Model reference adaptive control of a two axes hydraulic manipulator

Gomes De Almeida, Fernando

Award date:
1993

Awarding institution:
University of Bath

Model Reference Adaptive Control

of a

Two Axes Hydraulic Manipulator

submitted by Fernando Gomes de Almeida

for the degree of PhD

of the University of Bath

1993

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Synopsis

Due to the highly non-linear nature of robot dynamics, there is scope to improve performance through the use of adaptive algorithms. One of the most promising types of algorithm for this kind of application is the model reference based controller. For a robot using hydraulic direct-driven axes, in a decentralized adaptive control scheme, each axis can be modelled as a third order time-varying SISO system subjected to disturbances from other axes. The use of a decentralized controller has the advantage of a linear increase of computational work with the number of axes.

The classical method used in the synthesis of model reference adaptive control systems is hyperstability theory. As this method is based on a positivity condition, unmodelled plant dynamics, particulary in the presence of external disturbances, may lead to the drift of the adaptive controller gains, and so, to instability.

A novel synthesis method, based on the theory of variable structure systems (VSS), is developed in this work. The resulting controller employs memory-less adaptive gains, so they cannot drift, and an auxiliary signal is used to counteract the effects of external disturbances. A series-parallel model for the dynamic behaviour of the model-following error is used in order to circumvent the order

reduction that always happens with a classic VSS controller. This technique avoids the undesirable chattering phenomena associated with VSS controllers.

The new method is used in the synthesis of a decentralized controller for a two axes hydraulic manipulator. Due to its nature the system presents a free unity gain integrator between speed and position enabling the use of a second order adaptive velocity controller in an inner loop and a fixed outer loop for position control. An extensive set of simulation and experimental results is presented, showing the effectiveness of the proposed controller.

Acknowledgements

I would like to express my deepest appreciation to Prof. Kevin Edge, my supervisor, for all the invaluable help, support and advice he provided me, as well as all the enlightening discussions we had, that were vital for the accomplishment of this work.

I am also grateful to Dr. Francisco Freitas of the Universidade do Porto for his help and all the logistic support he provided me during the course of this work.

I also want to thank Dr. Will Richards for his advice on some mathematically related subjects, as well as to Peter Prest for the design of the resolver-to-digital converter cards.

I cannot forget all the help that the technicians of the workshop, and the electronics and fluid power laboratories gave me in the actual building and commissioning of the experimental manipulator used in this work.

To my colleagues Jonathan Gamble, Anthony Mercer, Andrew Plummer and Markus Gubeli I want to thank for all the companionship, help, discussions and social integration they provided me during the periods I was living in Bath.

I want to thank the Calouste Gulbenkian Foundation for all the financial support they provided, making this work possible. I also want to express my recognition of the work of Miss Maria Antónia da Silva and her staff of the Portuguese Affairs Department.

I want to dedicate this work to Angela, my wife, and ask for her forgiveness for all the time I could not give her.

Table of Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix A

Notation

a_i - elements of the last row of $(A_m - A_p)$

A_m, A_p - model, plant coefficient matrices

b_m, $\boldsymbol{b}_m$, B_m - model input gain, gain vector, gain matrix

b_p, $\boldsymbol{b}_p$, B_p - plant input gain, gain vector, gain matrix

$\boldsymbol{c}$ - vector of coriolis and centripetal torques

d, $\boldsymbol{D}$ - linear time-invariant filter vector, matrix

$\boldsymbol{e}$ - generalized state error vector

$\boldsymbol{e}_r$ - reference state error vector

E_p - plant external disturbance input matrix

$\boldsymbol{g}$ - vector of gravity torques

$\boldsymbol{g}_p$, $\hat{\boldsymbol{g}}_p$ - actual, estimated vectors of plant external disturbances

h, $\boldsymbol{h}$ - disturbance cancelation signal, signal vector

$\boldsymbol{I}$ - identity matrix

$\boldsymbol{I}^*$ - a vector, $\boldsymbol{I}^{*T} \triangleq [0\ 0\ \cdots\ 0\ 1]$

$\boldsymbol{J}$ - inertia matrix

K, k - scalar gain

k - sample number

K_g - external disturbance feedforward matrix

k_u, K_u - input gain, gain matrix

k_x, K_x - state feedback gain vector, gain matrix

l - auxiliary variable for the sample number

l_1, l_2 - lengths of links 1 and 2

l_{c1}, l_{c2} - distances between the centres of mass and the joints of links 1 and 2

n - system order

m - number of system inputs

m_1, m_2 - masses of links 1 and 2

s - Laplace operator

t - time

T - sampling period

u_m, $\boldsymbol{u}_m$ - model input signal, signal vector

u_p, $\boldsymbol{u}_p$ - plant input signal, signal vector

v, $\boldsymbol{v}$ - filtered generalized state error, error vector

v - distance of the state error to a datum sliding surface

v_r - distance of the reference state error to a datum sliding surface

v_e - difference between the reference and the actual state errors

w, $\boldsymbol{w}$ - non-linear time-varying feedback block output signal, signal vector

$\boldsymbol{x}_m$, $\boldsymbol{x}_p$ - model, plant state vectors

z - external input signal vector to the generalized error feedback system

α - adaptation gain, discontinuous components of an adaptive gain

β - constant component of an adaptive gain

γ - finite positive constant. $-\gamma^2$ is a lower bound of the Popov integral or summation

ε - error bound

ζ - damping ratio

η - value of the Popov integral or summation

θ - angular position

μ - thickness of the boundary layer around the switching surface

τ - torque, auxiliary variable for time

φ - external disturbances referred to the plant input

ϕ - elements of Φ

Φ - gain adaptation matrix functional

ω - angular frequency

ω_n - natural frequency

Subscripts and Superscripts

I - integral action

i, j, n - integer numbers

P - proportional action

T - transpose

$\dagger$ - left Penrose pseudo-inverse; $B_p^{\dagger} = (B_p^T B_p)^{-1} B_p^T$, which exists if $B_p^T B_p$ is a nonsingular matrix. $B_p^{\dagger}$ has the property that $B_p^{\dagger} B_p = I$

Chapter 1 | Introduction

The development of production techniques and the introduction of flexible manufacturing concepts puts a harder demand on automation systems. One of the most challenging tasks to perform is the flexible handling of materials and tools. It was the need to perform this task using a programmable and flexible machine (one that is not dedicated to a single manipulation task, but can be programmed to perform a broad class of manipulation tasks) that was on the origin of the industrial robot.

The word *robot* originated from the Czech word *robota*, meaning work. It was introduced into the English language in 1921 by the play-wright Karel Capek in his satirical drama, *R.U.R.* (Rossum's Universal Robots) [1]. In this work, robots are machines that resemble people, but work tirelessly.

Nowadays industrial robots are used in materials handling, like loading and unloading machine-tools, tool changing, assembly, welding, painting, and many other jobs that could only be performed by human operators, sometimes in situations of high risk for his or her physical and psychological integrity, and in many cases with a better performance (faster, with better repeatability, without fatigue problems, etc.)

Although the structure of a robot can vary considerably, most of them present the following four major components [2]:

1. The Manipulator. A collection of mechanical linkages connected by joints to form an open-loop kinematic chain. The manipulator is capable of movement in various directions and is said to do "the work" of the robot. Some times the terms "robot" and "manipulator" are used with the same meaning, although, strictly speaking, this is not correct.

2. Sensory Devices. These are the elements used to inform the controller about the status of the manipulator. Sensors can provide instantaneous position, velocity, acceleration, force, or other physical quantities as feedback information about the links to the control unit to produce the proper control of the mechanical system. Other devices such as video cameras can be used for tracking, object recognition or object grasping in a higher level control loop.

3. The Controller. This device (commonly microcomputer based) is responsible for the motion control of the manipulator, the storage of position and sequence data in its memory and the interface of the robot with other devices that cooperate with the robot in the task being performed. Controllers must do the necessary arithmetic computations for determining the correct manipulator path, speed and position. They must also send signals to the joint actuators and use the information provided by the robot's sensors. They are also responsible by the communication between peripheral devices and the manipulator.

4. The Power Conversion Unit. The purpose of this part of the robot is to provide the necessary energy to the actuators of the manipulator. It can be a power amplifier in the case of systems actuated by electric servo-motors, or a hydraulic power-pack or a compressor when hydraulic or pneumatic actuators are used.

An industrial robot is a general-purpose, computer-controlled manipulator consisting of several rigid links connected in series (although there is some research work being done on parallel connected links) by revolute or prismatic joints. One end of the chain is attached to a supporting base, while the other end is free and equipped with a tool to manipulate objects or perform assembly tasks. The

manipulator is composed of an arm and a wrist subassembly plus a tool (end effector). The arm subassembly generally can move with three degrees of freedom (DOF). The combination of its joints movements positions the wrist unit within its work volume. The wrist subassembly unit usually consists of three rotary motions. The combination of these motions orients the tool as needed. Hence, for a six DOF manipulator the arm subassembly is the positioning mechanism, while the wrist subassembly is the orientation mechanism.

The need for higher operating speeds has lead to the current trend for direct drive robots. In addition greater payloads are easily attained with the use of hydraulic actuators.

Why are direct drive manipulators potentially faster? The main factor affecting the cycle time of a given manipulation task is the acceleration capability of the manipulator links. So, if this capability can be increased, the cycle time can, potentially, became faster. To accomplish this, for a given actuating torque on the link, the choice between a small geared motor and a bigger direct-drive unit must be the one that minimizes the moment of inertia reflected on the link. Empirically, the relation between torque (T) and moment of inertia of a motor (J) can be given by [3]:

$$J = kT^p \tag{1.1}$$

with the parameters k and p dependent of the actuator technology.

The ratio of reflected inertias on the link for the same link actuating torque is:

$$\frac{n^2 J_{geared}}{J_{direct}} = \frac{n^2 k(T/n)^p}{kT^p} = \frac{n^2}{n^p} \tag{1.2}$$

where n is the gear ratio. The reducer inertia is considered small compared with the others and, as such, is not taken into consideration.

For gear ratios greater or equal to unity, the minimum of this function is obtained with unity gear ratio (direct-drive) as long as p is less than 2. Typical values of p are around 1.3 for electrical servo-motors and 1.5 for hydraulic ones. So, to maximize the link acceleration capability for a given link actuating torque, direct-drive actuators should be used.

Other important advantages of the use of direct-drive actuators are:

- Simpler design due to the elimination of the mechanical transmissions.

- Elimination of backlash, transmission induced compliance and mechanical noise (which can excite high-frequency structural vibration modes) due to non ideal transmission components (that present clearances, eccentricities, friction, etc.), many times the cause of degraded manipulator performance.

- Greater power efficiency (manipulator mechanical transmissions present efficiency values around 80%).

The direct-drive actuation of manipulators also presents its own set of problems such as:

- Increase in the control difficulty due to a greater sensitivity to manipulator configuration and payload induced inertial changes.

- The actuators of the links nearer the base must provide the torque necessary to support the weight of the following ones (in the direction of the end-effector) and their reaction torques.

- The actuators used to provide the necessary high torque for direct-drive operation are heavier and bulkier than the set of actuator plus reducer used in geared drives, specially in the case of electrical actuation.

Hydraulic actuators are usually lighter and smaller than electric ones, more so in the high torque range as needed for direct-drive operation. Meanwhile the control problem becomes more difficult when using a hydraulic servo-drive given its highly non-linear behaviour and because the actuators dynamics cannot be neglected as is usually the case with electric actuators.

The dynamic behaviour of a manipulator can be described by [4]:

$$\tau = \boldsymbol{J}(\theta)\ddot{\theta} + \boldsymbol{c}(\theta,\dot{\theta}) + \boldsymbol{g}(\theta) \tag{1.3}$$

where θ is the vector of link positions, $\boldsymbol{J}$ is the inertia matrix of the manipulator, $\boldsymbol{c}$ is a vector of coriolis and centripetal torques, $\boldsymbol{g}$ is a vector of gravity torques and τ is the vector of the links actuating torques.

In the particular case of a two DOF manipulator with rotary joints (see fig.1.1), corresponding to the one that will be used in this work,

$$\tau_1 = J_{11}\ddot{\theta}_1 + J_{12}\ddot{\theta}_2 - h_c\dot{\theta}_2^2 - 2h_c\dot{\theta}_1\dot{\theta}_2 + g_1 \tag{1.4}$$

$$\tau_2 = J_{21}\ddot{\theta}_1 + J_{22}\ddot{\theta}_2 + h_c\dot{\theta}_1^2 + g_2 \tag{1.5}$$

where

$$J_{11} = m_1 l_{c1}^2 + I_1 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos\theta_2) + I_2 \tag{1.6}$$

$$J_{22} = m_2 l_{c2}^2 + I_2 \tag{1.7}$$

$$J_{12} = m_2 l_1 l_{c2}\cos\theta_2 + m_2 l_{c2}^2 + I_2 \tag{1.8}$$

$$h_c = m_2 l_1 l_{c2}\sin\theta_2 \tag{1.9}$$

$$g_1 = m_1 l_{c1} g \cos\theta_1 + m_2 g (l_{c2} \cos(\theta_1 + \theta_2) + l_1 \cos\theta_1) \tag{1.10}$$

$$g_2 = m_2 l_{c2} g \cos(\theta_1 + \theta_2) \tag{1.11}$$

where I_1, I_2 are the centre of mass referred moments of inertia, m_1, m_2 are the masses, l_1, l_2 are the lengths and l_{c1}, l_{c2} are the distances between the centres of mass and the joints of respectively link 1 and link 2.

In the case of direct-drive actuation these are the torques and, what is more significant when the manipulator is moving, the torque variations that must be provided by the actuators.

The hydraulic servo-drive can be modelled after linearization, as presented in Appendix A, by a third order transfer function from servo-valve spool position (x_v) and disturbance torque (τ_L) to link position (θ):

$$\mathcal{L}(\theta) = \frac{K_{xv}\omega_n^2}{s(s^2+2\zeta\omega_n s+\omega_n^2)} \mathcal{L}(x_v) + \frac{K_L(as+1)}{s(s^2+2\zeta\omega_n s+\omega_n^2)} \mathcal{L}(\tau_L) \tag{1.12}$$

with all the transfer function parameters strongly dependent on the position and velocity of the link, and the coupling and gravity effects represented by the disturbance torque.

A hydraulic manipulator can, as such, be modelled as a set of third-order time-varying systems, one per link with its position dependent moment of inertia (the diagonal terms of the inertia matrix), perturbed by inter-axis inertia couplings, centripetal, coriolis and gravity torques.

Robot control can be divided into three levels [5]:

- strategic
- tactical
- executive

At the strategic level the task that the robot has to do is analyzed, leading to the definition of the required trajectory of the end effector due to functional aspects, avoidance of collisions with the environment and other robots, etc. This should not be specified in a too rigid manner, or else any scope for optimization will disappear. This stage is done off-line and is a heavy user of simulation tools.

At the tactical level the desired end effector trajectory is translated into axes trajectories. Optimization of time response, energy consumption or any other desired criteria may be performed taking into consideration the saturation limits of the actuators. This stage can be done off-line or in real time. If it is done off-line, a reference vector must be recorded for each sampling interval (thinking in terms of digital control), for the time interval the resulting trajectory takes to be performed. This usually leads to the creation of a big information array, needing large amounts of storage memory. If this task is to be run in real-time, enough processing power is needed to run the kinematic (or dynamic, if optimization is to be done) equations of the robot model in real-time. This is usually a centralized process, but some authors [5] propose decentralized ways of doing it.

At the executive level the robot is driven in such a way that it performs according to the demands of the other levels. In essence, it is a more-or-less sophisticated servo-controller. This real-time process can be centralized or decentralized. Centralized algorithms usually use the robot model in order to compute the required control actions. These are nominal actions because imperfections of the model and payload variations make it impossible to control a robot in open-loop. There is always some sort of feedback (position, velocity, acceleration, force, etc.) closing the loop and increasing the robustness of the controller as shown in fig.1.2. A controller of this type must run the robot dynamic model in real-time, which represents, computationally, a very heavy burden. The use of decentralized controllers aims to solve this problem. Each axis is controlled independently from the others, and the dynamic interactions are treated as disturbances. An advantage of using decentralized controllers is that this decentralization can be easily mirrored by the hardware implementation, resulting in a parallel system

with one microprocessor controller per axis. Even if only one processor is used to implement all the axis controllers, the use of decentralized controllers has the advantage of offering proportional scaling between number of axes and computational load.

Presently the executive level control of industrial robots is implemented using decentralized joint controllers which control the joint angles independently through conventional fixed gain servo-loops. Generally these controllers provide an acceptable level of performance due to the fact that most industrial robots are indirectly driven. The gearing mechanism used to increase the motor torque has a typical gear ratio in the order of 100:1. As the inertial changes due to configuration and payload variations vary inversely with the square of the gear ratio, when referred to the motor shaft, the effect of these changes is negligible. Without gear reduction, the conventional fixed gain servo-loop is no longer capable of producing acceptable performance [6]. Moreover, as said before, when using hydraulic actuators their dynamic behaviour cannot be neglected. This means that more advanced controllers must be used.

