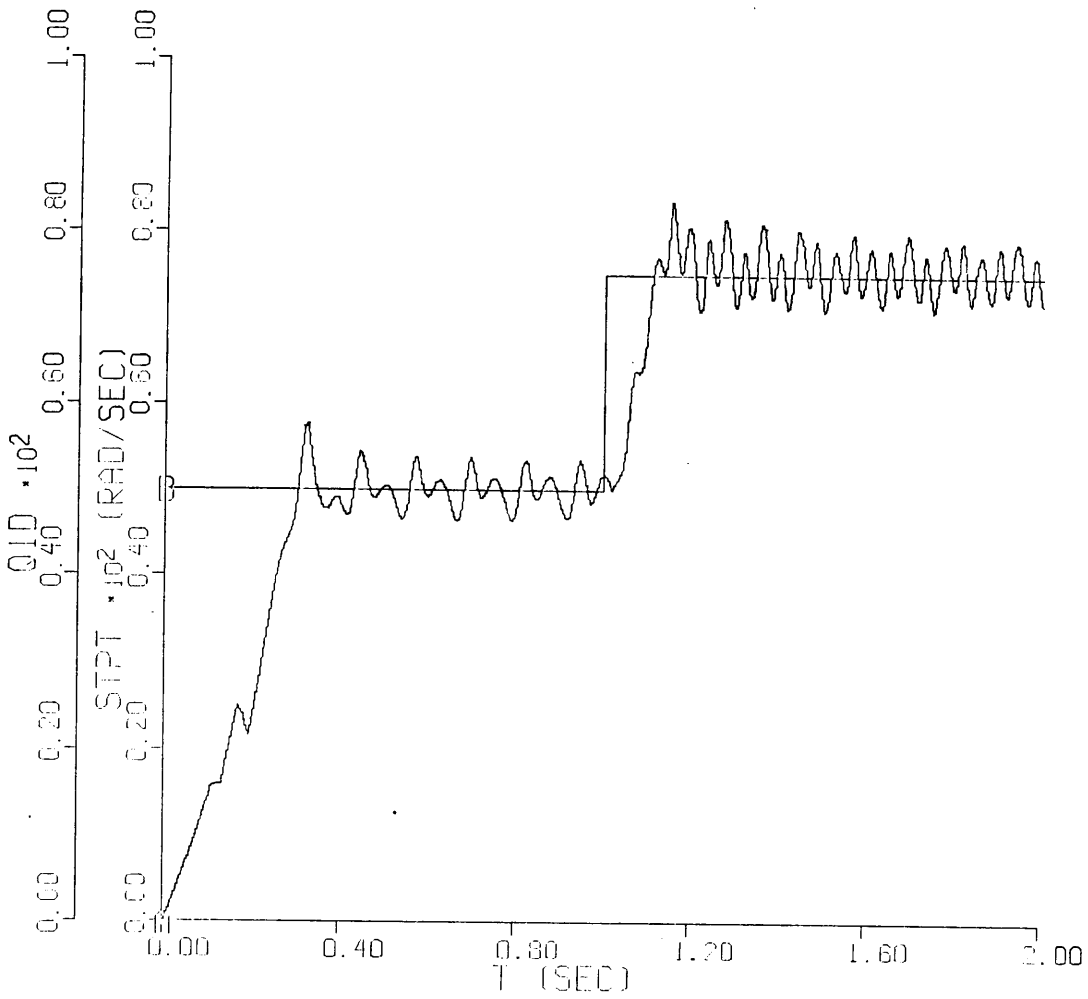


Fig. 6.2. ARMA Velocity: TSAMP = 12 ms, EPSL = .0001, APWR = 183, ALPHAP = .001, Step

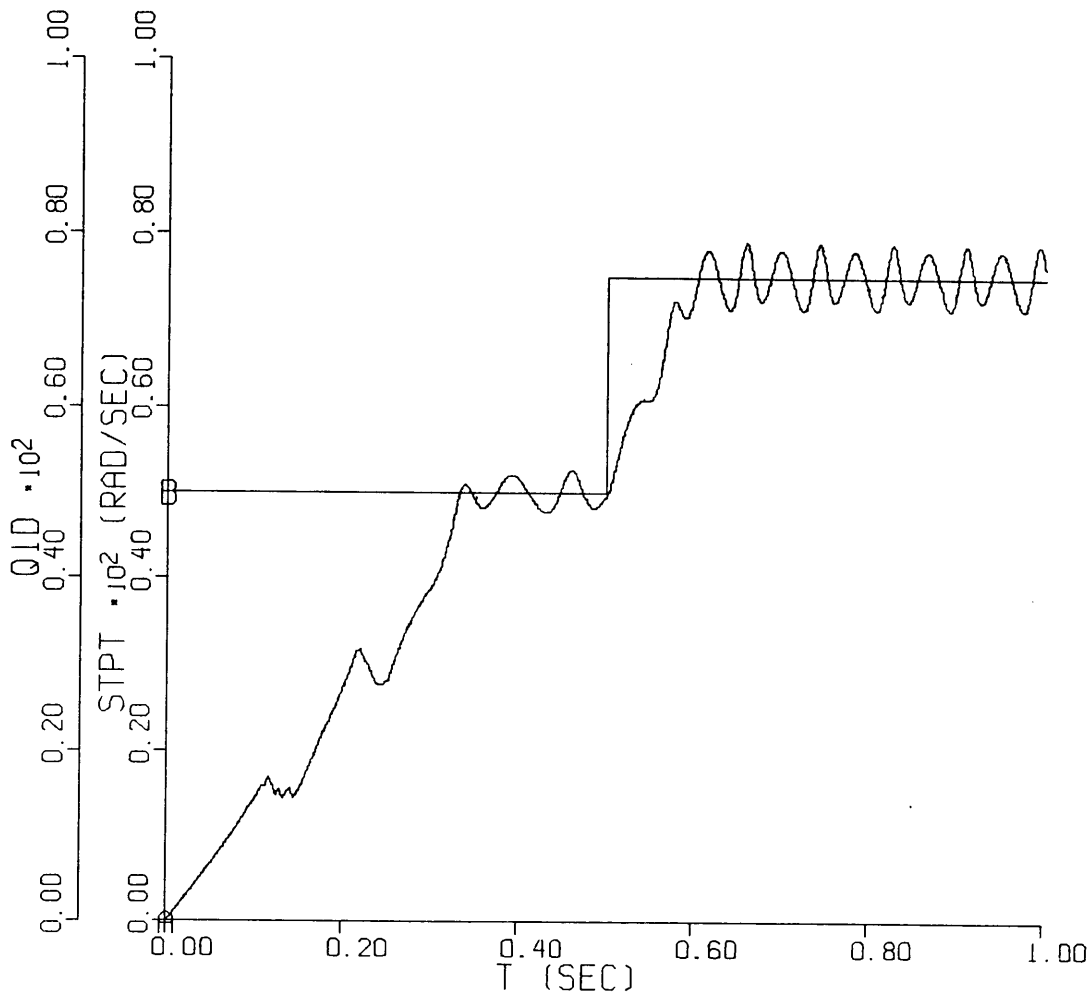


Fig. 6.3. ARMA Velocity: TSAMP = 4 ms, EPSL = .0001, APWR = 183, ALPHAP = .01, Step

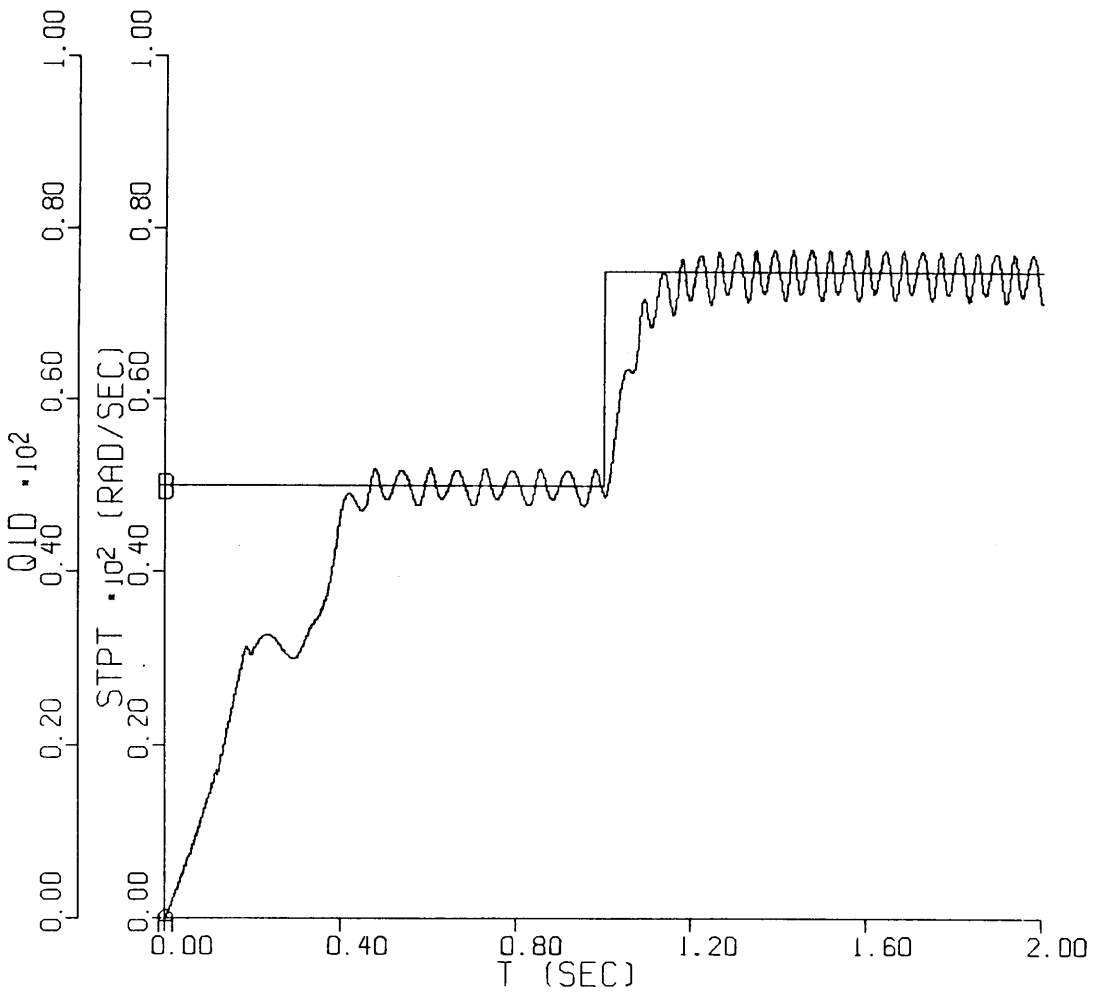


Fig. 6.4. ARMA Velocity: TSAMP = 2 ms, EPSL = .0001, APWR = 183, ALPHAP = .02, Step

overall trend is the decreasing sample interval decreases the velocity fluctuations. The observed decrease at 50 rad/sec is 14% to 8%. The decrease at 75 rad/sec is 9% to 7%. (The actual fluctuation for the 4 millisecond sample interval (Fig. 6.3) is greater than 9%. However, there are several factors which could cause this. Here, we are only considering trends, therefore this observation will be ignored.) This is an expected result. The closer spacing of samples taken around the input revolution should produce more current data to the control which should make the parameters "vary slower". In other words, the effective rate the controller can adapt is faster. Also, the sample data effects are reduced.

The rise time is not effected by the sample interval. Again using the velocity plots (Fig. 6.2,3,4), the observed rise time is approximately 100 milliseconds.

The velocity plots (Fig. 6.3,5,6) are for decreasing values of the control torque weight λ. The effect of decreasing λ is to allow more control torque to be used without increasing the performance index. The trend shown is for $\lambda = 1 \times 10^{-4}$, 1×10^{-5}, 1×10^{-7}. The simulations were run with the 4 ms sample interval. The general trend is for decreasing fluctuations. As would be expected, the more corrective torque used the better the performance. The torque plots, Fig. (6.7,8,9) illustrate the increase in torque use. However, with the smallest value for λ the maximum torque is not used to correct the fluctuation. Therefore, better performance may be expected with the minimum variance control where the control energy is unweighted ($\lambda = 0$). The disadvantage of small λ values is the control loop tends

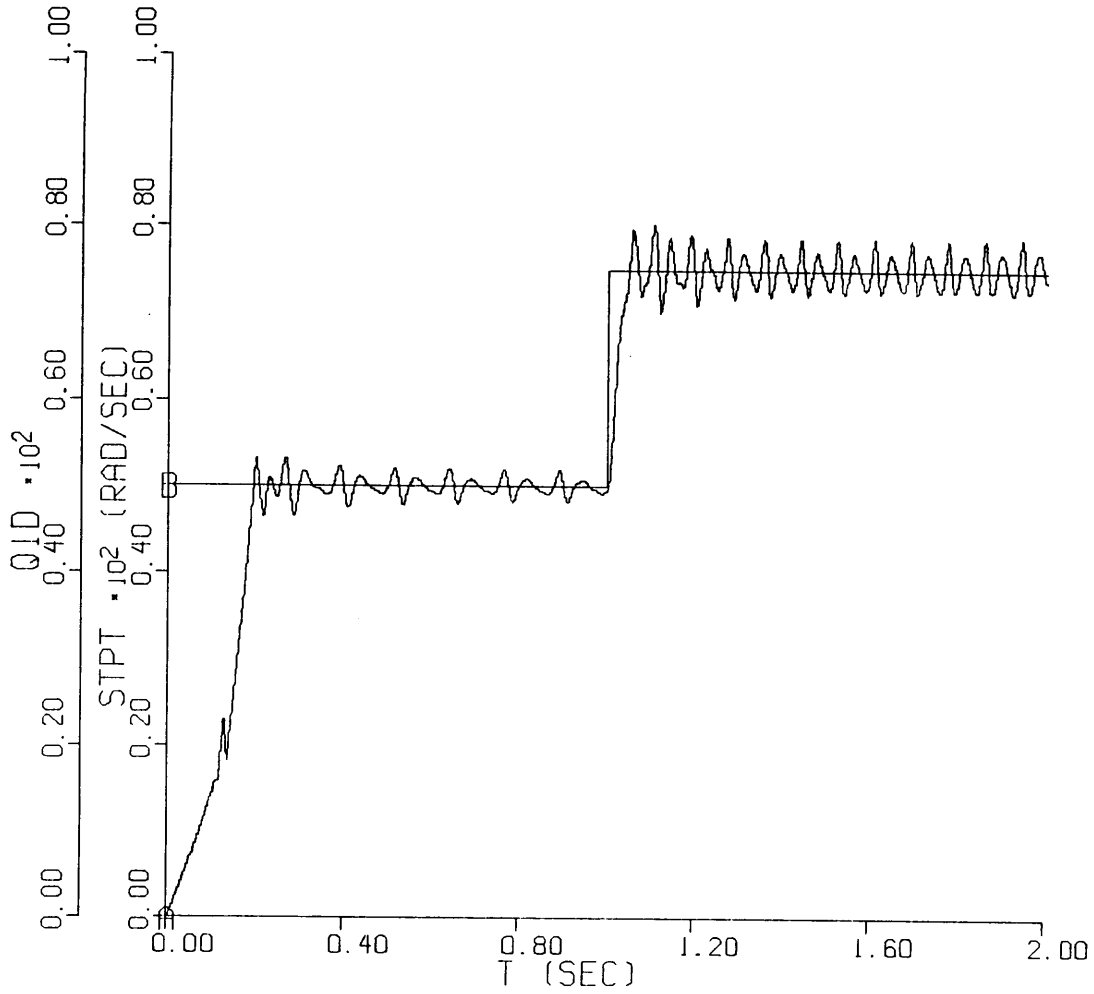


Fig. 6.5. ARMA Velocity: TSAMP = 4 ms, EPSL = .00001, APWR = 396, ALPHAP = .02, Step

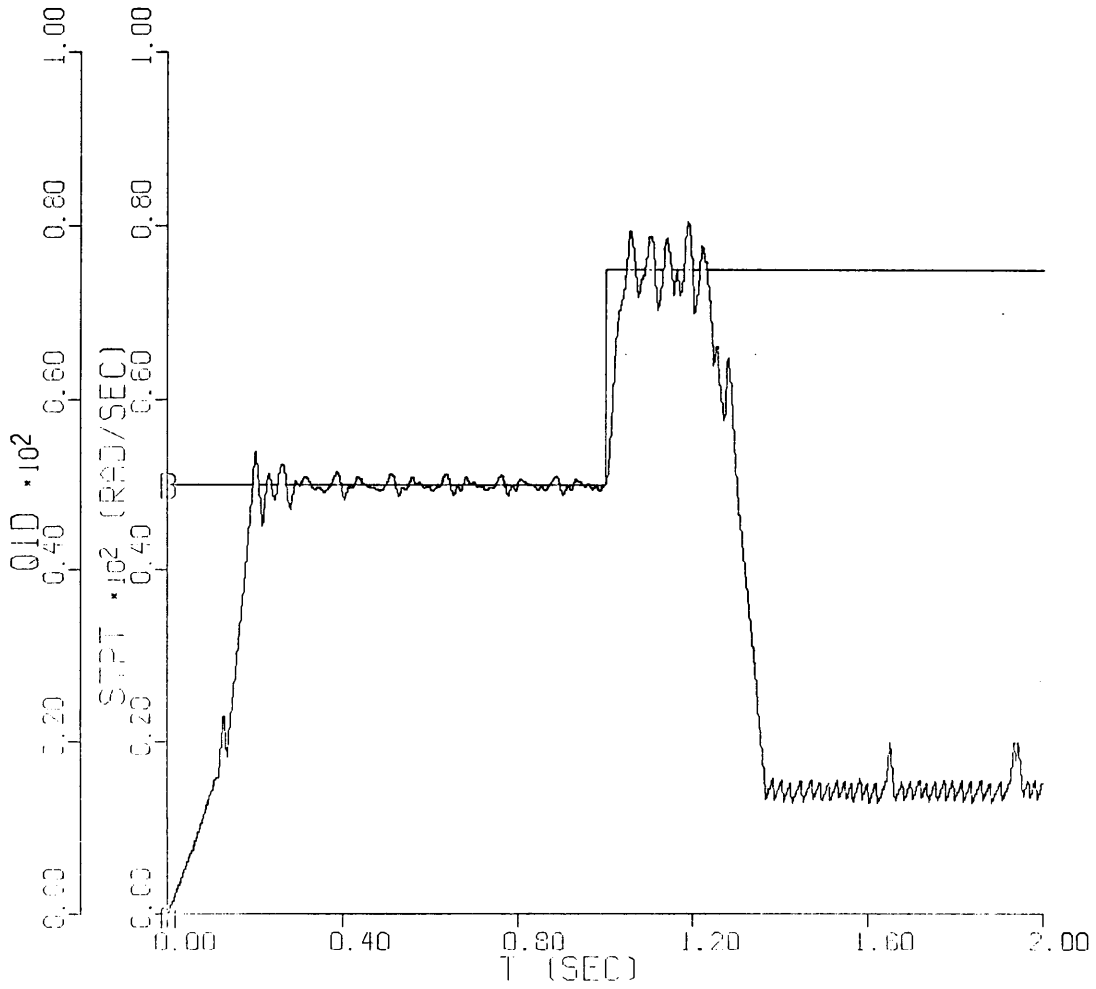


Fig. 6.6. ARMA Velocity: TSAMP = 4 ms, EPSL = 1 x 10^{-7}, APWR = 396, ALPHAP = .02, Step

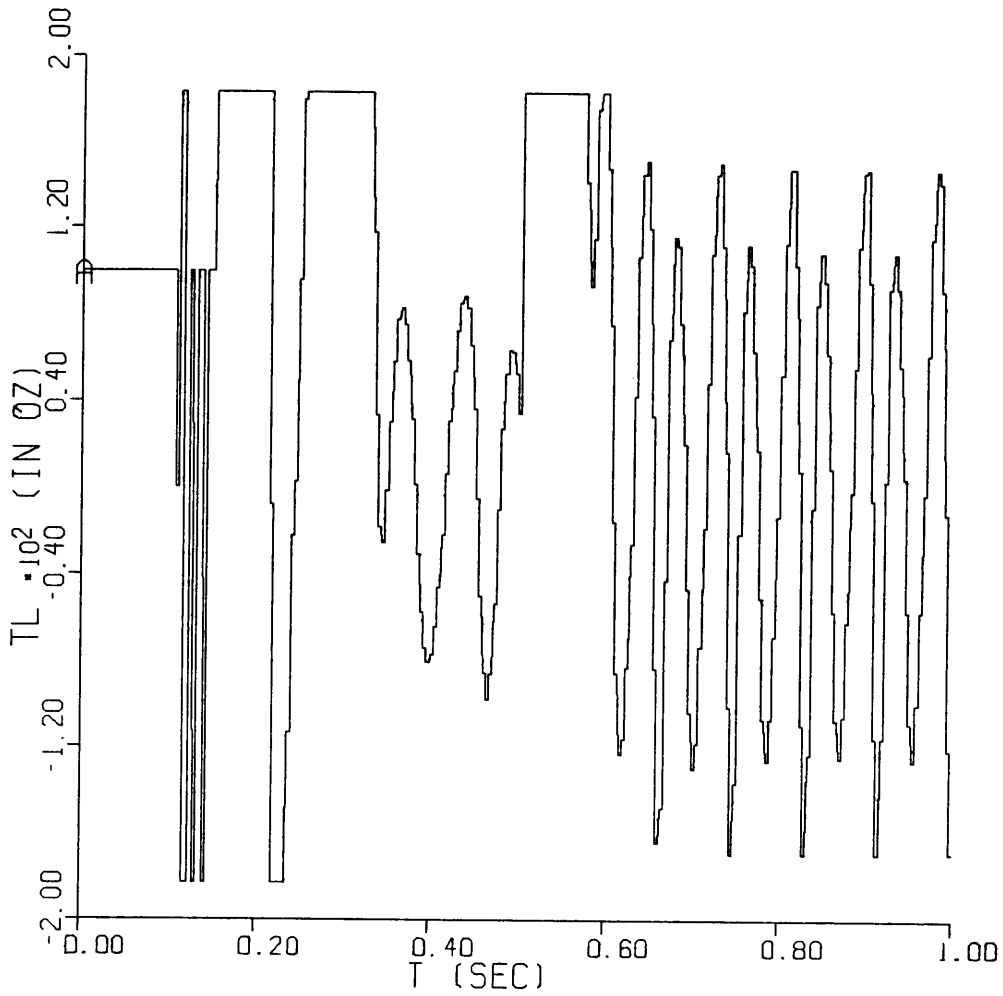


Fig. 6.7. ARMA Torque: TSAMP = 4 ms, EPSL = .0001, APWR = 183, ALPHAP = .01, Step

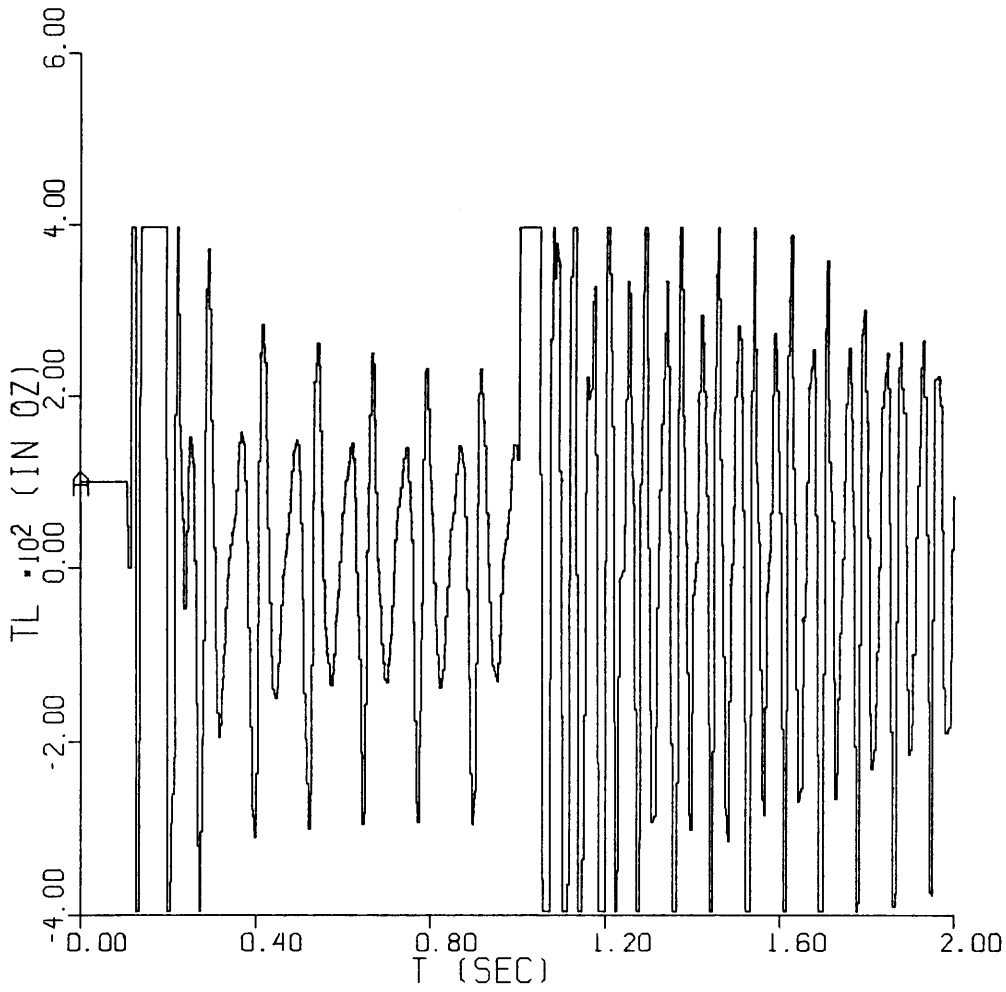


Fig. 6.8. ARMA Torque: TSAMP = 4 ms, EPSL = .00001, APWR = 396, ALPHAP = .02, Step

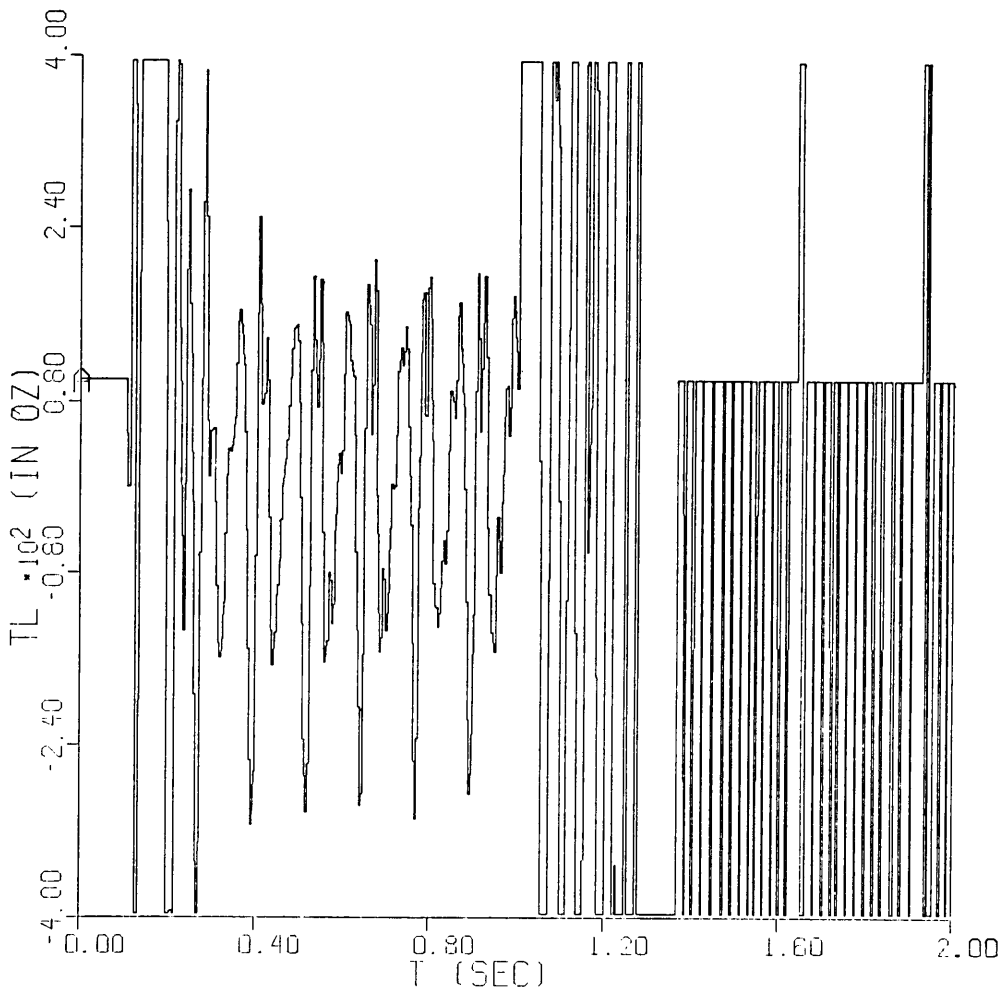


Fig. 6.9. ARMA Torque: TSAMP = 4 ms, EPSL = 1 x 10^{-7}, APWR = 396, ALPHAP = .02, Step

to become unstable as illustrated in Fig. 6.6 at 1.2 seconds. The decreasing control weight shows a decrease in the fluctuations at 50 rad/sec from 11% to 3%. The control does not adapt for the 75 rad/sec velocity and $\lambda = 1 \times 10^{-7}$, however, the other cases show a 9% to 8% decrease.

The global result for the fluctuation performance would be to sample as fast as possible using the smallest value for control weights without the algorithm going unstable.

The tracking performance for the ramp input is good. Velocity plots (Figs. 6.10,11) for 12 and 4 ms, respectively, show some minor steady-state lag. The step response, as mentioned, is related to the control weight. However, the smaller sample intervals produce a smoother step transition (Fig. 6.3,4) without the tendency to slightly overshoot as the 12 ms sample interval (Fig. 6.2).

The control loop stability depends on the stability of the parameter estimator. Since the open loop velocity control is basically a first order system, the only way to force it unstable is to apply a negative feedforward gain. The velocity, control gain, and ARMA parameter plots (Figs. 6.12,13,14,15,16) illustrate an example of the unstable behaviors encountered. For the first one second, the control is unstable. The estimated ARMA coefficients of equations 6.1-4 and 6.1-7, A0, A1, A2 (Fig. 6.14), B0, B1, B2 (Fig. 6.15), and D (Fig. 6.16) drastically fluctuate until the input steps-up at time = 1 sec. With this new information, the estimates stabilize and the controller begins to track the input function. The same trends are seen in the forward gain CGAIN, (Fig. 6.13). Here, the gain fluctuates with negative values

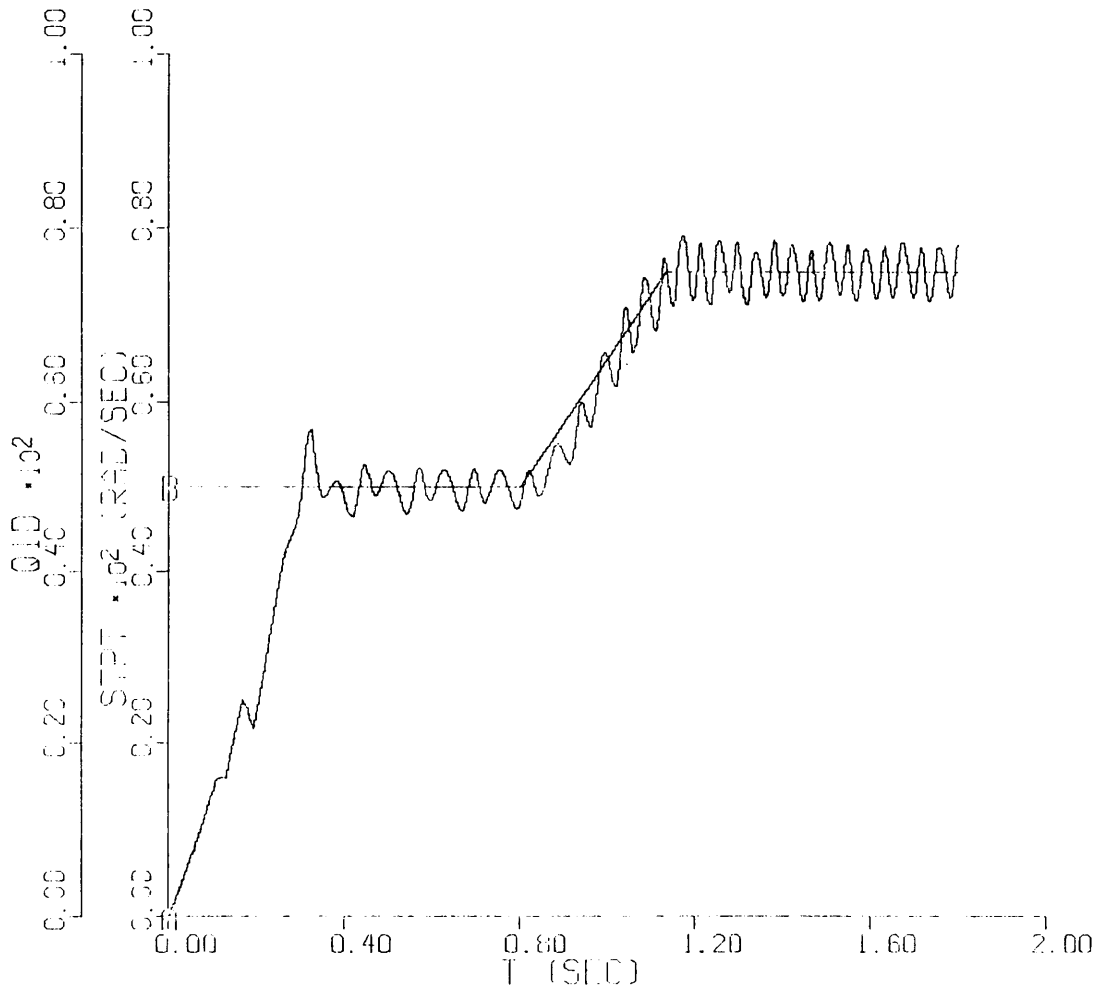


Fig. 6.10. ARMA Velocity: TSAMP = 12 ms, EPSL = .0001, APWR = 183, ALPHAP = .02, Ramp

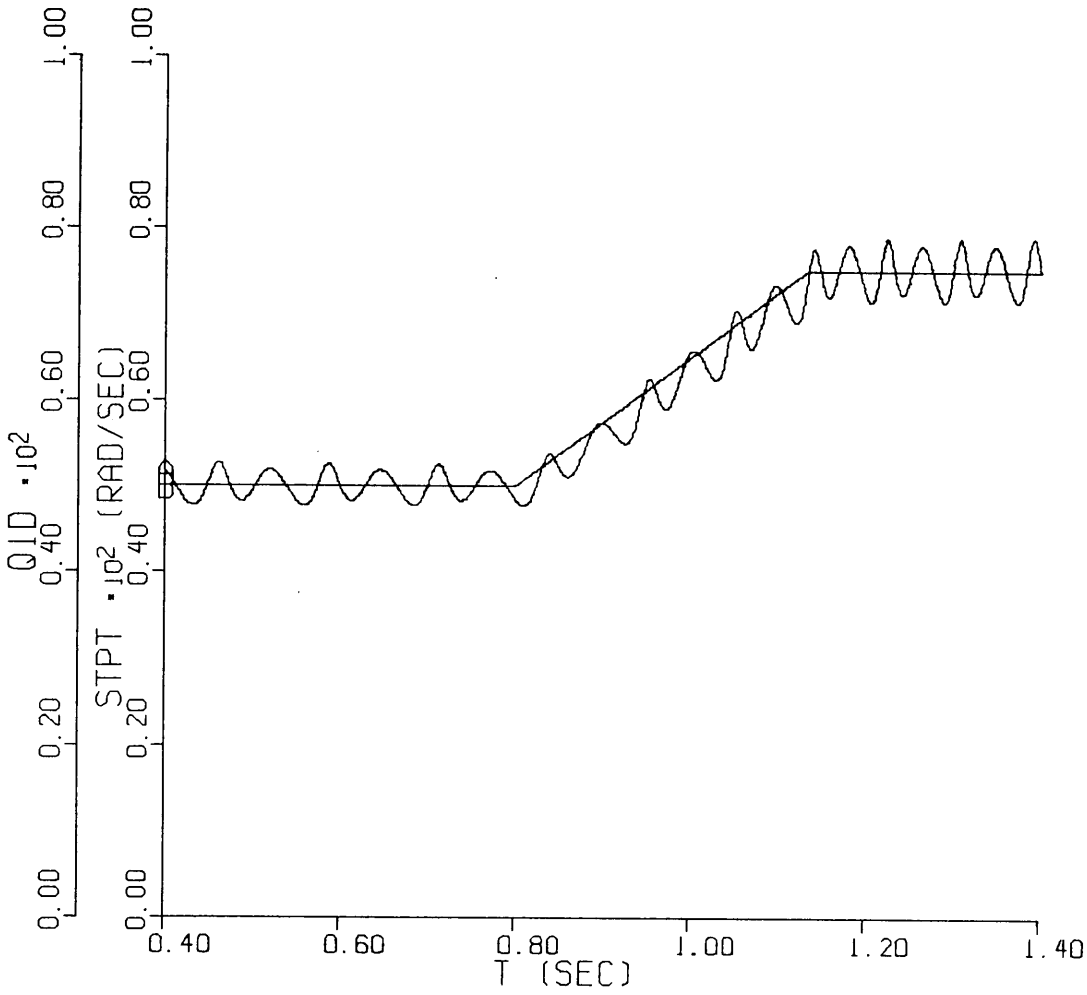


Fig. 6.11. ARMA Velocity: TSAMP = 4 ms, EPSL = .0001, APWR = 183, ALPHAP = .01, Ramp