Due to the highly non-linear nature of robot dynamics, there is scope for the use of adaptive algorithms. One of the most promising types for this kind of application is the model reference based controller. For a robot using hydraulic direct-driven axes, in a decentralized adaptive control perspective, each axis can be modelled as a third order time-varying SISO system. This is the perspective to be explored in this work. Plus, if the adaptive controller performs well, it is possible to build in front of the controlled axis system a filter with an inverted transfer function of the axis dynamics. Differentiation of the reference signal may be avoided if the tactical controller delivers the necessary information reference [7], that is, the complete desired state vector, plus the derivative of the higher order state variable. In this way there is some scope of reducing the axis dynamics to virtually a gain, and maybe a delay, as long as the frequency content of the reference signal is inside the bandwidth of the system.

A brief outline of the content of this work is as follows:

In the next chapter a discussion on adaptive controllers in general will be presented. The two fundamental approaches, self-tuning and Model Reference Adaptive Control (MRAC), are introduced. This is followed by a detailed review of model reference adaptive controllers. The Perfect Model Following (PMF) conditions are deduced and the concept of error system is presented. A new PMF condition, the disturbance (or load) invariance condition is introduced. The most commonly used synthesis methods are presented and briefly discussed, as well as the ideal design hypotheses and their most frequent violations.

Chapter 3 presents in detail "the" classical method of Model Reference Adaptive Control system design: Hyperstability theory. The concept of a external disturbances rejection signal is introduced in the classic design. This is followed by a discussion about the robustness properties of the obtained controller: the problems associated with the presence of unmodelled dynamics that lead to the drift of the adaptive gains are presented.

From that discussion the need of a robust design method emerges. Chapter 4 starts with an overview of this method, Variable Structure Systems (VSS) theory. VSS theory is then applied to the development of a new robust MRAC algorithm with memory-less adaptation gains and an external disturbances cancelation signal. The use of a series-parallel model of the error behaviour enables smooth control actions and a definable adaptation bandwidth.

A simulation case study is described in Chapter 5. The new adaptive control algorithm is applied in the control of a two axes directly driven hydraulic manipulator. Controller design is presented in detail. The simulation results obtained are displayed and the adaptive controller performance is evaluated. A brief simulation study using a state feedback controller is presented for comparison purposes.

In Chapter 6 the new algorithm is implemented as the controller of a purposely built two axis manipulator. The manipulator is directly driven by hydraulic rotary actuators. Controller design and implementation are covered in detail. The numerical benefits of the use of the delta operator in the discrete implementation of the controller filters are discussed. The obtained experimental results are presented and discussed.

A detailed overview of this work is presented in Chapter 7. Some general conclusions on the effectiveness of the developed adaptive control algorithm are drawn, taking into account the simulation and experimental results obtained.

The model of a servo-valve driven hydraulic rotary actuator is developed in Appendix A. The mathematical deductions of the continuous and discrete time hyperstable controllers discussed in Chapter 3 are presented in Appendix B and C respectively. These results are the starting point for the discussion and development of the new MRAC algorithm.

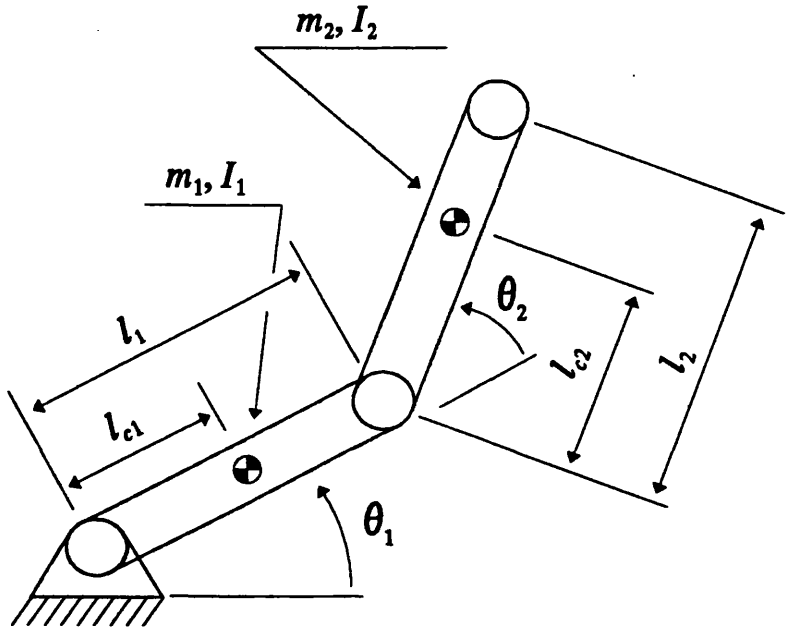


Figure 1.1 - Manipulator diagram

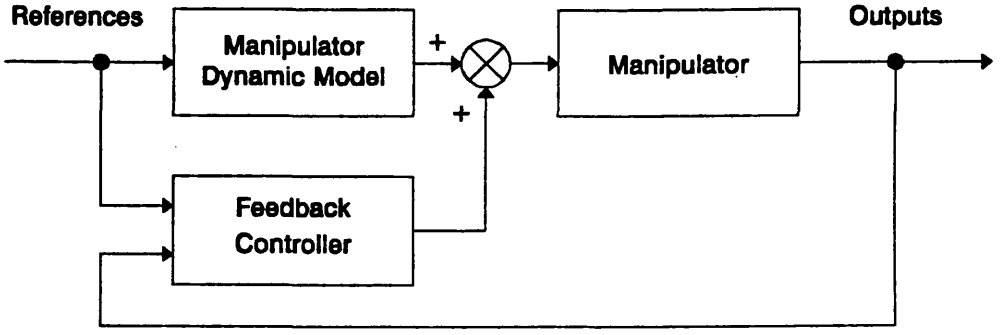


Figure 1.2 - Block diagram of a model based manipulator controller

Chapter 2 | A Review of Adaptive Control

2.1 Adaptive Control

It is difficult to give a definition of an adaptive control system and there is no definitive agreement in the literature about this subject. Landau [8] proposes what seems to be a useful one:

"An adaptive system measures a certain index of performance (IP) using the inputs, the states and the outputs of the adjustable system. From the comparison of the measured index of performance (IP) values and a set of given ones, the adaptation mechanism modifies the parameters of the adjustable system or generates an auxiliary input in order to maintain the index of performance (IP) values close to the set of given ones."

Within adaptive control theory there are two fundamental approaches. The first is based on the identification of a model of the plant by on-line parameter estimation techniques. This model is then used, also on-line, in the synthesis of a controller. The well-known self-tuning control strategy is an example of this type of controller (fig. 2.1). The second approach in adaptive control theory is called Model Reference Adaptive Control (MRAC). The controller is adjusted so that the dynamic behaviour of a system matches that of a preselected model (fig. 2.2). These two approaches can be shown to be in some cases equivalent [9].

Self-tuning controllers, also known as indirect adaptive controllers, use the input and output plant signals to estimate a plant model. One of the most commonly used estimation algorithms is recursive least squares. This method updates the model parameter estimates in a way to minimize the sum, over time, of the squares of differences between the actually observed and computed values of the model output [10]. In order to ensure the convergence of the parameters of the estimated model to the plant parameters, the plant must be persistently excited. If not, some undesirable behaviour, such as controller "bursting", may occur. A controller is then designed, on-line, based on the estimated model using the certainty equivalence principle, that is the control is determined as if the estimated model is equal to the true model. For servo problems a controller based on pole-zero placement may be used. This type of controller is very well suited to the servo problem given the easiness of specification of the desired time behaviour of a system by its poles and zeros location on the complex plane. A review of self-tuning controllers based on pole-zero placement can be found in Åström and Wittenmark [11]. An application of this type of controller to hydraulic servo-systems is presented by Vaughan and Plummer [12].

A self-tuning controller at each time step updates the model of the plant; a controller based on this model is designed and the control action is computed. In most cases, there is no other use for the estimated model of the plant but to design the controller (exceptions do exist, however, as for example the use of the plant model for condition monitoring purposes). So, if the adaptive control problem is reformulated in a way that the controller is directly updated, the number of numerical calculations needed on each time-step can be significantly reduced. An example of this kind of controller is the model reference adaptive controller [13].

Model reference adaptive controllers lead to relatively easy-to-implement systems with a high speed of adaptation. An auxiliary dynamic system, the reference model, is excited by the same external inputs as the plant-plus-controller system ("controller" in this context being the adjustable feedback and

feedforward filters), the adjustable system. This reference model is used to generate the desired dynamic behaviour of the adjustable system. The difference between the state of the reference model and that of the adjustable system is used by the adaptation algorithm to modify the parameters of the controller or to generate an auxiliary input signal in order to minimize this difference. In some cases both approaches can be used at the same time [14].

One of the most important advantages of this type of adaptive controller is its high speed of adaptation. This is due to the fact that a measure of the adjustable system performance (the IP) is obtained directly from the comparison of its state with the one of the reference model. As a counterpart, a certain a priori knowledge of the structures of the model and of the adjustable system is necessary for its implementation. Examples of applications of this type of adaptive controller are presented by Courtiol and Landau [15] for electromechanical systems and Edge and Figueredo [16,17] for electrohydraulic servo-systems. It has also been widely used as a centralized manipulator controller as described by, for example, Dubowsky and DesForges [18] and Nicosia and Tomei [19].

2.2 The Model Reference Approach

Classic optimal control techniques are based on finding an optimal control action that minimizes a quadratic index of performance. One of the greatest problems on the use of this technique is the choice of the weighting matrices that are used to compute the index. It is very difficult to find a correlation between the usual dynamic performance indices like rise time, overshoot, and damping and the weighting matrices. This problem can be avoided if the desired behaviour of the system is supplied in the form of a reference model. This model can be used implicitly or explicitly. In the former category the model is only used for the calculation of the control law. In the latter category the reference model is a part of the controller itself. Model Reference Controllers are in the second category.

Model Reference Controllers are based on the solution of the Perfect Model Following problem. In a linear time invariant system with known coefficients the solutions proposed by Erzberger [20], Wang and Desoer [21] and Morse [22] can be used. The control systems that result from the application of these solutions are known as Linear Model Following Control (LMFC) systems. The performance analysis of the behaviour of LMFC systems when in the presence of poor knowledge about the plant parameter values or their variation, leads to the conclusion that in order to assure the desired performance, expressed by the reference model, an adaptive controller must be used.

2.2.1 The Perfect Model Following Conditions

In order to use a Model Reference Adaptive Controller (MRAC) some conditions relating system and model structure must obeyed. These are the Perfect Model Following (PMF) conditions. The PMF conditions impose some restrictions on the choice of the model as well as on the choice of the state variables used to represent system and model behaviour. These are the structural conditions upon which it is possible for a plant to behave exactly as a given reference model. To evaluate these

conditions, PMF is actually supposed. Then, when plant and model are equated the PMF conditions can be calculated.

The model will be represented by:

$$\dot{x}_m = A_m x_m + B_m u_m \tag{2.1}$$

and the plant by:

$$\dot{x}_p = A_p x_p + B_p u_p + E_p g_p \tag{2.2}$$

with the control action

$$u_p = K_x x_p + K_u u_m + K_g \tilde{g}_p \tag{2.3}$$

where x_m and x_p are the model and plant state vectors, A_m and A_p are the model and plant coefficient matrices, u_m and u_p are the model and plant input vectors, B_m and B_p are the model and plant input matrices, g_p is a vector of external disturbances acting on the plant and E_p is the plant external disturbance input matrix. K_x is the state feedback matrix, K_u is the input gain matrix, K_g is the external disturbance feedforward matrix and $\tilde{g}_p$ is an estimated vector of external disturbances (see fig.2.3).

Assuming that

$$\dot{x}_m = \dot{x}_p \quad \text{and} \quad x_m = x_p = x \tag{2.4}$$

then,

$$0 = (A_m - A_p)x + B_m u_m - B_p(K_x x_p + K_u u_m + K_g \tilde{g}_p) - E_p g_p \tag{2.5}$$

$$0 = (A_m - (A_p + B_p K_x))x + (B_m - B_p K_u)u_m - E_p g_p - B_p K_g \tilde{g}_p \tag{2.6}$$

If $g_p \approx \tilde{g}_p$ or if the load can be measured, making $g_p = \tilde{g}_p$, then

$$0 = ((A_m - A_p) - B_p K_x)x + (B_m - B_p K_u)u_m - (E_p + B_p K_g)g_p \tag{2.7}$$

This means that matrices $(A_m - A_p)$, B_m and E_p must be reachable by the control action; that is, they must belong to the linear space defined by B_p (range of B_p). So,

$$\text{rank } B_p = \text{rank } [B_p \mid (A_m - A_p)] \tag{2.8}$$

$$= \text{rank } [B_p \mid B_m] \tag{2.9}$$

$$= \text{rank } [B_p \mid E_p] \tag{2.10}$$

These conditions, with a special emphasis to the last one, are also called the invariance conditions [23,24]. If these conditions are obeyed, by a proper choice of model structure and plant state variables (the choice of plant state variables is most of the time determined by the last invariance condition), then it is possible to have a model that specifies the desired state trajectory in a way independent (invariant) of changes on system dynamics and external disturbances (typically, a load).

Some "feel" can be gained about the meaning of this invariance to the load by looking at an example: in order to control the position and dynamic behaviour of an hydraulic servo-cylinder three state variables must be used, as the system has a third order transfer function. Restricting ourselves to the use of state variables that are physical quantities, two "natural" sets of state variables emerge: position, velocity and differential pressure, or position, velocity and acceleration.

As can be easily seen, the desired dynamic behaviour of the system translates to a desired behaviour of the position, velocity and acceleration, independently of any changes of the system or its load. The same cannot be said about the differential pressure, because it is not possible to specify a desired behaviour of this variable. The differential pressure must change with changes in the system and its disturbances in order to make the system behave in the desired manner.

It must be noticed that MRAC developments have traditionally disregarded the use of specific disturbance-rejection signals, and as such, disregard the need for the third PMF condition. This

condition has mostly been used within the framework of Variable Structure Systems (VSS). These two control strategies have in common the fact that they try to force the state of the system to follow a predefined trajectory (the model in the MRAC system case and the sliding surface in the VSS case) and as such they use the same invariance conditions. The choice of phase variables as state variables is a good one (as shown by the previous example) because it makes the system naturally obey the invariance conditions.

When the PMF conditions can be achieved, a MRAC system based on this approach can be built, having the advantage of not requiring an explicit identification of the plant parameters, and, as will be shown next, the adaptation obtained laws have an explicit form, and do not require as such the solution of a set of equations in real-time. A general block diagram of this type of controller is presented in fig.2.4.

2.2.2 The MRAC Synthesis Problem

There are several ways of synthesizing a MRAC system. Some popular ones are: gradient techniques, based on local parametric optimization theory, the use of Lyapunov functions, hyperstability theory and Variable Structure Systems (VSS) theory. All of these methods are based on the stabilization of a generalized error system composed by the model reference system and the adjustable system. The two most frequently used structures for the realization of the generalized error system are the parallel (fig.2.5) and the series-parallel (fig.2.6) model reference adaptive systems. The first one is a more general structure, usually used in tracking problems, while the second one is more oriented to regulation problems. Given that the control of a robotic manipulator is essentially a tracking problem, this work will focus on the use of parallel model reference systems.

The basic equations describing the generalized error system obtained by the combination of a parallel reference model with the adjustable system are:

- the reference model

$$\dot{x}_m = A_m x_m + B_m u_m \tag{2.11}$$

- the adjustable system

$$\dot{x}_p = A_p x_p + B_p u_p + E_p g_p \tag{2.12}$$

$$u_p = K_x x_p + K_u u_m + h \tag{2.13}$$

with h representing a disturbance cancelation signal.

- the generalized error system

$$e = x_m - x_p \Rightarrow \dot{e} = \dot{x}_m - \dot{x}_p \tag{2.14}$$

so,

$$\dot{e} = (A_m x_m - A_m x_p) + (A_m x_p - A_p x_p) - B_p K_x x_p + (B_m - B_p K_u) u_m - E_p g_p - B_p h \tag{2.15}$$

$$\dot{e} = A_m e + ((A_m - A_p) - B_p K_x) x_p + (B_m - B_p K_u) u_m - E_p g_p - B_p h \tag{2.16}$$

in which K_x, K_u and h are themselves functions of the generalized error, providing for the adaptation capabilities.

From the third invariance condition (equ.2.10) it can be said that

$$E_p g_p = B_p \varphi \tag{2.17}$$

where φ represents a vector of external disturbances when referred to the plant input.

So, the generalized error system can be described by:

$$\dot{e} = A_m e + ((A_m - A_p) - B_p K_x) x_p + (B_m - B_p K_u) u_m - B_p(\varphi + h) \tag{2.18}$$

Assuming that the reference model is stable, the MRAC system will try to stabilize (regulate) this generalized error system. To do this an adaptation algorithm must be used to find matrices K_x and K_u as well as the signal h in order to cancel the disturbances due respectively to the difference between model and plant dynamics (A_m - A_p), the input signal u_m and the difference between the model and plant input matrices (B_m - B_p), and the external disturbances φ.