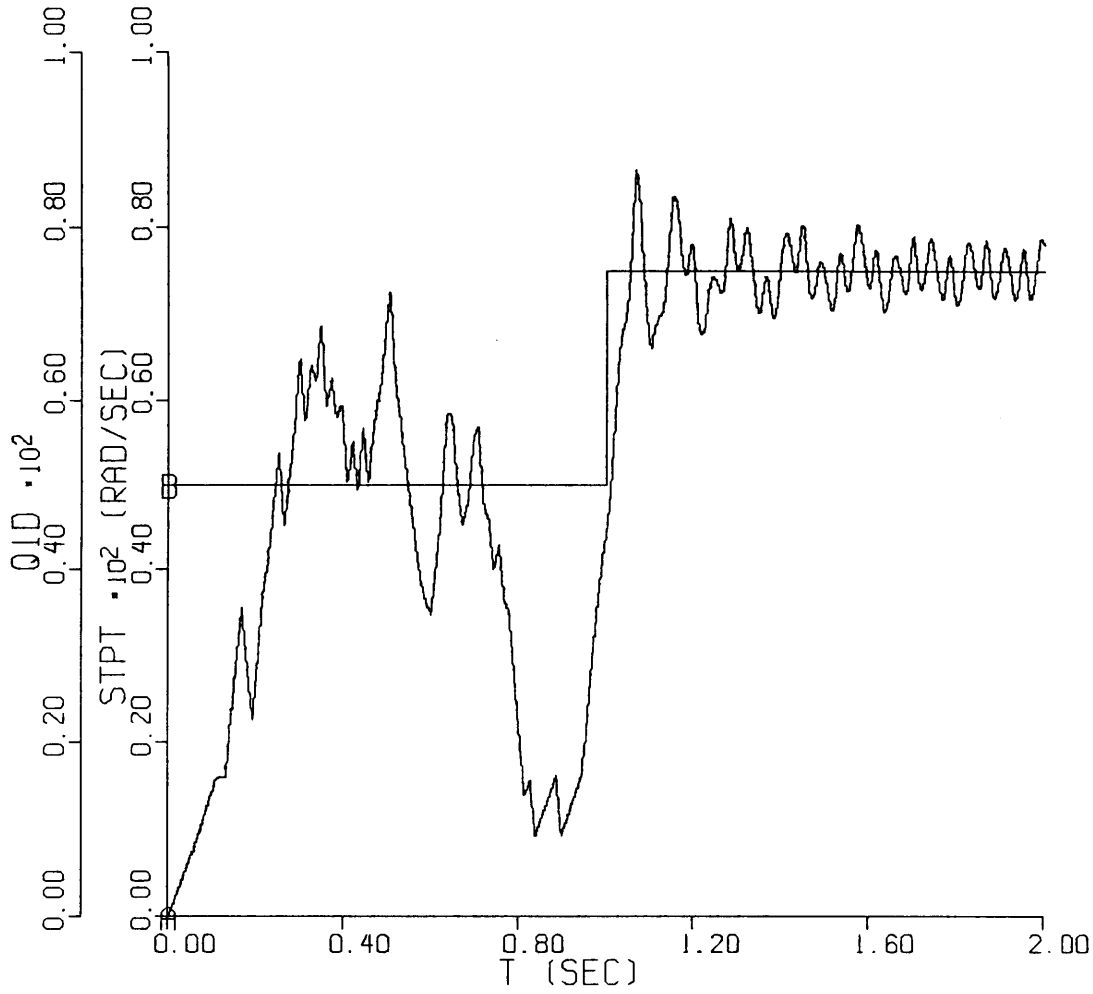


Fig. 6.12. ARMA Velocity: TSAMP = 12 ms, EPSL = .0001, APWR = 396, ALPHAP = .01, Step

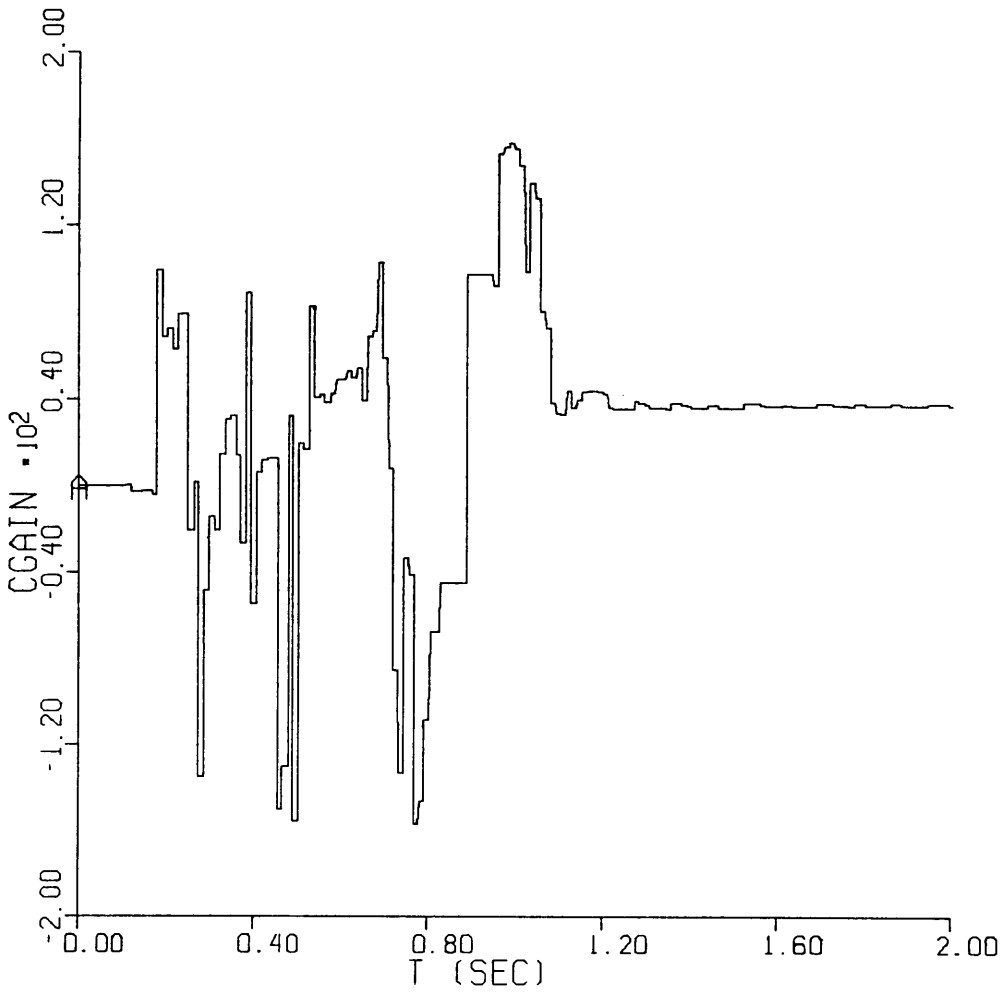


Fig. 6.13. ARMA Control Gain: TSAMP = 12 ms, EPSL = .0001, APWR = 396, ALPHAP = .01, Step

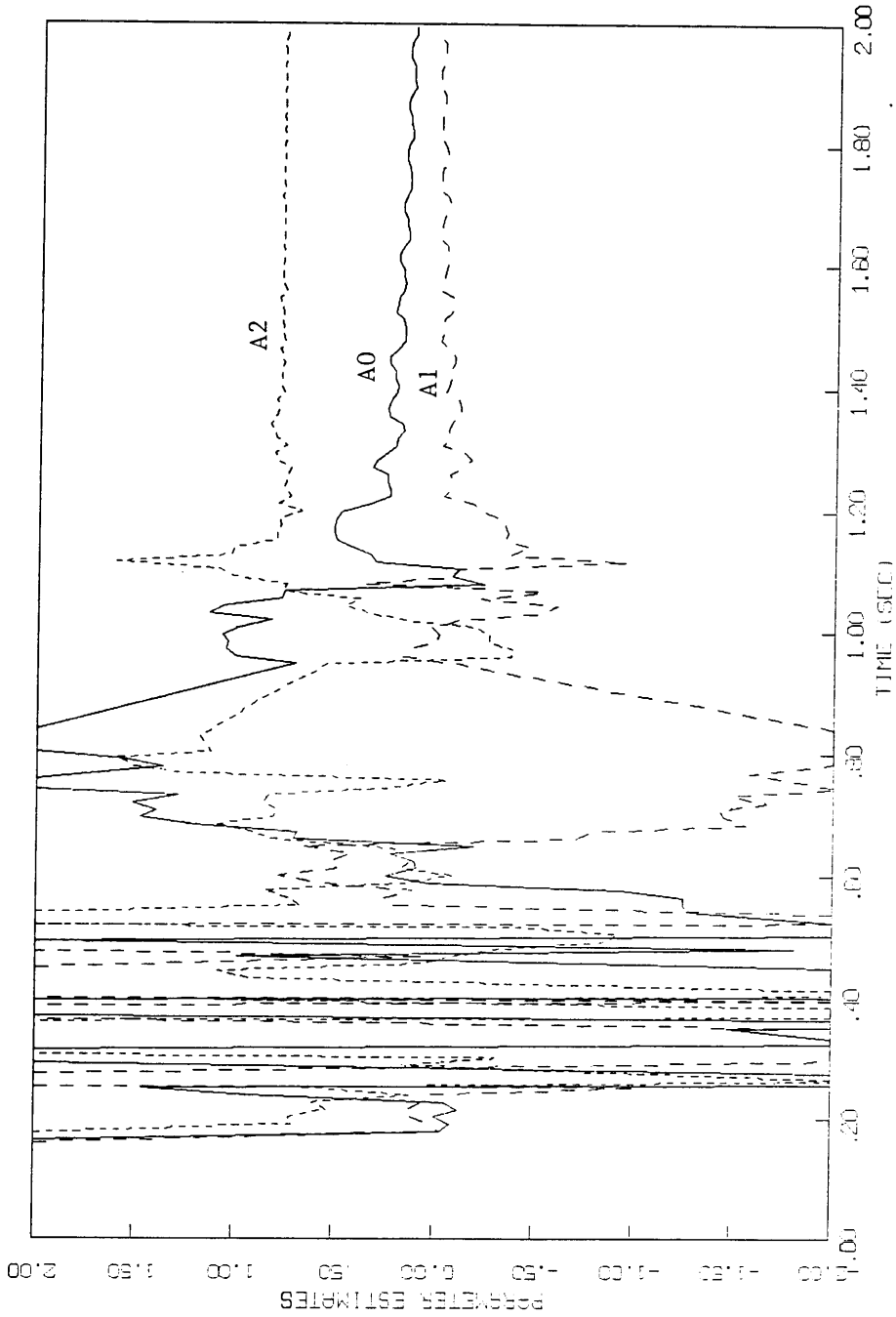


Fig. 6.14. ARMA Coefficients A0, A1, A2: TSAMP = 12 ms, EPSL = .0001, APWR = 396, ALPHAP = .01, Step

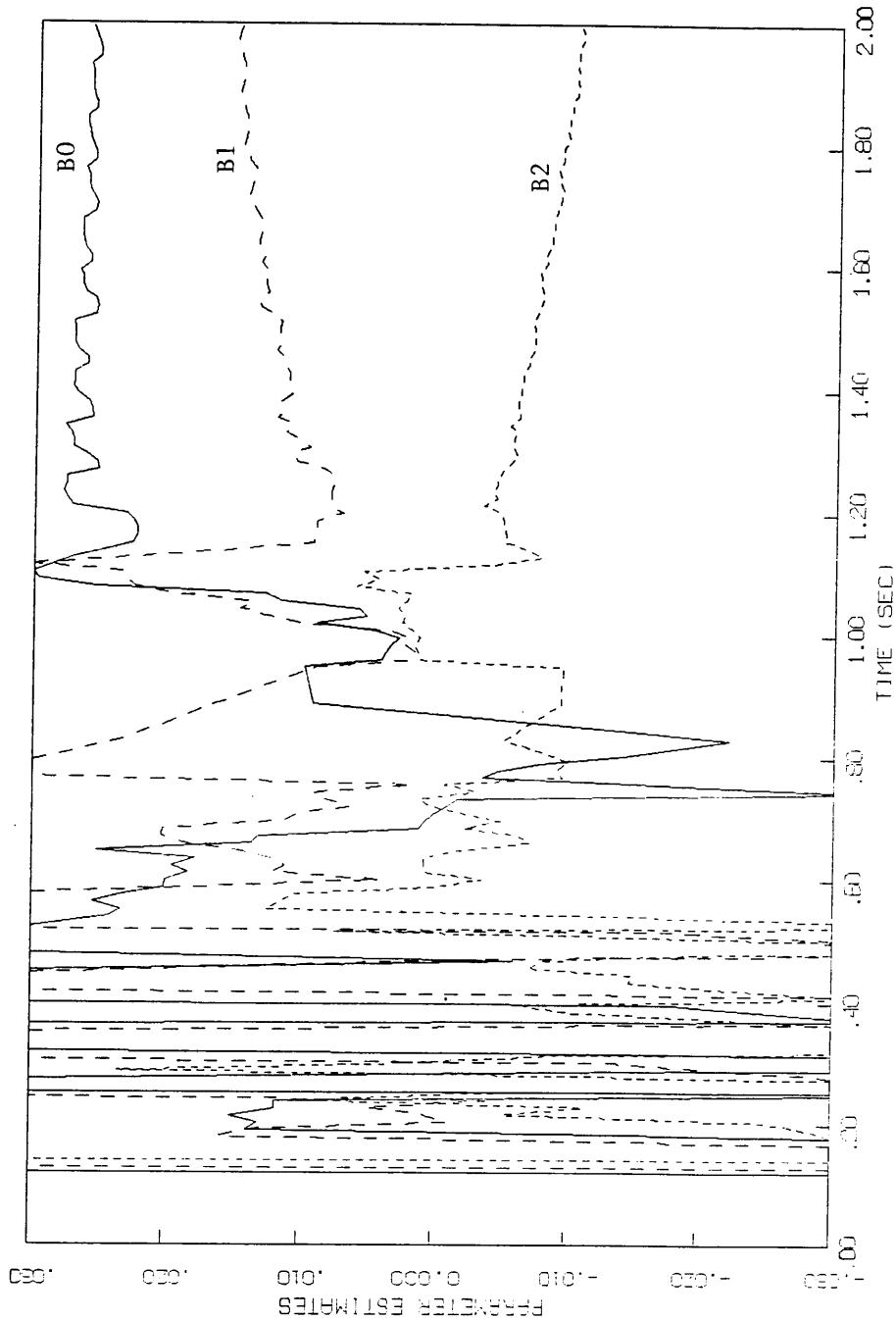


Fig. 6.15. ARMA Coefficients B0, B1, B2: TSAMP = 12 ms, EPSL = .0001, APWR = 396, ALPHAP = .01, Step

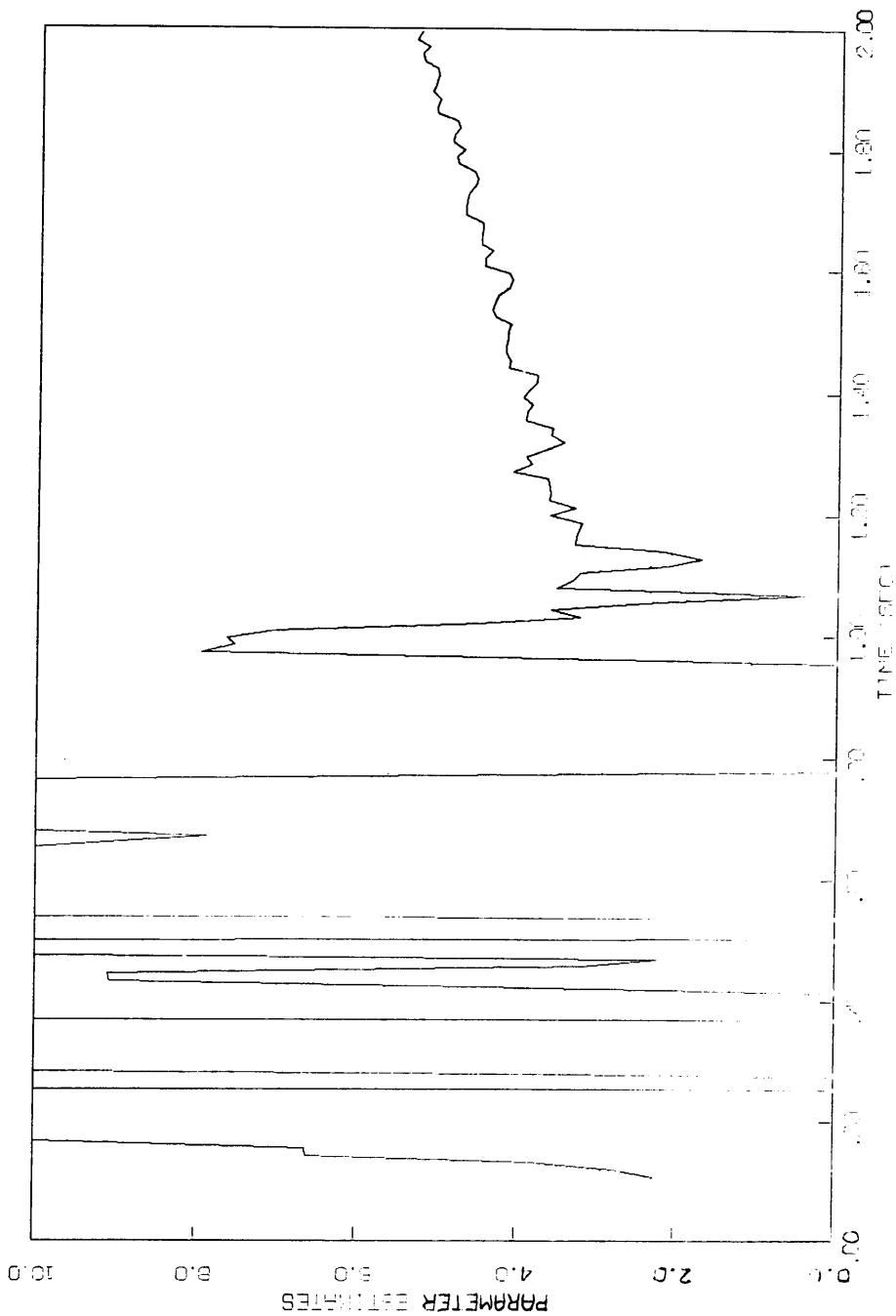


Fig. 6.16. ARMA Coefficient D: TSAMP = 12 ms, EPSL = .0001, APWR = 396, ALPHAP = .01, Step

during the first $t < 1$ second, then adapts to a steady-state value after time = 1 second. These problems are caused from instability in the parameter estimator due to the selection of the initial estimates and proper use of the exponential data weight ρ.

The parameter estimator was difficult to stablize. In general, the more samples taken per unit time, the higher the required value for $\rho(t)$ equation 5.2-13,14. Plots of the variable "forgetting" factor, $\rho(t)$, equation 5.2-15 for an $\rho' = 0.02$ and a 2 ms sample rate (Fig. 6.17) shows values closer to 1 than the $\rho(t)$ for $\rho' = .01$ (Fig. 6.18). This idea is more clearly illustrated for the Perturbation control, Section 7.

The general trend for the control gain, CGAIN, is illustrated for the step and ramp input functions (Fig. 6.19,20). The ARMA control gain adapts to a constant which shifts due to set-point changes. This is due to the requirement to minimize the performance index. Where the increase in control torque improves the velocity performance, thus minimizing the performance index.

The general trends for stable ARMA coefficients for the step input (Fig. 6.21,22,23) corresponding to (Figs. 6.5, 6.8) and ramp input (Fig. 6.24,25,26) corresponding to (Fig. 6.10, 6.20) are stable at one velocity setpoint then shifting to become stable at the next velocity setpoint.

The per revolution variation of the parameter estimates was not observed for different sample rates. However, the estimates for the 12 ms sample rate change (Fig. 6.24,25,26) show more fluctuation than the faster 4 ms interval coefficients (Fig. 6.21,22,23).

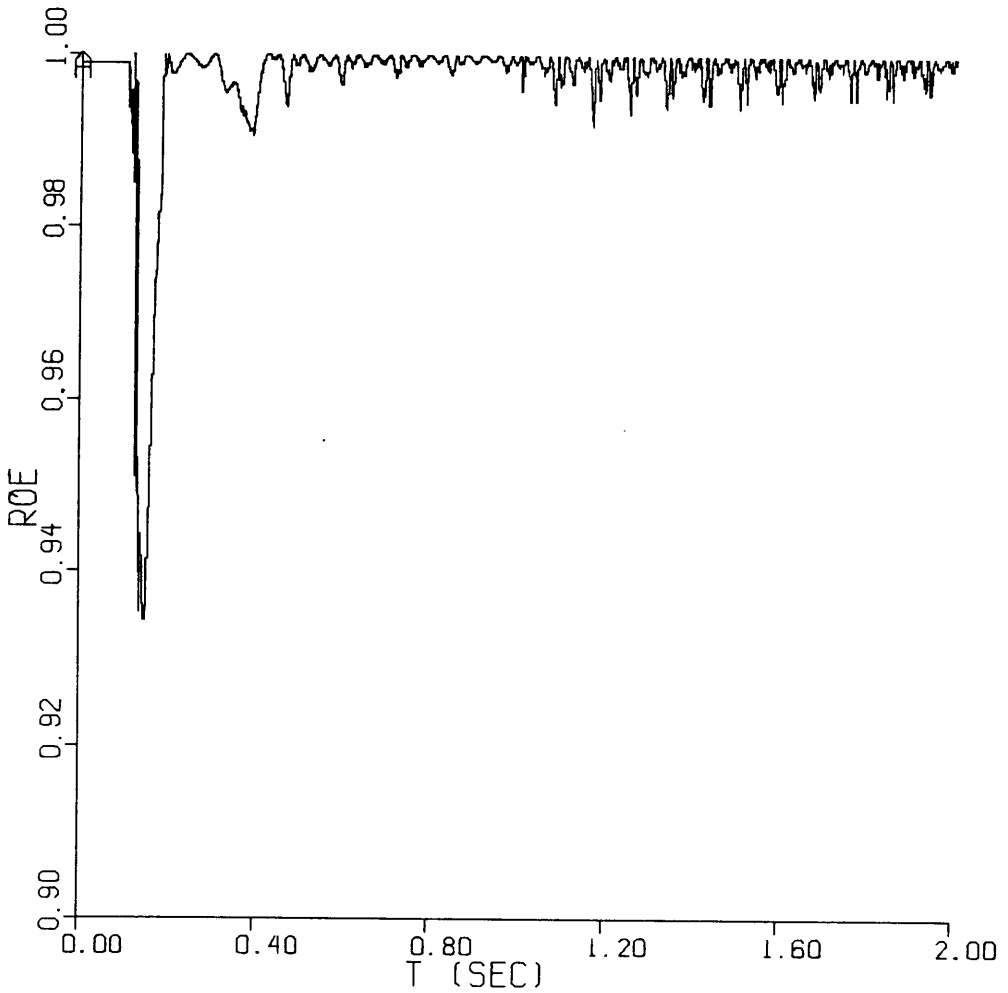


Fig. 6.17. ARMA Forgetting Factor ROE: TSAMP = 2 ms, EPSL = .0001, APWR = 183, ALPHAP = .02, Step

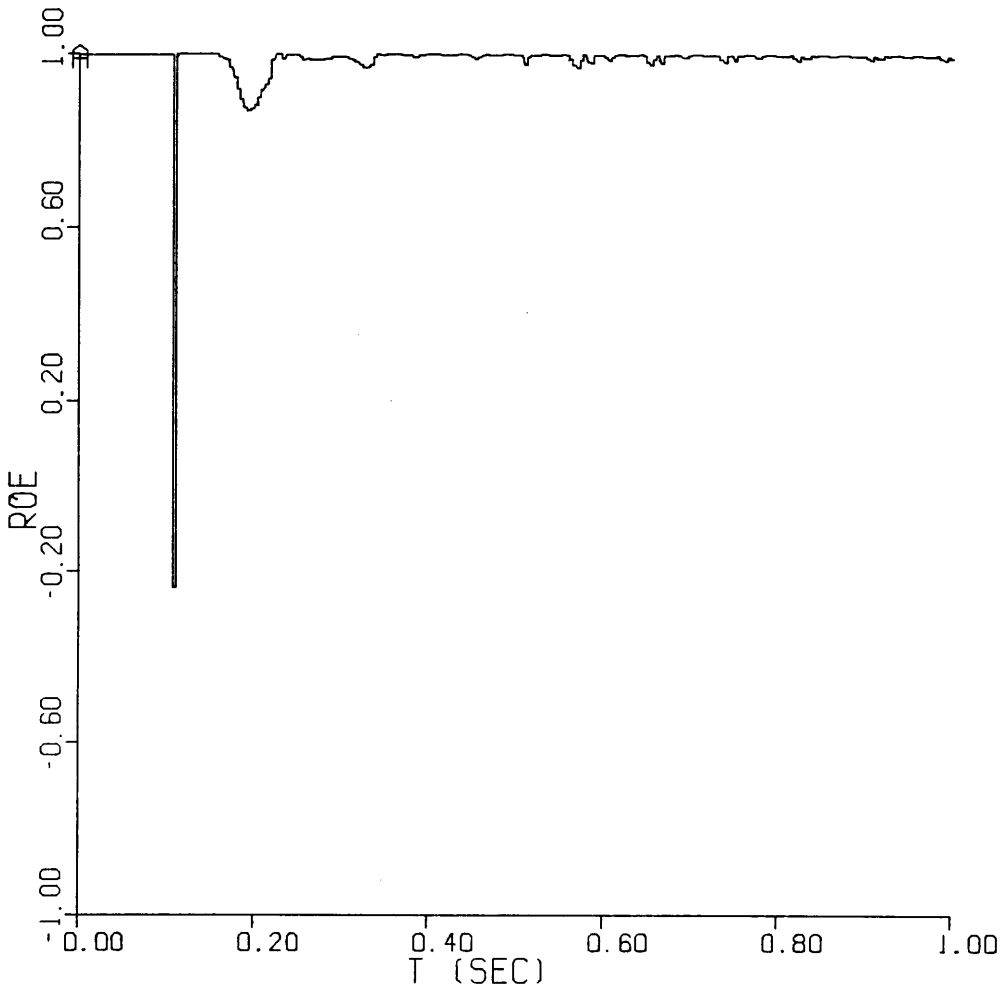


Fig. 6.18. ARMA Forgetting Factor ROE: TSAMP = 4 ms, EPSL = .0001, APWR = 183, ALPHAP = .01, Step

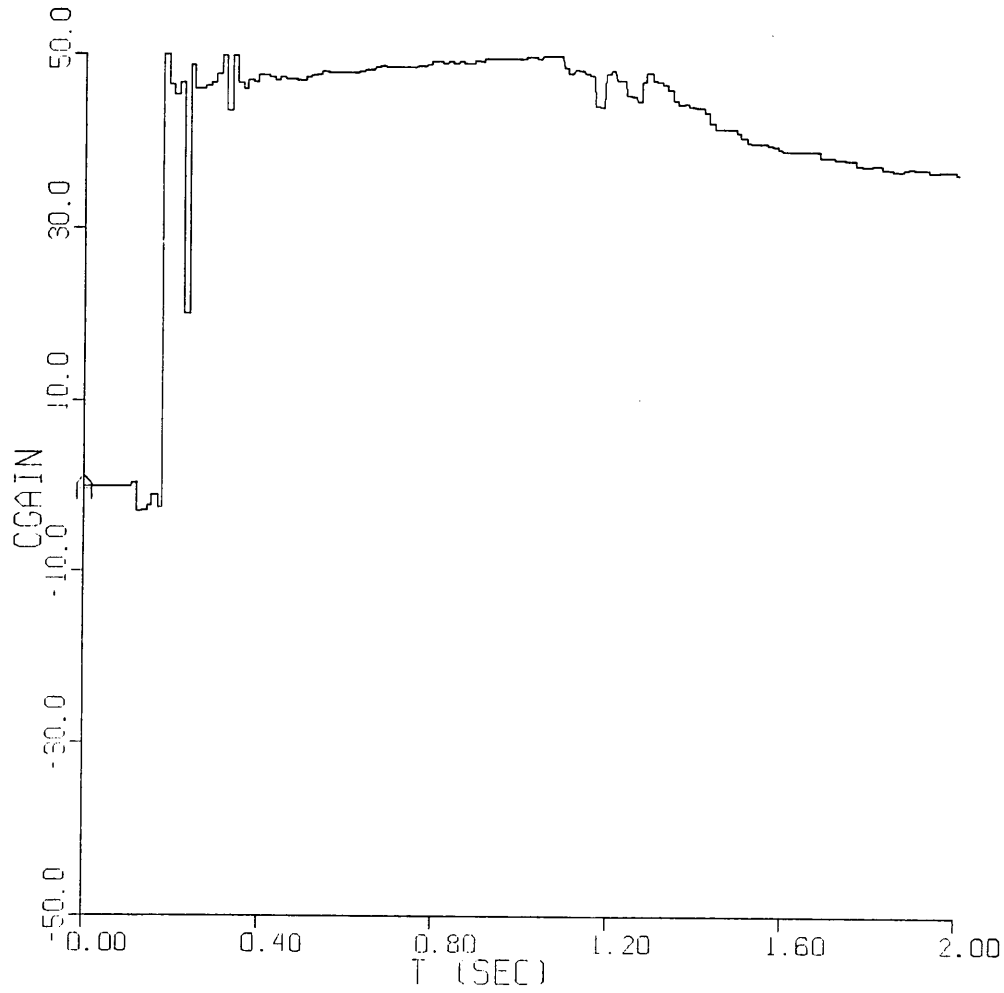


Fig. 6.19. ARMA Control Gain: TSAMP = 12 ms, EPSL = .0001, APWR = 183, ALPHAP = .001, Step

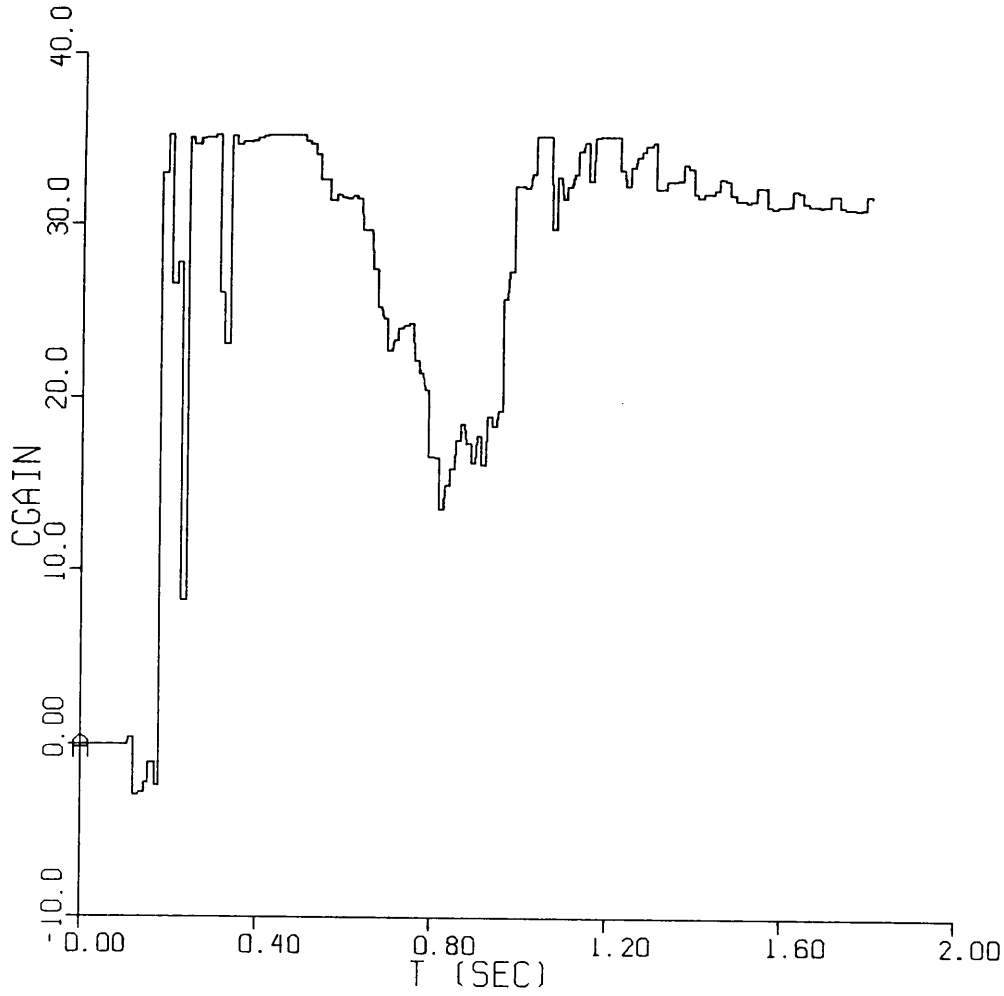


Fig. 6.20. ARMA Control Gain: TSAMP = 12 ms, EPSL = .0001, APWR = 183, ALPHAP = .02, Ramp

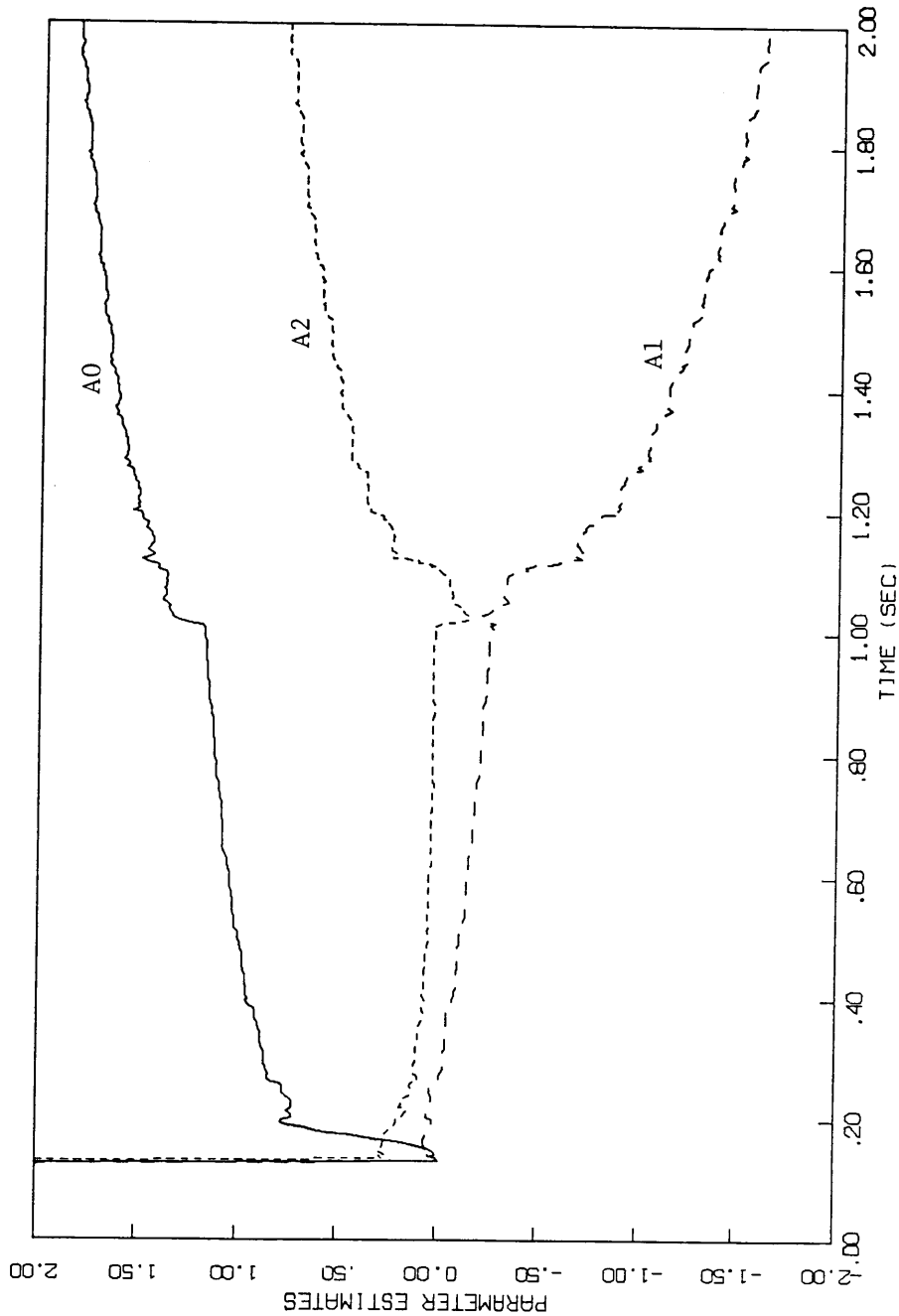


Fig. 6.21. ARMA Coefficients A0, A1, A2: TSAMP = 4 ms, EPSL = .00001, APWR = 396, ALPHAP = .02, Step