At the beginning of this section several methods of MRAC systems synthesis were briefly mentioned. The first one, gradient techniques, of which the MIT rule is a well known example, can lead to instability when fast adaptation is desired [13]. The second one, the use of Lyapunov functions opens the question of how to chose the function that leads to the class of adaptation laws most suitable for a given application. Hyperstability and VSS theory offer an answer for a class (larger for the case of hyperstable systems) of stable adaptation algorithms overcoming the structural and design problems associated with the other two. They also offer an advantage inherent to stability methods: the conditions used to synthesize the controller are stability conditions, and as such the obtained controller is certainly stable.

Before discussing the problems related with the synthesis of an MRAC system one must define a set of design hypotheses corresponding to an "ideal" case (the "ideal" case presented here is based on the one presented by Landau [13] but relaxing his condition on external disturbances):

1 - The reference model is a time-invariant linear system.

2 - The reference model and the adjustable system are of the same dimension.

3 - All the parameters of the adjustable system are accessible for adaptation (in the case of parameter adaptation).

4 - During the adaptation process the parameters of the adjustable system depend only on the adaptation mechanism.

5 - The initial difference between the parameters of the model and those of the adjustable system is unknown.

6 - The generalized state error vector is measurable.

A problem associated with the use of hyperstability is that some of the design hypotheses used on the synthesis of an MRAC system are sure to be violated in a practical implementation:

- The dimension of the reference model is in many cases smaller than the adjustable system one (a physical plant has infinite order), violating condition 2.

- The plant is in many cases (as with the manipulator) nonlinear or time-varying which represents a violation of condition 4.

- The state of the plant is in many cases not completely measurable, violating condition 6.

This can lead in some cases, as will be seen in a following chapter, to the loss of stability. Nevertheless a great amount of research and development work has been done with this method, with some successful implementations. This together with the fact that the synthesis of a hyperstable MRAC system when modified in a way to increase its robustness (its capability to withstand violations of the design hypotheses) leads to a variable structure MRAC system, makes the study of hyperstable MRAC systems mandatory.

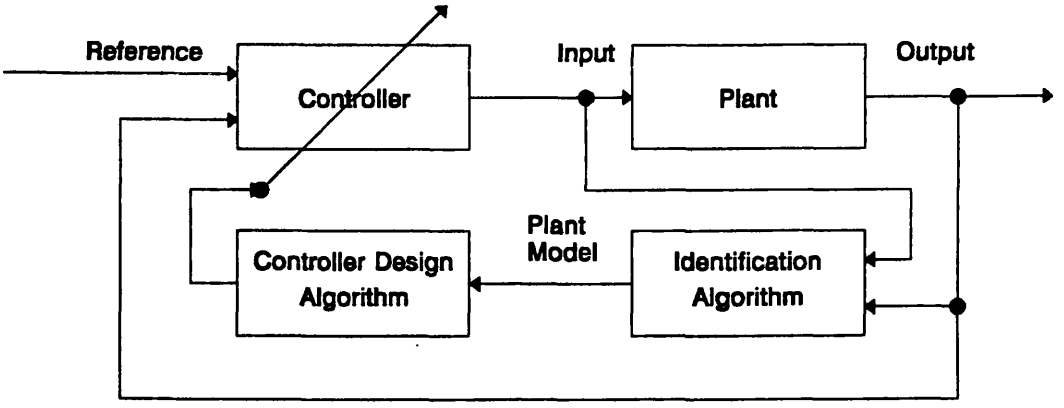


Figure 2.1 - Block diagram of a Self-Tuning Controller

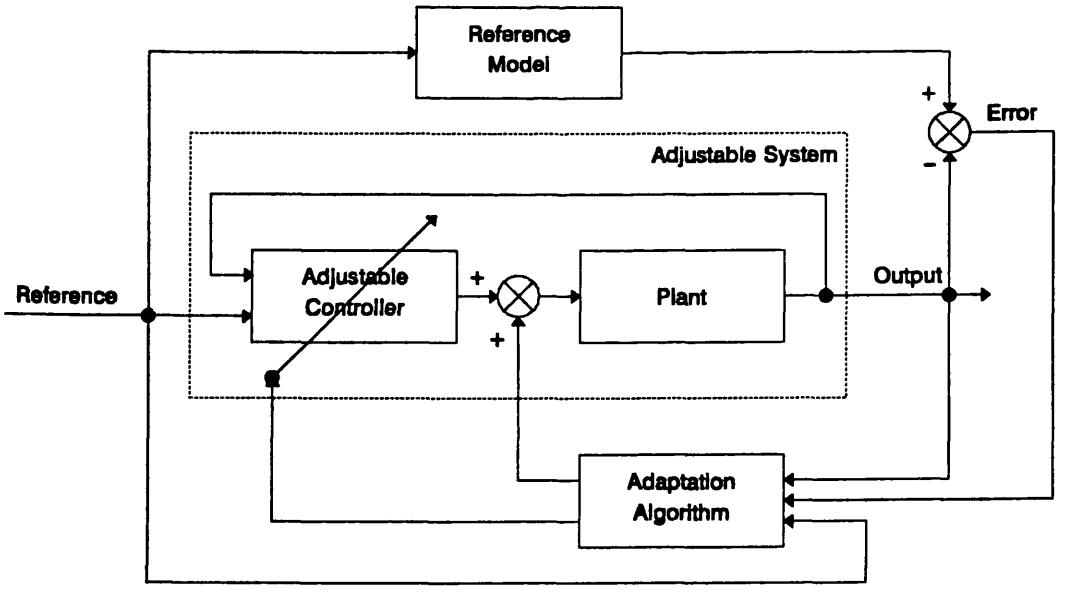


Figure 2.2 - Block diagram of a MRAC

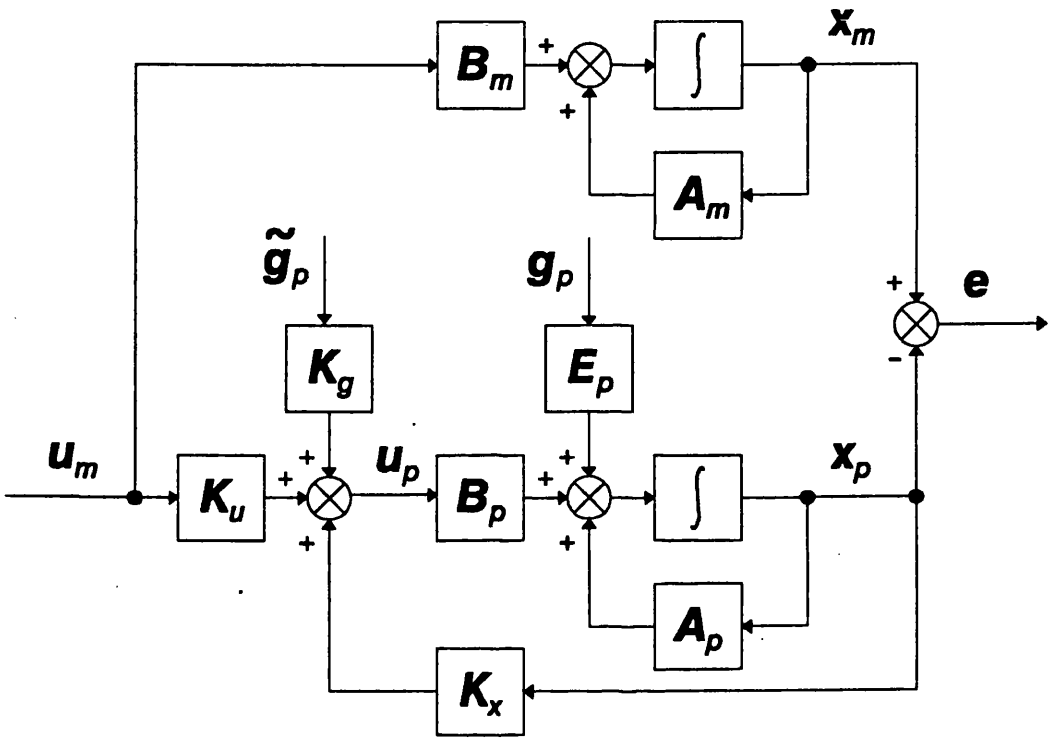


Figure 2.3 - Detailed block diagram of the reference model and the plant-plus-controller system

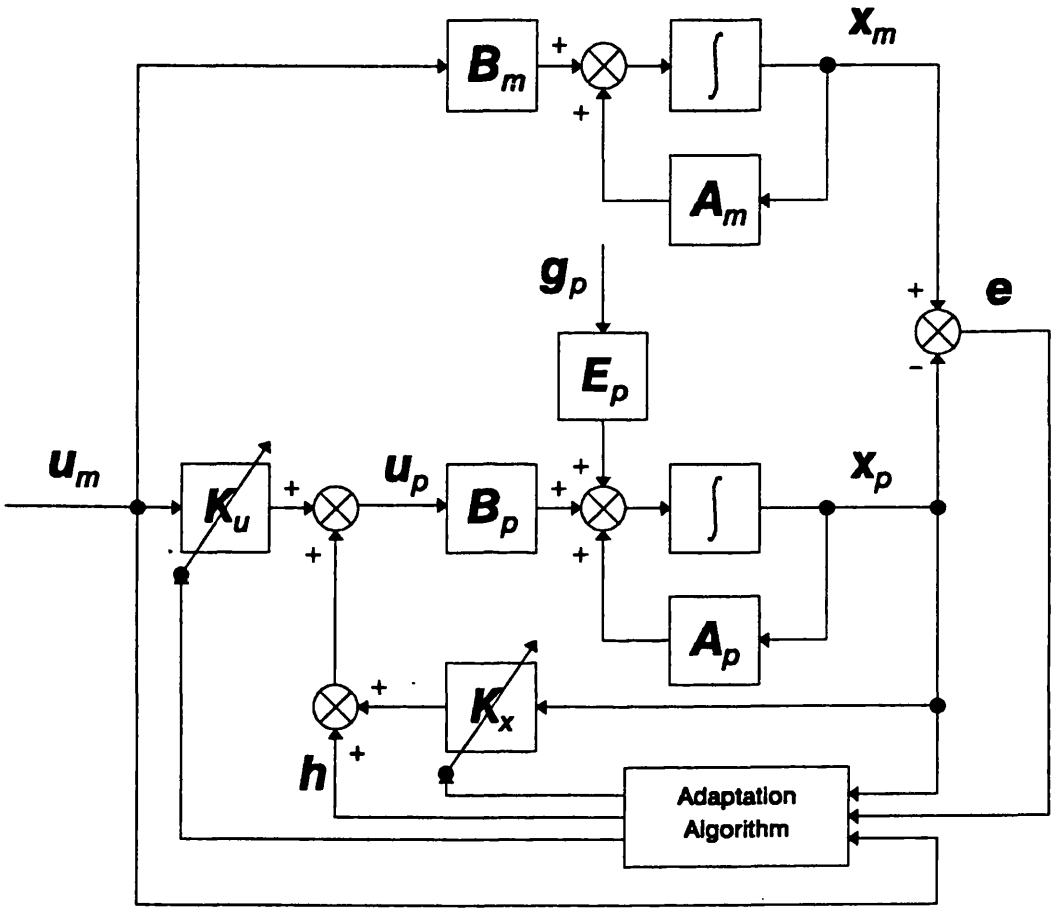


Figure 2.4 - General model reference adaptive controller

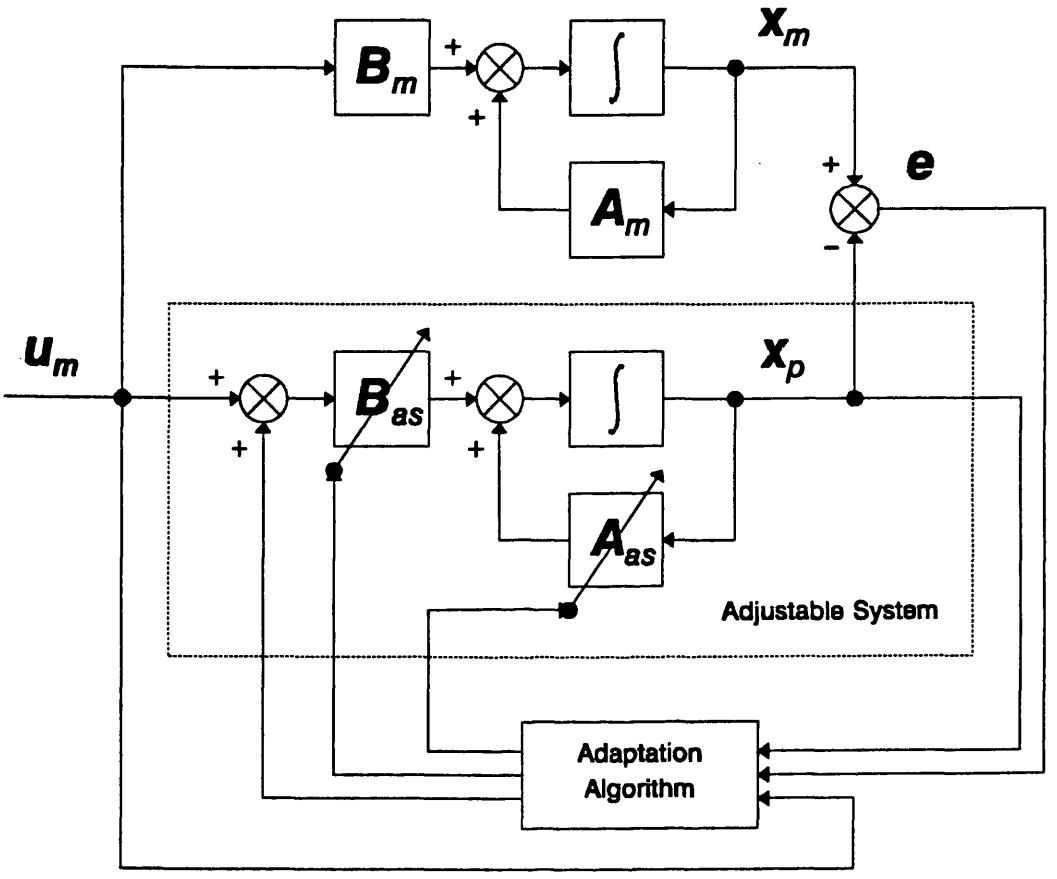


Figure 2.5 - Block diagram of a parallel MRAC

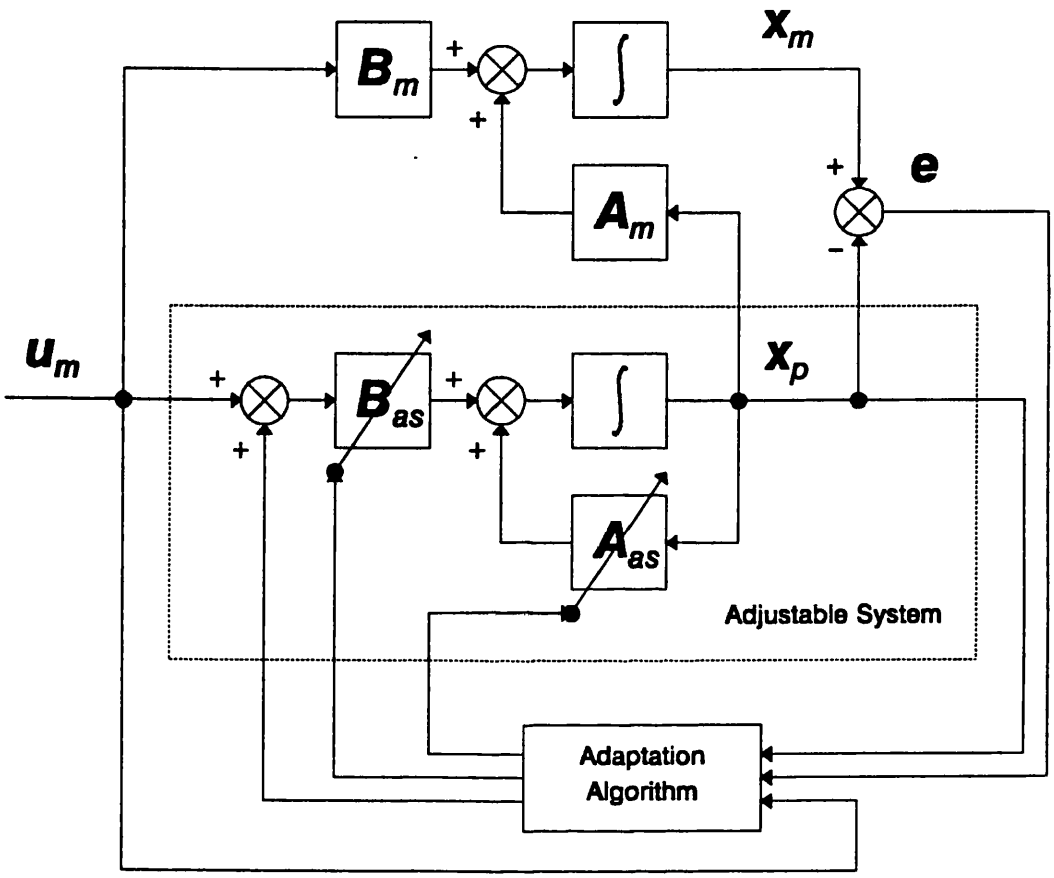


Figure 2.6 - Block diagram of a series-parallel MRAC

Chapter 3 | Hyperstable Model Reference Adaptive Controllers

3.1 Hyperstable Systems

Hyperstability theory is concerned with the stability properties of a class of feedback systems that can be decomposed into two blocks [13]: a linear time invariant feedforward block and a nonlinear time varying feedback block, as presented in fig.3.1.

The feedback block belongs to a class of systems satisfying an input/output relation of the form

$$\eta(0,t_1) \triangleq \int_0^{t_1} v^T w \, dt \geq -\gamma^2 \ , \ \forall t_1 \geq 0 \tag{3.1}$$

where v is the input vector of the block, w is its output vector, and γ is a finite positive constant. This integral inequality is known as the Popov integral (or summation in the discrete time case) inequality.

In order to make the feedback system an asymptotic hyperstable feedback system the feedforward block must have a strictly positive real transfer function, in the SISO case, or transfer matrix, in the MIMO case. For a transfer function, $h(s)$, to be strictly positive real, it must have the following properties:

1. $h(s)$ is real for real s.

2. The poles of $h(s)$ must lie in $\mathrm{Re}[s] < 0$.

3. For all real ω,

$$\mathrm{Re}[h(j\omega)] > 0 \ , \ -\infty < \omega < \infty$$

(the phase is always between plus and minus 90°)

A system obeying these conditions is said to be a hyperstable system, and as such, a stable one.

3.2 The Design of a Hyperstable MRAC System

In order to apply hyperstability theory to the design of a MRAC system, the time behaviour of the state error between model and plant (the generalized error) must be expressed by a hyperstable system. So, the generalized error system must be developed into a hyperstable loop. Then, applying the hyperstability conditions (solving the Popov integral), the adaptive laws can be found.

Following the method proposed by Landau [13], but with the introduction of some new terms taking external disturbances into account, the error system equation (equ.3.18) can be rewritten as:

$$\dot{e} = A_m e + B_p(B_p^{\dagger}(A_m - A_p) - K_x)x_p + B_p(B_p^{\dagger}B_m - K_u)u_m - B_p(\varphi + h) \tag{3.2}$$

This can now be developed into an hyperstable loop, producing a linear time invariant feedforward block

$$\dot{e} = A_m e - B_p w \tag{3.3}$$

$$v = De \tag{3.4}$$

and a nonlinear time varying feedback block

$$w = (K_x - B_p^{\dagger}(A_m - A_p))x_p + (K_u - B_p^{\dagger}B_m)u_m + (\varphi + h) \tag{3.5}$$

where $B_p^{\dagger}$ is the left Penrose pseudo-inverse of B_p and K_x, K_u and h are functions of time and v.

D is a linear time-invariant filter acting on the generalized error that must be defined in a way that the feedforward block transfer matrix

$$H(s) = D(sI - A_m)^{-1}B_p \tag{3.6}$$

is a strictly positive real transfer matrix. The resultant error loop is presented in fig.3.2.

To make this error loop a hyperstable one, the feedback block must fulfil the Popov condition. So, the following Popov integral inequality must be solved in order to find the adaptation laws:

$$\eta(0,t_1) \triangleq \int_0^{t_1} v^T((K_x - B_p^{\dagger}(A_m - A_p))x_p + (K_u - B_p^{\dagger}B_m)u_m + (\varphi + h))dt \geq -\gamma^2 \tag{3.7}$$

To solve this inequality all that must be proved is that the Popov integral is bounded from below, without any special conditions on the value of this bound ($-\gamma^2$).

The Popov inequality (equ.3.7) is true if each of the following inequalities is true:

$$\eta_1(0,t_1) \triangleq \int_0^{t_1} v^T(K_x - B_p^{\dagger}(A_m - A_p))x_p\, dt \geq -\gamma_1^2 \tag{3.8}$$

$$\eta_2(0,t_1) \triangleq \int_0^{t_1} v^T(K_u - B_p^{\dagger}B_m)u_m\, dt \geq -\gamma_2^2 \tag{3.9}$$

$$\eta_3(0,t_1) \triangleq \int_0^{t_1} v^T(\varphi + h)\, dt \geq -\gamma_3^2 \tag{3.10}$$

The third inequality (equ.3.10) results from the introduction of some new terms not included in Landau's method, in order to obtain a new component in the controller signal that will react against the effects of external disturbances.

The adaptation algorithm must now be chosen. Useful solutions of the Popov inequality are known for integral adaptation, integral plus proportional adaptation, integral plus relay adaptation, and integral plus proportional plus relay gain adaptation. The usually preferred type is integral plus proportional adaptation, resulting in controllers with fast adaptation capabilities and smooth control signals.

Using integral plus proportional gain adaptation, the following results are obtained (see Appendix B):

- For K_x,

$$K_x(t) = K_{x0} + \int_0^t \Phi_{1I}(v,t,\tau)\,d\tau + \Phi_{1P}(v,t) \tag{3.11}$$

$$\phi_{1Iij}(v,t,\tau) = \alpha_{xIij} v_i x_{pj} \ , \ \alpha_{xIij} > 0 \tag{3.12}$$

$$\phi_{1Pij}(v,t) = \alpha_{xPij}(t)\, v_i x_{pj} \ , \ \alpha_{xPij}(t) \geq 0 \ , \ \forall t \geq 0 \tag{3.13}$$

- For K_u,

$$K_u(t) = K_{u0} + \int_0^t \Phi_{2I}(v,t,\tau)\,d\tau + \Phi_{2P}(v,t) \tag{3.14}$$

$$\phi_{2Iij}(v,t,\tau) = \alpha_{uIij} v_i u_{mj} \ , \ \alpha_{uIij} > 0 \tag{3.15}$$

$$\phi_{2Pij}(v,t) = \alpha_{uPij}(t)\, v_i u_{mj} \ , \ \alpha_{uPij}(t) \geq 0 \ , \ \forall t \geq 0 \tag{3.16}$$

- For h,

$$h(t) = h_0 + \int_0^t \Phi_{3I}(v,t,\tau)\,d\tau + \Phi_{3P}(v,t) \tag{3.17}$$

$$\phi_{3Ii}(v,t,\tau) = \alpha_{hIi} v_i \ , \ \alpha_{hIi} > 0 \tag{3.18}$$

$$\phi_{3Pi}(v,t) = \alpha_{hPi}(t)\, v_i \ , \ \alpha_{hPi}(t) \geq 0 \ , \ \forall t \geq 0 \tag{3.19}$$

In all these equations the coefficients α represent adaption gains (the speed of adaptation increases with the value of the coefficients). The proportional adaptation gains can be made zero (making the adaptation algorithm of the integral type), but the integral adaptation gains must be time-invariant and strictly positive. A block diagram of the resultant system is presented in fig.3.3.

3.3 Some Considerations About Discrete-Time Hyperstable MRAC Systems

The discretization of continuous linear time-invariant systems does not usually present great difficulties. But when dealing with MRAC systems a more careful approach must be taken due to the fact that these are time-varying nonlinear systems and to the appearance of a delay in the adaptation loop. This delay is mainly a consequence of the time the controller needs to compute the control action (other delays like conversion delays are so small that they can be discarded). Two approaches can be taken when dealing with this problem in MRAC systems: one is to try to produce an algorithm in a way that the output of the controller at time kT depends only of data sampled in previous instants (k-1)T, (k-2)T,...,(k-n)T. In doing this the sampling time interval between (k-1)T and kT is available to compute the controller output at time kT. This was the option taken by Landau [13]. The other one is to use sampled values up to the instant kT and deal with the computing time in some other way. If with some code optimization the controller output delay can be made small (less then 10% of the sampling interval T) it can be just discarded. If not, this delay can be forced to be equal to the sampling interval. Doing this adds an extra delay to the plant-plus-controller system that may be seen as a plant delay. This delay can then be compensated for by the addition of an equal delay to the model in order to keep the model state synchronized with the plant state.

The former approach may look more attractive until the Popov summations (the discrete-time equivalent of the Popov integrals) are analyzed:

$$\eta(0,k_1) \triangleq \sum_{k=0}^{k_1} v(k)^T w(k) \geq -\gamma^2 \tag{4.20}$$

It must be noticed that the upper limit is k_1, that is, the present sampling instant kT. As the summation is dependent on the model-system error at this instant it is not possible solve it with sample data to the (k-1)T instant only.

Landau [13] tried to get around this problem using the concept of "a priori" and "a posteriori" plant state vectors. The first is the state vector at instant kT that is obtained with the values of the adjustable parameters at instant (k-1)T (this is a measurable vector). The second is the state vector at instant kT, after the adaptation has taken place (this is not a measurable vector). He developed a way of finding the "a posteriori" state vector from the "a priori" one, but knowledge of the $\boldsymbol{B}_p$ matrix is required. Also, an auxiliary signal that acts on the reference model (adding to $\boldsymbol{u}_m$) is needed, but the "a posteriori" state vector must be predicted in order to synthesize it. The problem gets into a closed circle so that knowledge of the "a posteriori" state vector is needed in order to compute the signals and filters that convert the "a priori" state vector into the "a posteriori" one.

The second approach (to suppose zero computing time and compensate for it adding a delay to the model) is much more straightforward because there are no special problems in the computation of the Popov summations. The results obtained, shown in Appendix C, are completely analogous to the continuous-time case results.

3.4 The Robustness of Hyperstable MRAC Systems

So far, it was assumed that A_p, B_p, and $E_p g_p$ are unknown (with some restrictions on the structure) but constant (design hypothesis 4).

It is possible to show [25] that if A_p or $E_p g_p$ are time-varying but bounded, the error also remains bounded (using the property that a hyperstable feedback system has a bounded output for a bounded input). B_p must remain invariant, because it appears in the linear time-invariant feedforward block of the error system.

With

$$A_p(t) = A_p(0) + \Delta A_p(t) \tag{3.21}$$

and

$$B_p\varphi(t) = B_p\varphi(0) + B_p\Delta\varphi(t) \tag{3.22}$$

the generalized error system equation becomes:

$$\begin{aligned}\dot{e} = {} & A_m e + B_p(B_p^{\dagger}(A_m - A_p(0) - \Delta A_p(t)) - K_x)x_p + B_p(B_p^{\dagger}B_m - K_u)u_m \\ & - B_p(\varphi(0) + \Delta\varphi(t) + h)\end{aligned} \tag{3.23}$$

This equation can be rewritten as:

$$\dot{e} = A_m e - B_p w + B_p z(t) \tag{3.24}$$

with:

$$z(t) = -B_p^{\dagger}\Delta A_p(t) - \Delta\varphi(t) \tag{3.25}$$

$z(t)$ is an external input to the generalized error feedback system (see fig.3.4) and represents a disturbance to the adaptation process. As the free (unforced) error system is a hyperstable system, if $z(t)$ is bounded the error is also bounded. Landau [13] shows how to make an estimation of this bound.

Stoten [6,26] provides a study on the robustness of a MRAC controller subject to the time variations of both A_p and B_p matrices. In his work matrix B_p is excluded from the linear time-invariant feedforward block and included in the nonlinear time varying feedback block. Solutions of the Popov integral inequality can be obtained if the system is represented in phase-variable form, given some sign conditions on the values of the elements of B_p. The solutions found by Stoten are of the same type of those presented in equ.3.11 to 3.16. As he does not include a specific disturbance-cancelation signal, disturbances must be compensated for by changes in the K_x and K_u matrices. Consequently plant state and control signals cannot be identically zero along the closed-loop trajectories. He concludes that if the variations of A_p and B_p matrices occur at rates that are much slower than possible variations within the adaptive mechanism, then the MRAC laws are stable in the face of plant parameter variations.

Another type of design hypothesis violation is known as unmodelled dynamics (violation of design hypothesis 2). In this case, the plant contains high frequency dynamics that were disregarded in the design of the MRAC system. This situation is always present because plant models become highly inaccurate at high frequencies. So, as a plant model cannot contain all the plant dynamics, any assumption of plant order or relative degree in a broad (infinite) frequency spectrum is unrealistic. This means that parasitic high frequency dynamics (due to plant model ignorance, energy delivering elements, transducers, etc.) are always present.

If the plant is modelled as

$$\dot{x}_p = A_p x_p + B_p f(t,u_p) + E_p g_p \tag{3.26}$$

in which f is a linear time invariant function that represents all the unmodelled dynamics referred to the plant input, the generalized error system becomes:

$$\dot{e} = A_m e - B_p f(t,w^*) \tag{3.27}$$

in which

$$w^* = K_x x_p - f^{-1}(t,B_p^{\dagger}(A_m - A_p)x_p) + K_u u_m - f^{-1}(t,B_p^{\dagger}B_m u_m) + h + f^{-1}(t,\varphi) \tag{3.28}$$

Equ.3.27 and equ.3.4 define the real feedforward block of the generalized error system. This means that when unmodelled dynamics are present, the controlled plant can only match the reference model up to a certain frequency range, and the forward loop transfer function which determines the error dynamics loses its positive realness property. If the adaptive gains grow high enough due to the presence of disturbances on the plant, instability will certainly occur.

Rohrs *et al.* [27] shows in detail how this instability mechanism works and concludes that it is due to the presence of infinite-gain operators on the adaptation loop. He says that: "The pragmatic implication of the existence of such infinite-gain operators is that 1) sinusoidal reference inputs at specific frequencies and/or 2) sinusoidal output disturbances at any frequency (including dc), can cause the loop gain to increase without bound, thereby exciting the unmodelled high-frequency dynamics, and yielding an unstable control system."

Rohrs infinite-gain operators are due to the presence of free integrators in the adaptation algorithm. When the plant input signal is not persistently exciting of the appropriate order and, given the presence of unmodelled dynamics, with the proper frequency content [28], the plant parameters will not converge to a set of values, but will tend to drift in the presence of external disturbances. The occurrence of this drift can be verified even if the plant has no unmodelled dynamics [29].

Given the weak robustness of MRAC systems as presented, several modifications to the adaptation rules have been proposed in the literature [30,31]. All these modifications are based on the ideas of limiting the gain of the adaptation loop and stopping the integral part of the adaptation

algorithm preventing the drift of the adaptive gains. The most common modifications to the basic algorithm are:

- The use of dead-zones. The adaptive gains are adjusted only when the norm of v lies outside of a dead-zone. This dead-zone must be greater than the expected values of v due to external disturbances. Peterson and Narendra [32] show that the adaptation process takes place only during a finite time and the controller parameters converge to a set of values close to the ideal ones. A tradeoff exists in this scheme between robustness and model following error - as the size of the dead-zone is increased, robustness increases but so does model following error. Some variations of this idea have also been developed using relative error or variable dead-zones [16] with the objective of a better compromise between error and robustness.

- Constrained gains adaptation. In this approach the controller parameters are constrained to a given bounded set [33]. This set must be chosen in a way that the possible values of controller parameters needed to stabilize the plant are contained in it. This implies that some preknowledge about the plant parameters and external disturbance bounds exists. An advantage of this approach is that the model following error can be annulled by the controller in the disturbance free case.

- Integrator leakages (σ and e_1 modifications). In this type of modification a leakage (σ) is introduced in the integrators of the adaptation laws, transforming them into first order low-pass filters. Ionnou and Kokotovic [34] show that with a properly chosen value of σ the controller parameters converge to a residual set which contains the parameters needed for perfect model following. With this modification there is also a tradeoff between robustness and model tracking. If the parasitic dynamics are located at higher frequencies, smaller values of σ may be used, and closer model tracking is achieved. The e_1 modification proposed by Narendra an Annaswamy [35], offers a better compromise by making the integrator leakage proportional to the modulus of the model following error. An

advantage of e_1 over σ modification is that in the ideal case (no unmodelled dynamics) perfect tracking of the reference model is achievable.

- Normalization of the signals used in the adaptation process. The effects of unmodelled dynamics may be represented by an unbounded disturbance in the error equation [31]. Being a discrete-time method, the fundamental idea of normalization is to devise a normalization factor such that when applied to the error equation a normalized error equation with bounded disturbances results. Ortega *et al.* [36] and Prayly [37] show that with the use of a suitable normalizing factor, a discrete-time adaptive controller can be made stable, given the resulting boundness of all the signals within the controller (due to normalization), in the presence of unbounded disturbances.

- Combinations of several of these methods. The first three modifications to the adaptation algorithms when used independently lead only to local stability results, that is the obtained MRAC systems are stable for bounded (external) disturbances or if the effects of unmodelled dynamics are small (the parasitic dynamics are located at frequencies that are much higher than the dominant ones). Only when these modifications are used together with normalization can global stability results be achieved. Prayly [37] uses a combination of normalization and constrained gains adaptation, and Ioannou and Tsakalis [38] present an discrete-time adaptation algorithm in which a σ modification is used together with normalization of the signals used for parameter adaptation. Kreisselmeier and Anderson [39] present another discrete-time adaptive law combining a relative dead zone (using a normalized error signal) and constrained gains estimation. In this design preknowledge of the plant and controller parameters bounds is needed.