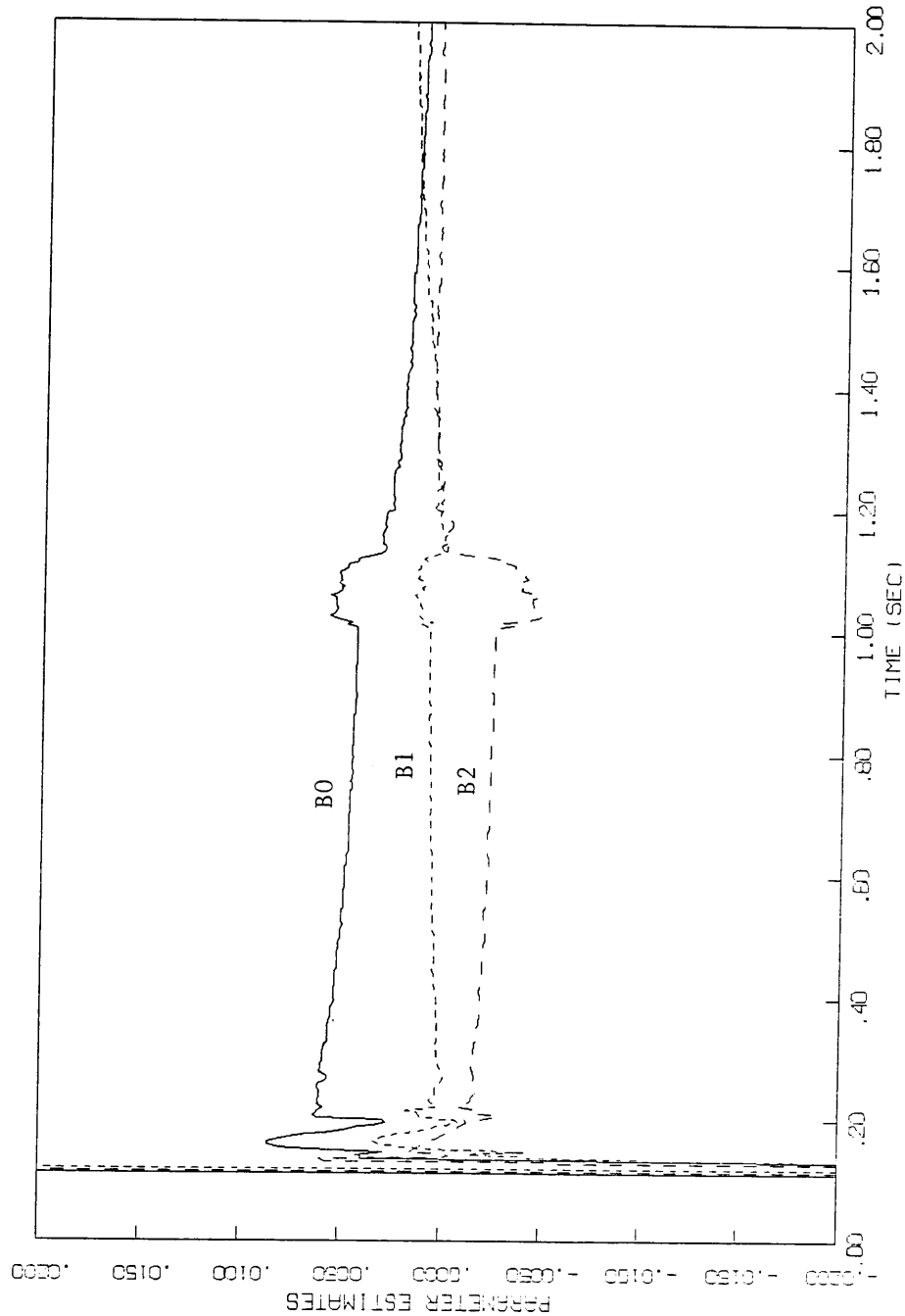


Fig. 6.22. ARMA Coefficients B0, B1, B2: TSAMP = 4 ms, EPSL = .00001, APWR = 396, ALPHAP = .02, Step

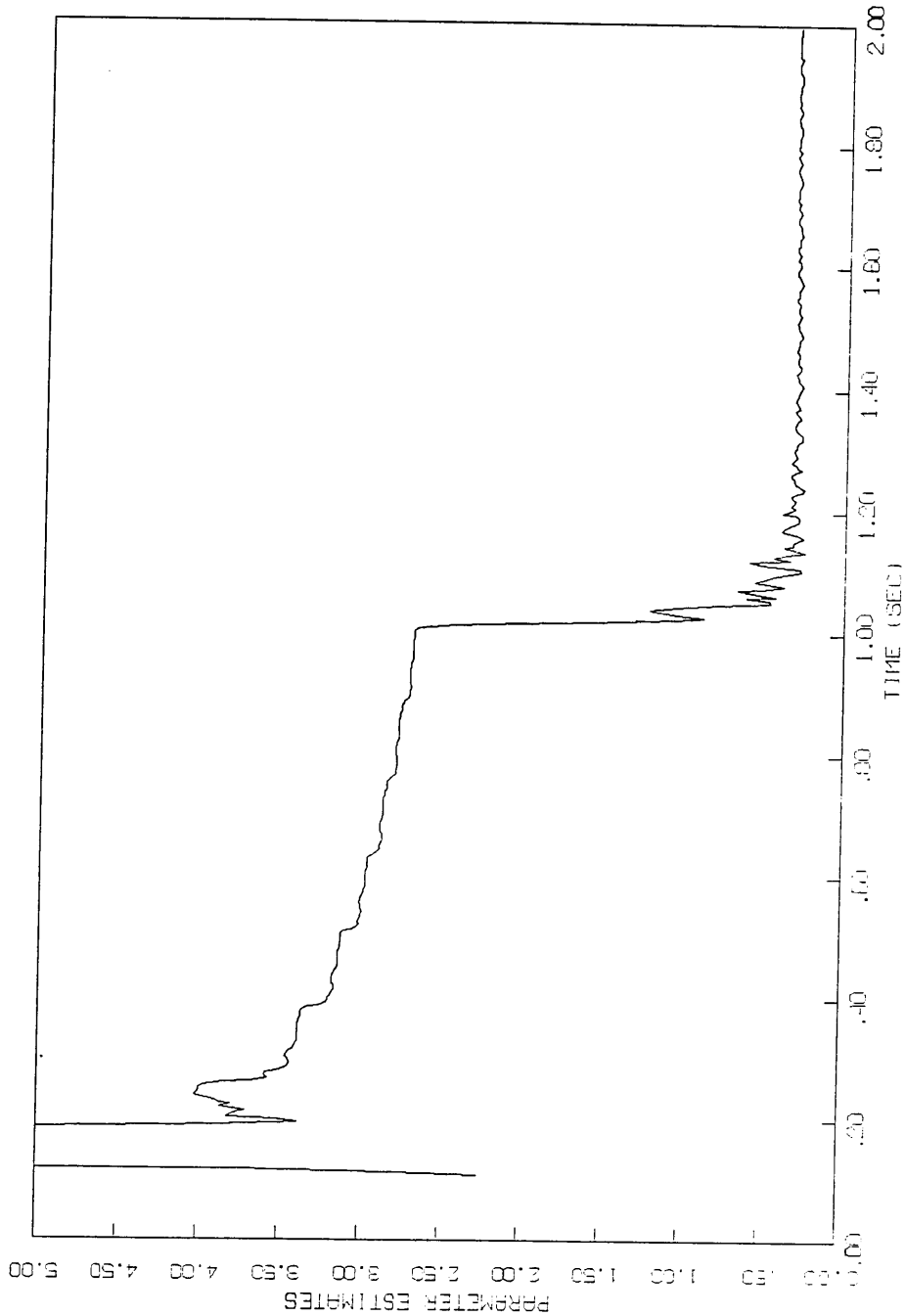


Fig. 6.23. ARMA Coefficient D: TSAMP = 4 ms, EPSL = .00001, APWR = 396, ALPHAP = .02, Step

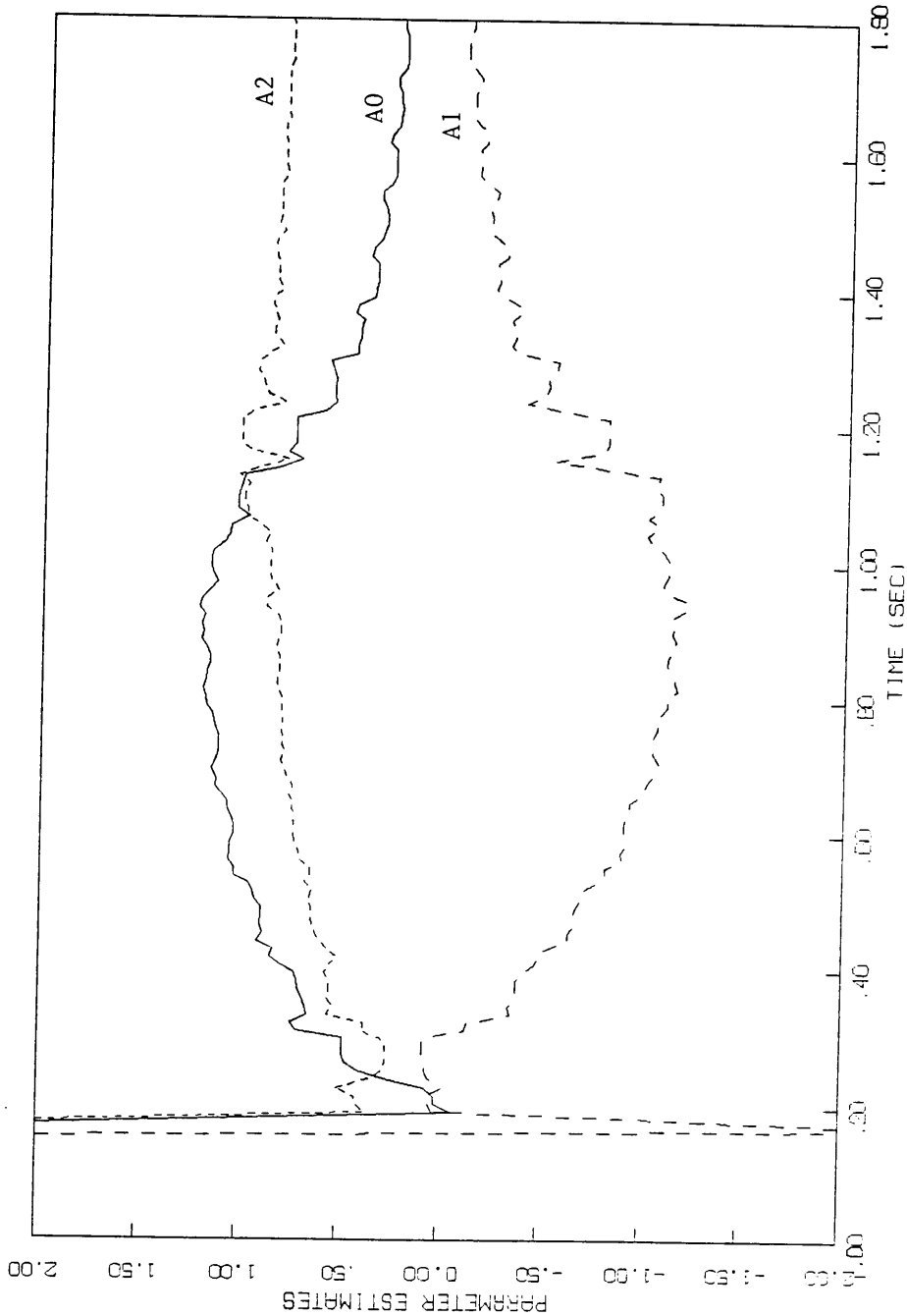


Fig. 6.24. ARMA Coefficients A0, A1, A2: TSAMP = 4 ms, EPSL = .0001, APWR = 183, ALPHAP = .02, Ramp

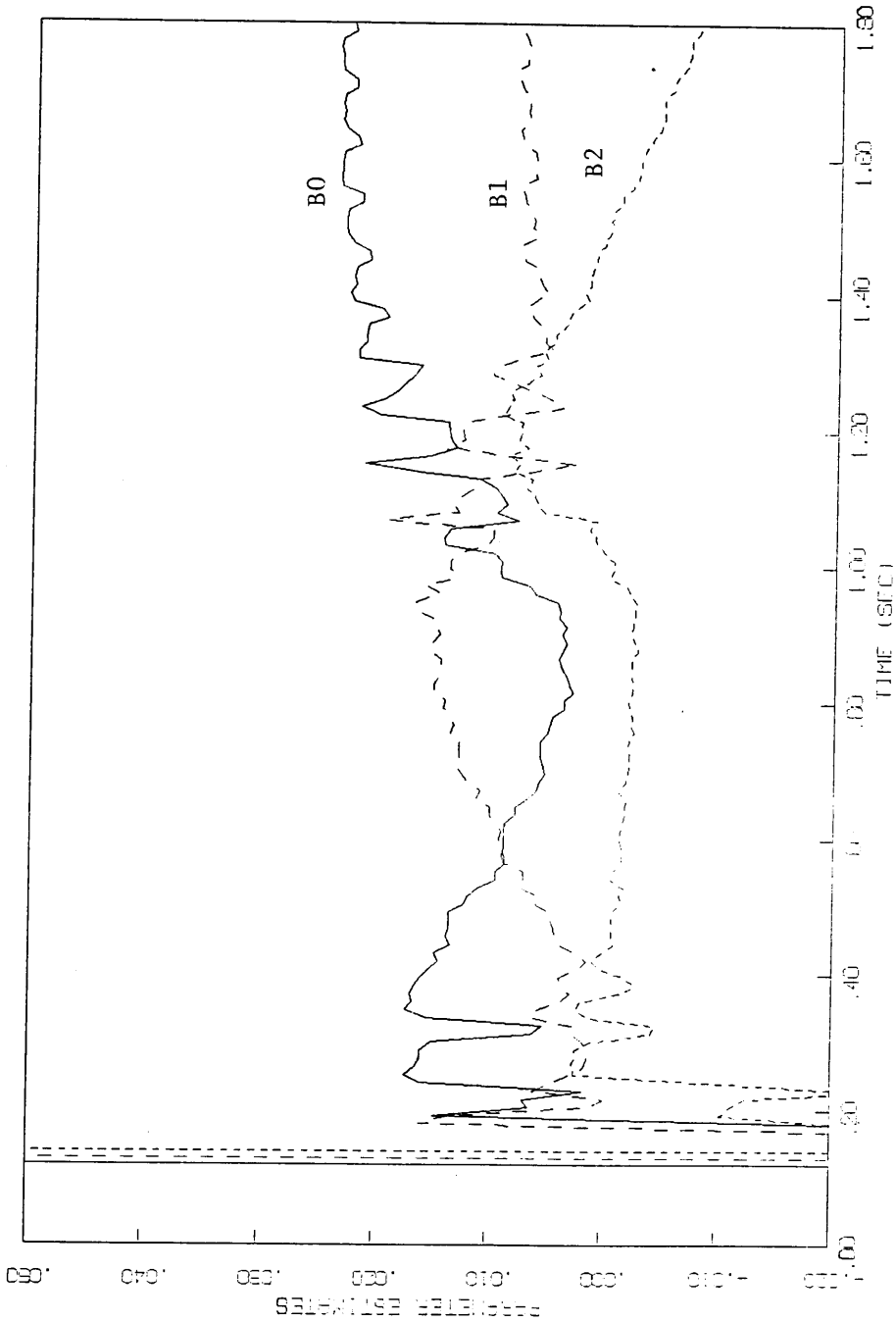


Fig. 6.25. ARMA Coefficients B0, B1, B2: TSAMP = 12 ms, EPSL = .0001, APWR = 183, ALPHAP = .02, Ramp

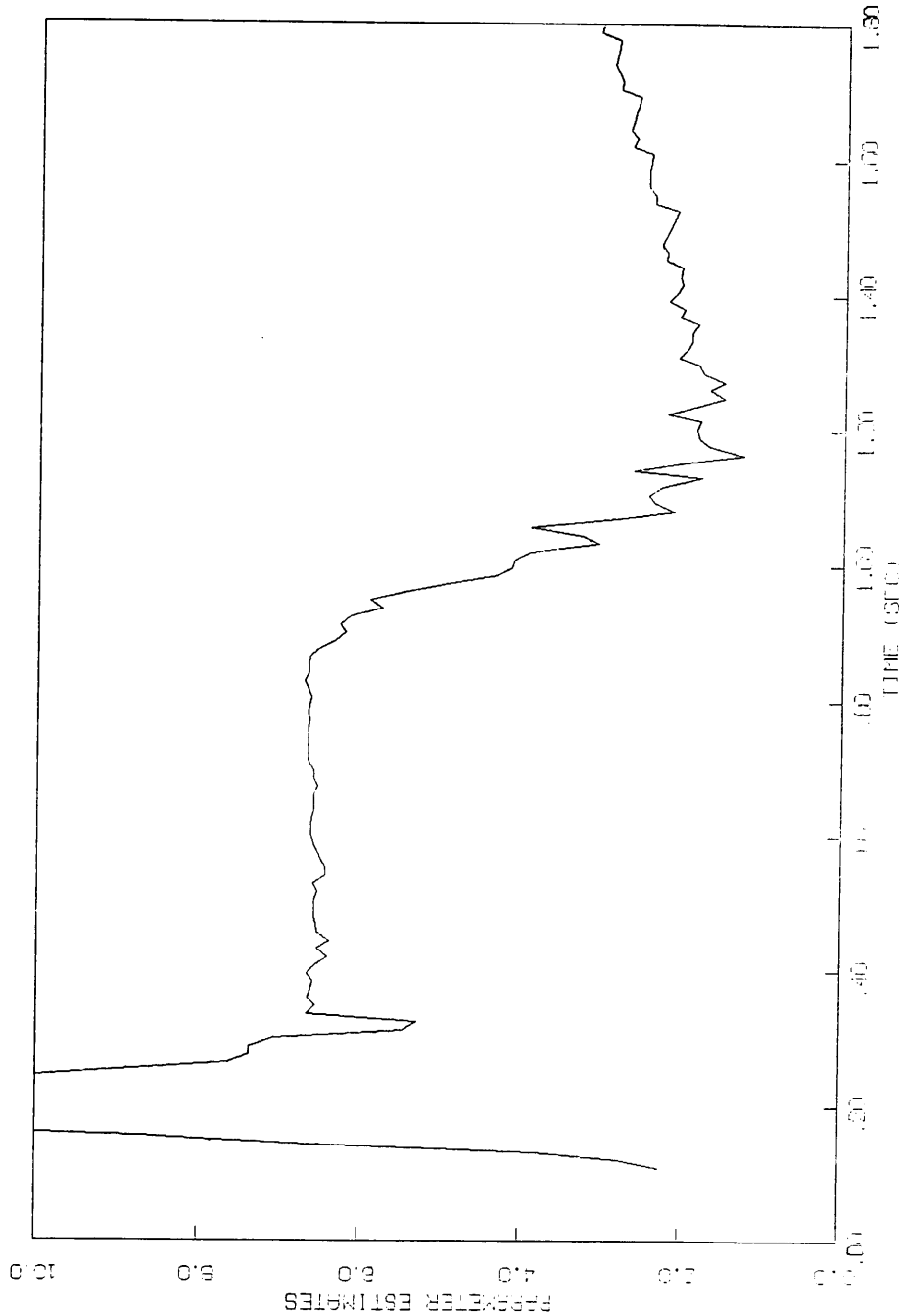


Fig. 6.26. ARMA Coefficient D: TSAMP = 12 ms, EPSL = .0001, APWR = 183, ALPHAP = .02, Ramp

As concluding observations, the ARMA adaptive control is not easy to implement. The required calculation times, for the third-order model, cause the algorithm to be sensitive to the selection of the estimator forgetting factor. As compared to the standard lead-lag control, the overall performance is not good and the stability over setpoint changes is questionable.

7.0 Perturbation Adaptive Control

The perturbation four-bar adaptive control uses the known mechanism dynamics and the ARMA adaptive control, Section 6. The control scheme is modeled after a robot position control developed by Lee and Chung [23]. The objective is to compute a required nominal input torque, T_n^*, around an input revolution using an open-ended computed torque model, the mechanism position and the desired velocity function. The nominal torque controls the gross mechanism dynamic nonlinearities. Any modeling errors or disturbances are then considered as linear perturbations to the nominal velocity trajectory. These perturbations are controlled using an ARMA adaptive control, Section 6. In general terms, the control scheme is very mechanism dependent, however it does provide the best overall performance for four-bar control.

The computed-torque model uses a pseudo position feedback as a means to separate the control of the nonlinear dynamic changes due to the mechanism geometry changes (kinetic changes) from the linear control required for velocity setpoint changes. This separates the nonlinear disturbance torque rejection (Section 2.3) from the control of the average mechanism inertia. In the perturbation control, the position dependence of the computed-torque model control the velocity fluctuation due to the kinetic changes at the desired velocity. The ARMA control, although adapts to account for computed-torque model errors, should only be required to control the steady-state velocity. The overall effect is: for a given (or assumed) constant velocity, the required torque (to reject the disturbance torques) will be applied by the computed-torque model given a zero velocity feedback error. Thus, the required torque

is applied due to the mechanism required motion and position only without regard to the actual mechanism velocity or the velocity feedback error.

7.1 Perturbation Derivation

The perturbation four-bar control derivation is straight forward. Following directly the derivation for the robot position control [23], we describe the desired nominal velocity path, ω_n, and a model of the four-bar dynamics (equation 3.1-2) repeated as

$$\dot{\omega} = \frac{T^* - \frac{dI^*}{d\phi}\omega^2 - T^*_G}{I^*} \qquad (7.1\text{-}1)$$

The functional dependence of I^*, T^*_G on the input position ϕ_i is implied. The linear Taylor series expansion for a small perturbation from the nominal velocity-torque path is

$$\dot{\omega}_n + \delta\dot{\omega} = \frac{1}{I^*}(T^* + \delta T^*) - \frac{2}{I^*}\frac{dI^*}{d\phi}\omega_n(\omega_n + \delta\omega) \qquad (7.1\text{-}2)$$

subtracting 7.1-1 from the displaced velocity ω yields the perturbation dynamic equation

$$\delta\dot{\omega} = \frac{1}{I^*}\delta T^* - \frac{2}{I^*}\frac{dI^*}{d\phi}\omega_n\,\delta\omega \qquad (7.1\text{-}3)$$

where the departure from the nominal velocity path is

$$\delta\omega = \omega - \omega_n$$

and the departure from the nominal computed torque is

$$\delta T^* = T^* - T_n^*$$

The overall control operates (Fig.7.1) by computing the nominal torque T_n^* using the computed-torque model

$$T_n^* = I^* \dot{\omega}_n + \frac{dI^*}{d\phi} \omega_n^2 + T_G^* \qquad (7.1\text{-}4)$$

the actual mechanism position and the nominal (desired) velocity path. This nominal torque is applied to the mechanism to control the major dynamic nonlinearities. The linear perturbations, equation (7.1-3), are then controlled using a first order ARMA control, Section 6.

Several observations should be recognized. The first is the better the computed-torque model, the better the overall control. If the complete requirements of the mechanism are known, usually not the case, and can be applied the control should be exact. There are two limiting cases for this fact. The digital control introduces sampled-data errors into the computed torque wave form, and the torque actuator has a limited output. There also exists a problem at discontinuities in the velocity path. At a discontinuity, a step, the computed torque is infinite. Since this amount of torque is not available the mechanism does not accelerate to follow the step. However, the computed-torque model continues to calculate torque values based on the greater velocity. Therefore, the computed torque is incorrect relative to the actual torque required to correct the mechanism velocity. This leads to

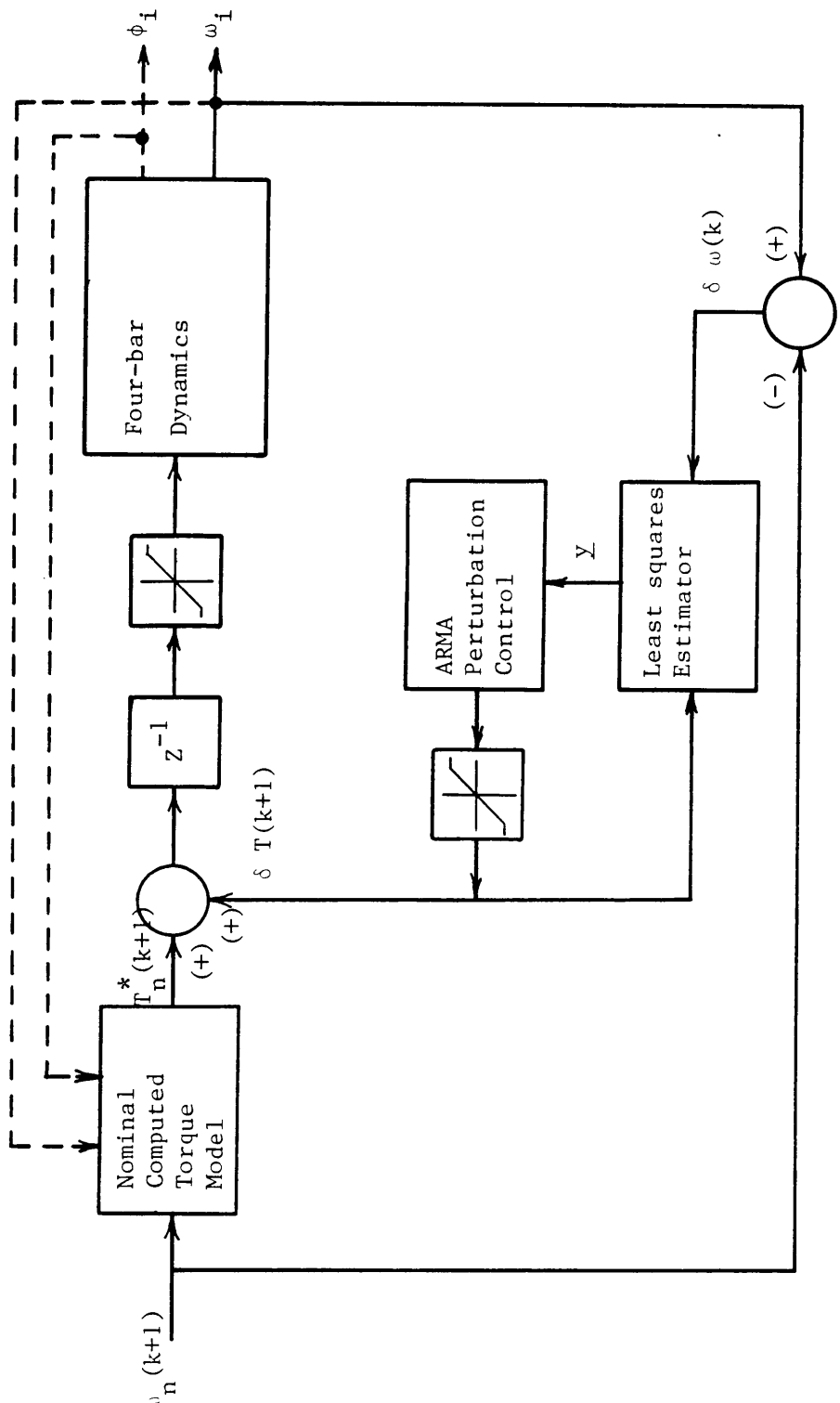


Fig. 7.1 Perturbation Adaptive Control Block Diagram

problems when the mechanism fails to follow the desired velocity path.

7.2 Control Algorithm

The perturbation control loop block diagram (Fig. 7.1) shows the layout of the control system. Note here that the velocity "feedback" to the computed-torque model is only used to estimate the next time (k + 1) position of the mechanism. There is no true velocity feedback to the computed-torque model. The ACSL simulation PERTURBATION appear in Appendix B.

The computed-torque model is derived using kinematic influence coefficients, Section 2. Since the influence coefficient model is a function of the input position, it can be precomputed and stored in look-up tables. The required coefficients can then be recalled using the mechanism position which saves controller computing time. Therefore, the computed torque $T^*(k + 1)$ is calculated using the nominal input acceleration

$$\dot{\omega}_n = \frac{\omega_n(k + 1) - \omega_n(k)}{\delta}$$

and the predicted position

$$\phi^p(k + 1) = \phi(k) + \delta\ \omega(k) \ .$$

Here $\phi(k)$, and $\omega(k)$ are the "feedback" position and velocity. This yields the computed-torque model.

$$T_n^* = I^*(k+1)\left(\frac{\omega_n(k+1) - \omega_n(k)}{\delta}\right) + \frac{dI^*(k+1)}{d\phi}\,\omega_n^2 + T^*_G(k+1)\ .$$

The linear pertubations (equation 7.1-3) are then controlled using a first order, N = 1, ARMA adaptive control, Section 6.

Two torque limits are imposed in the Perturbation control. The first is a limit on the available motor torque to satisfy the specifications of Section 3.3. The second limit is applied to the ARMA perturbation control torque δT^* to prevent the introduction of false data into the ARMA control loop. Recall, the ARMA prediction model is based on the input/output data, $\delta\omega$, δT^*. If at a given sample the controller actuator can not supply the total required torque $T^* + \delta T^*$ the perturbation correction torque δT^* is clipped. However, the ARMA loop is internal to the overall control. Therefore, the actual applied portion of δT^* is different than the computed, and stored, value of δT^*. This constitutes entering false data into the ARMA prediction model. The internal limit on δT^* is to circumvent, or control, the maximum data errors the ARMA prediction receives thus improving its operation.

For the simulations, the influence coefficient model was "tabulated" every 6 degrees of rotation and the adaptive control estimated the three parameters. The look-up table memory requirement is estimated using 4 influence coefficients evaluated at 60 points using single precision numbers. This requires less than 1024 8 bit bytes of memory using 4 bytes per single precision floating point number. The total calculation time for operating the look-up table and estimating 3 parameters is approximately 3 milliseconds.

This page intentionally left blank

7.3 Simulation Studies

The Perturbation control was simulated for the step and ramp input functions, sample intervals of 4 and 2 milliseconds using the 183 and 396 in-oz (12.9×10^3, 27.9×10^3 cm-dyn) motor torque limits. The tabulated influence coefficients are used as exact values, however, there is approximately 2% error introduced into the values for the computed torque model values for the link inertias and masses. The ARMA perturbation control has the same start-up problems as before (Section 6) and the same torque application was used to "start" the controller. Several observations can be made from the simulation data. The most substantial is that this controller provides the best performance for maintaining a constant velocity. The fluctuations are less than 2% at 50 rad/sec and less than 4% at 75 rad/sec.

The velocity plots for eight simulations (Fig. 7.2-9) show the general performance of the perturbation control. The best performance to guarantee the kinematic input condition is for the control using a 2 ms sample interval, 396 in-oz (27.9×10^3 cm-dyn) torque limit and a 50 in-oz (3.5×10^2 cm-dyn) ARMA torque limit (Fig. 7.8). The best performance in this respect at the 4 ms sample interval is also for the higher torque limit (Fig. 7.5). Considering the computed torque plot (Fig. 7.10), we can easily see that the computed torque oscillates with a 390 in-oz (27.5×10^3 cm-dyn) magnitude. With a motor only capable of producing 183 in-oz (12.9×10^3 cm-dyn) of torque, this demand cannot be met. Therefore, to gain the extra performance increase, a larger motor must be used. However, even if the difference between the computed torque and the output torque is limited (Fig. 7.11), the loss in fluc-

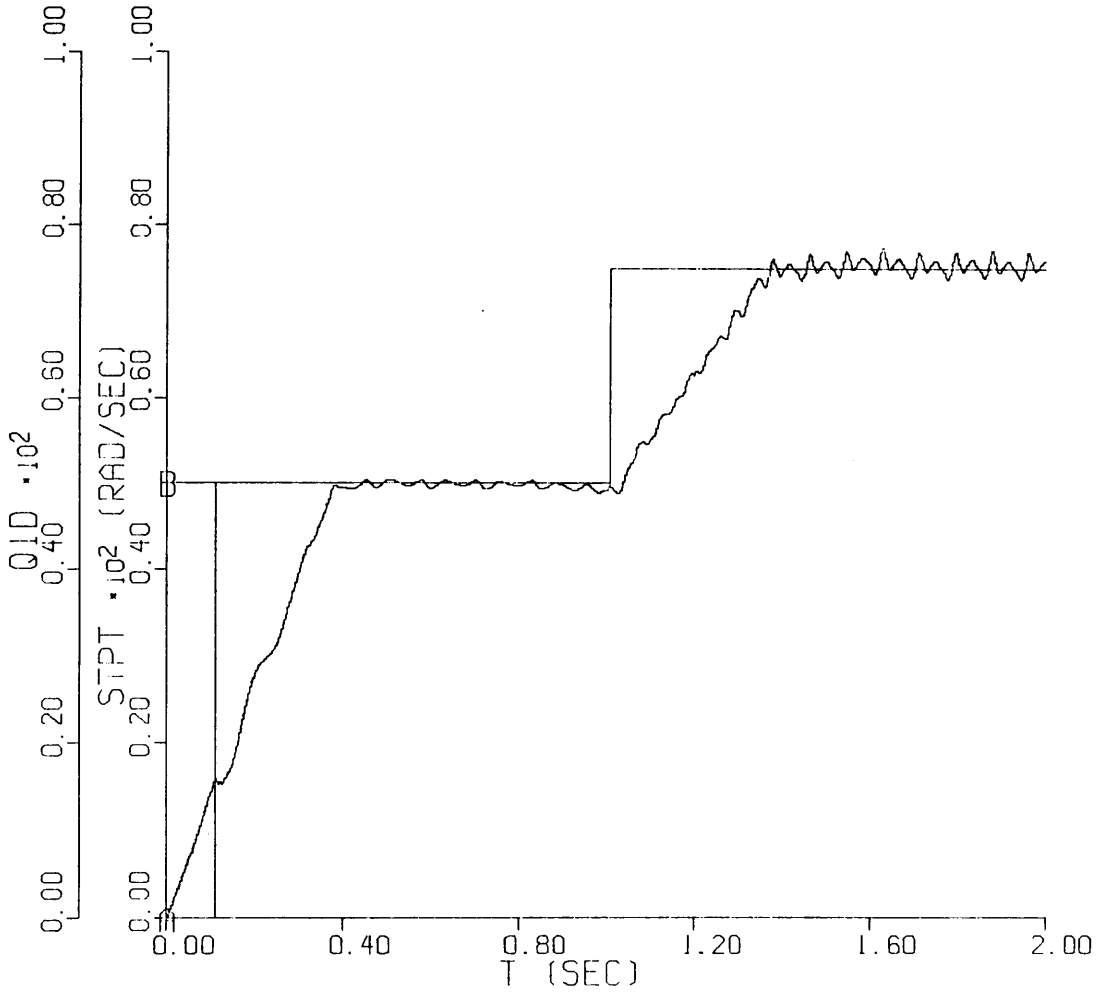


Fig. 7.2. Perturbation Velocity: TSAMP = 2 ms, EPSL = .0002, APWR = 183, ROE = .95, ARMA Torque = 100, Step

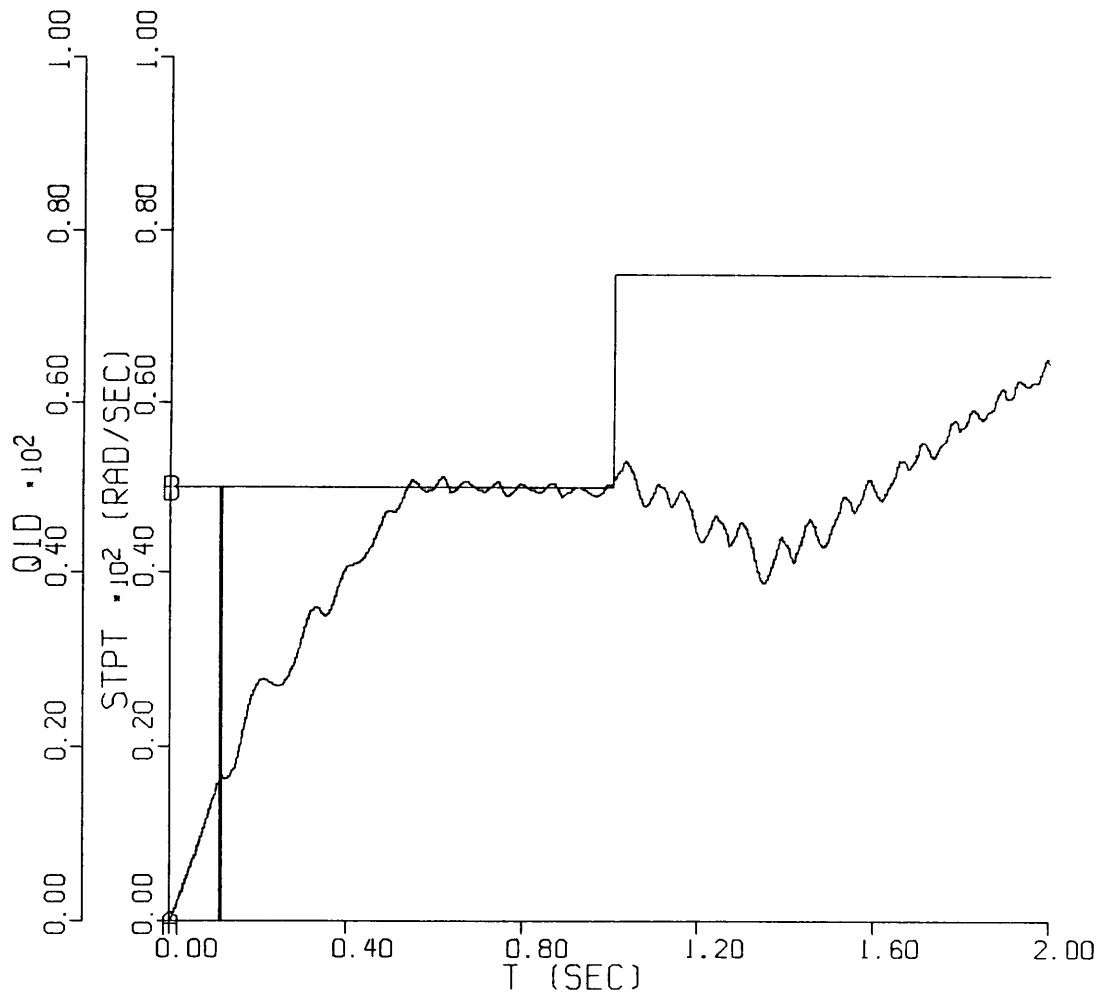


Fig. 7.3. Perturbation Velocity: TSAMP = 4 ms, EPSL = .0002, APWR = 183, ROE = .92, ARMA Torque = 50, Step

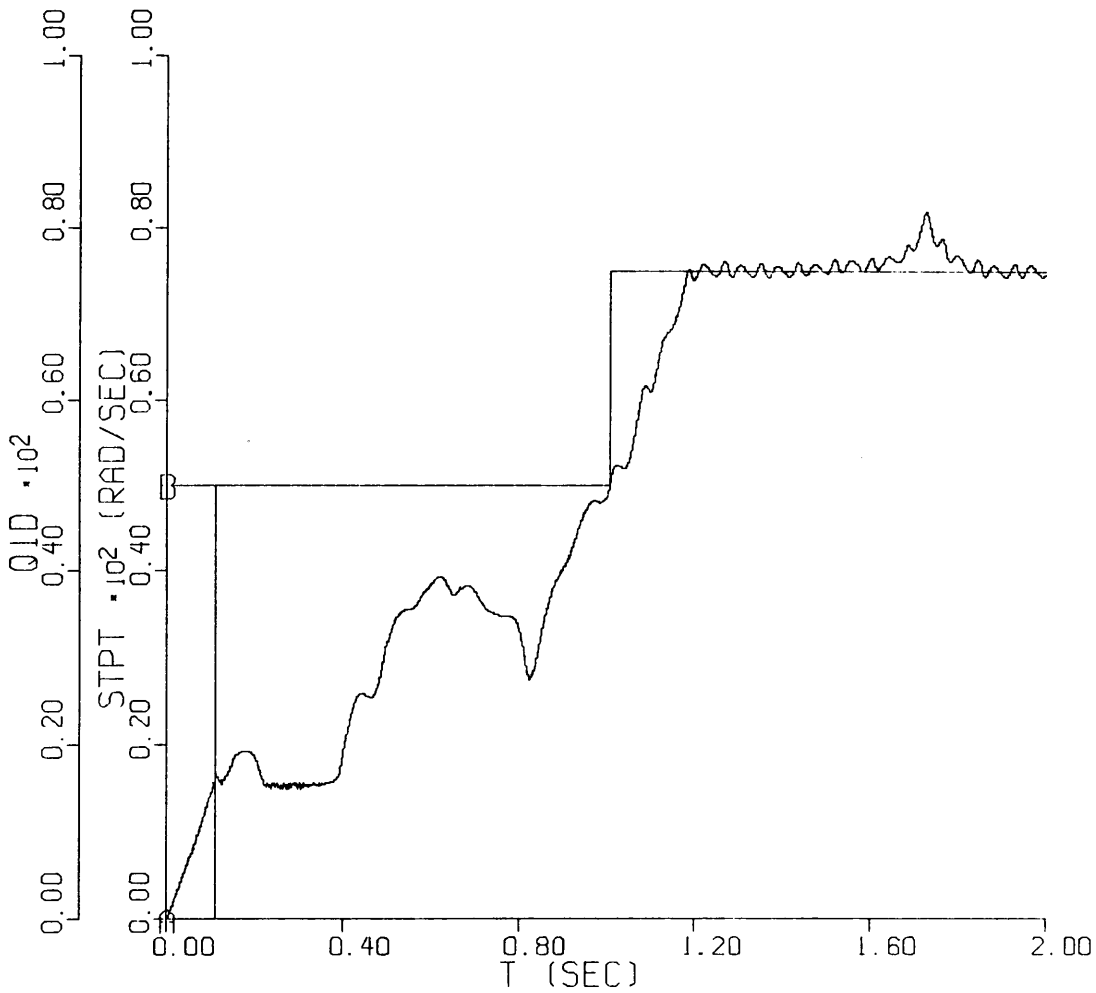


Fig. 7.4. Perturbation Velocity: TSAMP = 2 ms, EPSL = .0002, APWR = 396, ROE = .95, ARMA Torque = 100, Step

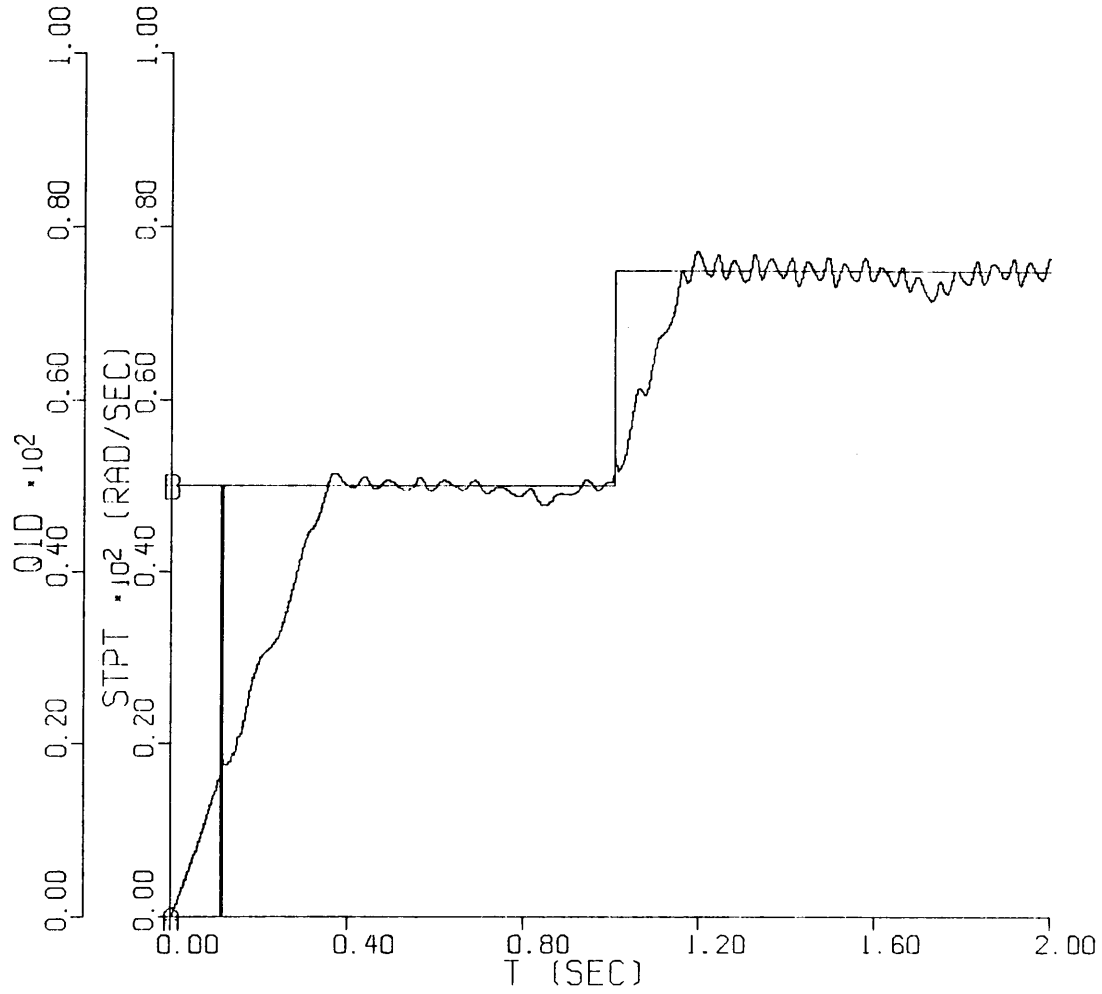


Fig. 7.5. Perturbation Velocity: TSAMP = 4 ms, EPSL = .0002, APWR = 396, ROE = .92, ARMA Torque = 100, Step

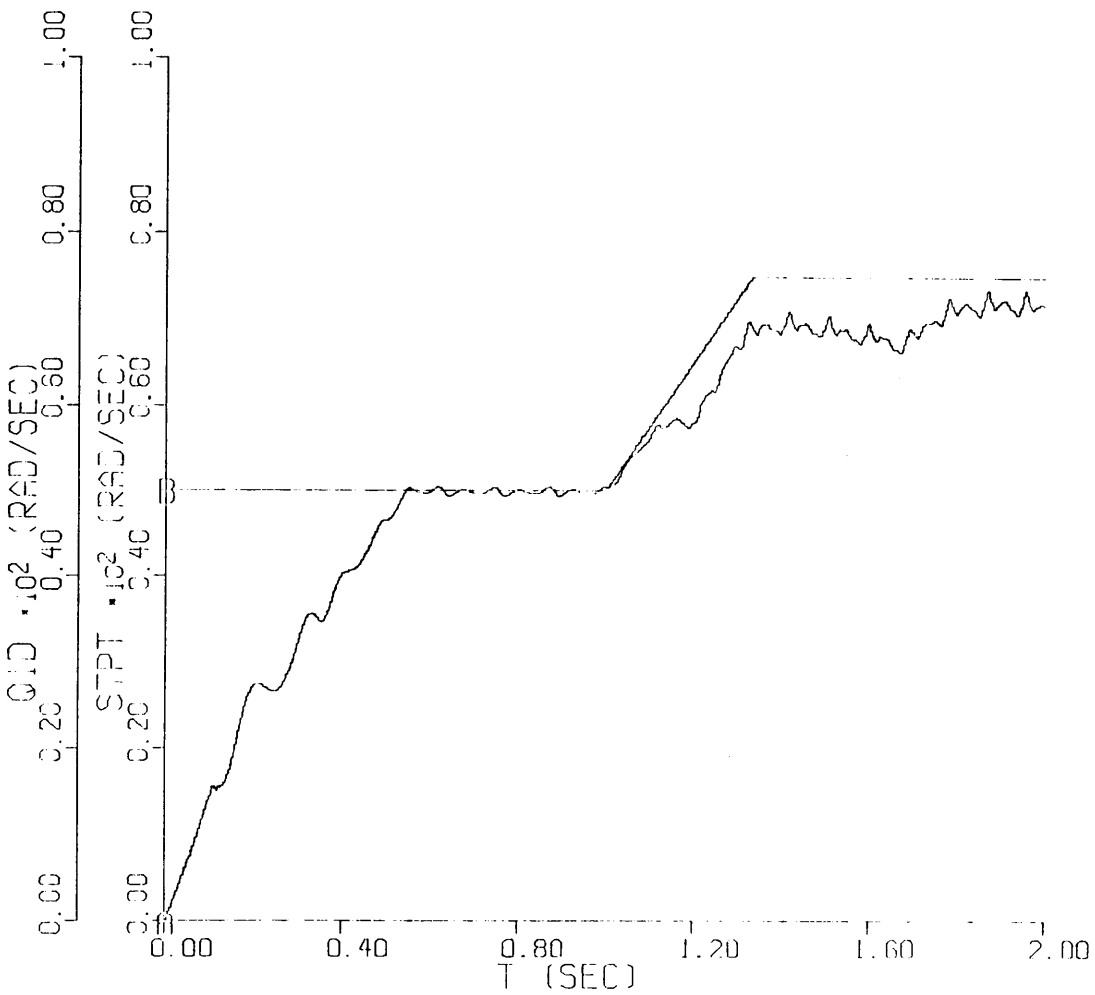


Fig. 7.6. Perturbation Velocity: TSAMP = 2 ms, EPSL = .0002, APWR = 183, ROE = .95, ARMA Torque = 50, Ramp

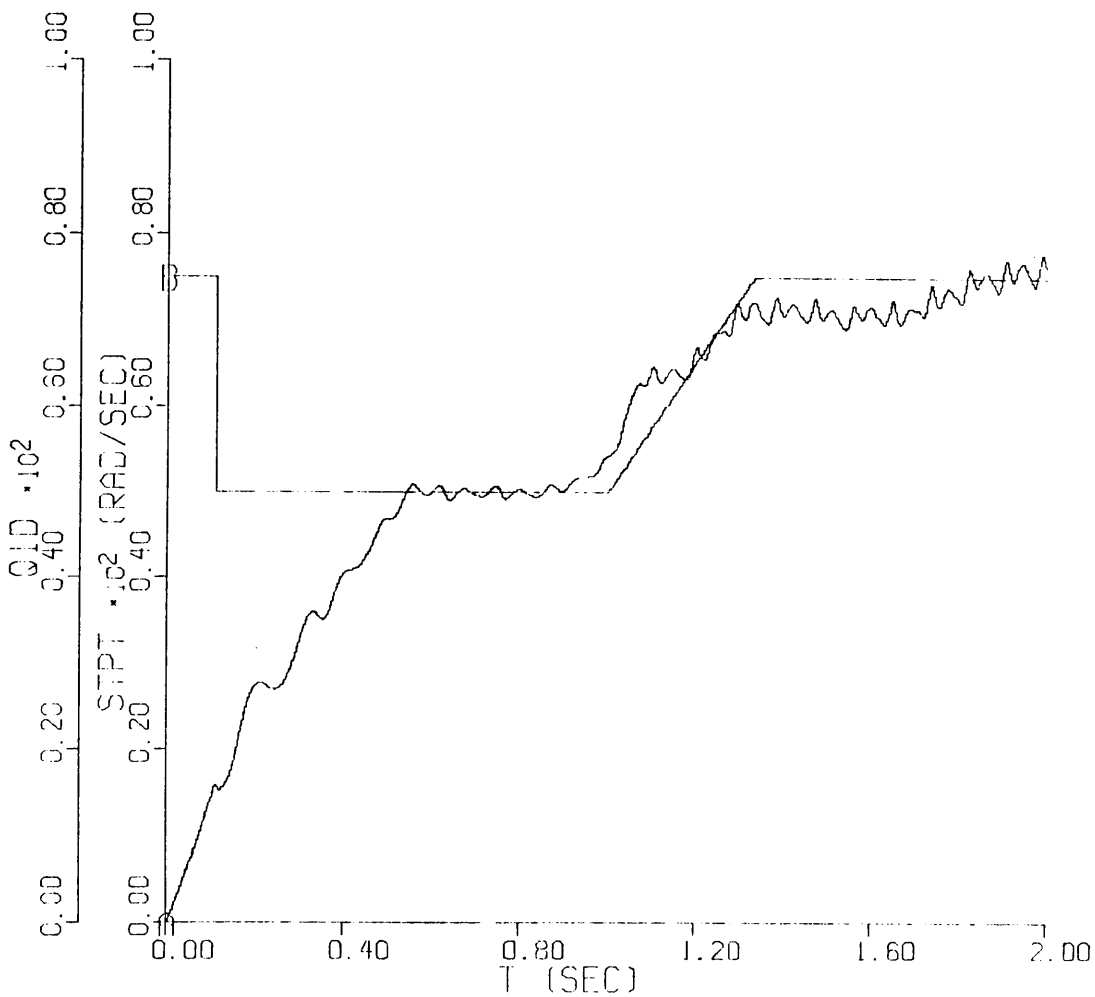


Fig. 7.7. Perturbation Velocity: TSAMP = 4 ms, EPSL = .0002, APWR = 183, ROE = .92, ARMA Torque = 50, Ramp

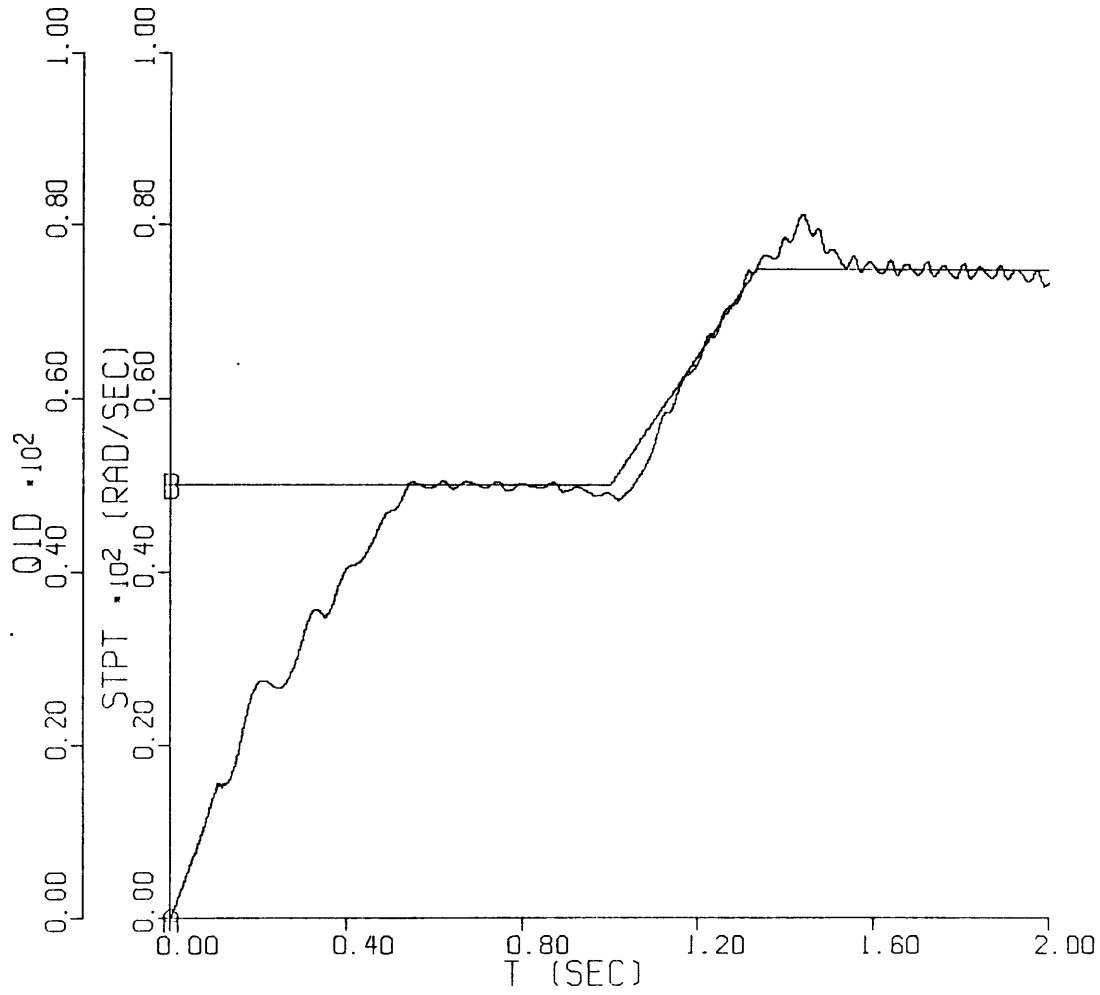


Fig. 7.8. Perturbation Velocity: TSAMP = 2 ms, EPSL = .0002, APWR = 396, ROE = .95, ARMA Torque = 50, Ramp

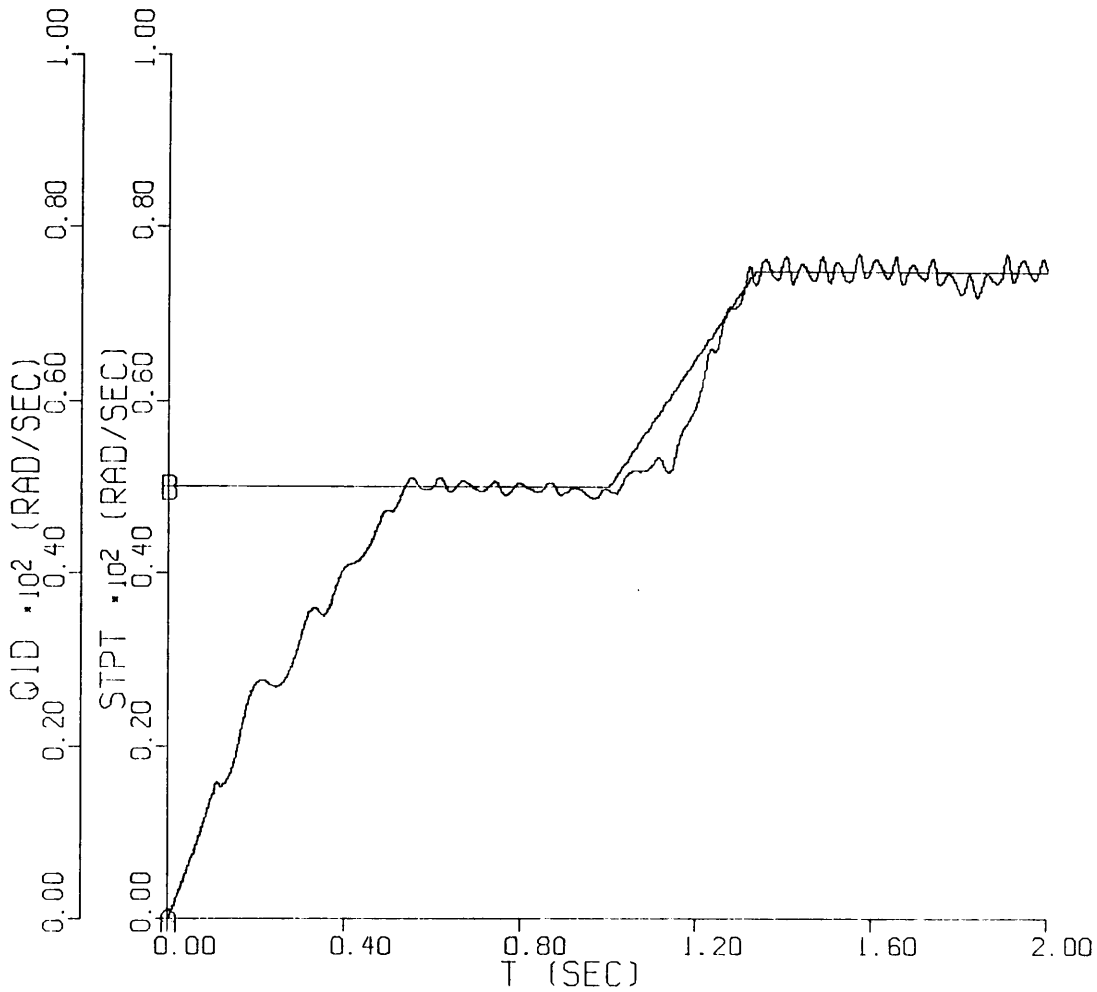


Fig. 7.9. Perturbation Velocity: TSAMP = 4 ms, EPSL = .0002, APWR = 396, ROE = .92, ARMA Torque = 50, Ramp

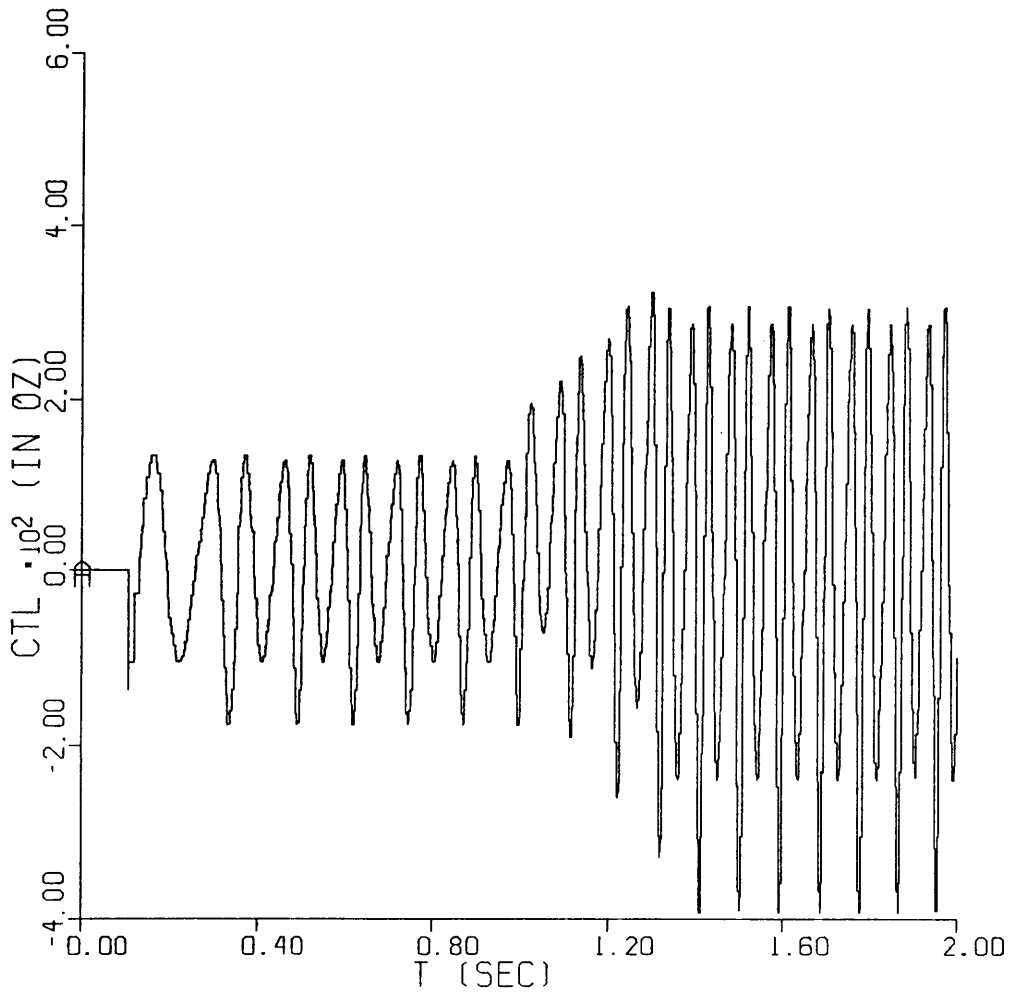


Fig. 7.10. Perturbation Computed Torque: TSAMP = 2 ms, EPSL = .0002, APWR = 183, ROE = .95, ARMA Torque = 50, Ramp

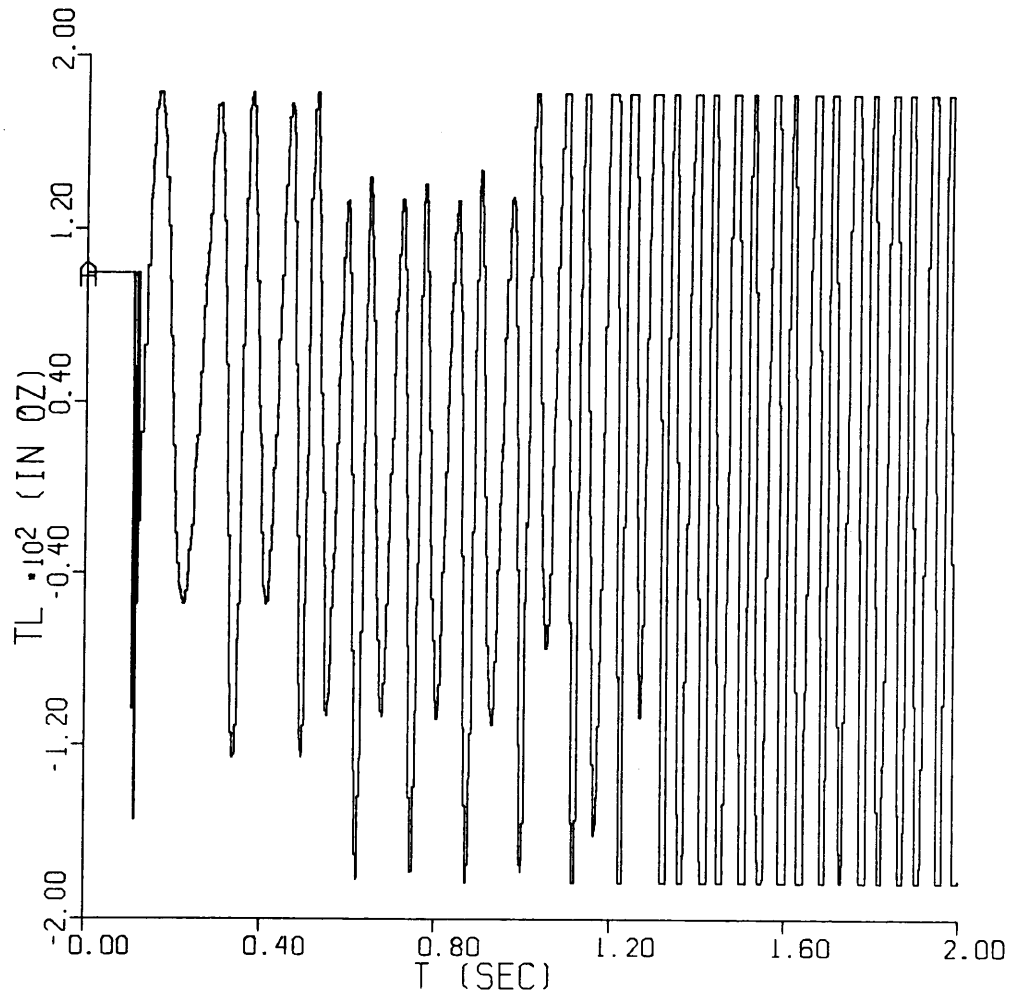


Fig. 7.11. Perturbation Applied Torque: TSAMP = 2 ms, EPSL = .0002, APWR = 183, ROE = .95, ARMA Torque = 50, Ramp

tuation performance is less than 2%.

For both torque limit cases, the best fluctuation performance is for the 2 ms sample interval. As illustrated in Figs. (7.2, 7.8), the velocity fluctuations are less than 2% at 50 rad/sec and less than 4% at 75 rad/sec. This is an expected result. The shorter sample interval reduces the sample-data effects, thus the discrete-time control performs more like a perfect continuous computed-torque control.

The rise time to a step input is controlled by the ARMA perturbation control loop. The open-ended computed-torque control "sees" the step function as an input discontinuity and only produces an accelerating torque for one sample interval. After the step increase, it computes the torque for the new nominal velocity without regard to the actual mechanism velocity. Therefore, this "perturbation" error has to be controlled by the ARMA loop. The overall rise time is then limited by the torque limit placed on the ARMA correction torque δT^*. Considering the velocity plots, (Figs. 7.2, 7.5), the observed rise times are .17, .13 seconds, respectively. Since the ARMA controls are well-behaved, the different sample intervals can be overlooked. Therefore, the different rise times are directly related to the ARMA control torque applied.

The tracking ability is better behaved. Here, the computed-torque model is able to follow the finite 75 rad/sec^2 acceleration to produce the gradually increasing computed torque (Fig. 7.10). Good ramp tracking is illustrated in Fig. 7.8 for a 2 ms sample interval and the 396 in-oz (27.9×10^3 cm-dyn) torque limit. The perturbation torque plot (Fig. 7.12) shows the proper correction action. The maximum torque

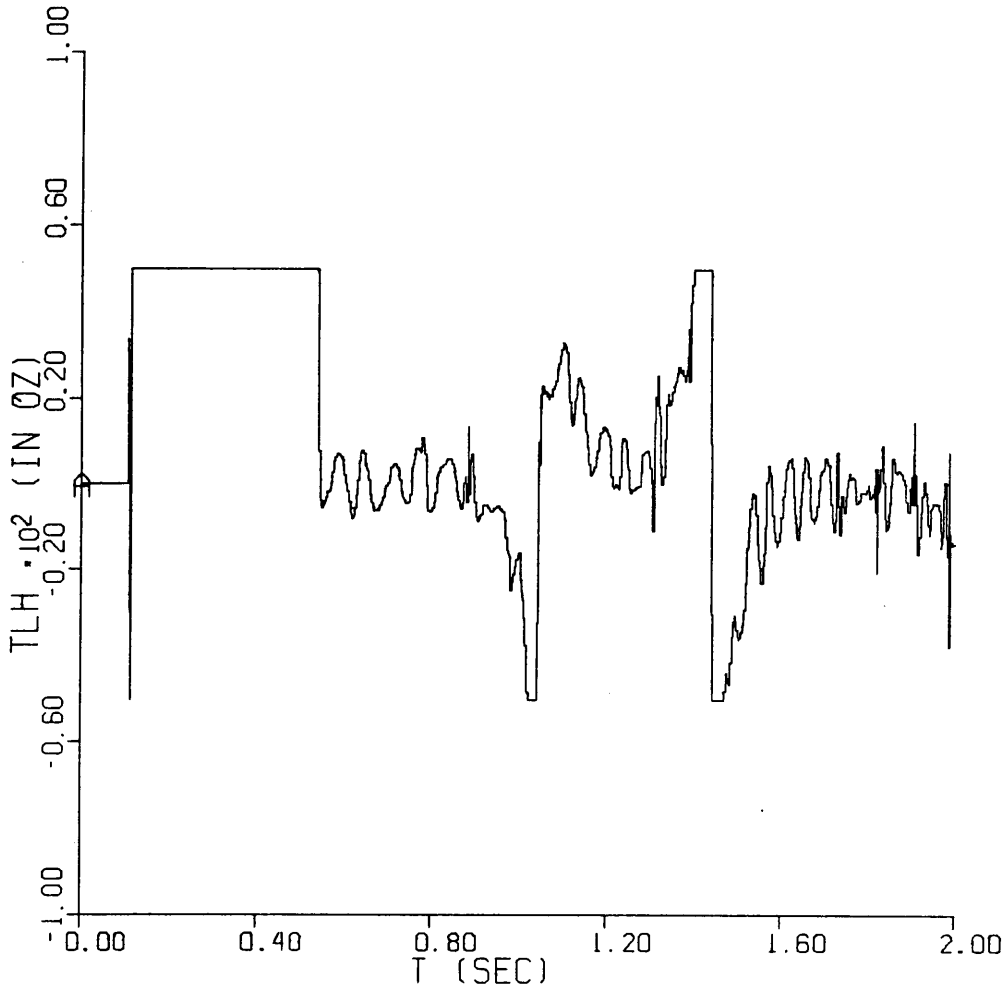


Fig. 7.12. Perturbation ARMA Torque: TSAMP = 2 ms, EPSL = .0002, APWR = 396, ROE = .95, ARMA Torque = 50, Ramp

is used from .1 to .5 seconds to accelerate the mechanism. The sinusoidal corrections are applied from .5 to .9 seconds. The unstable correction torque which causes the velocity sag and overshoot at .9 and 1.3 seconds, respectively, are shown.

The velocity plot (Fig. 7.6) shows poor tracking. Here the problems are caused by the limited motor torque and instabilities in the ARMA control loop.