In most practical cases (as in the manipulator control problem) some preknowledge about the plant parameter variations is available. Limits on those variations can usually be defined. These limits on the plant parameters range imply limits on the controller parameters range. So, the search of controller parameters, which is done by the adaptation algorithm, can be easily constrained to a given

bounded set, increasing the robustness of the adaptive controller. MRAC system instability is always connected to the drift of the controller parameters. This drift happens because the adaptation rules contain memory elements (the integrators). Also there is no simple way of defining the adaptation bandwidth, that is, the adaptation gains. If the adaptation bandwidth is made too large, the extra phase-shift introduced by unmodelled dynamics may turn the system unstable due to an insufficient phase margin.

Taking all this into account, it would be interesting to develop a new adaptation algorithm. This new algorithm shall be free to use the available preknowledge about the bounds on the plant parameters values and the adaptation rules shall be of the memory-less type, so that the adaptive controller parameters cannot drift. There must be also a way of establishing the adaptation bandwidth so that the controller does not try to adapt to high frequency modes it cannot control.

3.5 Design of a Robust Hyperstable MRAC System With Memory-less Adaptation Rules

As discussed in the Introduction of this work the control of the manipulator will be a decentralized one, that is, there will be a completely independent controller per axis. This implies that the manipulator is seen as a set of Single Input Single Output (SISO) systems, the links, dynamic interactions being treated as external disturbances. So, for the purpose of this work, only the SISO case of the controller design needs to be developed.

The design of a memory-less hyperstable MRAC system will follow the same steps as presented in 3.2. The main differences will be the inclusion of b_p in the feedback block of the hyperstable error system, the assumption that the reference and plant models are represented in phase-variable form, and the exclusion of integrators from the adaptation rules. The inclusion of b_p in the feedback block decreases the effects of the design hypotheses violations, leaving the feedforward block linear and time-invariant even if the plant is changing, if the use of linear and time-invariant reference model is assumed. The preknowledge of bounds on plant parameters values as well as the sign as the plant input gain (usually positive) is also assumed. Although the adaptation rules are memory-less asymptotic stability will nevertheless be achieved with the use of a sliding mode, circumventing the need of integral adaptation gains.

The error equation (equ.3.2) can be rewritten in the SISO case as:

$$\dot{e} = A_m e + b_p(b_p^{\dagger}(A_m - A_p) - k_x^T)x_p + b_p(b_p^{\dagger}b_m - k_u)u_m - b_p(\varphi + h) \qquad (3.29)$$

where

$$A_m = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \\ a_{n1} & a_{n2} & \cdots & \cdots & \cdots & a_{nn} \end{bmatrix} \quad , \quad b_m = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ b_m \end{bmatrix} \tag{3.30}$$

$$(A_m - A_p) = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ a_1 & a_2 & \cdots & a_n \end{bmatrix} \quad , \quad b_p = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_p \end{bmatrix} \quad , \quad b_p^{\dagger} = \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{b_p} \end{bmatrix} \tag{3.31}$$

b_p can be factorised as

$$b_p = I^* b_p \quad , \quad I^* = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{3.32}$$

The error equation can now be developed into a hyperstable loop, originating a linear time-invariant feedforward block

$$\dot{e} = A_m e - I^* w \tag{3.33}$$

$$v = d^T e \tag{3.34}$$

and a nonlinear time-varying feedback block

$$w = b_p((k_x^T - b_p^{\dagger}(A_m - A_p))x_p + (k_u - b_p^{\dagger} b_m)u_m + (\varphi + h)) \tag{3.35}$$

d^T is a linear time-invariant filter on the generalized error chosen in a way that the feedforward block transfer function

$$h(s) = d^T(sI - A_m)^{-1}I^* \tag{3.36}$$

is a strictly positive real transfer function.

Having b_p in the feedback block enables the accomplishment of the hyperstability conditions, even if the plant is time-varying, because in this situation the feedforward block is composed of linear and constant terms only.

In order to make the error loop a hyperstable one, the Popov integral inequality must be solved. The Popov inequality will be true if each of the following integral inequalities is true:

$$\eta_1(0,t_1) = \int_0^{t_1} v b_p (k_x^T(v) - b_p^\dagger(A_m - A_p))x_p\,dt \geq -\gamma_1^2 \tag{3.37}$$

$$\eta_2(0,t_1) = \int_0^{t_1} v b_p (k_u(v) - b_p^\dagger b_m)u_m\,dt \geq -\gamma_2^2 \tag{3.38}$$

$$\eta_3(0,t_1) = \int_0^{t_1} v b_p (h(v) + \varphi)\,dt \geq -\gamma_3^2 \tag{3.39}$$

where k_x^T, k_u and h are functions of v only (memory-less adaptation rules).

The first inequality (equ.3.37) can be rewritten in scalar form as:

$$\eta_1(0,t_1) = \sum_{i=1}^{n} \int_0^{t_1} v b_p \left(k_{xi}(v) - \frac{a_i}{b_p} \right) x_{pi}\,dt \geq -\gamma_{1i}^2 \tag{3.40}$$

This inequality holds if each of the i inequalities hold. One way of achieving this is to guarantee the positiveness of all the i integrands.

The adaptation law

$$k_{xi}(v) = \alpha_{xi}\,\mathrm{sign}(v x_{pi}) + \beta_{xi} \tag{3.41}$$

is able to fulfil that requirement. This type of adaptation law will create a sliding mode that compensates for the absence of memory, guaranteeing asymptotic stability.

In order to find the values of the adaptation coefficients (α_{xi}, β_{xi}), preknowledge of the sign of the plant input gain (sign of b_p) as well as its range and the range of the plant parameters (range of a_i / b_p) will be needed. If it is assumed that b_p is positive (as it usually is) and as such making α_{xi} also positive, the following inequality results:

$$b_p v x_{pi}\left(\alpha_{xi}\,\mathrm{sign}(v x_{pi}) + \beta_{xi} - \frac{a_i}{b_p}\right) \geq 0 \quad , \quad \forall t \geq 0 \tag{3.42}$$

That is:

- If $\mathrm{sign}(v x_{pi}) > 0$ then,

$$\alpha_{xi}\,\mathrm{sign}(v x_{pi}) + \beta_{xi} - \frac{a_i}{b_p} \geq 0 \;\Rightarrow\; \alpha_{xi} \geq \left(\frac{a_i}{b_p} - \beta_{xi}\right) \tag{3.43}$$

- If $\mathrm{sign}(v x_{pi}) < 0$ then,

$$\alpha_{xi}\,\mathrm{sign}(v x_{pi}) + \beta_{xi} - \frac{a_i}{b_p} \leq 0 \;\Rightarrow\; \alpha_{xi} \geq \left(\beta_{xi} - \frac{a_i}{b_p}\right) \tag{3.44}$$

- So,

$$\alpha_{xi} \geq \max_{x_{pi},t}\left|\frac{a_i}{b_p} - \beta_{xi}\right| \tag{3.45}$$

From the second Popov Inequality (equ.3.38) the following result is obtained:

$$k_u(v) = \alpha_u \text{sign}(v u_m) + \beta_u \tag{3.46}$$

with

$$\alpha_u \geq \max_{u_m,t} \left| \frac{b_m}{b_p} - \beta_u \right| \tag{3.47}$$

which uses the preknowledge of the range of b_m / b_p.

In the same way, the following results are obtained from the third inequality (equ.3.39):

$$h(v) = \alpha_h \text{sign}(v) + \beta_h \tag{3.48}$$

with

$$\alpha_h = \max_{\varphi,t} | \varphi + \beta_h | \tag{3.49}$$

which uses the preknowledge of the range of φ.

The resultant controller (a block diagram of it is presented in fig.3.5) has discontinuous gains. This type of gain adaptation usually leads to a kind of undesirable system behaviour: chatter. It is very interesting to notice the similarities with a Variable Structure Control (VSC) system [40]. Indeed, this system is a VSC system in which the controller works in the space of the model following error, as suggested by Balestrino *et al.* [41]. Here $v = 0$ defines a sliding plane for the error state. As the system is asymptotically stable and contains switching gains, the error state will slide to the origin of the error state space chattering along this plane, as will be shown in the next chapter.

In the synthesis of this controller, the need to use preknowledge about plant parameters bounds, the sign of the plant input gain, and external disturbances bounds emerged. The first and second conditions are not too restrictive because for a given working plant it is fair to assume that some knowledge of plant parameter bounds exists, and usually the sign of the plant input gain does not

change with the plant state. These two conditions are easily obeyed by a robotic manipulator. The third condition (preknowledge of external disturbances bounds) is more restrictive. Even if it can be argued that some design bounds do exist, there is no way to be sure they will not be exceeded. It is acceptable that when faced with an overload the system will not behave as desired (it usually saturates), but the possibility of unstable behaviour under these circumstances must be carefully examined.

As the controller gains are bounded (actually they can only present one of two values) and the adaptation rules are memory-less there is no scope for drift or instability due to non-persisting excitation and external disturbances. Nevertheless, the fact that this controller induces chattering is an undesirable feature, because chattering may excite high frequency modes of the system (usually structural vibration modes) and reduces the life expectancy, due to fatigue, of actuators, mechanical couplings, transmissions, etc. There is also no way to specify the bandwidth of the adaptation loop (in an ideal implementation it would be infinite) and so the strength of the influence of the unmodelled dynamics is not known.

From this discussion it can be concluded that this controller needs to be modified. The use of memory-less adaptation rules should be kept in the dynamic adaptation gains (k_x and k_u), although the switching gains must be smoothed. Furthermore a synthesised disturbance cancelation signal h must be generated by a dynamic controller in order to avoid the need for external disturbance bounds. It must be also possible to establish a desired (or possible) adaptation loop bandwidth, that may be critical in the generation of h with a dynamic controller, in order to ensure the stability of the system.

Given the similarities of this hyperstable controller with a variable structure controller there is scope to use variable structure systems theory in order to accomplish the modifications needed, because VSS theory offers greater freedom and insight in the synthesis of memory-less adaptive controllers. This is the approach that will be followed in the next chapter.