The perturbation four-bar control stability is determined from the ARMA stability, which is controlled by adjusting the estimator forgetting factor ρ. A particular stable forgetting factor required several simulations to find. However, the general trend is a shorter sample interval requires a higher, closer to 1, "forgetting" factor. This stands to reason. With the faster sample rate, more data is collected over an interval of the mechanism's revolution. Therefore, the data must be retained for more estimator cycles. The opposite is true for the large sample interval. There are fewer estimation cycles for the given arc of revolution, therefore, the data must be "forgotten" faster. The end result makes the selection of ρ a time consuming task. There are methods to relieve this task. One method, variable forgetting, is discussed in Section 5.2.

The ARMA instability is caused from negative estimates for the perturbation feedback loop gain, CGAIN. Observing the velocity and control gain plots (Fig. 7.3, 7.13), the input function is followed until the step increase. At this point negative control gain values produce counter productive correction torque (Fig. 7.14). The ARMA control has readjusted itself at 1.3 seconds to produce the proper

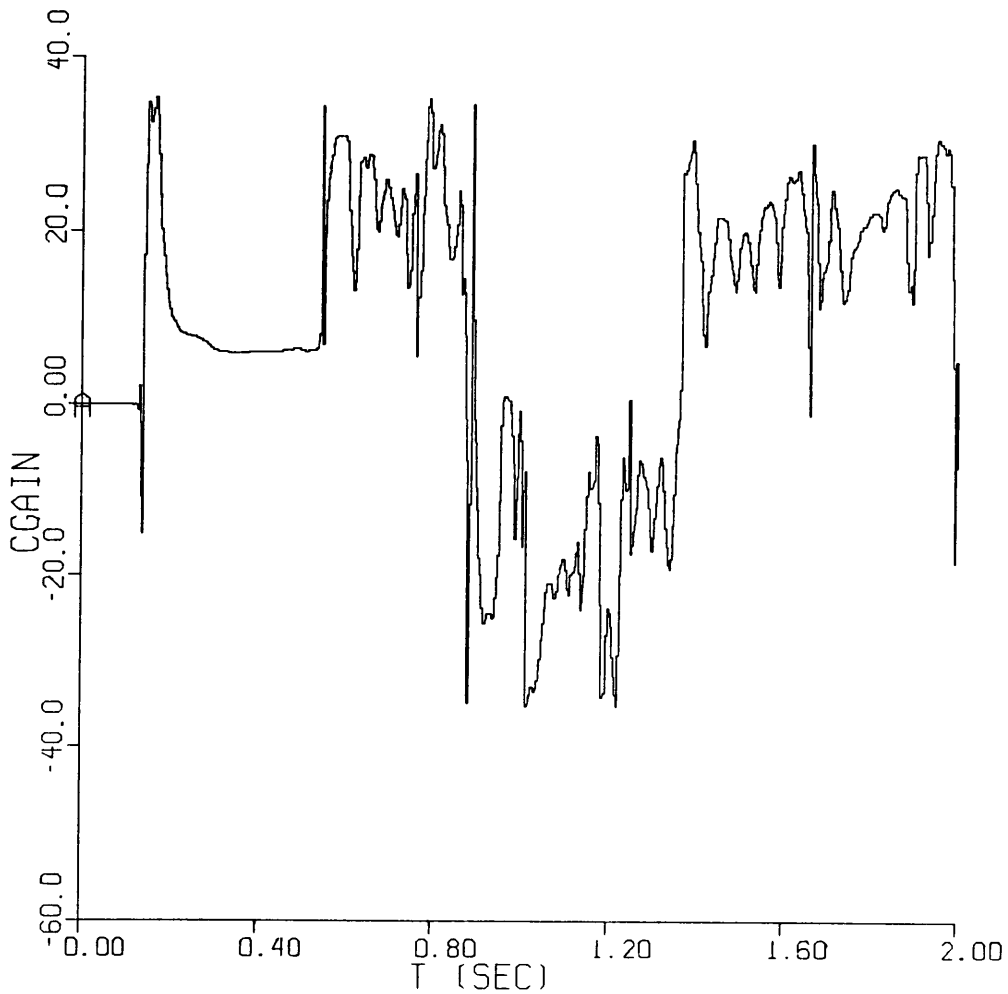


Fig. 7.13. Perturbation ARMA Control Gain: TSAMP = 4 ms, EPSL = .0002, APWR = 183, ROE = .92, ARMA Torque = 50, Step

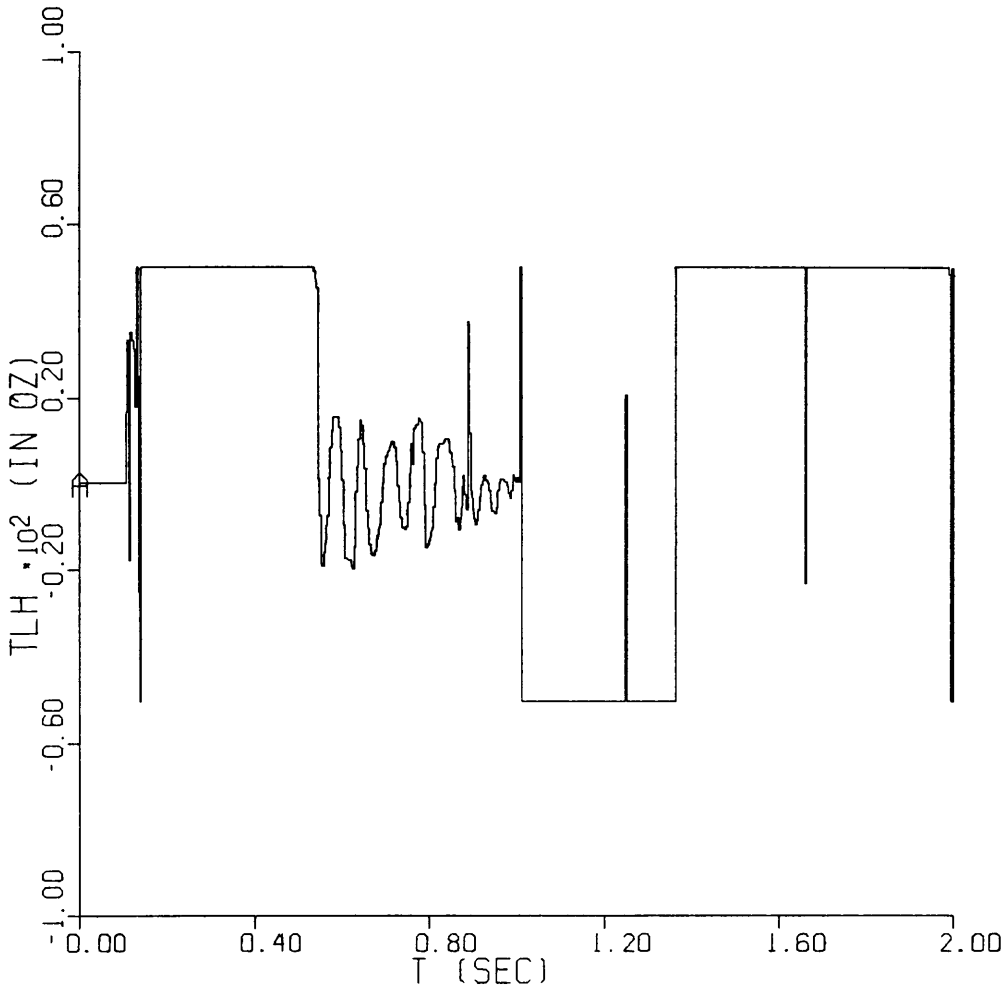


Fig. 7.14. Perturbation ARMA Torque: TSAMP = 4 ms, EPSL = .0002, APWR = 183, ROE = .92, ARMA Torque = 50, Step

corrective action. The velocity and control gain plots (Fig. 7.5, 7.15) show the same effects. The negative control gain estimates produce glitches in the velocity profile at .8 and 1.6 seconds.

Although perturbation four-bar control works well, it would be desirable to have full velocity feedback to the computed-torque model and to be able to use the control over a broad range of mechanisms. The nonlinear control presented in Section 8 is a possible scheme to implement these ideas.

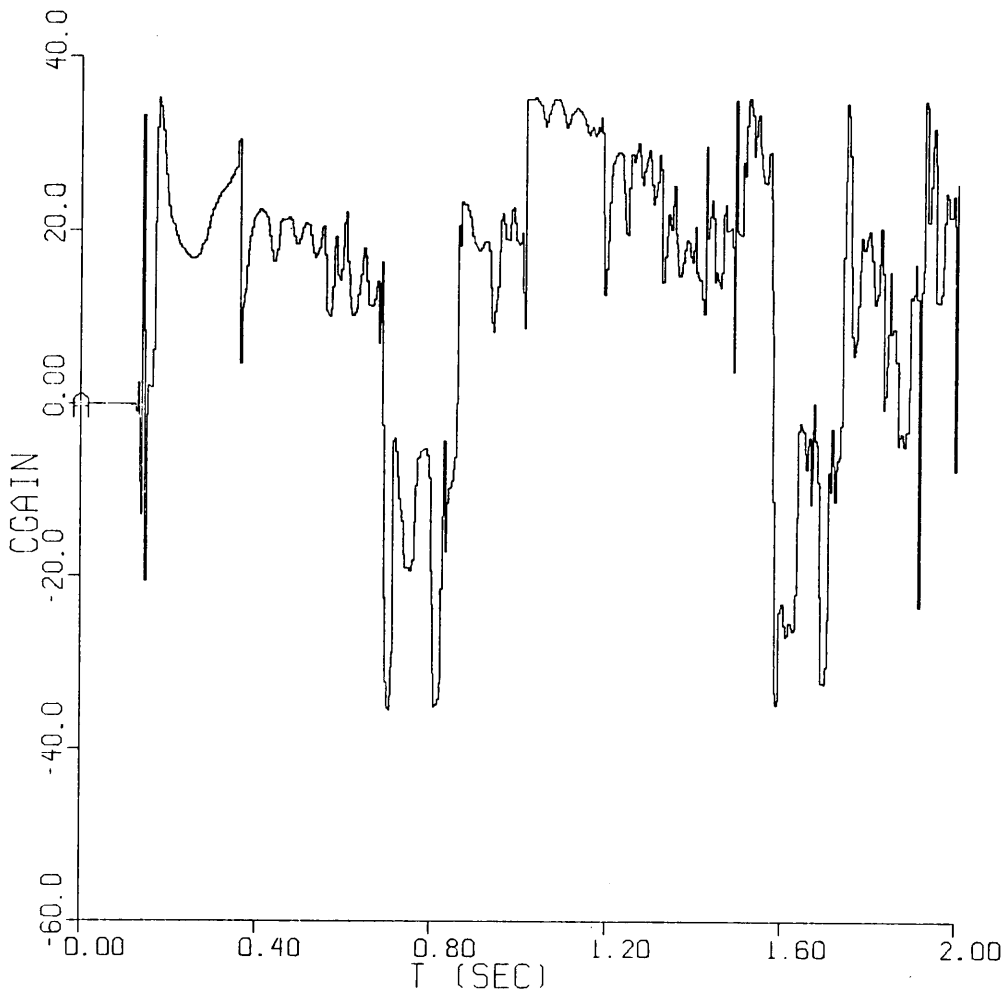


Fig. 7.15. Perturbation ARMA Control Gain: TSAMP = 4 ms, EPSL = .0002, APWR = 396, ROE = .92, ARMA Torque = 100, Step

8.0 Nonlinear Adaptive Control

The nonlinear four-bar control is an attempt to improve the perturbation four-bar control. The objectives are to enhance the transient response, and avoid the Perturbation control mechanism dependency while maintaining the steady-state velocity performance. This is accomplished by incorporating full velocity feedback into the "computed-torque" model where the "computed-torque" model can be adapted on-line to best fit the driven mechanism.

The nonlinear adaptive control scheme is based on the optimal step-ahead control and least-squares estimator. Here, the inverse/prediction model is interpreted as the perturbation control "computed-torque" model and is a linear combination of mechanism position dependent basis functions. This redefines the "slowly time-varying" parameters as the basis function coefficients. Since the mechanism presents the same equivalent inertia for each revolution, the basis coefficients should tend to be constant. This allows the velocity feedback control loop and the parameter estimation to be calculated as separate problems. Therefore, processor time-sharing or multiple processors can be used to increase the nonlinear control velocity loop sample rate.

The final form of the nonlinear control uses both the current velocity feedback error and the known desired velocity function to calculate the step-ahead control torque. The control formulation follows the development of Depkovich [24] and Goodwin [18]. The nonlinear control is simulated for the formulation using the exact four-bar nonlinear dynamic model and for the separated velocity control loop/estimation calculations using a general dynamic model. The exact

nonlinear four-bar control behaves as desired. However, the general split-calculation case has some performance problems related to the general "computed-torque" model and the parameter estimation. The nonlinear four-bar control is derived in section 8.1.

8.1 Nonlinear Control Derivation

The nonlinear four-bar control derivation follows a similar development as the single link position controller [24] and the bilinear adaptive control [18]. The departures are the use of the mechanism position in the prediction model and the optimal control formulation using a two step-ahead approach. The objectives are to "learn" about the mechanism dynamics over a revolution and provide a means to control using higher order velocity error derivatives.

First consider the inverse model required for least-squares parameter estimation. Since the inverse model will be used to derive the optimal control prediction model, it should model the nonlinear dynamics as closely as possible. Therefore, the general nonlinear four-bar equation of motion

$$T^* = I^* \alpha + \frac{dI^*}{d\phi} \omega^2 + T^*_G + D\,\omega \qquad (8.1\text{-}1)$$

is used as the inverse model. The applied torque, T^*, and the velocity, ω, are easily measured, and the value of I^* is mechanism position dependent. However, the acceleration, α, must be filtered using the velocity sequence and a central difference derivative.

From the discussion, Section 2.3, the functions for the equivalent

inertia I^*, and the gravity torque T^*_G can be represented by a general model. The exact models would be the influence coefficient models equations 3.1-3, 3.1-5. However, these are mechanism dependent. To relieve this depenancy, introduce the general models

$$\tilde{I}^*(\phi) = a_1 + a_2 \cos\phi + a_3 \sin\phi + a_4 \cos 2\phi + a_5 \cos 2\phi \qquad (8.1\text{-}2)$$

and

$$\tilde{T}_G(\phi) = a_6 \sin\phi + a_7 \cos\phi \, . \qquad (8.1\text{-}3)$$

Equation 8.1-1 can then be used to represent a general rotating device where the inertia and torque parameter vary as functions of the input position sine and cosine. The model choice is due to the (2ϕ) sinusoidal-like variations of the equivalent inertia and torques (Fig. 2.6, 2.8). The least-square parameter estimator adjusts the parameters a_1 - a_7 and the friction coefficient D to best fit the controlled mechanism.

The other objectives are to include velocity feedback and use the input function derivative information to aid input tracking. The combined effect is to calculate the torque applied at time (k + 1) using the current, (k), velocity and the input function at (k + 2). This yields a two step-ahead performance index, and contains the required sample interval delay for the control calculation.

The optimal step-ahead prediction model is derived from the inverse model 3.1-1. Considering computed torque from the perturbation control, the required torque at the (k + 1) sample instant is

$$T^*(k+1) = I^*(k+1)\left(\frac{\omega(k+2)-\omega(k)}{2\delta}\right) + \frac{dI^*(k+1)}{d\phi}\,\omega^2(k+1) + D\,\omega(k+1) + T^*_G(k+1) \tag{8.1-2}$$

using

$$\alpha(k+1) = \frac{\omega(k+2) - \omega(k)}{2\delta} .$$

Solving 8.1-2 yields the prediction model.

$$\omega^p(k+2) = \frac{2\delta}{I^*(k+1)}\, x$$

$$\left[T^*(k+1) + \frac{I^*(k+1)}{2\delta}\,\omega(k) - \frac{dI^*(k+1)}{d\phi}\,\omega^2(k+1) - D\omega(k+1) - T^*_G(k+1)\right]. \tag{8.1-3}$$

The performance index (5.2-1) for the (k + 2) sample instant

$$J = E\left[\frac{1}{2}\left(\omega^d(k+2) - \omega^p(k+2)\right)^2 + \frac{1}{2}\,\lambda\, u^2(k+1)\right]$$

is solved for the optimal control

$$T^*(k+1) = \frac{\frac{2\delta}{I^*(k+1)}}{\left(\frac{2\delta}{I^*(k+1)}\right)^2 + \lambda}\, x$$

$$\left[\omega^d(k+2) - \left(\omega(k)\,\frac{I^*(k+1)}{2\delta} - \frac{dI^*(k+1)}{d\phi}\,\omega^2(k+1) - D\omega(k+1) - T^*_G\right)\frac{2\delta}{I^*(k+1)}\right]. \tag{8.1-4}$$

The control provides full velocity feedback and uses the first

derivative of the input function.

As previously discussed, the value of I^* and T^*_G are described using the mechanism position so are not time-varying parameters in the adaptive control sense. An interpretation would be to consider the control 8.1-4 as a gain schedule control where the gain is scheduled according to the mechanism position. Therefore, the control loop can be calculated without updating the estimates for the general rotational models $\tilde{I}^*$, $\tilde{T}^*_G$. Adaptation then arises from the time-varying parameter adjustment in the rotational models.

The block diagram (Fig. 8.1) illustrates the nonlinear control operation. The mechanism position (ϕ_i) is used to "schedule" the forward loop control gain using the estimated parameter of the nonlinear model and least-squares estimator. The velocity is fedback as an acceleration modifier (state-variable feedback) to the feed forward gain and used to estimate the rotational model coefficients.

To benchmark the control performance, the nonlinear control was simulated using the influence coefficient inertia and gravity equivalents and the exact frictional drag value D. This case is used as the limiting case for the general rotational model.

3.2 Nonlinear Control Algorithm

The Nonlinear control algorithm was programmed using the optimal control law 8.1-4 and the rotational models, equation 8.1-2, 8.1-3. The least-square parameter estimator with a constant exponential data weight is used to estimate the inverse model parameters. The ACSL simulation program Nonlinear appears in Appendix B.

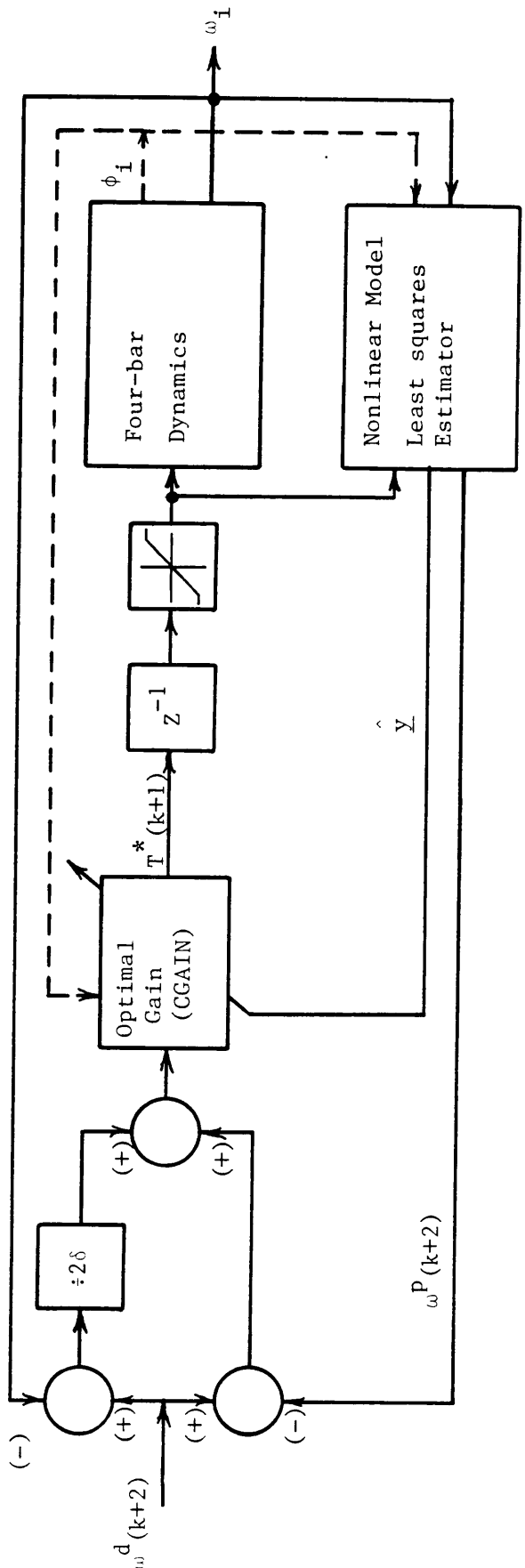


Fig. 8.1 Nonlinear Adaptive Control Block Diagram

The least-squares estimator requires the estimated parameters $\hat{\underline{y}}$ to appear as linear combinations of the basis $\underline{v}$. For the nonlinear inverse model, equations 8.1-1, 8.1-2, 8.1-3, are written in this form after resolving the position derivative of $\tilde{I}^*$ as

$$T^*(k-1) = \underline{v}\,\hat{y} \tag{8.2-1}$$

where

$$\underline{v}^T = \begin{bmatrix} \alpha \\ \alpha\cos(\phi) - \omega^2\sin(\phi) \\ \alpha\sin(\phi) + \omega^2\cos(\phi) \\ \alpha\cos(2\phi) - \omega^2 2\sin(2\phi) \\ \alpha\sin(2\phi) + \omega^2 2\cos(2\phi) \\ \sin(\phi) \\ \cos(\phi) \\ \omega \end{bmatrix} \quad (k-1) \tag{8.2-2}$$

and

$$\hat{\underline{y}} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ D \end{bmatrix} \tag{8.2-3}$$

evaluated at time (k - 1). Here, the acceleration is calculated using

the central difference

$$\alpha(k - 1) = \frac{\omega(k) - \omega(k - 2)}{2\delta} , \qquad (8.2\text{-}4)$$

and the estimation is delayed until the actual instrument measurements for velocity and torque are available. The estimated parameters $\hat{\underline{y}}$ are used in equations 8.1-4 to implement the adaptive control.

The control algorithm is simulated for a time-sharing calculation technique for the velocity control loop and the parameter estimator. The estimator is computed as a background task to the control loop. In this mode, the estimator operates by grabbing a snapshot, three consecutive samples of the instrument values, and calculating the parameter estimator for that particular position. The parameter updates are then passed to the velocity control loop which operates at a constant 4 ms sample interval.

The velocity control loop acts as a gain schedule control. This relies on the ability for the general model to converge to a "good" representation of the actual mechanism equivalent inertia. In this event, the estimation is no longer required and could be stopped.

The velocity control loop calculation requires approximately 2 ms. Given the 4 ms sample interval, this leaves 2 ms to execute part of the least-square estimator in the background. The eight parameter estimator will require approximately 16 ms to calculate and will get 2 ms of execution time over a sample. Therefore, the simulation was written to portray the 4 ms sample interval with the new parameter updates occuring every tenth pass through the control loop or 40 ms.

The position dependent functions were "tabulated" every 3^{o} of arc to simulate the look-up table required to achieve the real time calculation speeds. The overall performance and the estimation properties are discussed in section 8.3.

8.3 Simulation Studies

The simulation studies were made using the exact four-bar model and the time-sharing computation scheme. The velocity control loop has a simulated 4 ms sample interval with a 40 ms parameter update interval. Simulations were made for the velocity step and ramp input functions using both the motor torque limits. However, the control torque weight λ was not changed. Therefore, the different torque limits had no observable effect. The main differences observed is for two different values of the least-square parameter estimation forgetting factor, ρ. This change caused some differences in the inertia and gravitational model parameter estimates. The overall performance is not good. The velocity fluctuations are greater than 10 percent at 50 and 75 rad/sec velocities. However, the step and ramp tracking is better showing rise times of .1 seconds but with some steady-state offsets. The simulation output presented represent the overall nonlinear control study results.

The exact case uses the exact influence coefficient inertia and gravity models and the exact values for the link parameters. This case is used to verify the control derivation and establish a reference to compare the general inertia and gravity models (equations 8.1-2,3). This case was simulated using the 4 ms sample interval and a control

weight $\lambda = .0001$. Since there is no parameter estimation, the exact control is interpreted as a gain schedule control where the gain varies as a function of the mechanism position. This case is the limiting case for the best fit general inertia and gravity models.

The velocity plot (Fig. 8.2) shows the exact case control performance. The step response at start-up is nearly a dead-beat response and the velocity tracks the input function with no average velocity overshoots. The velocity performance to maintain the kinematic input conditions, constant speed, is 4% fluctuation at 50 rad/sec and 6% fluctuation at 75 rad/sec. This performance could be improved. The torque plot (Fig. 8.3) shows the maximum (396 in-oz, 27.9×10^3 cm-dyn) torque is not used. This can be increased with a smaller control torque weight λ which would improve the velocity performance.

The forward control gain is the expected variation the generalized inertia models should produce if the models and the parameter estimates are good. The recognizable features are (Fig. 8.4) the sinusoidal fluctuations between 48.9 and 49.9 at twice the input rotational frequency.

The velocity plots using the general inertia and gravity models (Fig. 8.5,6,7) show the poor velocity fluctuation performance, with (Fig. 8.7) showing some mean velocity offset. In general, the tracking performance is much better showing a rise time of .1 seconds (Fig. 8.5) and good mean velocity tracking (Fig. 8.6). The effect of changing the value of the forgetting factor, ρ, typically does not effect the velocity performance except for the required estimation time at start up. However, Fig. 8.7 does show a mean offset at the 75 rad/sec