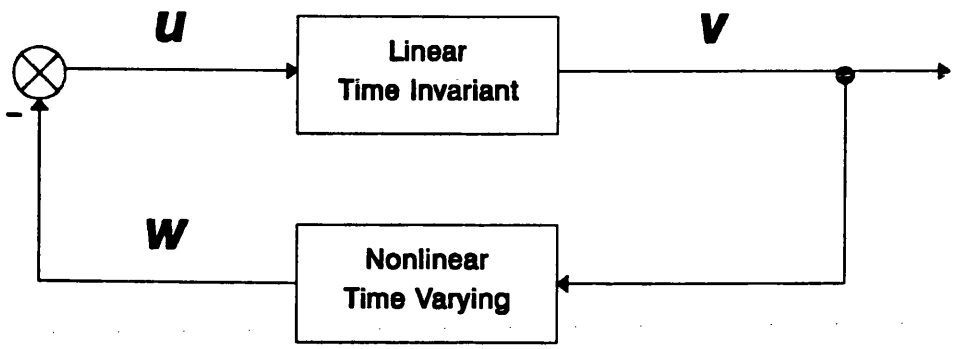


Figure 3.1 - Structure of a hyperstable system

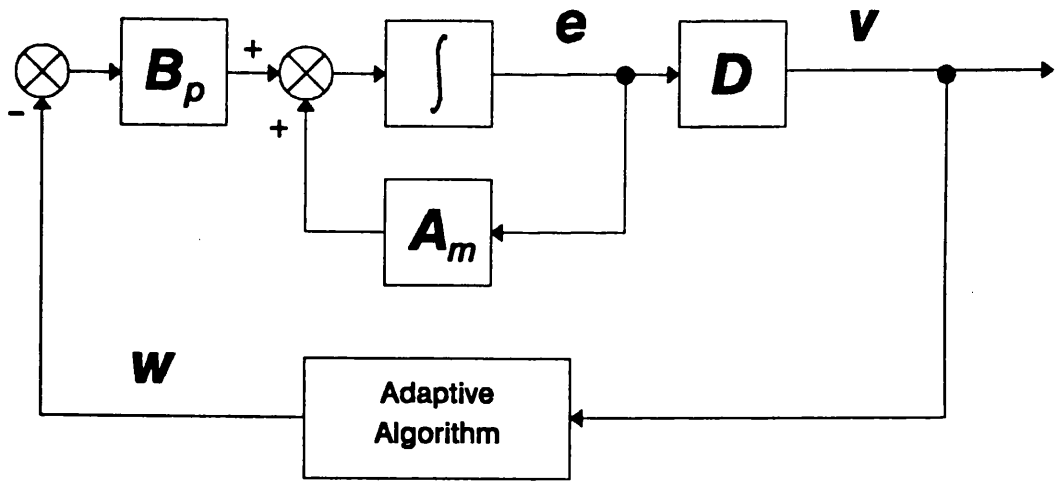


Figure 3.2 - MRAC error loop

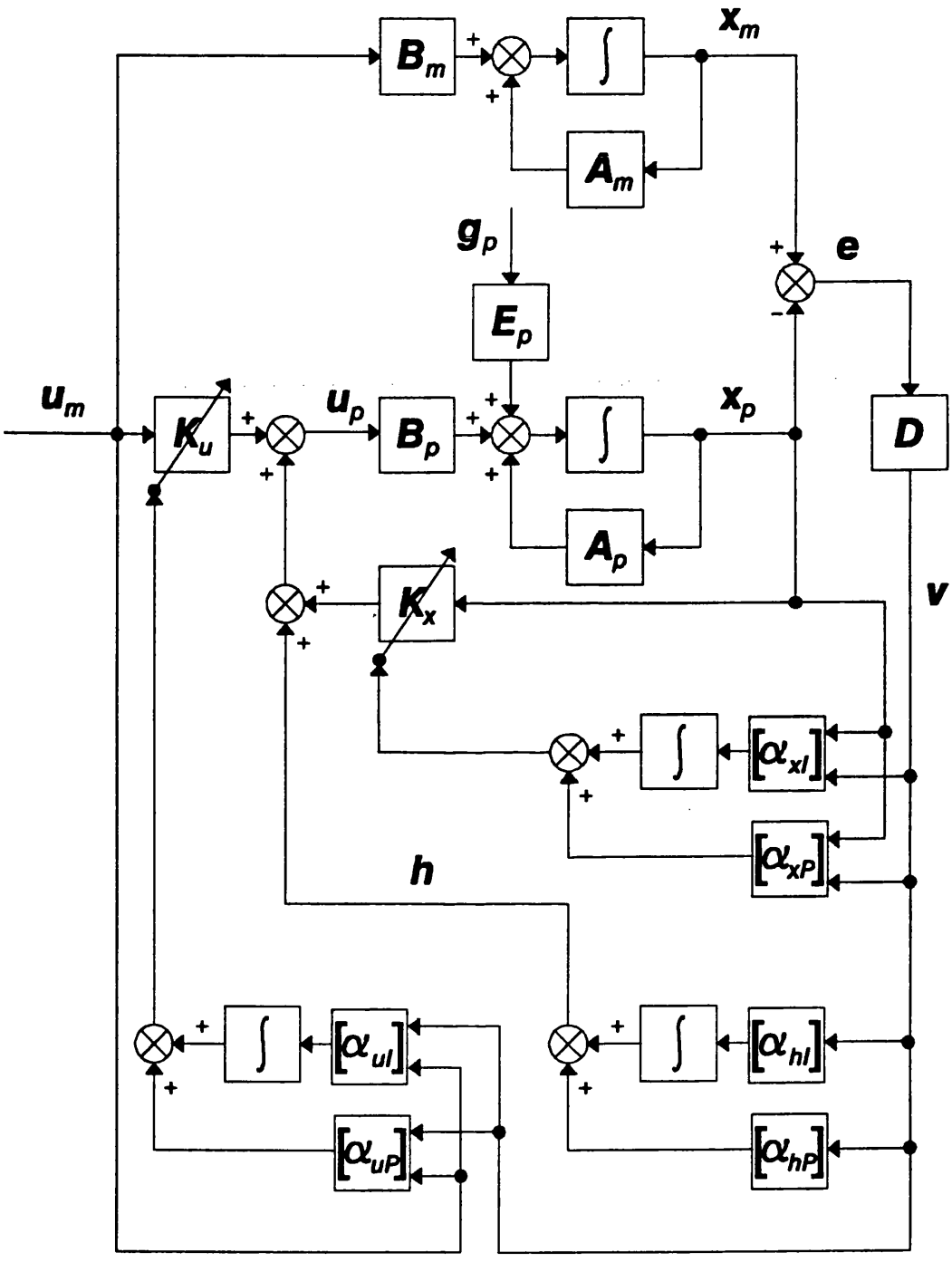


Figure 3.3 - Block diagram of a hyperstable MRAC

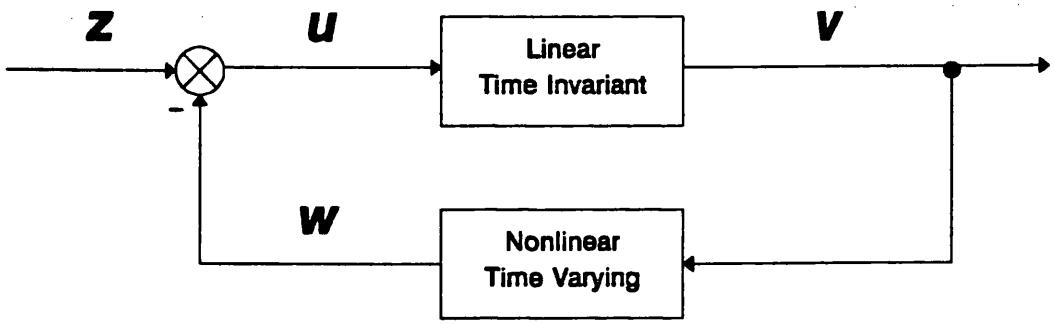


Figure 3.4 - The effect of time-varying parameters on the hyperstable loop

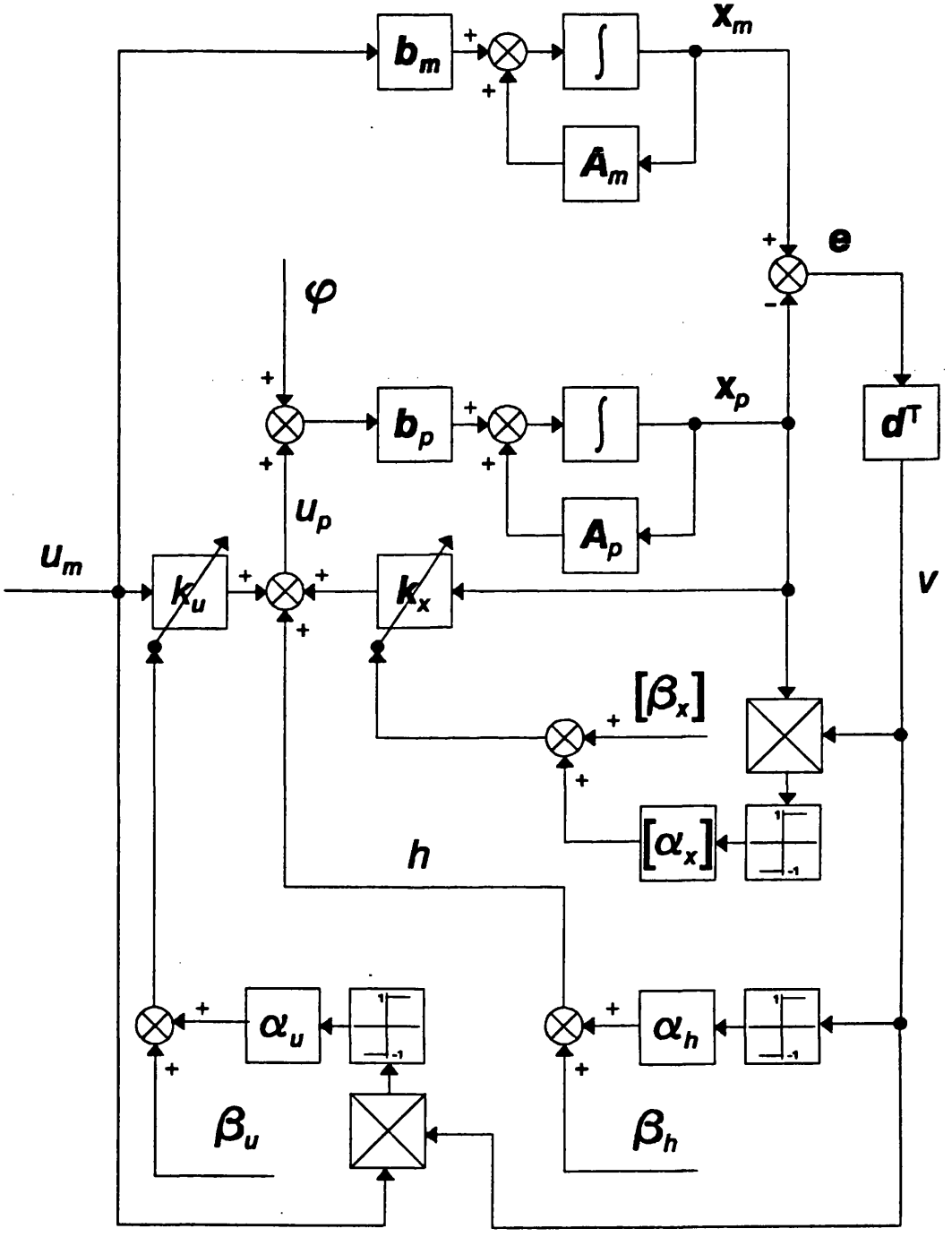


Figure 3.5 - Block diagram of a hyperstable MRAC for time-varying plants

Chapter 4 | A Variable Structure Model Reference Adaptive Controller

4.1 Variable Structure Systems

In a variable structure system two different control structures (not necessarily stable) are combined by switching between them when the describing point of the state trajectory (the present state) passes trough a previously defined surface, the switching surface [40,42].

If each of the control structures forces the state trajectory to be directed towards the switching surface, then once on the surface, the describing point (state) evidently cannot move along any trajectory adjacent to that surface over any period, however short (see fig.4.1). Indeed, in response to any deviation, a motion always starts that returns the describing point to the discontinuity surface. Consequently, in a system of this kind, the describing point of the state trajectory can ideally only move along the discontinuity surface; this motion is conventionally referred to as the sliding mode. What has been described is an idealised model of a sliding mode. In an actual system, the discontinuous change of structure presents imperfections such as delays, hysteresis, etc. Consequently, the imperfections make the switching occur at a finite frequency and the describing point oscillates in a certain finite vicinity of the discontinuity surface. This oscillatory movement, occurring at the switching frequency, is commonly referred to as chatter.

When a system is in a sliding mode its state trajectory is completely defined by the switching surface. As the state trajectory is nothing more than a representation of the time behaviour of the system, system dynamics are completely defined by the switching surface.

As an example the stabilization of a second order unstable plant is presented. Let the system be described by the following equation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \qquad (4.1)$$

where x_1 and x_2 are state variables, a_{21} and a_{22} are positive constant parameters, and u is the control action. Let u be a piecewise linear function of x_1:

$$u = -kx_1 \qquad (4.2)$$

and assume that there are two linear structures in the this system which are associated with the values k^+ or k^- of the gain k. Let k^+ be selected so that the system with $k = k^+$ has complex eigenvalues, and k^- so that the system with $k = k^-$ has real eigenvalues. The phase portraits of both structures are represented in figs.4.2 and 4.3. It must be noticed that each of these structures is unstable.

Let the system structure be changed on the straight lines $x_1 = 0$, $v = d \cdot x_1 + x_2 = 0$, with d constant and strictly positive. The coefficient d will be selected such that the straight line $v = 0$ will be between the axis x_1 and the stable eigenvalue associated with the structure $k = k^-$ (fig.4.3).

Fig.4.4 is a phase portrait of the system with the switching law:

$$k = \begin{cases} k^+ \Leftarrow x_1 v > 0 \\ k^- \Leftarrow x_1 v < 0 \end{cases} \qquad (4.3)$$

The phase portrait shows that the state trajectory invariably reaches the switching line $v = 0$ from any initial condition. In the vicinity of the straight line the trajectories of both structures are directed towards it; therefore further motion will proceed in the switching or sliding mode along the straight line $v = 0$.

The behaviour of the system is then described by the switching line equation. As $x_2 = \dot{x}_1$ the switching line equation can be rewritten as:

$$\dot{x}_1 + d \cdot x_1 = 0 \tag{4.4}$$

which is a first order differential equation.

So, combining two unstable structures, a stable motion resulted. This motion has first order dynamics (there is always an order reduction of one when a variable structure controller is used) and is independent of plant parameters and external disturbances.

The stability criteria of having the state trajectories directed towards the switching line in all state space can be described by the expression:

$$v\dot{v} < 0 \quad , \quad \forall x \neq 0 \tag{4.5}$$

If this condition is verified in all state space (with the exception of the origin) the system will enter in sliding mode and, as such, is stable. This is the inequality commonly used in the design and stability proof of variable structure control systems.

4.2 The Design of a Variable Structure MRAC System

A new model reference adaptive controller algorithm for time-varying plants will now be developed, making use of Variable Structure Systems (VSS) theory. This route is strongly suggested by the results obtained in 3.5. In the same way the controller algorithm will be developed for the Single Input Single Output (SISO) case only. Some assumptions about the system and reference model structures are made, such us the use of phase variables as state variables, the preknowledge of plant parameters bounds and the sign of the plant input gain. In order to avoid the problem associated with the way external disturbances were dealt in 3.5, external disturbances will be treated somewhat independently from the plant parameters variation.

One of the main problems presented by variable structure systems is the induction of chatter due to non-ideal switching between control structures. System chatter may also be attributed to the fact that the system is being forced to perform a physically impossible task: to behave as a dynamic system of an order that is one less than its own. In the controller to be presented the chattering problem will be solved by modifications to the basic variable structure algorithm:

- Switching gains will not be used. The relay function will be substituted by a finite gain function. When using this modification it can only be guaranteed that the system state will converge and stay inside of a boundary layer of a given thickness around the sliding surface.

- System order reduction will be avoided. This will be accomplished by having the sliding surface generated by a dynamic system. This dynamic system is implemented in the controller algorithm as a series-parallel model of the model following error behaviour. If this model is chosen to be equal to the state reference model the system will never be forced to have a dynamic behaviour faster, or of smaller order, than the reference model. Although it is not compulsory to make the two models (state and model following error behaviour) equal, that assumption will be made here, leading to a simpler

controller structure. It can also be argued that, at least in the manipulator control case, if one of the models can be made faster than the other it is because the plant is able to follow it. So why not chose the other model equally fast? That will lead to a greater bandwidth of either reference tracking or model following.

The basic equations describing the generalized error system in the SISO case are:

- the reference model

$$\dot{x}_m = A_m x_m - b_m u_m \tag{4.6}$$

- the adjustable system

$$\dot{x}_p = A_p x_p + b_p u_p + b_p \varphi \tag{4.7}$$

$$u_p = k_x^T x_p + k_u u_m + h \tag{4.8}$$

- the generalized error system

$$\dot{e} = A_m e + ((A_m - A_p) - b_p k_x^T) x_p + (b_m - b_p k_u) u_m - b_p(\varphi + h) \tag{4.9}$$

Now, a distance to a datum sliding surface is introduced:

$$v = d^T e \quad , \quad d = \begin{bmatrix} d_1 \\ \vdots \\ d_{n-1} \\ 1 \end{bmatrix} \tag{4.10}$$

In order to avoid the order reduction of the system, as well as to generate the external disturbance cancellation signal h a model reference series-parallel regulator of the error behaviour will be used. The reference model of this regulator is described by:

$$\dot{e}_r = A_m e \tag{4.11}$$

The distance of this reference error to the datum sliding surface is:

$$v_r = d^T e_r \tag{4.12}$$

The controller will try to make v equal to zero, not by forcing the error to converge to the datum sliding surface as fast as possible (which creates the order reduction problem), but by following the reference sliding surface distance given by v_r. So, v_r can be seen as a new, dynamically generated, sliding surface.

The time behaviour of the difference between the two distances (the distance between the actual error and the reference error) is described by:

$$\dot{v}_e = \dot{v}_r - \dot{v} = d^T(\dot{e}_r - \dot{e}) \Rightarrow \tag{4.13}$$

$$\dot{v}_e = d^T((b_p k_x^T - (A_m - A_p))x_p + (b_p k_u - b_m)u_m + b_p(\varphi + h)) \tag{4.14}$$

$$\dot{v}_e = \sum_{i=1}^{n} (b_p k_{xi} - a_i)x_i + (b_p k_u - b_m)u_m + b_p(\varphi + h) \tag{4.15}$$

In order to find the adaptation laws for k_x and k_u (h will be dealt independently) the VSS stability condition will be used:

$$v_e \dot{v}_e < 0 \quad , \quad \forall e \neq 0 \tag{4.16}$$

That is:

$$v_e \sum_{i=1}^{n} (b_p k_{xi} - a_i) x_i < 0 \tag{4.17}$$

and

$$v_e (b_p k_u - b_m) u_m < 0 \tag{4.18}$$

The first condition (equ.4.17) is true if each of the i equations

$$v_e (b_p k_{xi} - a_i) x_i < 0 \tag{4.19}$$

is true.

So, supposing b_p to be positive as before,

$$\text{sign}(v_e x_i) > 0 \Rightarrow k_{xi} < \frac{a_i}{b_p} \Rightarrow k_{xi}^{+} < \min\left(\frac{a_i}{b_p}\right) \tag{4.20}$$

$$\text{sign}(v_e x_i) < 0 \Rightarrow k_{xi} > \frac{a_i}{b_p} \Rightarrow k_{xi}^{-} > \max\left(\frac{a_i}{b_p}\right) \tag{4.21}$$

The second condition (equ.4.18) is true if:

$$\text{sign}(v_e u_m) > 0 \Rightarrow k_u < \frac{b_m}{b_p} \Rightarrow k_u^{+} < \min\left(\frac{b_m}{b_p}\right) \tag{4.22}$$

$$\text{sign}(v_e u_m) < 0 \Rightarrow k_u > \frac{b_m}{b_p} \Rightarrow k_u^{-} > \max\left(\frac{b_m}{b_p}\right) \tag{4.23}$$

Without surprise the obtained adaption laws are similar to those produced by the use of hyperstability theory. But now, the use of VSS theory gives the freedom to modify the relay type adaptation, avoiding the chattering problem, while retaining the stability properties.

The adaptation laws obtained so far have the form:

$$k_{xi} = \begin{cases} k_{xi}^{+} \Leftarrow \text{sign}(v_e x_i) > 0 \\ k_{xi}^{-} \Leftarrow \text{sign}(v_e x_i) < 0 \end{cases} \tag{4.24}$$

These laws can be modified into (see fig.4.5):

$$k_{xi} = \begin{cases} k_{xi}^{+} \Leftarrow v_e \text{sign}(x_i) > \mu \\ v_e \text{sign}(x_i) \dfrac{k_{xi}^{+} - k_{xi}^{-}}{2\mu} + \dfrac{k_{xi}^{+} + k_{xi}^{-}}{2} \Leftarrow -\mu < v_e < \mu \\ k_{xi}^{-} \Leftarrow v_e \text{sign}(x_i) < -\mu \end{cases} \tag{4.25}$$

The same modification must be applied to k_u.

This modification defines a boundary layer around the switching surface $v_e = 0$ with a thickness equal to μ (see fig.4.6). In a classical variable structure controller which presents a fixed sliding surface the parameter μ and the slope of the surface define an error bound ε. In the present case, with the sliding surface being generated by a dynamic model, its slope is not constant. So, the relation with the error bound is lost. As the adaption laws are not changed outside this boundary, the boundary layer attractiveness is guaranteed. So, all state trajectories starting outside the boundary layer will converge to and go inside it [43]. The modification used here is a linear interpolation between the two extreme values of the gain, but other smoothing functions are possible, like the one presented by Ambrosino *et al.* [44].

Now, the external disturbances cancelling signal h must be found. As the effects due to the mismatches between plant and model coefficients were solved, only external disturbances effects must be taken into account. Even if this condition is not completely true, the invariance conditions (equ.2.8, 2.9, 2.10) imply that all the effects due to plant and model coefficients mismatch may be included in φ. So, it can be said that:

$$\dot{v}_e = b_p(\varphi + h) \tag{4.26}$$

Making

$$h = -k_h v_e \tag{4.27}$$

the following first order differential equation is obtained for v_e:

$$\dot{v}_e = -k_h b_p v_e + b_p \varphi \tag{4.28}$$

which is stable for all k_h strictly positive.

It must be noticed that it is not required that v_e goes to zero, in the presence of external disturbances, in order to obtain perfect model following. Due to the structure of the algorithm (fig.4.7), v_r will converge to a value that forces v to zero, as required in order to make the model following error converge to zero. So, an implicit integral action is introduced by the use of the series-parallel model of the error behaviour. As noticed before, any or all of the error-inducing components due to the differences between model and plant coefficients (the $b_m u_m$ and the n $a_i x_i$ terms) can be considered as components of φ. This is very convenient because it increases the robustness of the controller. If any parameter exceeds its bound, the quality of the controller action may deteriorate, but the system stability is assured. In the limit only the disturbance rejection part of the algorithm needs to be implemented.