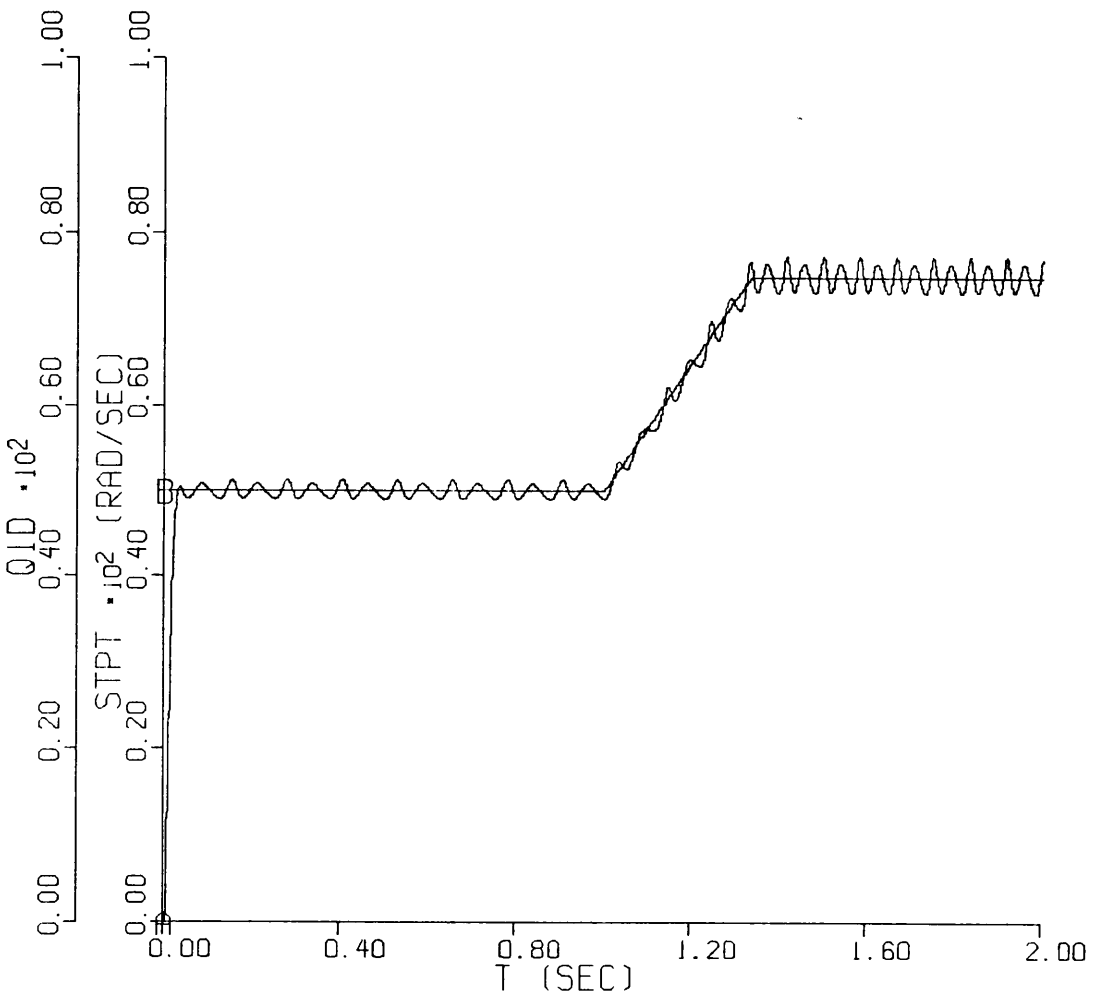


Fig. 8.2. Velocity Plot Nonlinear Exact Case: EPSL = .0001

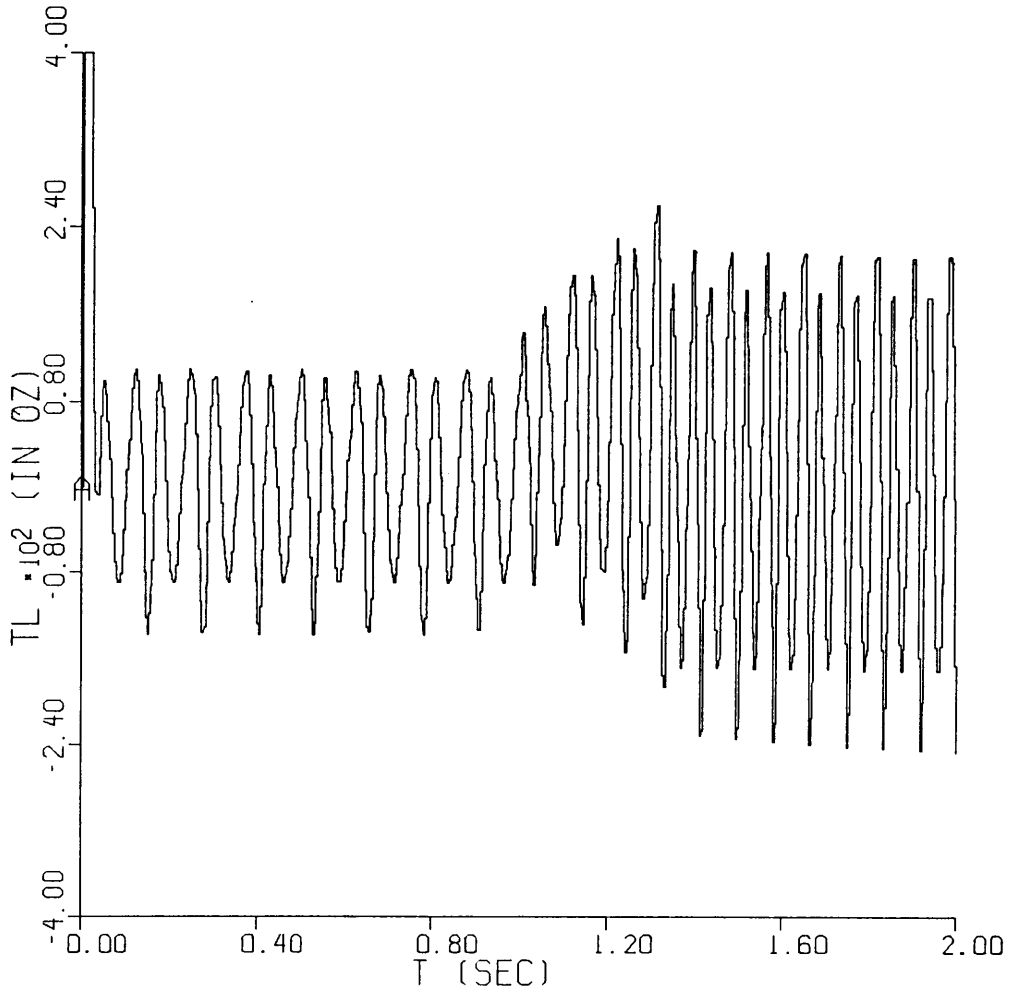


Fig. 8.3. Torque Plot Nonlinear Exact Case: EPSL = .0001

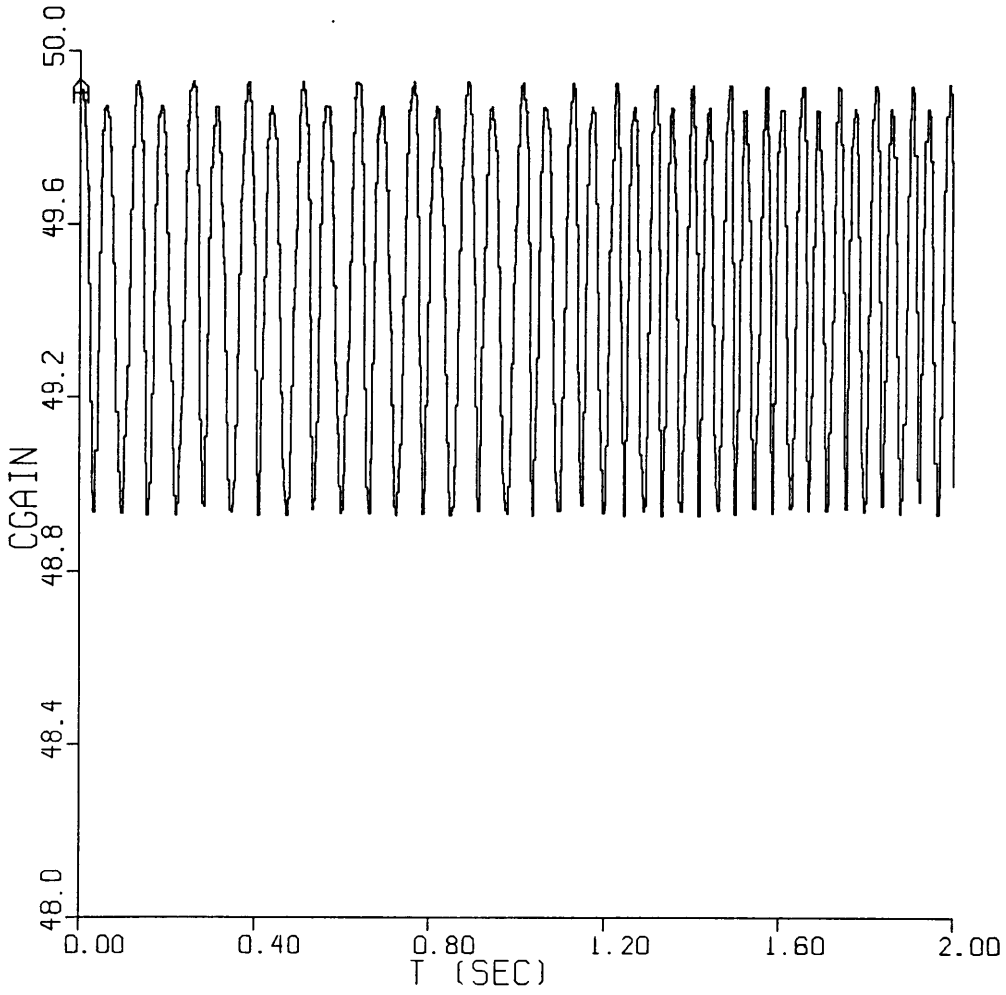


Fig. 8.4. Control Gain Plot Exact Case: EPSL = .0001

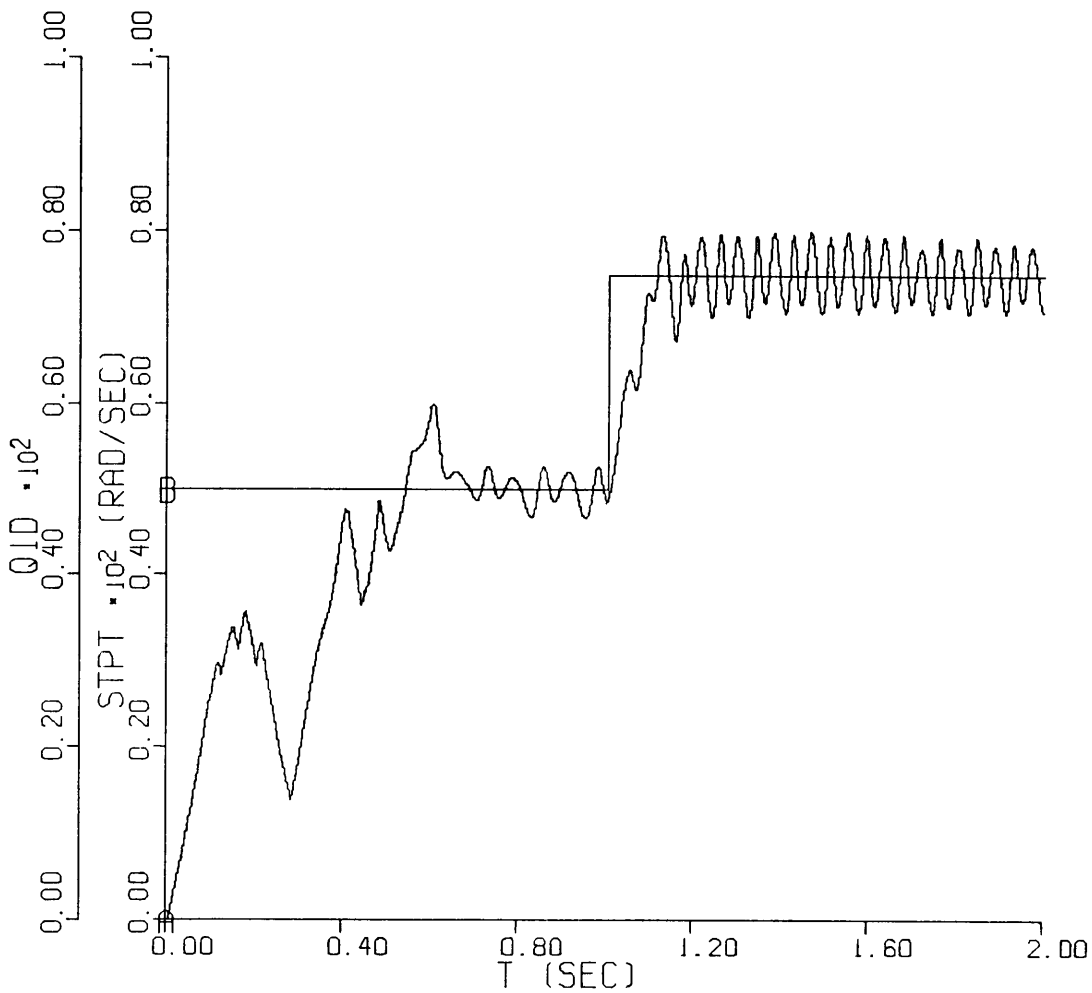


Fig. 8.5. Nonlinear Velocity: EPSL = .0001, APWR = 183, ROE = .9, Step

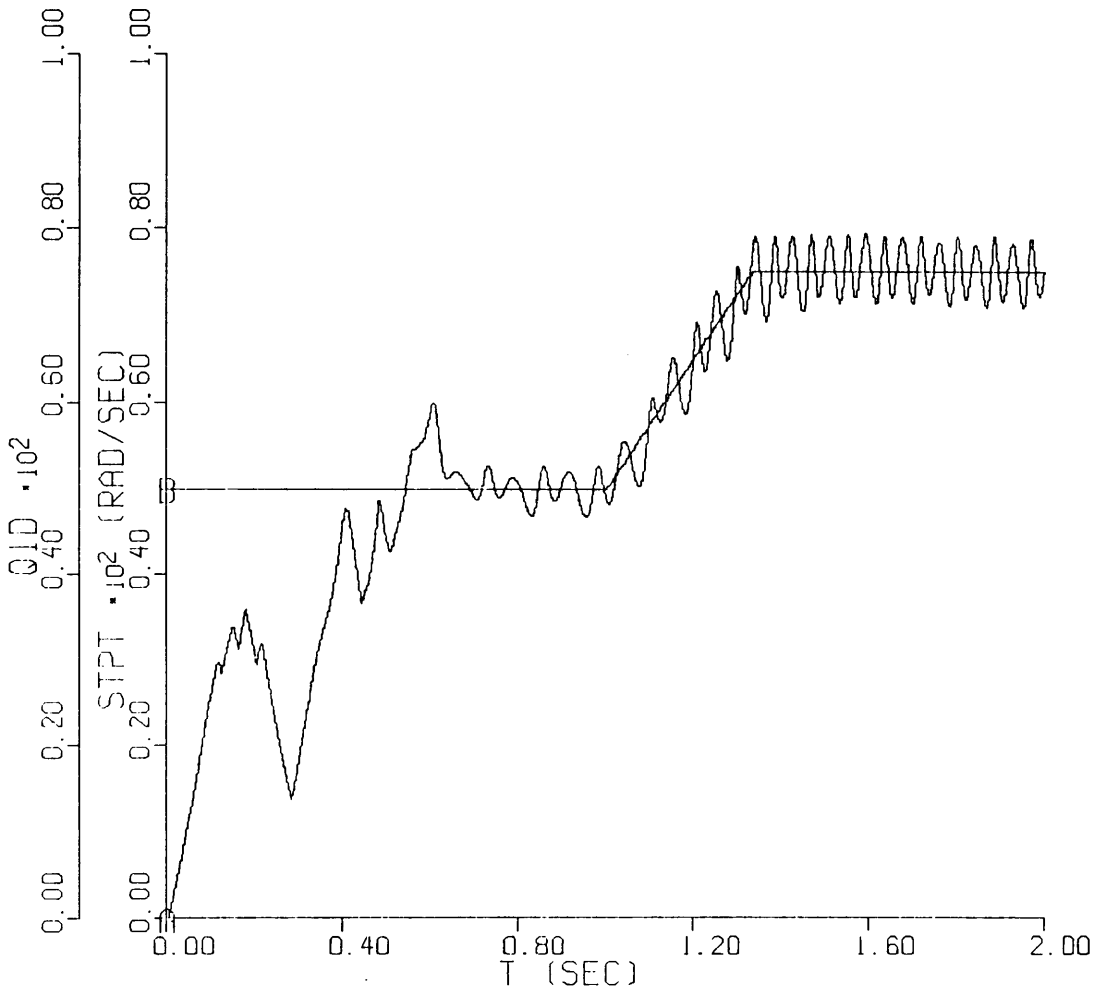


Fig. 8.6. Nonlinear Velocity: EPSL = .0001, APWR = 183, ROE = .9, Ramp

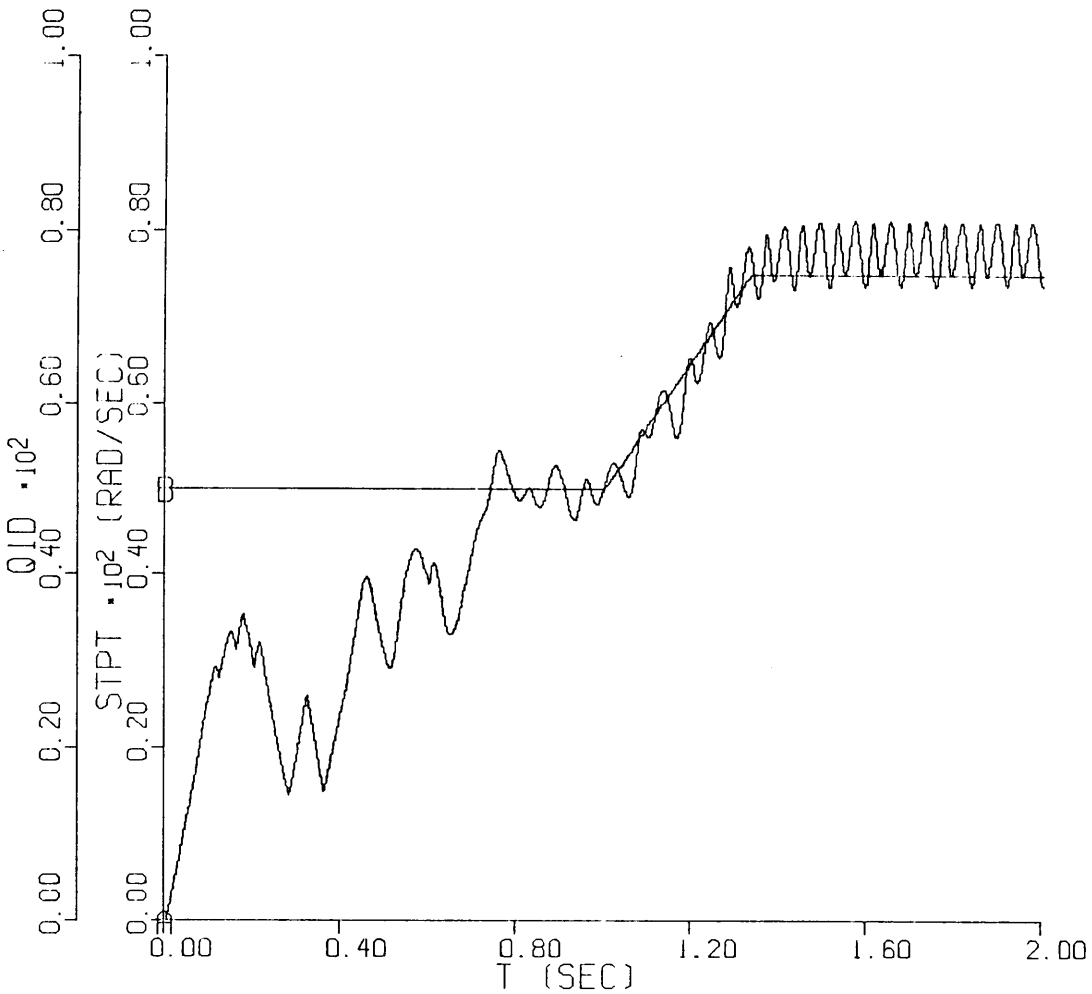


Fig. 8.7. Nonlinear Velocity: EPSL = .0001, APWR = 133, ROE = .95, Ramp

setpoint.

The control gain plots (Fig. 8.8,9,10) for the simulation cases do not show the desired fluctuations. The control gain, for a forgetting factor ROE = .9 (Fig. 8.10) does show some fluctuation. However, the fluctuations are too large in magnitude and at the wrong frequency. This fluctuation is due to the parameter estimation and not the position functions as anticipated.

The plots for the estimated parameters show the inertia model is not very good. All of the basis function coefficients A2 - A5 for the step input and the ramp input (Fig. 8.12) are estimated as zero with the offset coefficient A1 term estimated as a constant value. In the ideal situation, the A1 parameter should be the average mechanism inertia. Unfortunately, the average inertia is 0.7 (Fig. 2.6) and A1 is estimated at approximately 0.25. This indicates the basis functions do not fit the mechanism's inertia variations. As an overall observation, the parameters are estimated for a constant velocity input function with some shifts for a velocity setpoint change. In general, the parameter estimation should receive "rich data" from the centrifugal or other disturbances on the mechanism. This allows adaptation without dither signals.

Again for all cases, step (Fig. 8.13) and ramp (Fig. 8.14) shown, the gravity parameters A6, A7 fluctuate, however do settle down to a non-zero value. These parameters where expected to estimate the gravity torques (Fig. 2.8). However, the steady state values are large with respect to the actual gravity torque. The large values indicate these functions were more closely correlated with the position torque data,

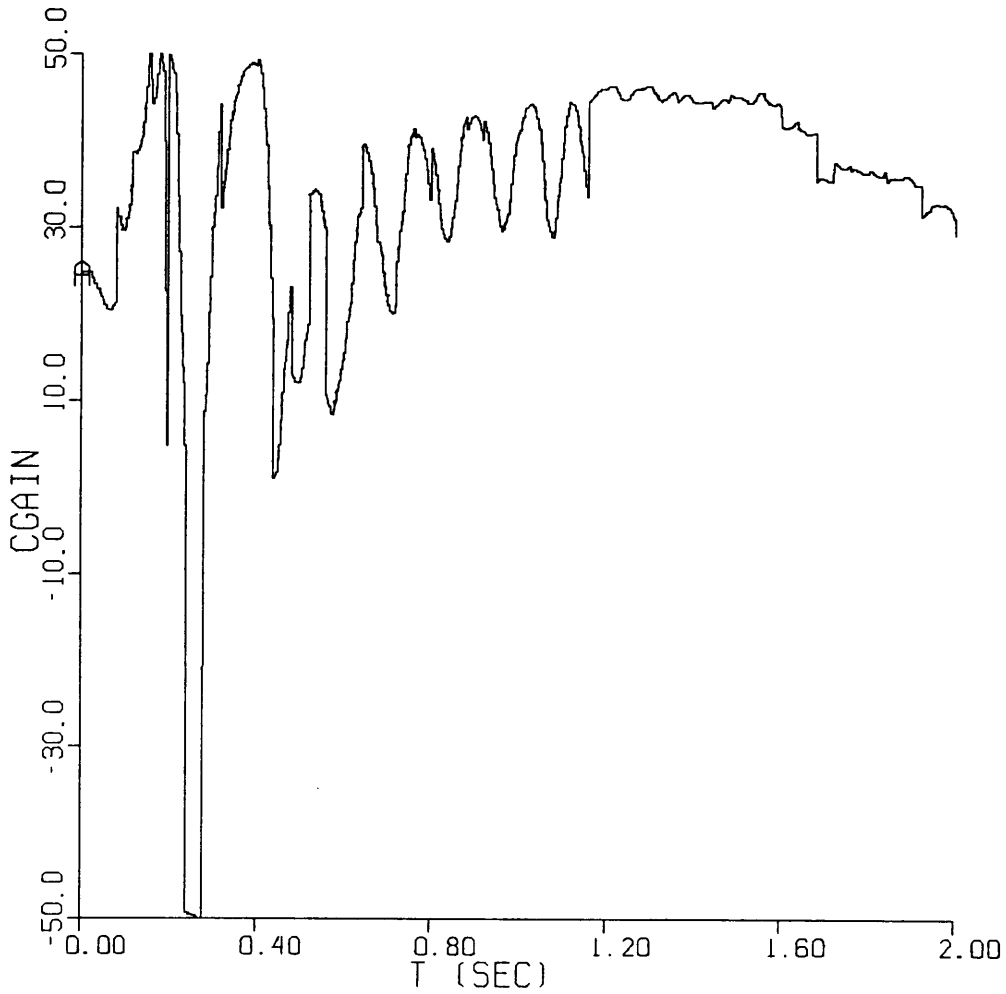


Fig. 8.8. Nonlinear Control Gain: EPSL = .0001, APWR = 183, ROE = .9, Step

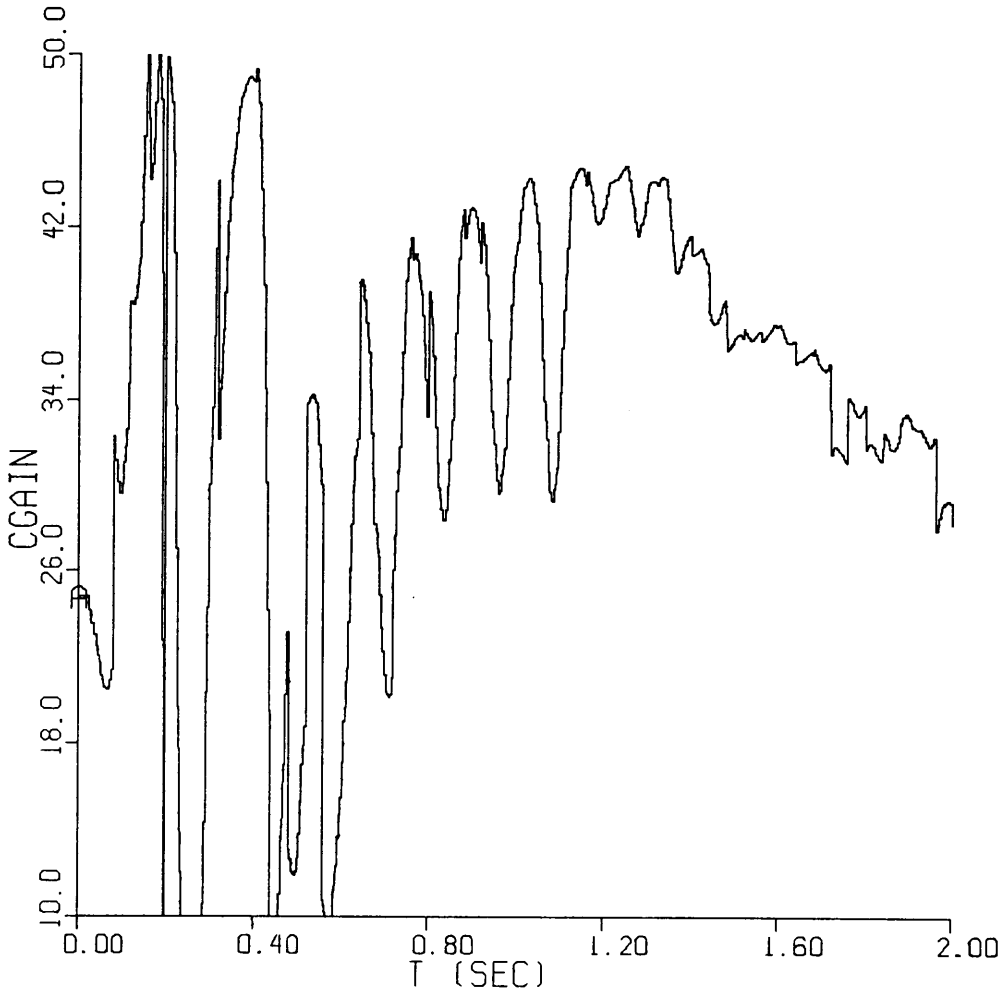


Fig. 8.9. Nonlinear Control Gain: EPSL = .0001, APWR = 183, ROE = .9, Ramp

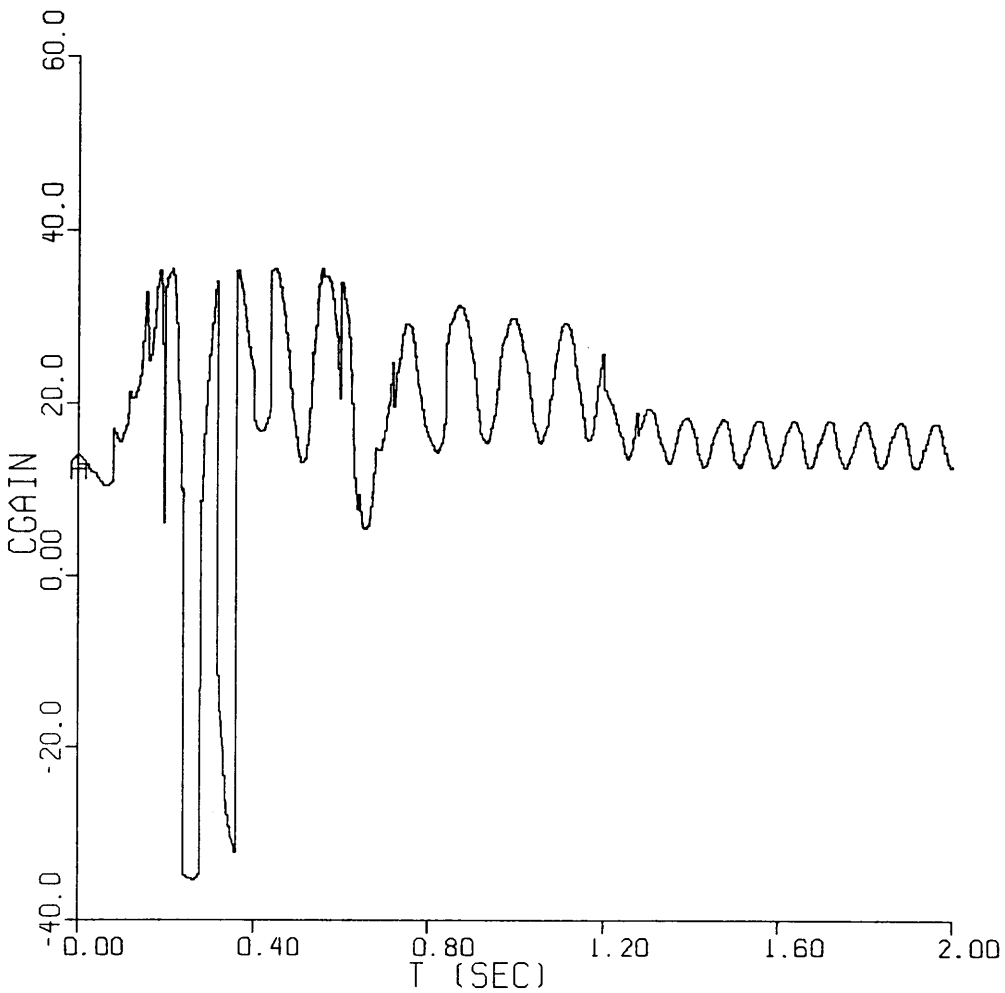


Fig. 8.10. Nonlinear Control Gain: EPSL = .0001, APWR = 183, ROE = .95, Ramp

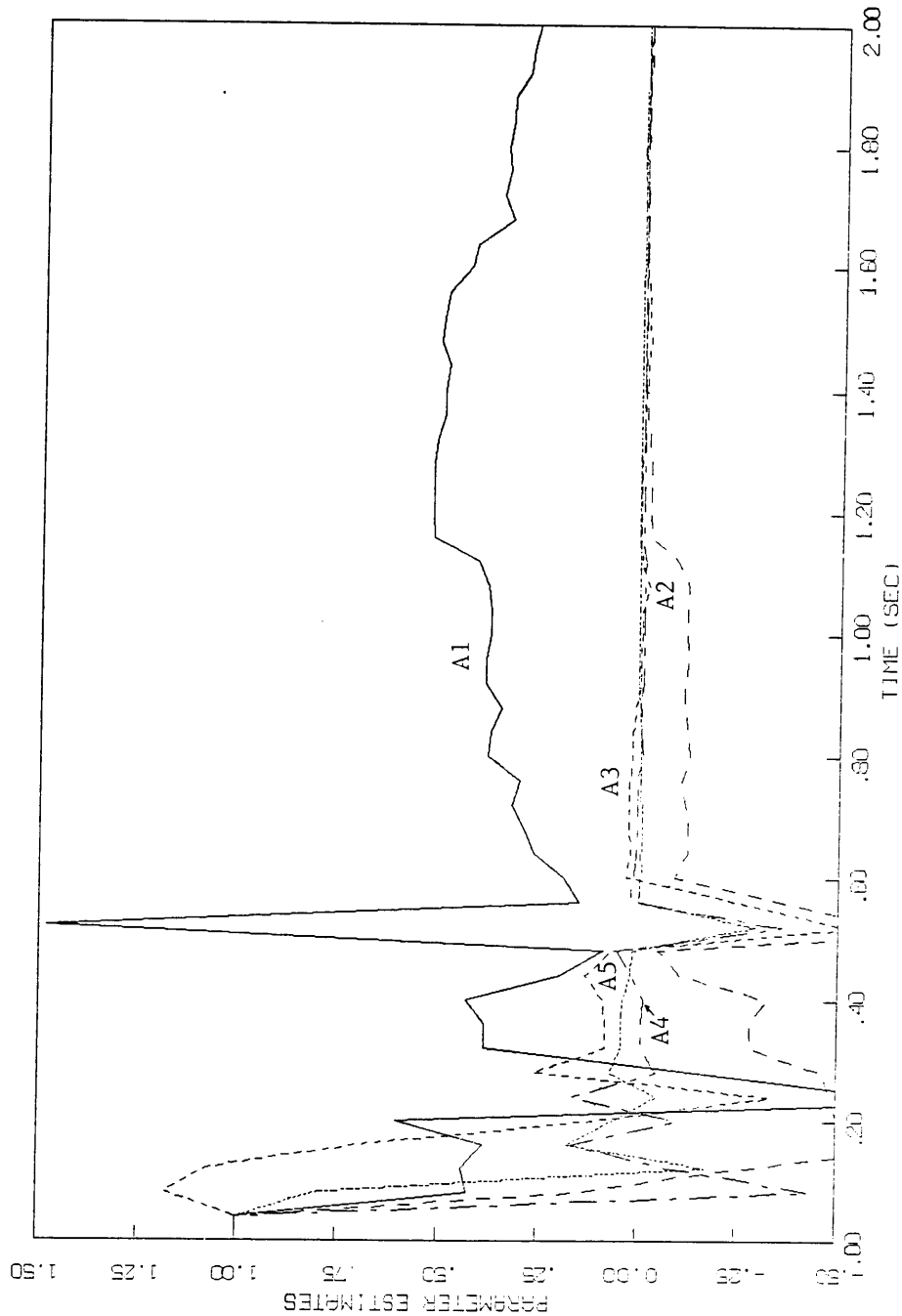


Fig. 8.11. Nonlinear Inertia Coefficients A1-A5: EPSL = .0001, APWR = 183, ROE = .9, Step

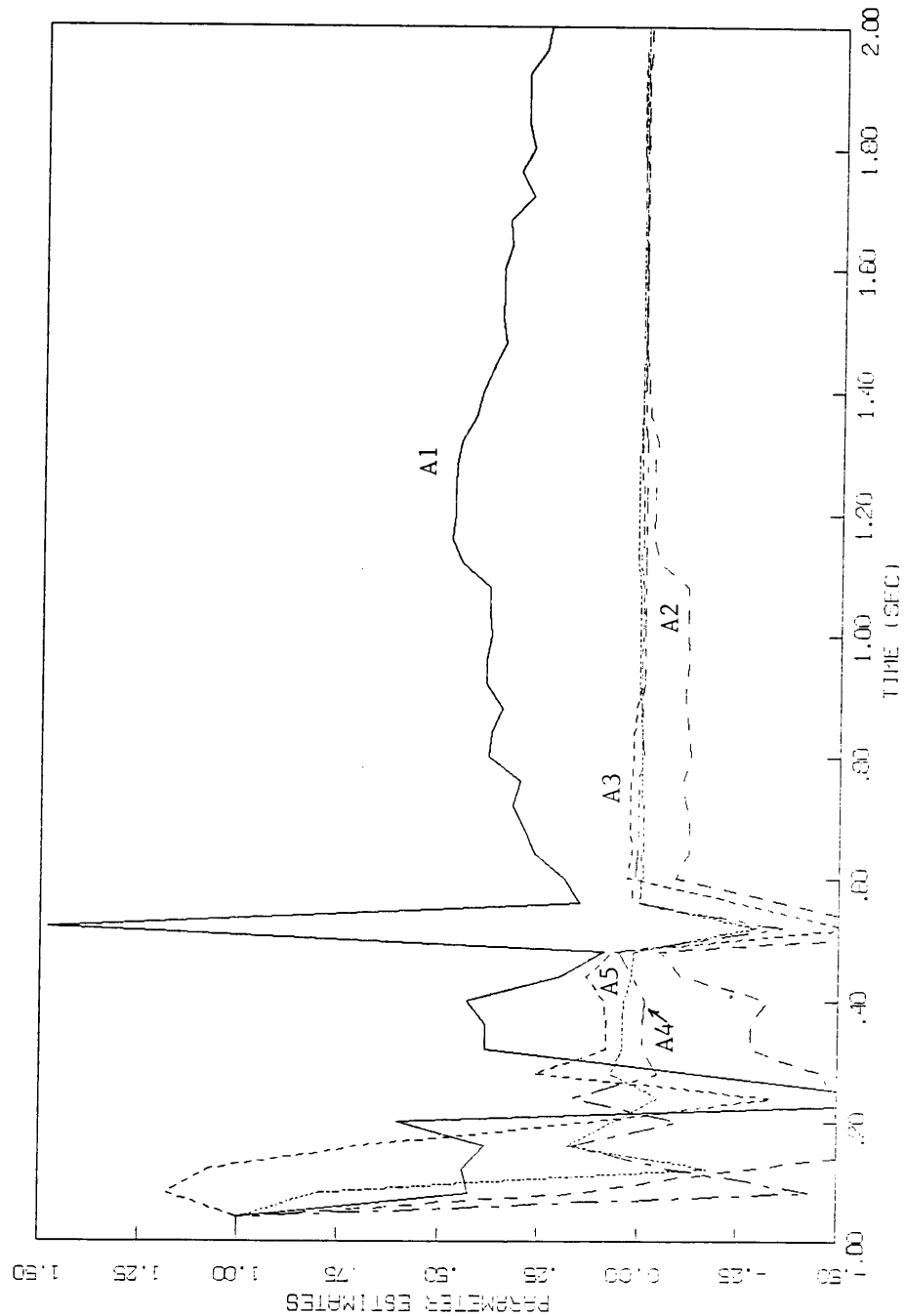


Fig. 8.12. Nonlinear Inertia Coefficients A1-A5: EPSL = .0001, APWR = 183, ROE = .9, Ramp

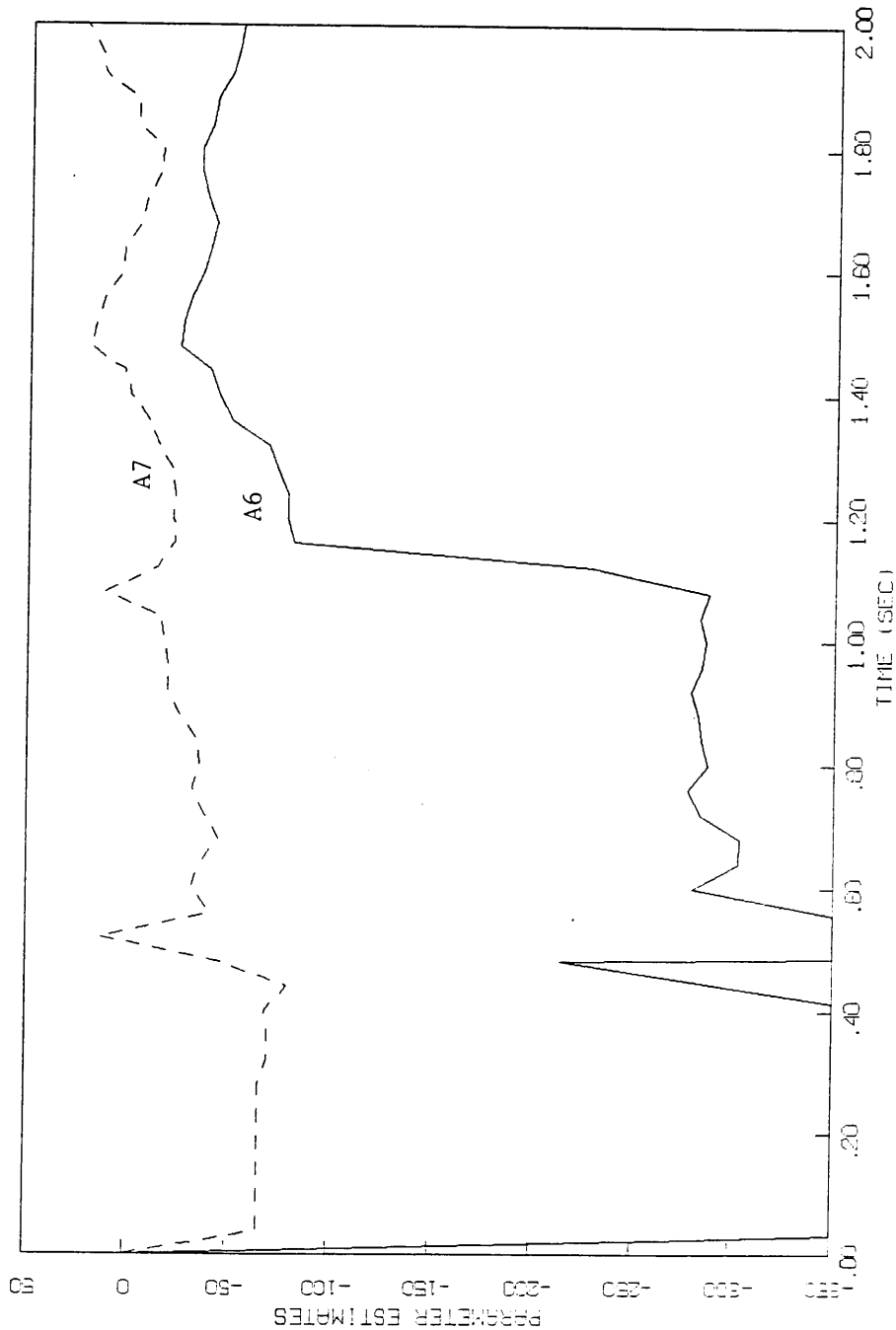


Fig. 8.13. Nonlinear Gravity Coefficients A6-A7: EPSL = .0001, APWR = 183, ROE = .9, Step

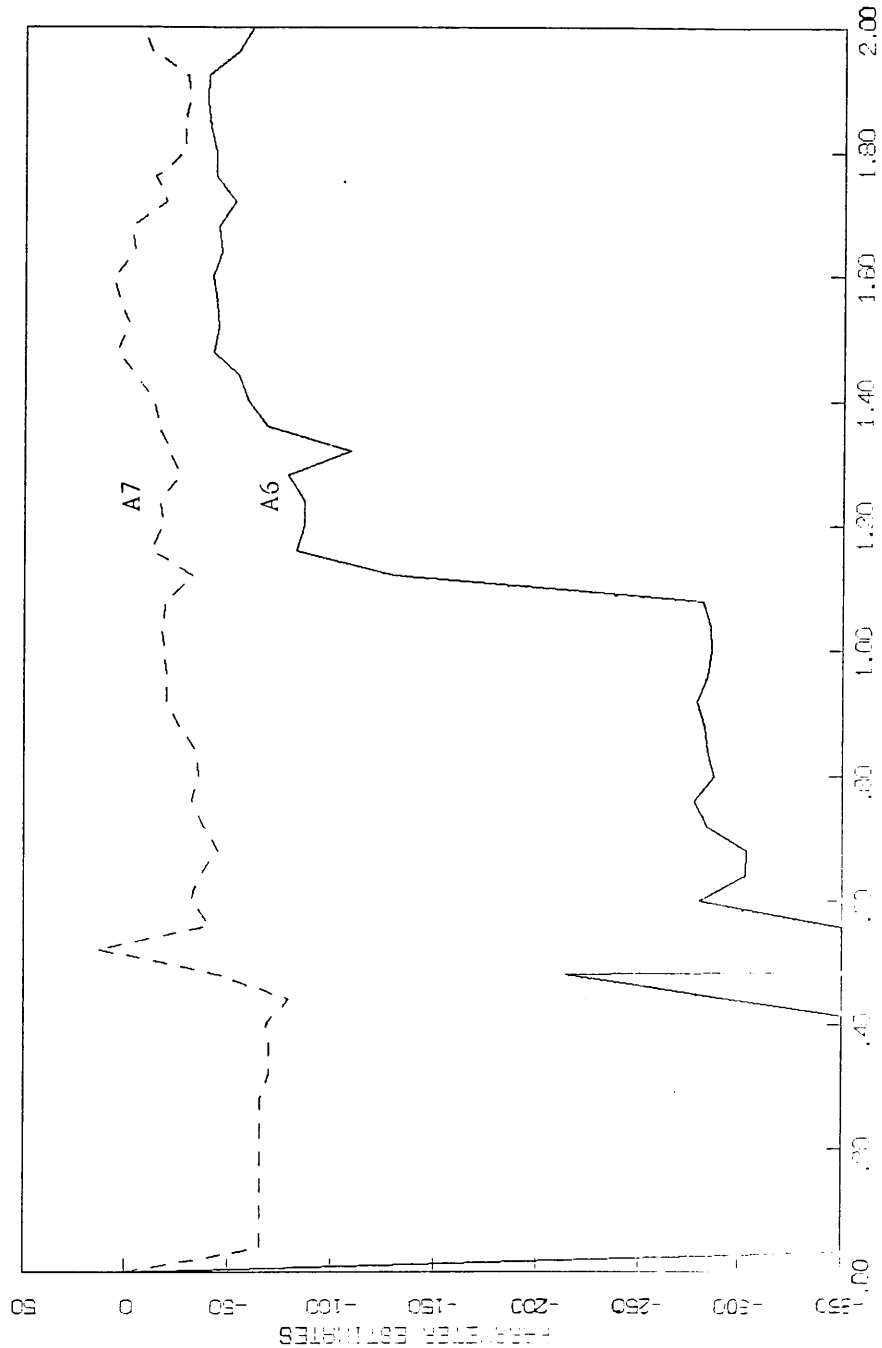


Fig. 8.14. Nonlinear Gravity Coefficients A6-A7: EPSL = .0001, APWR = 183, ROE = .9, Ramp

equation (8.2-1) and were "fitted" without regard to the velocity and acceleration. This would indicate the inverse model may be improved using position dependent functions to model the torque fluctuations without including the velocity and acceleration directly.

All cases (Fig 8.15,16) the friction coefficient, D, estimate is a small positive number. Ideally, D should be estimated as the frictional motor damping D_p. This is true by order of magnitude comparison.

There are several problems present for the general mechanism nonlinear control. These deal mostly with the choice for the inertia modelling functions. However, it should be noted that the exact case does produce an improved transient response over the perturbation control.

A reasonable choice for the nonlinear inertia models would be the influence coefficient functions. These are position dependant functions where the link masses and inertias appear as the linear coefficients. Thus, the inertia and gravitational models would be correct and the problem would then be to tune the estimator to estimate the proper link masses and inertias as the linear coefficients.