In the controller algorithm presented a datum sliding plane v was used. How can this sliding plane be chosen? This happens to be a false problem. The fact is this sliding plane is just a datum and, as such, it does not matter at all what particular plane is chosen. Actually, as may be seen in equ.4.15, v_e is completely independent of d. Some insight about this may be gained by observing what happens

with a second order system in a phase plane diagram, as represented in fig.4.8. In this figure, $v_1 = 0$ and $v_2 = 0$ are the sliding lines that would result from two different choices of $\boldsymbol{d}$. The actual sliding line, obtained using the new method, is given by $v_e = 0$. The signal v_e is obviously independent of the datum chosen for $v = 0$. Given this freedom of choice, an obvious candidate for $\boldsymbol{d}$ is $\boldsymbol{I}^*$.

As the new sliding line (desired error trajectory) is not rigidly predetermined, but is generated by a dynamic model of the desired error behaviour, there is no order reduction of the error dynamics. In this way the system is never forced to behave with dynamics faster than the model dynamics. This characteristic avoids the chattering problem, enabling the use of smaller interpolation bands on the discontinuous gains smoothing functions (μ), with the correspondent increase of the state tracking capabilities; that is, obtaining smaller model following errors.

An advantage of the controller structure presented is the extra freedom given to the choice of model, the only restrictions being that the PMF conditions must be obeyed. The model can be linear time-varying causing no problems to the algorithm.

Another interesting point is, that apart from the model, there are only two parameters that must be chosen by the controller designer: k_h and μ. As any residuals of the parameter adaptation may be considered as components of φ, the value of μ is not critical and is not too rigidly related with the adaptation error. Its main function is to provide enough smoothing on the discontinuous gains in order to eliminate the chattering phenomena, and it is with that purpose in mind that it should be chosen. As there is no formal way of computing this parameter, it must be chosen by trial and error, ideally with the help of simulation studies. At first sight it seems that there are no restrictions on the value of k_h. In practice the dynamic behaviour of v_e is limited by parasitic (unmodelled) dynamics, like the dynamic characteristics of the energy delivering elements (servo-valve, power amplifier, etc. as in fig.4.9). So, k_h should be chosen in a way that the bandwidth of v_e is kept inside the bandwidth of these unmodelled dynamics. Nevertheless it should always be the aim to make the value of k_h as high as possible, given

that the robustness of the controller to external disturbances and residuals of parameter adaptation is proportional to it. In some cases the introduction of some additional compensator increasing the phase lead can be beneficial, as long as the noise transmission ratio is kept small, in the sense that as the bandwidth of the adaptation loop is increased, it enables the use of a higher value for k_h, increasing the robustness of the controller.

Saturation is more difficult to deal with, when using this new controller design. In a classical hyperstable MRAC all that is needed under saturation is to make the state of the model equal to that of the plant, so that, while the system is saturated it is the model that follows the plant [6,13]. In a variable structure controller there is no need to do anything, due to the fact that the adaptation rules are completely memory-less. With the proposed controller, the use of an internal antiwindup loop is compulsory, in order to provide the right initial condition to v_e when the plant desaturates, avoiding any sort of saturation-related misbehaviour. A fixed gain, k_{aw}, acting on the difference between the saturated and the unsaturated control actions, obtained with the help of a saturation block in the controller, is used to determine the time response of this loop.

A block diagram of the complete system, including the compensator and the antiwindup loop, is presented in fig.4.10.

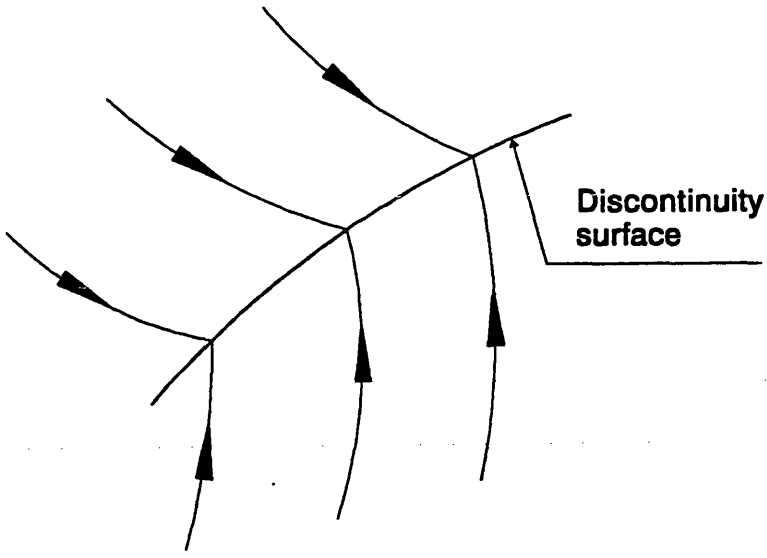


Figure 4.1 - Sliding surface

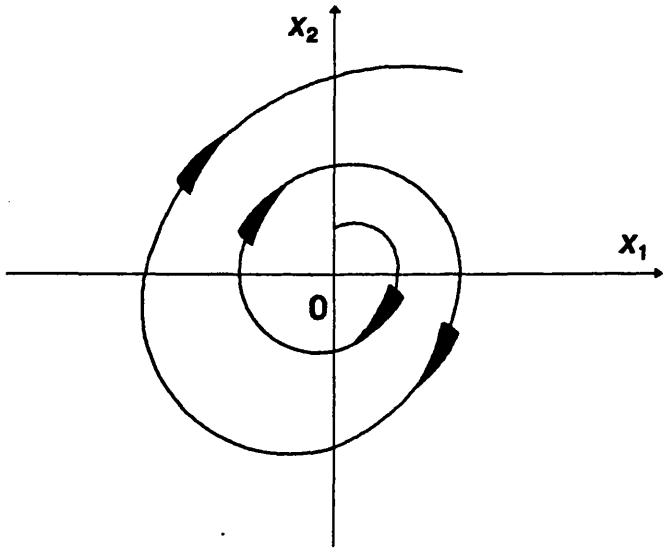


Figure 4.2 - Phase portrait with $k = k^+$

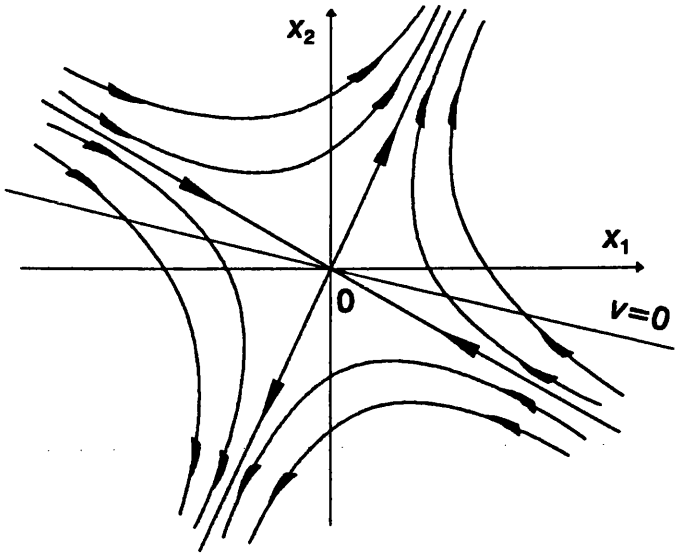


Figure 4.3 - Phase portrait with $k = k^-$

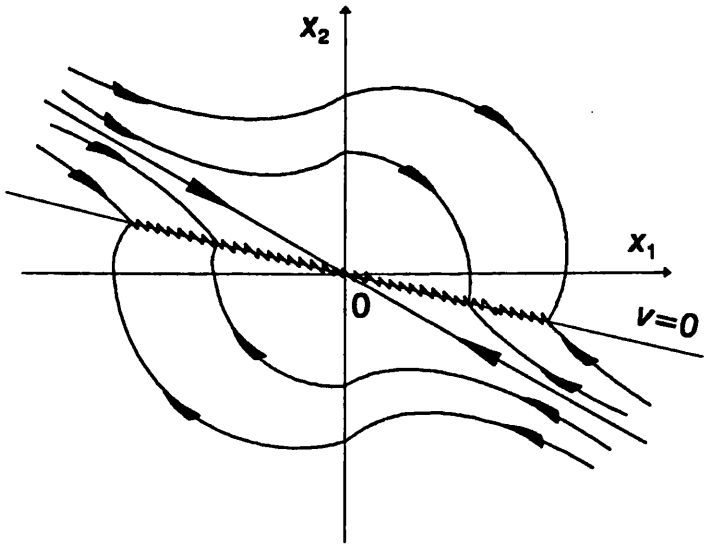


Figure 4.4 - Phase portrait with switching law

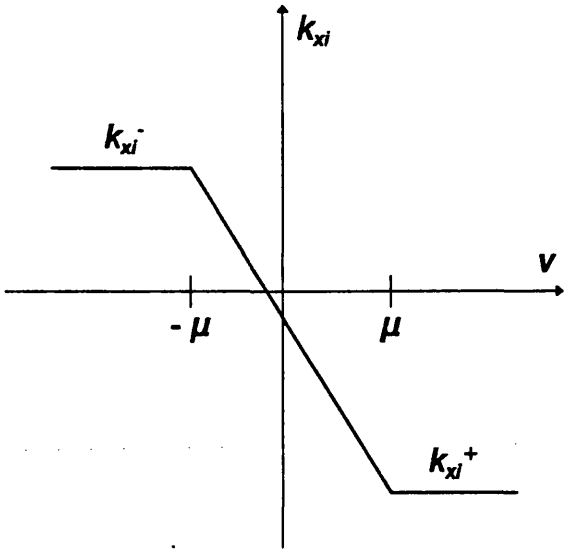


Figure 4.5 - Linear gain interpolation

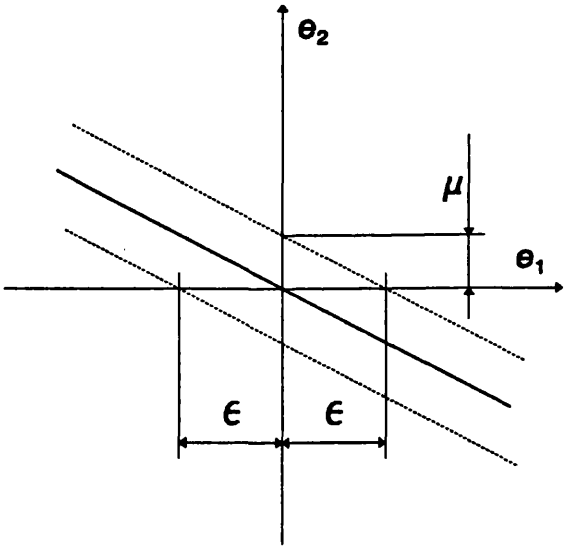


Figure 4.6 - Boundary layer around
the switching line

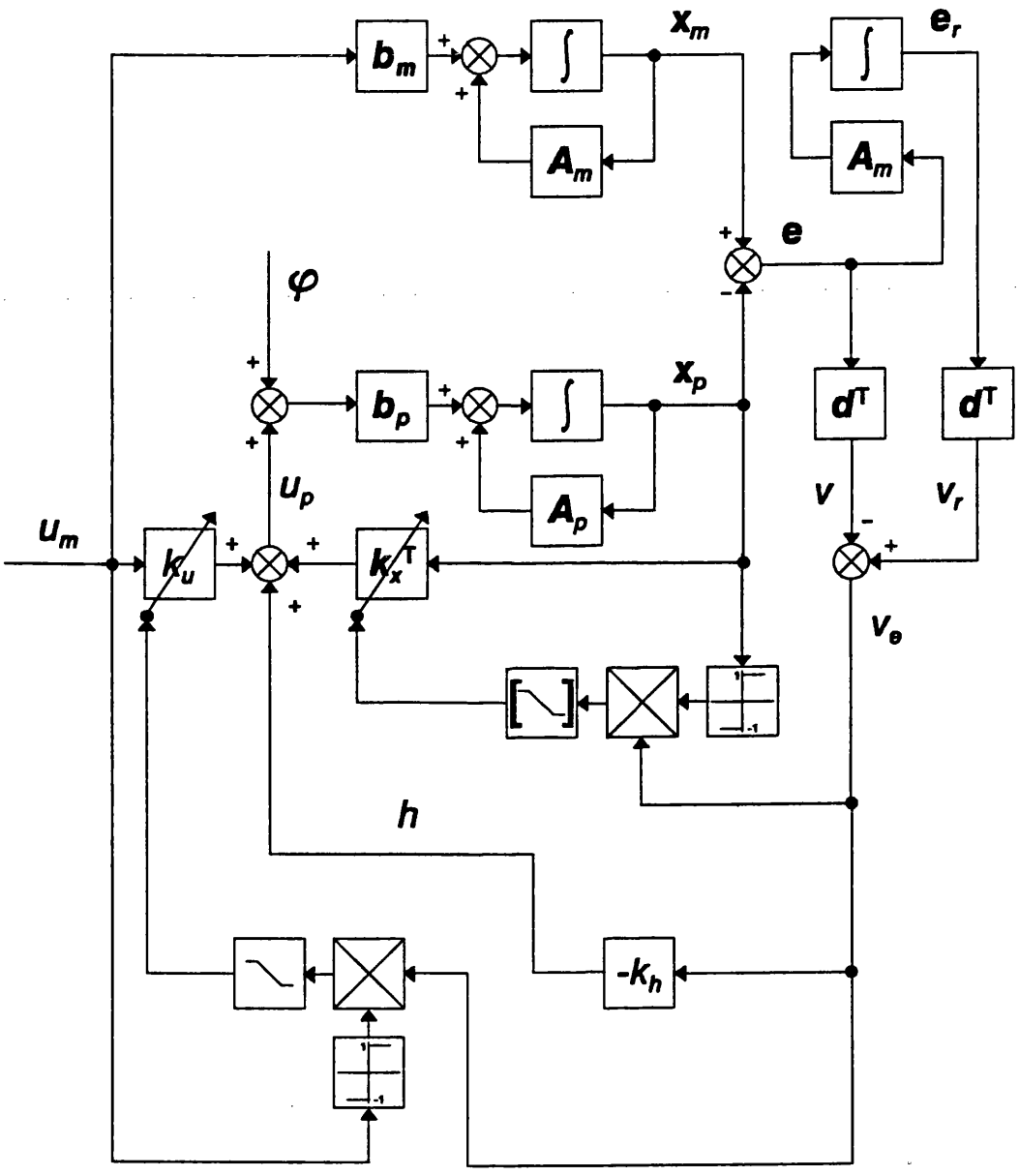


Figure 4.7 - Block diagram of the new MRAC algorithm

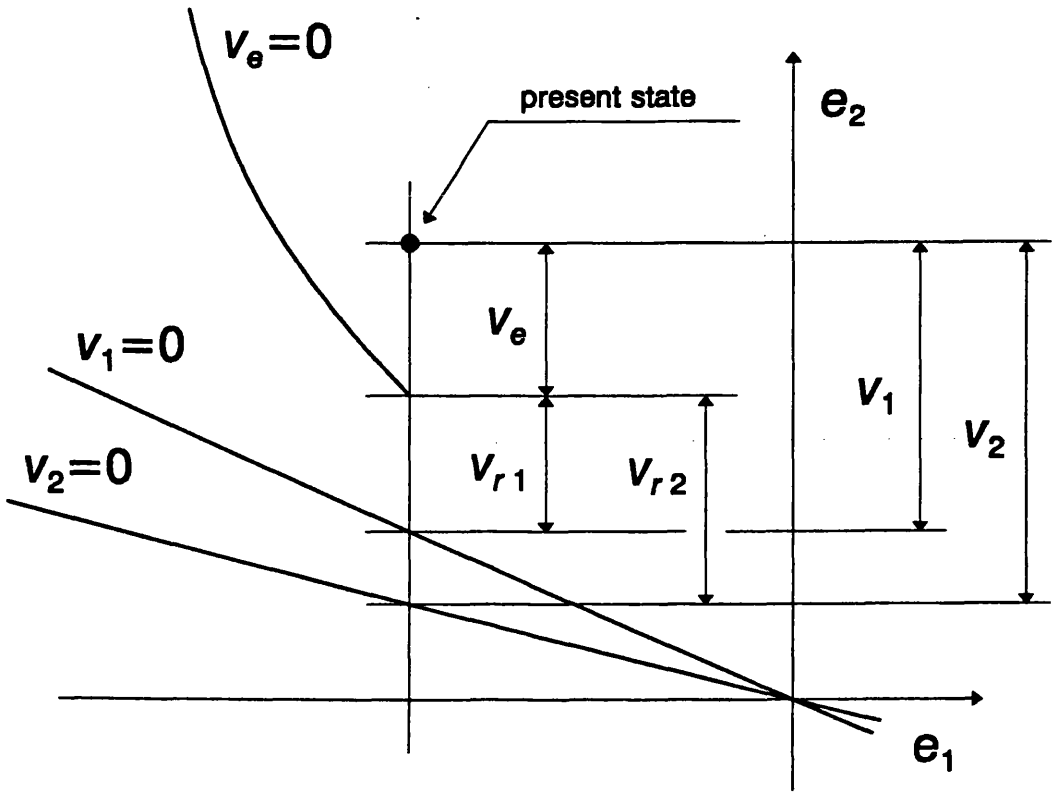


Figure 4.8 - The new sliding line $v_e=0$ and its independence from the $v=0$ datum