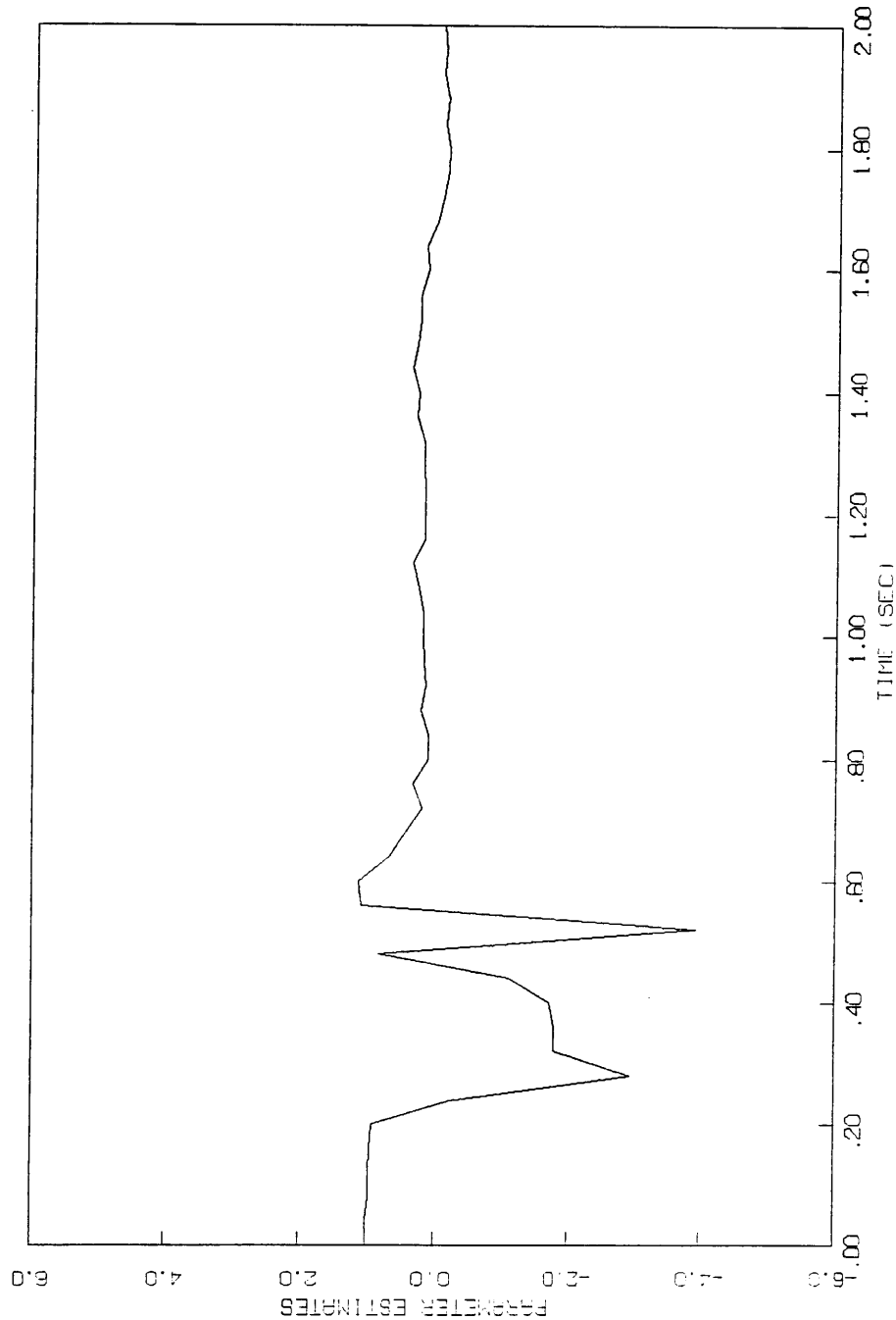


Fig. 8.15. Nonlinear Friction Coefficient D: EPSL = .0001, APWR = 183, ROE = .9, Step

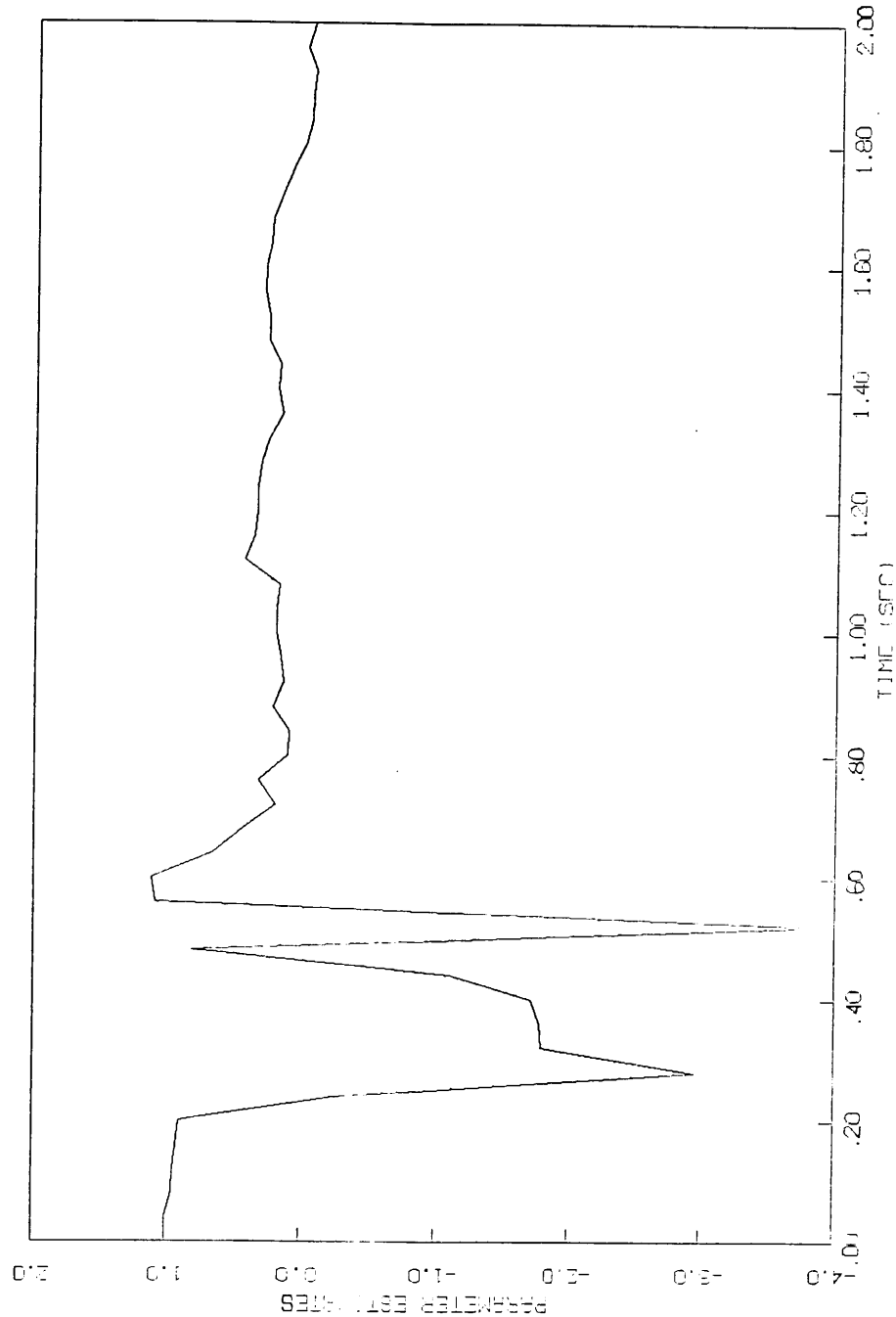


Fig. 8.16. Nonlinear Friction Coefficient D: EPSL = .0001, APWR = 183, ROE = .9, Ramp

9.0 Results

In order to maintain a constant velocity, the required motor torque must be available. From Fig. (3.9) the simulated four-bar requires a motor with a 390 in-oz (27.5×10^3 cm-dyn) torque capability. The results from the simulation studies (Table 9.1) show the velocity fluctuations were reduced for the greater available torque. Thus to maintain the desired linkage input condition the motor torque must be available.

From aspects of overall performance, the perturbation four-bar control is the best adaptive control to implement. However, it has to be redefined or reprogrammed to be effective with different mechanisms. Otherwise the lead-lag control tuned as a proportional control using the maximum feedback gain, is the most effective for a general mechanism. Since the lead-lag control system is essentially first order, the high gain will not cause any loop instability.

Since large motors are required to meet the performance requirements, the easiest solution is to use input link flywheels and the large motor. The flywheel will serve to reduce the fluctuation and the greater available torque will accelerate the larger inertia. However, the velocity transient response will always be faster for the lowest inertia and the maximum motor torque regardless of the controller.

Table 9.1 Summary of Performance Measurements

	183 in-oz			396 in-oz		
	%F 50	%F 75	rise	%F 50	%F 75	rise
LEAD-LAG	6%	8%	.1 sec	4%	6%	.04 sec
ARMA						
Tsamp = .002	8%	7%	.17 sec	-	-	-
Tsamp = .004	11%	9.4%	.1 sec	7%	8%	.04 sec
Tsamp = .012	14%	9%	.1 sec	-	10.6%	-
PERTURBATION						
Tsamp = 2 ms	< 2%	5%	.37 sec	< 2%	2%	-
Tsamp = 4 ms	3.4%	-	-	5%	4%	.17 sec
NONLINEAR						
Exact	4%	6%	-	4%	6%	-
Roe = .9	12%	12%	.1 sec	-	-	-
Roe = .95	9%	10%	-	-	-	-

- not obtained

10.0 Areas for Continuing Research

Two areas show some promise to improve the four-bar velocity control. These are hybrid systems, using both continuous time and discrete time components, and further development of the Nonlinear control, Section 8. Hybrid systems, one based on the pertubation control and one based on an adaptive velocity control loop are discussed. The next logical steps for the development of the Nonlinear control, Section 8, are presented.

The pertubation control, Section 7, readily lends itself to a hybrid system. The hybrid system would use the nominal computed-torque calculations as presented. However, the ARMA perturbation control would be replaced by a lead-lag or other continuous control. This sytem would still have the mechanism dependent properties inherent in the computed-torque model. The advantage is avoiding the stability and calculation problem of the ARMA adaptive control. This system would no longer be an adaptive system but the increase in overall performance would be a better design goal.

Another approach to a hybrid solution is a closed-loop continuous velocity control using discrete-time adaptive gains, (Fig. 10.1). This hybrid system would use an integral control on the continuous closed-loop velocity with an adjustable forward gain. The integral control would resolve any steady-state effects and a high forward loop gain to reduce the velocity fluctuations. However, the input transient response would be second order and with high forward gains lightly damped. To resolve these new problems, a digital adaptive system is used to adjust the forward loop gain to improve the transient response and to maintain

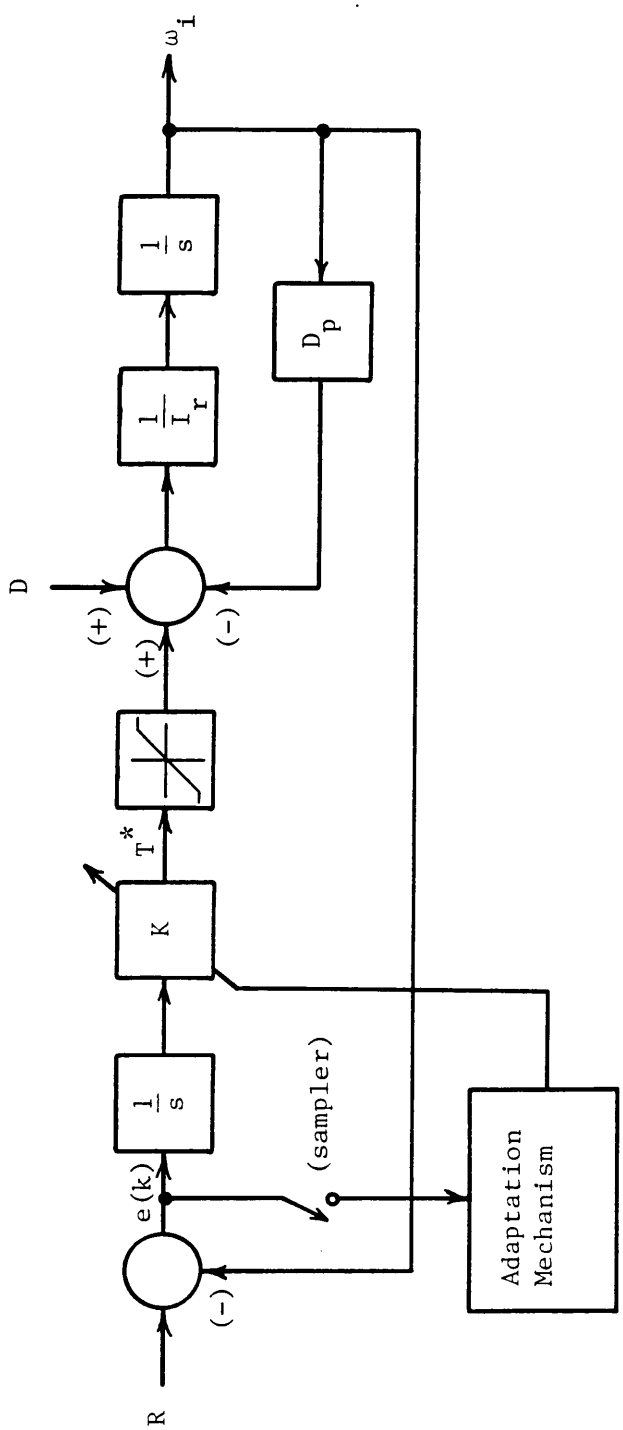


Fig. 10.1 Hybrid Adaptive Control Block Diagram

the velocity performance. This adaptive control may possibly be formulated using an ARMA model and the performance index

$$e(k) = \sum_{i=1}^{N} a_i \, e(k - i) + b_i \, K(k - i) + d$$

and

$$J = E\{e^2(k) + \lambda \, (K(k) - K(k - 1)^2\}$$

where e(k) is the feedback error on the block diagram.

Further research is required to fulfill the nonlinear adaptive control potential. This control is viewed as the best method studied for general mechanism control. The areas that should yield the most productive result are in the inertia model selection (equation 8.1-2). For a given linkage, the kinematic influence coefficient model is a logical choice. The estimated parameters are then the link inertia parameters, $\bar{I}_1$, $\bar{I}_2$, ..., $\bar{M}_2$, $\bar{M}_3$. Therefore, the same kinematic mechanism could be controlled for different links. This would allow the links to have "slowly-varying" link parameters.

Some preliminary work was performed to support this idea. The link inertial parameters were estimated using the mechanism applied torque, T^*, the dynamic state ϕ, ω, α, and the kinematic influence coefficient model (equation 3.2-2). The least-squares estimator was able to estimate the values for $\bar{M}_2$, $\bar{I}_2$ given the other four inertial parameters. However, efforts to estimate all six parameters were not successful. The estimation had a tendancy to estimate negative values for some of

the parameters. Since this is not possible, the estimator could be constrained to only estimate positive values. Therefore, estimate better parameters thus provide better control.

The most general inertial model (equation 8.1-2) would possibly require a means to adjust the frequency of the inertial variation with respect to the input position. If the inertial model basis function

$$V_1 = \sin \hat{\beta}\, \phi_i$$

is chosen, it is easily seen the estimated parameter, β, is nonlinear in V_1. Therefore, simple least-squares can not estimate the value $\hat{\beta}$. However, a nonlinear form for sequential least-squares can estimate $\hat{\beta}$ at the cost of calculation time. The design trade-offs are the required model complexity for performance, and the required time to calculate the parameter estimator. However, it should be noted that the parameter estimation is a separate calculation from the control law. In a long term operating environment, the parameters once estimated would not change. Therefore, the assumed several minute adaptation time would be a small loss compared to long term operation for a wide range of mechanisms.

The main objective for other research would center around making the actuator as "stiff" as possible. This could be accomplished using a motor control which will reduce, in a stochastic minimum variance sense, the allowed velocity feedback error. Therefore, the velocity fluctuations will be rejected by the driving motor without regard to the driven mechanism.

11.0 Conclusions and Recommendations

A dynamic model for the four-bar mechanism was developed using kinematic influence coefficients. This derivation yields one linear and one nonlinear state-space equations with algebraic coefficients. The algebraic coefficient equations are developed as functions of the input link position. The model is easily inverted to calculate the required torque for a given input link state. Test data show the model dynamic behavior is similar to the dynamic behavior of an experimental mechanism.

The four-bar dynamic model was used as the basis to model an open-loop mechanism velocity control system. This includes a reduced model for a D.C. servomotor and control amplifier as the system actuator. An incremental optical encoder was modeled as the feedback instrument. The encoder yields both position and velocity information with good overall accuracy. These models were used to develop a computer simulation for the open-loop system. This simulation was used to evaluate a standard linear controller and three adaptive control schemes.

The first adaptive control is an Autoregressive Moving Average self-tuner. This control would be difficult to implement in real time due to the controller calculation requirements. Other problems involving adaptation rates make this controller a poor candidate as a four-bar controller. Further study using this arrangement for direct four-bar control is not recommended.

The second adaptive control is the Perturbation control. This control uses a position-dependent computed-torque technique to reject the known torque disturbances. The technique operates by applying the

required correction torque based on the input position and desired mechanism velocity. The control studied is mechanism dependent, however it clearly demonstrates the advantage of using the known dynamics and the mechanism position to improve input velocity control. The steady-state velocity performance indicates this is the best control to guarantee the kinematic input condition for constant velocity. Using the proper configuration for the nominal torque model, this control is very effective in reducing the position-varying centrifugal acceleration disturbances. However, the lack of velocity feedback gives poor transient response to a discontinuous control input function. The perturbation control and hybrid control using this technique warrant further study.

The Nonlinear control, the third adapative control, has the potential to operate as a general adaptive control for any mechanism. This configuration incorporates velocity feedback to improve the velocity transient response, an improvement over the Perturbation control. The position "feedback" technique of the Perturbation control is incorporated to improve the constant velocity performance. For the Nonlinear control, the adaptation extends from the adjustment of the computed-torque model on-line to best fit the driven mechanism. In its true form, the control does not perform well. However, using the exact or a best approximation to the mechanism dynamics under control yields and improved constant velocity performance while maintaining the transient response. This control configuration then provides the best performance considering the velocity transient response the steady-state fluctuations. Further research to develop this control is

recommended. The algorithms are feasible to implement in real time and should provide a means to obtain the performance goals for constant velocity and transient response.

At present, the most effective control is the perturbation four-bar control. Otherwise the simplest control for the four-bar is a proportional control using an input flywheel and a large motor to achieve the required transient response.

References

1. Myklebust, A., "Computer Control of Non-Linear Mechanism Systems to Achieve Minimum Input Link Speed Fluctuations," Report on Phase I, Engineering and Industrial Experiment Station, College of Engineering, Florida Atlantic University, October 1983.

2. Myklebust, A., "Dynamic Response Analysis of Electric Motor-Mechanism System," Royal Norwegian Council for Scientific and Industrial Research, Institute for Machine Design, The Technical University of Norway, July 1975.

3. Myklebust, A., "Dynamic Response of an Electric Motor-Linkage System During Startup," Journal of Mechanical Design, Transactions of ASME, January 1981.

4. Benedict, C. E., and D. Tesar, "Dynamic Response Analysis of Quasi-Rigid Mechanical Systems Using Kinematic Influence Coefficients," Journal of Mechanisms, Vol. 6, 1971, pp. 383-403.

5. Hirschorn, Jeremy, Kinematics and Dynamics of Plane Mechanisms, McGraw-Hill, 1962.

6. PMI Switching Servo Amplifier, Instruction Manuel, PMI Motors Division of Kollmorgen Corporation, New York, July 1981.

7. Luh, J. Y. S., W. D. Fisher, and R. P. C. Paul, "Joint Torque Control by a Direct Feedback for Industrial Robots," School of Electrical Engineering, Purdue University, April 1977.

8. Snyder, W. E., and Jorg Schott, "Using Optical Shaft Encoders," Robotics Age, Vol. 2, No. 3, Fall 1980.

9. Advanced Continuous Simulation Language User Guide/Reference Manual, Mitchell and Gauther Assoc. Inc., 1981.

10. Landau, Y. D., Adaptive Control the Model Reference Approach, Marcel Dekker, Inc., New York, 1979.

11. Dubowsky, S., and D. T. DesForges, "The Application of Model-Reference Adaptive Control to Robotic Manipulators," Journal of Dynamic System, Measurement and Control, Vol. 101, September 1979.

12. Astrom, K. J., and B. Wittenmark, "Self-Tuning Controllers Based on Pole-Zero Placement," Proceedings IEE, Vol. 127, May 1980.

13. Clarke, D. W., and P. J. Gawthrop, "Self-Tuning Control," Proceedings IEE, Vol. 126, 1979.

14. Astrom, K. J., and B. Wittenmark, "On Self-Tuning Regulators," IFAC Congress, Paris, France, June 1972.

15. Astrom, K. J., and B. Wittenmark, "Theory and Applications of Self-Tuning Regulators," 6th IFAC Congress on Control Technology in the Service of Man, Boston, MA, August 1975.

16. Gawthrop, P. J., "Some Interpretations of the Self-Tuning Controller," Proceedings of the Institute of Electrical Engineers, Control and Science, Vol. 124, 1977.

17. Clarke, D. W., and P. J. Gawthrop, "Self-Tuning Controller," Proceedings of the Institute of Electrical Engineers, Control and Science, Vol. 122, 1975.

18. Goodwin, G. C., and K. S. Sin, Adaptive Filtering Prediction and Control, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1984.

19. Bryson, A. E., and Y.-C. Ho, Applied Optimal Control, Hemisphere Publishing Company, John Wiley and Sons, 1975.

20. Junkins, J. L., An Introduction to Optimal Estimation of Dynamical Systems, Sijthoff and Noordhoff International Publishers, 1978.

21. Koivo, A. J., and R. P. Paul, "Manipulator with Self-Tuning Controller," Proceedings of the International Conference Cybernetics Society, October 8-10, 1980.

22. Startz, R., 8087 Applications and Programming for the IBM PC and other PC's, Robert J. Brady Co., Bowie, MD, 1983.

23. Lee, C. S. G., and M. J. Chung, "An Adaptive Control Strategy for Computer-Based Manipulators," Decision and Control, Vol. 2, 1982.

24. Depkovich, T., and H. Elliott, "Experiments in Nonlinear Adaptive Control of Mechanical Linkage Systems," Proceedings of the Meeting of the Coordinating Group on Modern Control Theory (4th), October 1982.

Appendix A Four-Bar Link Models

For the simulated mechanism, the following models are used to estimate the experimental mechanism. Here, the kinematic lengths and link masses are the same as the experimental mechanism.

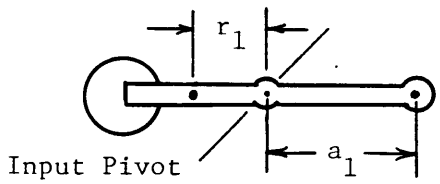
The base form for each link model is a rod with counterweights attached on the ends to represent the lumped portions of the real links. Flywheels are used to represent large inertial link sections. The gravity centers all lie on the link centerline axis.

The input link was the most difficult to estimate. The actual link is a Brigg's and Straton three horsepower motor crank shaft. For this study, it is modeled as a rod with a counterweight opposite the coupler joint (Fig. A.1a). The kinematic length a_1 was measured. However, the distance r_2 was chosen as one-half the length a_1. The moment of inertia was chosen based on observations of the other links.

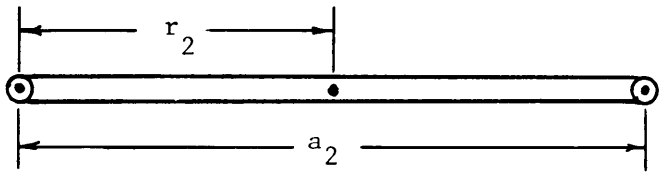
The coupler link was represented by a rod of length a_2 and a sufficient diameter to have the same mass. The material is aluminum (Fig. A.1b). For a rod the moment of inertia is

$$\bar{I}_{CG} = \bar{M} \frac{1}{48} (3\ d^2 + 4\ a^2)$$

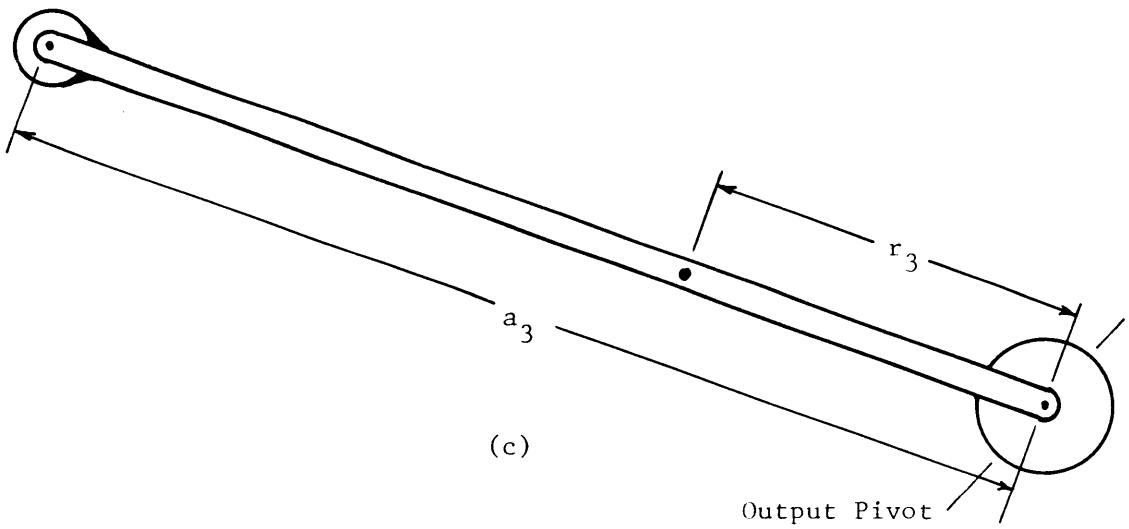
The output link can be loaded using a detachable flywheel on the pivot shaft. Attaching the flywheel only results in a change of the link inertia. The net gravitational effects remain the same. The permanent parts of the output link are used to determine the required parameters. The addition of the flywheel is then a step increase in the moment of inertia.



(a)



(b)



(c)

Fig. A.1. Simulated Mechanism Link Models

a. Input Link
b. Coupler Link
c. Output Link

The output rocker is modeled as a rod with a lumped mass on the moving end to represent the coupler-rocker pin, and a flywheel for the pivot shaft. Again, the masses are kept the same as the actual link, and the inertias are estimated from the simplified model (Fig. A.1c). Summing the moments about the pivot axis yields the value for r_3. The parallel axis theorem yields the estimate for I3. The attachable flywheel has a six inch diameter and a moment of inertia of 4.3 oz-in-s^2 (302 dyn-cm-s^2). The link final parameters are given in Table 3.1.

Appendix B

ACSL Simulation Program Listings

```
PROGRAM STANDARD CONTROL
" This is an ACSL program which will simulate the      "
" response of a FOUR-BAR linkage driven by a           "
" D.C. SERVO-MOTOR.                                    "
"                                                      "
" The control system simulates the response of a       "
" Kollmorgen Corporation PMI Servo-Motor and switching "
" servo amplifier.                                     "
"                                                      "
" The nonlinear four-bar linkage dynamics are described"
" using Kinematic Influence Coefficients.              "
"                                                      "
" CONSTANT PARAMETERS FOR THE CONTINUOUS MODEL         "

"--------------- INERTIAL PARAMETERS -----------------"

CONSTANT  M1 = 0.0827 ,  M2 = 0.007   ,  M3 = 0.104  , ...
          I1 = 0.648  ,  I2 = 0.00635 ,  I3 = 5.0    , ...
          R1 = 0.375  ,  R2 = 1.625   ,  R3 = 2.06,    ...
         TH1 = 3.1415 , TH2 = 0.0     , TH3 = 0.0

"------------------- LINK LENGTHS -----------------"

CONSTANT  AA1 = 0.75,  AA2 = 3.25,  AA3 = 5.5,  AA4 = 6.0

"-----------------  MOTOR PARAMETERS  -------------"

CONSTANT  KPP = 6.10,  DP = 0.01146
CONSTANT  AAMP = 30.0

"-------------- CONTROLLER PARAMETERS ---------------"

CONSTANT  KFG = 5.0, P = 5.684E-3, Q = 6.6314E-3


"-------------- SIMULATION CONTROL PARAMETERS -------"

CONSTANT  Q1D0 = 0.0, Q10 = 0.0,  TMAX = 1.0,  G = 386.0

"-------------- INPUT CONDITION PARAMETERS ----------"

CONSTANT  STPT = 75.0

"                                                      "
"  INITIALIZE THE FIRST POSITION OF THE MECHANISM      "
"                                                      "
INITIAL
SGN3=-1.
SGN2=1.
```

```
 AI=SIN(Q10)
 BI=COS(Q10)-AA4/AA1
 CI=-AA4/AA3*COS(Q10)+(AA1**2+AA3**2+AA4**2-AA2**2)/2/AA1/AA3
 CC3=(AA1**2+AA3**2+AA4**2-AA2**2)/2./AA1/AA3
 Q30=2.*ATAN((AI+SGN3*SQRT(AI**2+BI**2-CI**2))/(BI+CI))
 CII=(AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2-AA4/AA2*COS(Q10)
 CC2= (AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2
 Q20=2.*ATAN((-AI+SGN2*SQRT(AI**2+BI**2-CII**2))/(CII-BI))
 END   $"END OF INITIAL"
"                                                    "
 CINTERVAL CINT = 0.001  $" CINT THE INTEGRATION INTERVAL "

 DERIVATIVE

 "-- INPUT FUNCTION GENERATION ------------------------"
 PROCEDURAL (STPT = T,STPT)
       IF (T .GT. 0.7333) GOTO LBL1
       IF (T .GT. 0.4   ) GOTO LBL2
        STPT = 50.0
        GOTO LBL3
  LBL2..STPT = 50.0+75*(T -0.4)
        GOTO LBL3
  LBL1..STPT = 75.0
  LBL3..CONTINUE
 END $" OF INPUT GENERATION "

 "  CONTROL EQUATION FOR THE PMI LEAD-LAG CONTROLLER   "

 ERR=(STPT-Q1D)*KFG
 DQ4= STATE+P*ERR/Q

 "-- 30 AMP MOTOR CURRENT LIMIT ----------------------"

 Q4 = BOUND (-AAMP,AAMP,DQ4)
 STATE=INTEG((ERR-DQ4)/Q,0.0)
 TL  = KPP*Q4-DP*Q1D

 "--  Q1DD IS THE INPUT LINK ACCELERATION ------------"
 Q1DD= (TL-                                          ...
       Q1D*Q1D*(I2*G2*H2+I3*G3*H3+M2*                ...
       GBHB)                                         ...
       -G*(M1*X1+M2*X2+M3*X3))/                      ...
      (I1+I2*G2*G2+I3*G3*G3+M2*G2B)

 "-- Q1D IS THE INPUT LINK VELOCITY -----------------"
 Q1D= INTEG(Q1DD,Q1D0)

 "-- Q1 IS THE INPUT LINK POSITION ------------------"
 Q1=  INTEG(Q1D,Q10)
```

```
"-- Q1 RANGE LIMITED FROM 0.0 TO 2*PI --------------"
Q1 = RSW(Q1 .GE. 6.283186, Q1-6.283186, Q1)

"-- CALCULATE THE CURRENT POSITION OF THE MECHANISM -"
A = SIN(Q1)
B = COS(Q1)-AA4/AA1
C = CC2-AA4/AA2*COS(Q1)
FF2 = (-A+SGN2*SQRT(A*A+B*B-C*C))/(C-B)
Q2 = ATAN(FF2)*2.
B3 = COS(Q1)-AA4/AA1
C3 = CC3-AA4/AA3*COS(Q1)
FF3 = (A+SGN3*SQRT(A*A+B3**2-C3**2))/(B3+C3)
Q3 = 2*ATAN(FF3)

"-- CALCULATE THE INFLUENCE COEFFICENTS ------------"

"-- ROTATIONAL INFLUENCE COEFFICIENTS --------------"
G3 = AA1/AA3*SIN(Q1-Q2)/SIN(Q3-Q2)
G2 = AA1/AA2*SIN(Q1-Q3)/SIN(Q3-Q2)

"-- ACCELERATION INFLUENCE COEFFICIENTS ------------"
H2 = AA1/AA2*(SIN(Q3-Q2)*COS(Q1-Q3)*(1.-G3)-      ...
     SIN(Q1-Q3)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2))**2
H3 = AA1/AA3*(SIN(Q3-Q2)*COS(Q1-Q2)*(1.-G2)-      ...
     SIN(Q1-Q2)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2)**2)

"-- LINK 2 TRANSLATIONAL INFLUENCE COEFFICIENTS ----"
G2B = AA1**2+R2*R2*G2*G2+ 2.*AA1*R2*G2*COS(-Q1+Q2+TH2)
GBHB = R2*R2*G2*H2+ AA1*R2*(H2*COS(-Q1+Q2+TH2)-   ...
       G2*SIN(-Q1+Q2+TH2)*(G2-1.))

"-- GRAVITATIONAL INFLUENCES COEFFICIENTS ----------"
X1 = R1*COS(Q1+TH1)
X2 = AA1*COS(Q1)+G2*R2*COS(Q2+TH2)
X3 = R3*COS(Q3+TH3)*G3

"-- STOP AT TIME TMAX -------------------------------"
TERMT (T .GE. TMAX)
END $" OF DERIVATIVE "
END $" OF PROGRAM"
```

```
PROGRAM   ARMA ADAPTIVE CONTROL
"This ACSL program simulates the Four-bar       "
"response using Koivo's Autoregressive Step-    "
"ahead joint control (Section 6).  This is also "
"discussed in Goodwin and Sin.                  "
"                                               "
"The simulation uses the instrument and actuator"
"models.  (Section 3.4)
"                                               "
"The FOUR-BAR model is discribed using the full "
"influence coefficient nonlinear dynamic        "
"equations.
" CONSTANT PARAMETERS FOR THE CONTINUOUS MODEL  "

CONSTANT  M1 = 0.0827 ,  M2 = 0.007   ,  M3 = 0.104,    ...
          I1 = 0.648  ,  I2 = 0.00635 ,  I3 = 5.0  ,    ...
          R1 = 0.375  ,  R2 = 1.625   ,  R3 = 2.06 ,    ...
         TH1 = 3.1415 , TH2 = 0.0     , TH3 = 0.0

"------------------- LINK LENGTHS ---------------------"

CONSTANT  AA1 = 0.75,  AA2 = 3.25,  AA3 = 5.5,  AA4 = 6.0

"-----------------  MOTOR PARAMETERS  -----------------"

CONSTANT  KPP = 6.10,  DP = 0.01146
CONSTANT APWR = 183.0

"-------------- ADAPTIVE CONTROL PARAMETERS -----------"

CONSTANT ALPHAP = 0.01
CONSTANT EPSL  = 0.0001, ROE=.999
CONSTANT CGAIN=0

"------------- SIMULATION CONTROL PARAMETERS ----------"

CONSTANT  Q1D0 = 0.0, Q10 = 0.0,  TMAX = 2.0,  G = 386.0

"------------- INPUT CONDITION PARAMETERS -------------"

CONSTANT  STPT = 75.0

INITIAL $" THE TIME = ZERO VALUES OF PARAMETERS -------"
TL=0.0
SGN3=-1.
SGN2=1.

"-- CALCULATE THE TIME = 0 POSITION OF THE MECHANISM --"

AI=SIN(Q10)
```

```
BI=COS(Q10)-AA4/AA1
CI=-AA4/AA3*COS(Q10)+(AA1**2+AA3**2+AA4**2-AA2**2)/2/AA1/AA3
CC3=(AA1**2+AA3**2+AA4**2-AA2**2)/2./AA1/AA3
Q30=2.*ATAN((AI+SGN3*SQRT(AI**2+BI**2-CI**2))/(BI+CI))
CII=(AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2-AA4/AA2*COS(Q10)
CC2= (AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2
Q20=2.*ATAN((-AI+SGN2*SQRT(AI**2+BI**2-CII**2))/(CII-BI))

"-- INITIALIZE THE ADAPTIVE CONTROLLER PARAMETERS -----"
CALL CTLINT
END   $"END OF INITIAL"

"-- INTEGRATION INTERVAL FOR THE CONTINUOUS SECTION ---"
CINTERVAL CINT=.0005

DERIVATIVE $" DEFINE THE STATE VARIABLE EQUATIONS -----"

"--  DEFINE THE INPUT FUNCTION ------------------------"
 PROCEDURAL (STPT = T,STPT)
      IF (T .GT. 1.1333) GOTO LBL1
      IF (T .GT. .8    ) GOTO LBL2
       STPT = 50.0
       GOTO LBL3
 LBL2..STPT = 50.0+ 75*(T-.8)
       GOTO LBL3
 LBL1..STPT = 75.0
 LBL3..CONTINUE
 END $"OF INPUT FUNCTION"

"--  Q1DD IS THE INPUT LINK ACCELERATION  -------------"
Q1DD= (TL-DP*Q1D-                                          ...
      Q1D*Q1D*(I2*G2*H2+I3*G3*H3+M2*                       ...
      GBHB)                                                ...
      -G*(M1*X1+M2*X2+M3*X3))/                             ...
      (I1+I2*G2*G2+I3*G3*G3+M2*G2B)

"-- INTEGRATE TO GET THE INPUT LINK VELOCITY ----------"
Q1D= INTEG(Q1DD,Q1D0)

"-- INTEGRATE TO GIVE THE INPUT LINK POSITION ---------"
Q1=  INTEG(Q1D,Q10)

"-- LIMIT RANGE OF Q1 FROM 0.0 TO 2*PI ----------------"
Q1 = RSW(Q1 .GE. 6.283186, Q1-6.283186, Q1)

"-- CALCULATE THE MECHANISM POSITION ------------------"
A = SIN(Q1)
B = COS(Q1)-AA4/AA1
C = CC2-AA4/AA2*COS(Q1)
FF2 = (-A+SGN2*SQRT(A*A+B*B-C*C))/(C-B)
```

```
Q2 = ATAN(FF2)*2.
B3 = COS(Q1)-AA4/AA1
C3 = CC3-AA4/AA3*COS(Q1)
FF3 = (A+SGN3*SQRT(A*A+B3**2-C3**2))/(B3+C3)
Q3 = 2*ATAN(FF3)

"-- CALCULATE THE INFLUENCE COEFFICIENTS --------------"

"-- ROTATIONAL INFLUENCE COEFFICIENTS -----------------"
G3 = AA1/AA3*SIN(Q1-Q2)/SIN(Q3-Q2)
G2 = AA1/AA2*SIN(Q1-Q3)/SIN(Q3-Q2)

"-- ACCELERATION INFLUENCE COEFFICIENTS ---------------"
H2 = AA1/AA2*(SIN(Q3-Q2)*COS(Q1-Q3)*(1.-G3)-      ...
     SIN(Q1-Q3)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2))**2
H3 = AA1/AA3*(SIN(Q3-Q2)*COS(Q1-Q2)*(1.-G2)-      ...
     SIN(Q1-Q2)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2)**2)

"-- TRANSLATIONAL INFLUENCE COEFFICIENTS -------------"
G2B = AA1**2+R2*R2*G2*G2+ 2.*AA1*R2*G2*COS(-Q1+Q2+TH2)
GBHB =  R2*R2*G2*H2+ AA1*R2*(H2*COS(-Q1+Q2+TH2)- ...
        G2*SIN(-Q1+Q2+TH2)*(G2-1.))

"-- GRAVATATIONAL INFLUENCE COEFFICIENTS -------------"
X1 = R1*COS(Q1+TH1)
X2 = AA1*COS(Q1)+G2*R2*COS(Q2+TH2)
X3 = R3*COS(Q3+TH3)*G3

TERMT (T .GE. TMAX)
END $"OF DERIVATIVE"

DISCRETE $" ADAPTIVE CONTROL "

INTERVAL TSAMP=.004 $" CONTROLLER SAMPLE INTERVAL  ---"
PROCEDURAL (TL=T,Q1D,TL,TSAMP)
"-- SET UP INSTRUMENT VALUES FOR THE DISCRETE CONTROL "
"-- OPTICAL ENCODER MODEL ----------------------------"
     IF (Q1D .GT. 15.34) GO TO ELSE
     VEL = 0.0
     GOTO LABEL
ELSE..NN = INT (6.283E-3/Q1D/1.0E-7)
     VEL = 6.283E-3/NN/1.0E-7
"-- ADC MODEL FOR TORQUE MEASUREMENT -----------------"
LABEL..TORQ = QNTZR (0.179,TL)
"-- CALCULATE THE ADAPTIVE CONTROL -------------------"
CALL CNTRL
END $"OF PROCEDURAL"
END $"OF DISCRETE"
END $ "OF PROGRAM"
     SUBROUTINE CTLINT
```

```
C   THIS SUBROUTINE SETS UP THE COMMON VARIABLES FOR THE
C   ADAPTIVE CONTROL AND SETS THEM TO THEIR TIME = ZERO
C   VALUES.
C
C   THE VARIABLES THE DEFINED AS
C         P  --- IS THE COVARIANT MATRIX
C        XK  --- ARE THE ESTIMATED CONTROL PARAMETERS
C
      COMMON /CNTRL/ P(7,7), BASIS(7), XK(7),
     & TEMP1(7), TP(7)
      DO 30 K4=1,7
      P(K4,K4)=100.0
      XK(K4)=0.1
      BASIS(K4)=1.0
  30  CONTINUE
      BASIS(7)=1.
      RETURN
      END
C
      SUBROUTINE CNTRL
      COMMON /CNTRL/ P(7,7), BASIS(7), XK(7),
     & TEMP1(7), TP(7)
$

C  APPLY 100 OZ-IN UNTIL OPTICAL ENCODER "STARTS"
      IF (VEL - 1.0) 110,120,120
 110  TL=100
       RETURN
 120  CONTINUE
C
C  APPLY THE CONTROL SIGNAL AFTER A SAMPLE INTERVAL
C  DELAY
       TL=TLHOLD
C  CALCULATE THE LEAST SQUARES PARAMETER ESTIMATOR
      CALL MULA(BASIS,P,TEMP1)
      CALL MLT(TEMP1,BASIS,SCLR)
      CALL MULP(P,BASIS,TP)
      CALL MLT(BASIS,XK,SCLR2)
C  WITH VARIABLE FORGETTING FACTOR
       ROE = 1.-(ALPHAP*(VEL-SCLR2)**2/(SCLR+1.))
       IF (ROE .GT. 0.9999) ROE = 1.0
       SCLR = 1.0/(SCLR+ROE)
       DO 10 K1=1,7
  10  XK(K1)= XK(K1)+ TP(K1)*SCLR*(VEL-SCLR2)
      WRITE(7,*) (XK(K1),K1=1,7),T
C
C   CALCULATE THE AUTOREGRESSIVE STEP AHEAD CONTROL
C
      STO=0.0
```

```
      STO1=0.0
      DO 40 KJ=2,3
      STO=STO+BASIS(KJ-1)*XK(KJ)
      STO1=STO1+BASIS(KJ-1+3)*XK(KJ+3)
 40   CONTINUE
      ERR=STO+STO1+VEL*XK(1) +XK(7) - STPT
      CGAIN= XK(4)/(XK(4)**2+EPSL)
      TLHOLD=CGAIN* -ERR
C CLIP OUTPUT TORQUE FOR MOTOR MODEL
      IF (TLHOLD.LT.-APWR) TLHOLD=-APWR
      IF (TLHOLD.GT. APWR) TLHOLD= APWR
C
C UPDATE THE P MATRIX USING THE ALREADY CALCULATED NUMBERS
C
       DO 20 K3=1,7
        DO 20 K2=1,7
 20   P(K3,K2)= 1.0/ROE*(P(K3,K2)-SCLR*TP(K3)*TEMP1(K2))
C
C  UPDATE THE BASIS VECTOR A
      DO 15 K=1,6
 15   BASIS(7-K)=BASIS(6-K)
      BASIS(1)=VEL
      BASIS(4)=TORQ
      RETURN
      END
C
C   SUBROUTINES
C   PERFORM THE REQUIRED MATRIX MATH FOR THE CONTROL
C
      SUBROUTINE MULA(AFCTR,FCTR,ANS)
C MULTIPLIES A 1x7  AND A   (7x7)
      DIMENSION AFCTR(7), FCTR(7,7), ANS(7)
       DO 10 KR=1,7
      STO = 0.0
        DO 20 KC=1,7
 20   STO = AFCTR(KC)*FCTR(KC,KR) + STO
 10   ANS(KR)= STO
      RETURN
      END
C
      SUBROUTINE MLT(FTR1,FTR2,ANS)
C THIS MULTIPLIES A (1x7) TO A (7x1)  RETURNS A SCALAR
      DIMENSION FTR1(7), FTR2(7)
      ANS=0.0
       DO 10 KR=1,7
 10   ANS=ANS+FTR1(KR)*FTR2(KR)
      RETURN
      END
C
```

```
      SUBROUTINE MULP(FTR1,FTR2,ANS)
C MULTIPLIES A (7x7) TO A (7x1)
      DIMENSION FTR1(7,7), FTR2(7), ANS(7)
       DO 10 KR=1,7
      STO =0.0
        DO 20 KC=1,7
  20  STO= STO+ FTR1(KR,KC)*FTR2(KC)
  10  ANS(KR)=STO
      RETURN
      END
```

```
PROGRAM PERTURBATION ADAPTIVE CONTROL
"This ACSL program simulates the response of the  "
"four-bar linkage under the perturbation control  "
"as described in section 7.                       "
"                                                 "
"The simulation uses the instrument and actuator  "
"models. (Section 3.4)                            "
"                                                 "
"The Four-bar dynamics are modelled using the     "
"Influence coefficient method.                    "
"                                                 "
" CONSTANT PARAMETERS FOR THE CONTINUOUS MODEL    "

CONSTANT  M1 = 0.0827 ,  M2 = 0.007   ,  M3 = 0.104,  ...
          I1 = 0.648  ,  I2 = 0.00635 ,  I3 = 5.0  ,  ...
          R1 = 0.375  ,  R2 = 1.625   ,  R3 = 2.06 ,  ...
         TH1 = 3.1415 , TH2 = 0.0     , TH3 = 0.0

"------------------- LINK LENGTHS ---------------------"

CONSTANT  AA1 = 0.75,  AA2 = 3.25,  AA3 = 5.5,  AA4 = 6.0

"-----------------  MOTOR PARAMETERS  -----------------"

CONSTANT  KPP = 6.10,  DP = 0.01146
CONSTANT APWR = 396.0

"-------------- ADAPTIVE CONTROL PARAMETERS -----------"

CONSTANT  EPSL = 0.0001,   ROE = 0.9

"-------------- SIMULATION CONTROL PARAMETERS ----------"

CONSTANT  Q1D0 = 0.0, Q10 = 0.0,  TMAX = 2.0,  G = 386.0
CONSTANT  CTL=0, TLH = 0
CONSTANT TBRK=1.0 ,CGAIN = 0.0

"-------------- INPUT CONDITION PARAMETERS -------------"

CONSTANT  STPT = 75.0
"                                                      "
" INITIALIZE THE TIME = ZERO SIMULATION PARAMETERS ----"
INITIAL
SGN3 = -1.0
SGN2 =  1.0
AI = SIN(Q10)
BI = COS(Q10)-AA4/AA1
CI = -AA4/AA3*COS(Q10)+(AA1**2+AA3**2+AA4**2-AA2**2)/2/AA1/AA3
CC3 = (AA1**2+AA3**2+AA4**2-AA2**2)/2./AA1/AA3
Q30 = 2.*ATAN((AI+SGN3*SQRT(AI**2+BI**2-CI**2))/(BI+CI))
```

```
CII = (AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2-AA4/AA2*COS(Q10)
CC2 = (AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2
Q20 = 2.*ATAN((-AI+SGN2*SQRT(AI**2+BI**2-CII**2))/(CII-BI))
CALL CTLINT
END   $"END OF INITIAL"

"-- CALCULATE THE CONTINUOUS FOUR-BAR DYNAMICS ---------"
CINTERVAL CINT=.0005
DERIVATIVE $" DEFINE THE STATE VARIABLE EQUATIONS -----"

"-- INPUT LINK ACCELERATION ---------------------------"
Q1DD = (TL-Q1D*DP-                                          ...
       Q1D*Q1D*(I2*G2*H2+I3*G3*H3+M2*                       ...
       GBHB)                                                ...
       -G*(M1*X1+M2*X2+M3*X3))/                             ...
       (I1+I2*G2*G2+I3*G3*G3+M2*G2B)

"-- INTEGRATE FOR VELOCITY ----------------------------"
Q1D = INTEG(Q1DD,Q1D0)

"-- INTEGRATE FOR POSITION ----------------------------"
Q1 =  INTEG(Q1D,Q10)

"-- LIMIT RANGE OF Q1 FROM 0.0 TO 2*PI ----------------"
Q1 = RSW(Q1 .GE. 6.283186, Q1-6.283186, Q1)

"-- CALCULATE THE MECHANISM POSITION ------------------"
A = SIN(Q1)
B = COS(Q1)-AA4/AA1
C = CC2-AA4/AA2*COS(Q1)
FF2 = (-A+SGN2*SQRT(A*A+B*B-C*C))/(C-B)
Q2 = ATAN(FF2)*2.
B3 = COS(Q1)-AA4/AA1
C3 = CC3-AA4/AA3*COS(Q1)
FF3 = (A+SGN3*SQRT(A*A+B3**2-C3**2))/(B3+C3)
Q3 = 2*ATAN(FF3)

"-- ROTATIONAL INFLUENCE COEFFICIENTS -----------------"
G3 = AA1/AA3*SIN(Q1-Q2)/SIN(Q3-Q2)
G2 = AA1/AA2*SIN(Q1-Q3)/SIN(Q3-Q2)

"-- ACCELERATION INFLUENCE COEFFICIENTS ---------------"
H2 = AA1/AA2*(SIN(Q3-Q2)*COS(Q1-Q3)*(1.-G3)-       ...
     SIN(Q1-Q3)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2))**2
H3 = AA1/AA3*(SIN(Q3-Q2)*COS(Q1-Q2)*(1.-G2)-       ...
     SIN(Q1-Q2)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2)**2)

"-- TRANSLATIONAL INFLUENCE COEFFICIENTS --------------"
G2B = AA1**2+R2*R2*G2*G2+ 2.*AA1*R2*G2*COS(-Q1+Q2+TH2)
GBHB = R2*R2*G2*H2+ AA1*R2*(H2*COS(-Q1+Q2+TH2)- ...
```

```
        G2*SIN(-Q1+Q2+TH2)*(G2-1.))

 "-- GRAVATATIONAL INFLUENCE COEFFICENTS ---------------"
 X1 = R1*COS(Q1+TH1)
 X2 = AA1*COS(Q1)+G2*R2*COS(Q2+TH2)
 X3 = R3*COS(Q3+TH3)*G3
 TERMT (T .GE. TMAX)
 END
 DISCRETE $" ADAPTIVE CONTROL "

 INTERVAL TSAMP = 0.004 $"THE CONTROLLER SAMPLE INTERVAL"
 PROCEDURAL (TL = T,Q1D,TL,TSAMP)
 "-- SET UP INSTRUMENT OUTPUT FOR THE DISCRETE CONTROL -"
 "-- CALCULATE THE OPTICAL ENCODER MEASUREMENTS --------"
 IF (Q1D .GT. 15.34) GOTO ELSE
 VEL=0.0
 GOTO LABEL
 ELSE..NN = INT(6.283E-3/Q1D/1.0E-7)
 VEL = 6.283E-3/NN/1.0E-7

 "-- CALCULATE THE ADC TORQUE MEASUREMENT --------------"
LABEL..TORQ = QNTZR(0.179,TL)

 "-- CALCULATE THE ADAPTIVE CONTROL --------------------"
 CALL CNTRL
 END $" OF PROCEDURAL "
 END $" OF DISCRETE   "
END $ "OF PROGRAM"
      SUBROUTINE CTLINT
C  SET UP THE INITIAL VALUES FOR THE CONTROL
      COMMON /CNTRL/ P(3,3), BASIS(3), XK(3),
     & TEMP(3), TP(3),CI1,CI2,CI3,CM1,CM2,CM3
      DO 30 K4=1,3
      P(K4,K4)=100.0
      XK(K4)=1.0
      BASIS(K4)=1.0
  30  CONTINUE
      BASIS(3)=1.
      CI1=0.64
      CI2=0.006
      CI3=5.1
      CM1 = 0.08
      CM2 = 0.007
      CM3 = 0.1
      RETURN
      END
C
      SUBROUTINE CNTRL
      COMMON /CNTRL/ P(3,3), BASIS(3), XK(3),
```

```
     & TEMP1(3), TP(3),CI1, CI2,CI3,CM1,CM2,CM3
$
C   APPLY 100 IN OZ TORQUE UNTIL OPTICAL ENCODER WORKS
C
       IF (VEL .GT. 0.1) GOTO 123
       TL = 100.0
       RETURN
 123   CONTINUE
C
C  APPLY THE CONTROL SIGNAL AFTER A SAMPLE INTERVAL
C  DELAY
       TL=TLHOLD
C
C CALCULATE THE ADVANCE VELOCITY PATH
C
       IF (T .GT. TBRK) GOTO 121
       STPTK = 50.0
       STPTK1 = 50.0
       STPTK2 = 50.0
        GOTO 122
 121   STPTK = 50.+75*(T-TBRK)
       STPTK1= 50.+75*(T-TBRK+TSAMP)
       STPTK2= 50.+75*(T-TBRK+2*TSAMP)
       IF(STPTK .GT. 75) STPTK=75
       IF(STPTK1.GT. 75) STPTK1=75
       IF(STPTK2.GT. 75) STPTK2=75
 122   STPT=STPTK
C
C  CALCULATE THE REQUIRED TORQUE AT K+1 USING THE
C  POSITION FOR THE COMPUTED TORQUE MODEL AT TIME K
C
C  THIS SECTION WILL SIMULATE THE LOOK UP TABLE
      Q1P=Q1 + TSAMP*VEL
       NNN=IFIX(Q1P/0.1047)
       Q1P = FLOAT(NNN)*0.1047
      AIL=SIN(Q1P)
      BIL=COS(Q1P)-AA4/AA1
      CIL=-AA4/AA3*COS(Q1P)+(AA1**2+AA3**2+AA4**2-AA2**2)/2/AA1/AA3
      Q3L=2.*ATAN((AIL+SGN3*SQRT(AIL**2+BIL**2-CIL**2))/(BIL+CIL))
      CLL=(AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2-AA4/AA2*COS(Q1P)
      Q2L=2.*ATAN((-AIL+SGN2*SQRT(AIL**2+BIL**2-CLL**2))/(CLL-BIL))
      G3L= AA1/AA3*SIN(Q1P-Q2L)/SIN(Q3L-Q2L)
      G2L= AA1/AA2*SIN(Q1P-Q3L)/SIN(Q3L-Q2L)
      H2L = AA1/AA2*(SIN(Q3L-Q2L)*COS(Q1P-Q3L)*(1.-G3L)-
     &SIN(Q1P-Q3L)*COS(Q3L-Q2L)*(G3L-G2L))/(SIN(Q3L-Q2L))**2
      H3L= AA1/AA3*(SIN(Q3L-Q2L)*COS(Q1P-Q2L)*(1.-G2L)-
     &SIN(Q1P-Q2L)*COS(Q3L-Q2L)*(G3L-G2L))/(SIN(Q3L-Q2L)**2)
      G2BL=AA1**2+R2*R2*G2L*G2L+ 2.*AA1*R2*G2L*COS(-Q1P+Q2L+TH2)
      GBHBL= R2*R2*G2L*H2L+ AA1*R2*(H2L*COS(-Q1P+Q2L+TH2)-
     &G2L*SIN(-Q1P+Q2L+TH2)*(G2L-1.))
```

```
      X1L= R1*COS(Q1P+TH1)
      X2L= AA1*COS(Q1P)+G2L*R2*COS(Q2L+TH2)
      X3L= R3*COS(Q3L+TH3)*G3L
C  END OF THE LOOK UP TABLE
C  CALCULATE THE COMPUTED TORQUE
      CNI1=CI1+CI2*G2L*G2L+CI3*G3L*G3L+CM2*G2BL
      CCENT=CI2*G2L*H2L+CI3*G3L*H3L+CM2*GBHBL
      CGRV= G*(CM1*X1L+CM2*X2L+CM3*X3L)
      CTL=(STPTK2-STPTK1)/TSAMP*CNI1+STPTK1**2*CCENT+
     &    CGRV+STPTK1*DP
C
C  UPDATE AUTOREGRESSION PERTURBATION CONTROL
C
      CALL MULA(BASIS,P,TEMP1)
      CALL MLT(TEMP1,BASIS,SCLR)
      SCLR= 1./(SCLR+ROE)
      CALL MULP(P,BASIS,TP)
      CALL MLT(BASIS,XK,SCLR2)
      DELX = VEL - STPTK
       DO 10 K1=1,3
 10   XK(K1)= XK(K1)+ TP(K1)*SCLR*(DELX-SCLR2)
C
C  STEP-AHEAD CONTROL LAW
C
      STO=0.0
      STO1=0.0
      ERR= XK(1)*DELX+XK(3)
      CGAIN= XK(2)/(XK(2)**2+EPSL)
      TLH=CGAIN* -ERR
C THE ARMA TORQUE LIMIT
      IF(TLH.GT.50) TLH=50.
      IF(TLH.LT.-50)TLH=-50.
      TLHOLD=TLH+CTL
C THE MOTOR TORQUE LIMIT
      IF (TLHOLD.LT.-APWR) TLHOLD=-APWR
      IF (TLHOLD.GT. APWR) TLHOLD= APWR
C
C UPDATE THE P MATRIX USING THE ALREADY CALCULATED NUMBERS
C
       DO 20 K3=1,3
        DO 20 K2=1,3
 20   P(K3,K2)= (P(K3,K2)-SCLR*TP(K3)*TEMP1(K2))/ROE
C
C  UPDATE THE BASIS VECTOR A
C
      BASIS(1)=DELX
      BASIS(2)=TLH
      RETURN
      END
C
```

```
C   SUBROUTINES  PERFORM REQUIRED MATRIX MATH
C
      SUBROUTINE MULA(AFCTR,FCTR,ANS)
C MULTIPLIES A 1x3  AND A   (3x3)
      DIMENSION AFCTR(3), FCTR(3,3), ANS(3)
       DO 10 KR=1,3
      STO = 0.0
        DO 20 KC=1,3
  20  STO = AFCTR(KC)*FCTR(KC,KR) + STO
  10  ANS(KR)= STO
      RETURN
      END
C
      SUBROUTINE MLT(FTR1,FTR2,ANS)
C THIS MULTIPLIES A (1x3) TO A (3x1)  RETURNS A SCALAR
      DIMENSION FTR1(3), FTR2(3)
      ANS=0.0
       DO 10 KR=1,3
  10  ANS=ANS+FTR1(KR)*FTR2(KR)
      RETURN
      END
C
      SUBROUTINE MULP(FTR1,FTR2,ANS)
C MULTIPLIES A (3x3) TO A (3x1)
      DIMENSION FTR1(3,3), FTR2(3), ANS(3)
       DO 10 KR=1,3
      STO =0.0
        DO 20 KC=1,3
  20  STO= STO+ FTR1(KR,KC)*FTR2(KC)
  10  ANS(KR)=STO
      RETURN
      END
```

```
PROGRAM  NONLINEAR ADAPTIVE CONTROL
"This ACSL program simulates the control response  "
"of the Nonlinear Adaptive Control using the gener-"
"al inertia and gravatational models. (Section 8)  "
"                                                  "
"The simulation uses the instrument and actuator   "
"models. (Section 3.4)                             "
"                                                  "
"The Four-bar response is simulated using the      "
"Influence Coefficient dynamic model.              "
" CONSTANT PARAMETERS FOR THE CONTINUOUS MODEL     "

CONSTANT  M1 = 0.0827 ,  M2 = 0.007   ,  M3 = 0.107,    ...
          I1 = 0.648  ,  I2 = 0.00635 ,  I3 = 5.0  ,    ...
          R1 = 0.375  ,  R2 = 1.625   ,  R3 = 2.06 ,    ...
         TH1 = 3.1415 , TH2 = 0.0     , TH3 = 0.0

"------------------- LINK LENGTHS ---------------------"

CONSTANT  AA1 = 0.75,  AA2 = 3.25,  AA3 = 5.5,  AA4 = 6.0

"-----------------  MOTOR PARAMETERS  -----------------"

CONSTANT  KPP = 6.10,  DP = 0.01146
CONSTANT APWR = 183

"--------------- ADAPTIVE CONTROL PARAMETERS -----------"

CONSTANT  EPSL = 0.0002
CONSTANT  ROE = 0.99

"------------- SIMULATION CONTROL PARAMETERS ----------"

CONSTANT Q1D0  = 0.0, Q10 = 0.0,  TMAX = 1.0,  G = 386.0
CONSTANT TRACE = 0.0
CONSTANT TBRK  = 0.7

"------------- INPUT CONDITION PARAMETERS -------------"

CONSTANT  STPT = 75.0

"-- INITIALIZE THE TIME = ZERO SYSTEM PARAMETERS ------"
INITIAL
TLHOLD = 0
CGAIN = 0.1
SGN3 = -1.0
SGN2 = 1.0
AI = SIN(Q10)
BI = COS(Q10)-AA4/AA1
CI = -AA4/AA3*COS(Q10)+(AA1**2+AA3**2+AA4**2-AA2**2)/2/AA1/AA3
```

```
CC3 = (AA1**2+AA3**2+AA4**2-AA2**2)/2./AA1/AA3
Q30 = 2.*ATAN((AI+SGN3*SQRT(AI**2+BI**2-CI**2))/(BI+CI))
CII = (AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2-AA4/AA2*COS(Q10)
CC2 = (AA1**2+AA2**2+AA4**2-AA3**2)/2./AA1/AA2
Q20 = 2.*ATAN((-AI+SGN2*SQRT(AI**2+BI**2-CII**2))/(CII-BI))
CALL CTLINT
END   $"END OF INITIAL"

"-- SIMULATE THE FOUR-BAR DYNAMICS --------------------"
CINTERVAL CINT=.0005

DERIVATIVE $" DEFINE THE STATE SPACE EQUATIONS --------"

"-- INPUT LINK ACCELERATION ---------------------------"
Q1DD = (TL-Q1D*DP-                                          ...
       Q1D*Q1D*(I2*G2*H2+I3*G3*H3+M2*                       ...
       GBHB)                                                ...
       -G*(M1*X1+M2*X2+M3*X3))/                             ...
       (I1+I2*G2*G2+I3*G3*G3+M2*G2B)

"-- INTEGRATE FOR VELOCITY ----------------------------"
Q1D = INTEG(Q1DD,Q1D0)

"-- INTEGRATE FOR POSITION ----------------------------"
Q1 = INTEG(Q1D,Q10)

"-- LIMIT RANGE OF Q1 TO 0.0 TO 2*PI ------------------"
Q1 = RSW(Q1 .GE. 6.283186, Q1-6.283186, Q1)

"-- CALCULATE THE MECHANISM POSITION ------------------"
A = SIN(Q1)
B = COS(Q1)-AA4/AA1
C = CC2-AA4/AA2*COS(Q1)
FF2 = (-A+SGN2*SQRT(A*A+B*B-C*C))/(C-B)
Q2 = ATAN(FF2)*2.
B3 = COS(Q1)-AA4/AA1
C3 = CC3-AA4/AA3*COS(Q1)
FF3 = (A+SGN3*SQRT(A*A+B3**2-C3**2))/(B3+C3)
Q3 = 2*ATAN(FF3)

"-- ROTATIONAL INFLUENCE COEFFICIENTS -----------------"
G3 = AA1/AA3*SIN(Q1-Q2)/SIN(Q3-Q2)
G2 = AA1/AA2*SIN(Q1-Q3)/SIN(Q3-Q2)

"-- ACCELERATION INFLUENCE COEFFICIENTS ---------------"
H2 = AA1/AA2*(SIN(Q3-Q2)*COS(Q1-Q3)*(1.-G3)-       ...
     SIN(Q1-Q3)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2))**2
H3 = AA1/AA3*(SIN(Q3-Q2)*COS(Q1-Q2)*(1.-G2)-       ...
     SIN(Q1-Q2)*COS(Q3-Q2)*(G3-G2))/(SIN(Q3-Q2)**2)
```

```
 "-- TRANSLATIONAL INFLUENCE COEFFICIENTS --------------"
 G2B = AA1**2+R2*R2*G2*G2+ 2.*AA1*R2*G2*COS(-Q1+Q2+TH2)
 GBHB = R2*R2*G2*H2+ AA1*R2*(H2*COS(-Q1+Q2+TH2)- ...
        G2*SIN(-Q1+Q2+TH2)*(G2-1.))

 "-- GRAVATATIONAL INFLUENCE COEFFICIENTS --------------"
 X1 = R1*COS(Q1+TH1)
 X2 = AA1*COS(Q1)+G2*R2*COS(Q2+TH2)
 X3 = R3*COS(Q3+TH3)*G3

 "-- PROGRAM TERMINATION --------------------------------"
 TERMT (T .GE. TMAX)
 END $"OF DERIVATIVE"

 DISCRETE $" ADAPTIVE CONTROL --------------------------"
 INTERVAL TSAMP=.004 $" CONTROLLER SAMPLE INTERVAL -----"
 PROCEDURAL (TL=T,Q1D,Q1DD,TL,TSAMP)
 "-- SET UP  INSTRUMENT VALUES FOR THE DICRETE CONTROL -"
      IF (Q1D .GT.    15.34) GOTO ELSE
      VEL = 0.0
      GOTO LABEL
 ELSE..NN=INT(6.283E-3/Q1D/1.0E-7)
       VEL  = 6.283E-3/NN/1.0E-7
 "-- ADC MODEL FOR TORQUE MEASUREMENT ------------------"
 LABEL..TORQ = QNTZR (0.179,TL)
      CALL CNTRL
 END $"OF PROCEDURAL"
 END $"DISCRETE"
END $ "OF PROGRAM"
      SUBROUTINE CTLINT
C  SET UP THE INITIAL VALUES FOR THE ADAPTIVE CONTROLLER
      COMMON/CNTRL/ P(8,8), BASIS(8), XK(8), TEMP1(8), TP(8)
      DO 30 K4=1,8
      P(K4,K4)=20.0
  30  CONTINUE
      XK(1)= .1
      XK(2)= .1
      XK(3)= .1
      XK(4)= .1
      XK(5)= .1
      XK(6)= .1
      XK(7)= .1
      XK(8)= .1
      RETURN
      END
C
       SUBROUTINE CNTRL
       COMMON/CNTRL/ P(8,8), BASIS(8), XK(8), TEMP1(8), TP(8)
$
C
```

```
C   CALCULATE THE VELOCITY PATH
C
       IF (T .GT. TBRK)  GOTO 121
       STK = 50.0
       STK1= 50.0
       STK2= 50.0
        GOTO 122
  121  STK = 50.0 + 75.0*(T-TBRK)
       STK1 = 50.0 + 75.*(T-TBRK+TSAMP)
       STK2 = 50.0 + 75.*(T-TBRK+2*TSAMP)
       IF(STK2 .GT. 75.) STK2 = 75.0
       IF(STK1 .GT. 75.) STK1 = 75.0
       IF(STK  .GT. 75.) STK  = 75.0
  122  STPT = STK
C
C  UPDATE THE ESTIMATES   XK   EVERY 10 TSAMPs
C
       IF (KSWT .LE. 0 ) GOTO 102
C
C  JUMP DOWN TO THE CONTROLLER
C
       GOTO 103
 102  KSWT = 10
C  CALCULATE THE OBSERVED ACCELERATION
      AKM1 = (VEL - VEL2)/2/TSAMP
C  CALCULATE THE BASIS FUNCTION VECTOR
      WSQ=VEL1*VEL1
      BASIS(1) = AKM1
      BASIS(2) = AKM1*COS(Q1M)+WSQ*SIN(Q1M)
      BASIS(3) = AKM1*SIN(Q1M)-WSQ*COS(Q1M)
      BASIS(4) = AKM1*COS(2.*Q1M)+WSQ*2.*SIN(2.*Q1M)
      BASIS(5) = AKM1*SIN(2.*Q1M)-WSQ*2.*COS(2*Q1M)
      BASIS(6) = -SIN(Q1M)
      BASIS(7) = -COS(Q1M)
      BASIS(8) = VEL1
C
C  UPDATE THE PARAMETERS TO XK (K)
C
      CALL MULA(BASIS,P,TEMP1)
      CALL MLT(TEMP1,BASIS,SCLR)
      SCLR= 1./(SCLR+ROE)
      CALL MULP(P,BASIS,TP)
      CALL MLT(BASIS,XK,SCLR2)
       DO 10 K1=1,8
  10  XK(K1)= XK(K1)+ TP(K1)*SCLR*(TLK1-SCLR2)
C  OUTPUT THE ESTIMATED PARAMETERS
       WRITE(7,*) (XK(KKK),KKK=1,8),T
       STO = 00.00
C   UPDATE THE  P  MATRIX
       DO 20 K3=1,8
```

```
        DO 20 K2=1,8
  20  P(K3,K2)= (P(K3,K2)-SCLR*TP(K3)*TEMP1(K2))/ROE
C  ADD THE P MATRIX DIAGONAL ELEMENTS
          DO 33 KK2=1,8
 33       STO = STO + P(KK2,KK2)
          TRACE = STO
C
C DECREMENT THE ESTIMATION DELAY
 103  KSWT = KSWT - 1
C OUTPUT THE DELAYED TORQ OUTPUT
      TL=TLHOLD
C
C  CALCULATE THE STEP AHEAD CONTROL LAW
C
       Q1P = Q1 + TSAMP*VEL
       NTAB = INT(Q1P/.0349)
       Q1P  = FLOAT(NTAB) * 0.0349
       RINK1=XK(1)+XK(2)*COS(Q1P)+XK(3)*SIN(Q1P)+XK(4)*COS(2*Q1P)
     &         +XK(5)*SIN(2.*Q1P)
       DINER= -XK(2)*SIN(Q1P)+XK(3)*COS(Q1P)-XK(4)*2.*SIN(2*Q1P)
     & +XK(5)*2.*COS(2.*Q1P)
       TGRV=XK(6)*SIN(Q1P)+XK(7)*COS(Q1P)
C
       CGAIN = 2.*TSAMP/RINK1/((2.*TSAMP/RINK1)**2+EPSL)
       D9=2.*TSAMP/RINK1
      ERR = STK2-(VEL/D9-XK(8)*STK1-DINER*STK1**2-TGRV)*D9
      TLHOLD=CGAIN*ERR
C LIMIT THE MOTOR POWER
       IF (TLHOLD .GT. APWR) TLHOLD = APWR
       IF (TLHOLD .LT.-APWR) TLHOLD =-APWR
C SAVE "SNAP-SHOT" OF MECHANISM STATE
      VEL2 = VEL1
      VEL1 = VEL
      TLK1 = TORQ
       Q1M=Q1
C
      RETURN
      END
C
C   SUBROUTINES  PERFORM THE REQUIRED MATRIX MATH
C
      SUBROUTINE MULA(AFCTR,FCTR,ANS)
C MULTIPLIES A 1x8  AND A   (8x8)
      DIMENSION AFCTR(8), FCTR(8,8), ANS(8)
       DO 10 KR=1,8
      STO = 0.0
        DO 20 KC=1,8
  20  STO = AFCTR(KC)*FCTR(KC,KR) + STO
  10  ANS(KR)= STO
      RETURN
```

```
      END
C
      SUBROUTINE MLT(FTR1,FTR8,ANS)
C THIS MULTIPLIES A (1x8) TO A (8x1)  RETURNS A SCALAR
      DIMENSION FTR1(8), FTR8(8)
      ANS=0.0
       DO 10 KR=1,8
  10  ANS=ANS+FTR1(KR)*FTR8(KR)
      RETURN
      END
C
      SUBROUTINE MULP(FTR1,FTR8,ANS)
C MULTIPLIES A (8x8) TO A (8x1)
      DIMENSION FTR1(8,8), FTR8(8), ANS(8)
       DO 10 KR=1,8
      STO =0.0
        DO 80 KC=1,8
  80  STO= STO+ FTR1(KR,KC)*FTR8(KC)
  10  ANS(KR)=STO
      RETURN
      END
```

The vita has been removed from the scanned document

ADAPTIVE CONTROL OF A FOUR-BAR LINKAGE

by

Stephen O. Carlson

(ABSTRACT)

Three discrete-time adaptive controllers are developed and applied to Four-bar linkage velocity control to reduce the input link velocity fluctuations without compromising the control system velocity transient response. The successful control techniques use the known mechanism kinematics and the mechanism input link position to control the nonlinear mechanism dynamics. The study shows that the adaptive controls are feasible to implement using current microprocessor technology, and the velocity control performance is improved when compared to an industry-standard analog servomotor control. However, more development is required to realize the full potential of the adapative control technique.

A nonlinear Four-bar dynamic model is developed using Kinematic Influence Coefficients. This model is used to develop the adaptive controls and to computer simulate the control scheme performances. The simulated model velocity response is compared qualitatively to experimental data and shown to be similar to an experimental device.