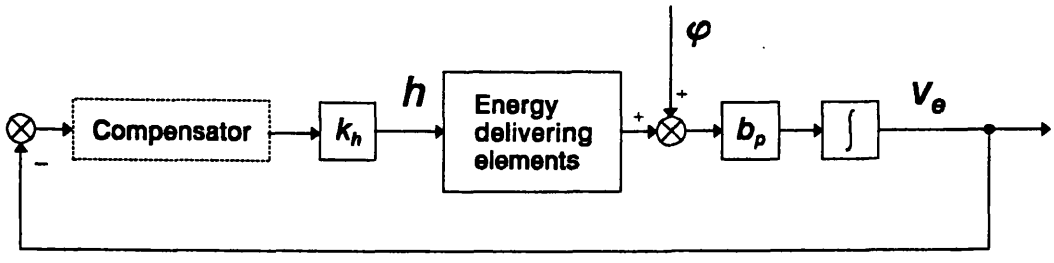


Figure 4.9 - The influence of the dynamic characteristics of the energy delivering elements on the v_e loop

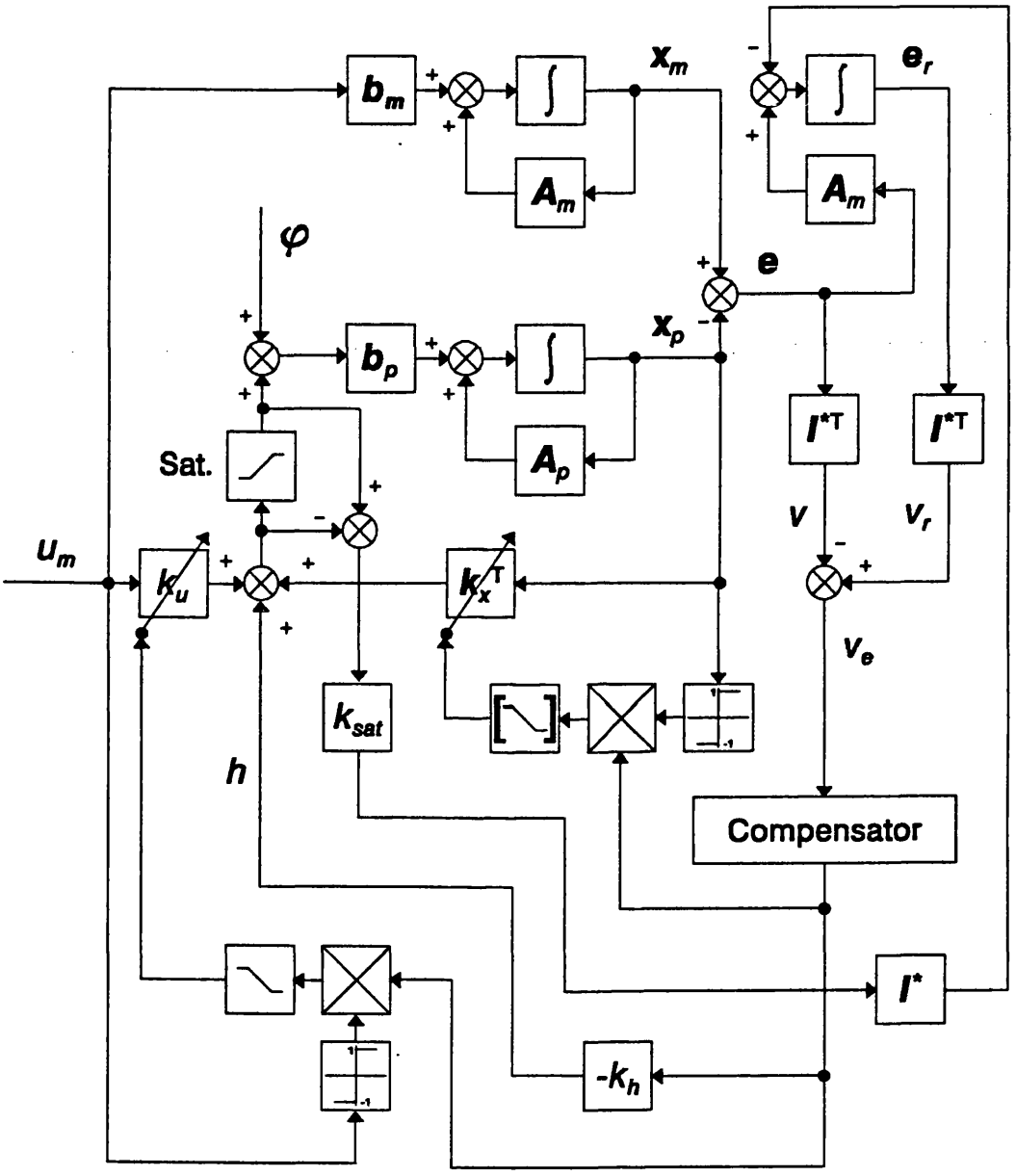


Figure 4.10 - Block diagram of the new adaptive controller, showing the v_e loop compensator and the antiwindup loop

Chapter 5 | Simulation of an Adaptively Controlled Two Axes Manipulator

The model reference adaptive control algorithm presented in the previous chapter was implemented, by a continuous time simulation, in the decentralized control of a two axis hydraulic driven manipulator. This simulation was done as a first step in the evaluation of the proposed control algorithm. It also provided some insight on the behaviour of the system and proved to be an essential tool in the choice of some controller parameters and reference model dynamics. This knowledge was subsequently used in the implementation of a similar controller on a physical system, the experimental results of which will be presented on the next chapter.

The simulation program was written using the Advanced Continuous Simulation Language (ACSL). This language works as a Fortran pre-processor, generating Fortran code, and simplifies the task of building numerical simulations of dynamic systems. Non-linear and discrete time systems are easily supported. A fourth order Runge-Kutta integration algorithm was used in the integration of the differential equations that compose the simulation. The simulation results were subsequently exported to Matlab for plot production.

5.1 The Manipulator Model

The manipulator can be modelled, as presented in Chapter 1, by the set of equations 1.3 to 1.11 which are repeated here for convenience:

$$\tau = \boldsymbol{J}(\theta)\ddot{\theta} + \boldsymbol{c}(\theta,\dot{\theta}) + \boldsymbol{g}(\theta) \tag{1.3}$$

where θ is the vector of link positions, $\boldsymbol{J}$ is the inertia matrix of the manipulator, $\boldsymbol{c}$ is a vector of coriolis and centripetal torques, $\boldsymbol{g}$ is a vector of gravity torques and τ is the vector of the links actuating torques.

In the present case of a two axis manipulator with rotary joints,

$$\tau_1 = J_{11}\ddot{\theta}_1 + J_{12}\ddot{\theta}_2 - h_c\dot{\theta}_2^2 - 2h_c\dot{\theta}_1\dot{\theta}_2 + g_1 \tag{1.4}$$

$$\tau_2 = J_{21}\ddot{\theta}_1 + J_{22}\ddot{\theta}_2 + h_c\dot{\theta}_1^2 + g_2 \tag{1.5}$$

where

$$J_{11} = m_1 l_{c1}^2 + I_1 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos\theta_2) + I_2 \tag{1.6}$$

$$J_{22} = m_2 l_{c2}^2 + I_2 \tag{1.7}$$

$$J_{12} = m_2 l_1 l_{c2}\cos\theta_2 + m_2 l_{c2}^2 + I_2 \tag{1.8}$$

$$h_c = m_2 l_1 l_{c2}\sin\theta_2 \tag{1.9}$$

$$g_1 = m_1 l_{c1} g\cos\theta_1 + m_2 g(l_{c2}\cos(\theta_1+\theta_2) + l_1\cos\theta_1) \tag{1.10}$$

$$g_2 = m_2 l_{c2} g\cos(\theta_1+\theta_2) \tag{1.11}$$

where I_1, I_2 are the centre of mass referred moments of inertia, m_1, m_2 are the masses, l_1, l_2 are the lengths and l_{c1}, l_{c2} are the distances between the centres of mass and the joints of respectively link 1 and link 2.

The manipulator model coefficients were chosen in order to represent the dynamic behaviour of the physical system used in the subsequent experimental work. So, the following set of values was used in the simulations: I_1=3Kgm^2, I_2=1.6Kgm^2, m_1=50Kg ,m_2=25Kg, l_1=0.5m, l_2=0.5m, l_{c1}=0.21m and l_{c2}=0.33m. Link working range is given by: $\theta_{1max}=2\pi/3$rad, $\theta_{1min}=\pi/6$rad, $\theta_{2max}=\pi/2$rad and $\theta_{2min}=-2\pi/3$rad (θ_1 and θ_2 are the links angular positions defined in fig.1.1).

The payload is supposed to be at the extremity of link 2. Its mass is introduced in the equations as an increase of m_2, together with a change in l_{c2}. The payload mass was allowed to change between 0 and 90Kg. Commercially available manipulators present typical maximum payload masses about 0.3 to 0.4 of the extreme link mass [45] (m_2 in the present case). That would dictate a maximum payload of 10Kg. The wider payload range used in this work effectively demonstrates the adaptation capabilities of the proposed control algorithm and is an example of what can be achieved with direct-drive hydraulic actuation. This wide payload range implies a large range of the moments of inertia of the links as seen by the actuators. Combining configuration and payload changes, J_{11max}=114Kgm^2, J_{11min}=10Kgm^2, J_{22max}=27Kgm^2 and J_{22min}=4Kgm^2.

The manipulator links are directly driven by hydraulic rotary actuators. The rotary actuators chosen for this work are vane type actuators. They present roughly the same external dimensions in both axes, although axis 1 actuator has a double vane and a cast iron body and axis 2 has a single vane and an aluminium body. So, axis 1 actuator has a volumetric displacement that is double of axis 2 actuator displacement (and for the same supply pressure twice the available torque), but less then half of axis 2 actuator angular travel: axis 1 actuator has a displacement of 124.9×10^{-6}m^3/rad and an angular travel of 100° while axis 2 actuator has a displacement of 62.4×10^{-6}m^3/rad and an angular travel of 280°.

Each hydraulic actuator is controlled by a servo-valve. The servo-valve model used in this simulation is based on the behaviour of a Dowty 4551 with a rated flow of 19 l/min at 70 bar pressure

drop. The servo-valve is current driven and a second order plus delay transfer function between reference input current and output spool position was fitted to the manufacturer's data. Although the works of de Pennington *et al.* [46] and Martin and Burrows [47] lead to the use of a third order valve model, the main difference between a second order model and the experimental results obtained by these authors shows up as an extra spool position phase lag. This extra lag, due mainly to the dynamics of the first stage of the servo-valve and non-linearities, can be accounted for with the introduction of a delay on the second order model. This model structure has the advantage of being easy to fit to the manufacturer's frequency response data. The model used has a natural frequency of 100Hz, a damping ratio of 0.85 and a delay of 0.2ms.

The servo-valves are linked to the actuators by hoses. The four hoses used are equal and each has a volume of $150\times10^{-6}m^3$. Given the presence of these hoses, the value of the effective bulk modulus of the hydraulic fluid was taken as 8000 bar as an educated guess.

From the physical model of a hydraulic actuator driving an inertial load, as presented in Appendix A, it is concluded that each axis can be described after linearization by the following third order continuous-time transfer function:

$$\frac{\mathcal{L}(\theta)}{\mathcal{L}(x_v)} = \frac{\omega_n K_q / D_{act}}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} \tag{5.1}$$

where K_q is the servo-valve flow gain, D_{act} is the hydraulic actuator volumetric displacement, ω_n is the natural frequency of the axis, ζ is the damping ratio of the axis and s is the Laplace operator.

Using the physical data presented above, it is concluded that the natural frequency of axis 1 can take values between 29 and 106 rad/s. The natural frequency of axis 2 can vary between 27 and 75 rad/s. Assuming a supply pressure of 150 bar and a minimum servo-valve pressure drop of 50 bar (which is realistic under normal work conditions but may become zero under saturation), the term $2\zeta\omega_n$ may take values between 1.4 and 194 rad/s in axis 1, and 1.2 and 166 rad/s in axis 2. The transfer

function numerator can vary between 1.6×10^6 and 58.7×10^6 rad^3/ms^3 for axis 1 and between 2.9×10^6 and 58.4×10^6 rad^3/ms^3 for axis 2.

It must be noticed that the linearized dynamic models of the axes behaviour was used for controller design purposes only. The simulation used the actual physical non-linear axes models. Nevertheless the linearized model gives a precious insight over the behaviour of the manipulator and how widely it can change.

5.2 Controller Design

The transfer function presented in equ.5.1 shows that the manipulator axes are type 1 systems. The free integration appears between velocity and position of the axis. This feature of plant dynamics can be used to simplify the controller. As the integrator has constant and known dynamic behaviour, only an adaptive velocity controller needs to be implemented, lowering by one the order of the adaptive controller, in an inner loop, enabling the use of a fixed outer loop for position control. This is how the controller is implemented in this work.

The design of a MRAC system starts with the choice of the reference model. The model must be followable by the plant. This means the PMF conditions presented in Chapter 2 must be obeyed and model bandwidth must not be to high for the plant dynamic capabilities:

- As servo-valve bandwidth is much higher than the expected axis bandwidth and the linearized transfer function between servo-valve spool position and axis velocity is a second order transfer function, the reference models used in the adaptive inner loop velocity controllers are second order models. Servo-valve dynamics behave as unmodelled parasitic dynamics. In order to obey the three PMF conditions (equ.2.8, 2.9 and 2.10), axis velocity and acceleration were chosen as state variables.

- In order to avoid large control signals that may easily drive the plant into saturation, model bandwidth should be similar to axis bandwidth. The expected axes model bandwidths should be between 1 and 10Hz (which are well below the servo-valve natural frequency of 100Hz), so that the parasitic dynamics, introduced by the servo-valve, are kept small. These values are in the same range of the axes natural frequencies, showing the adequacy of the servo-valves chosen to the desired control task.

After some trial and error studies using simulation, a model with a natural frequency of 5Hz and a damping ratio of 1 was chosen for both axis controllers. The DC gain of the model was chosen as unity, so that the output of the position outer loop controller is a reference of desired axis velocity. This choice enables the easy introduction in the controller of a feedforward of the velocity reference signal, if so desired, in order to decrease the tracking error of the system. The state space representation of the reference model is presented in the following equation:

$$\begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -987.0 & -62.83 \end{bmatrix} \cdot \begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix} + \begin{bmatrix} 0 \\ 987.0 \end{bmatrix} \cdot u_m \tag{5.2}$$

The value of k_h as well as any desired filter acting on v_e must now be chosen. The bandwidth of the adaptation loop (bandwidth of v_e) will be defined by this choice. The bandwidth of v_e must be kept inside the bandwidth of any unmodelled dynamics so that the controller does not try to adapt to high frequency modes it cannot control. In the same way the bandwidth of v_e must be smaller than the servo-valve bandwidth. As the servo-valve limits the maximum frequency at which any energy exchanges can be performed with the plant, control bandwidth is limited by servo-valve dynamic capabilities. In this continuous time simulation the only unmodelled dynamics present are due to servo-valve dynamics.

Using classical control techniques, with the help of Matlab, the value of k_h and a lead compensator were chosen as:

$$k_h \frac{(\tau_z s + 1)^2}{(\tau_p s + 1)^2} = 7.0\times10^{-6} \frac{(1.5\times10^{-3} s + 1)^2}{(0.5\times10^{-3} s + 1)^2} \tag{5.3}$$

Given the similarity between the maximum values of b_p in both axes, the two controllers use the same values of k_h and the same compensator. A Bode diagram of the open-loop frequency response

of the v_e loop, supposing a b_p value of 59×10^6 rad^3/ms^3, is presented in fig.5.1. From this Bode diagram it can be concluded that, with the choices made, the v_e loop has a minimum phase margin of 60°, a minimum gain margin of 5 and a maximum bandwidth of approximately 70Hz.

The feedback and feedforward gains, k_x and k_u, as well as the smoothing parameter μ may now be found. The gains must be calculated using equ.4.20 to 4.23. The parameter μ must be found by trial and error with the help of simulation. After several preliminary simulation studies, several conclusions emerged:

- In this particular system the contribution of the feedback path via the k_x gains is extremely small when compared with the contribution of h. It was found that adaptation time behaviour (the time behaviour of v_e) did not change when the k_x gains are omitted. So, it was decided not to use these feedback gains, in order to decrease the controller computing time when implemented digitally.

- The correctness of the gains k_u^+ and k_u^- is fundamental for a good model following performance. The simulations revealed that if this gain is omitted model following performance becomes very sluggish. It was also concluded that nothing is gained, in this system, by making k_u^+ smaller than its maximum value ($\min(b_m/b_p)$) or k_u^- greater than its minimum value ($\max(b_m/b_p)$).

- A value of μ equal to 30 rad/s^2 was found to be, in this system, a good compromise between fast adaptation with small model following errors and smooth control signals free of high frequency components that could lead to chattering.

In order to complete the definition of the adaptive velocity controller, the antiwindup loop proportional gain k_{sat} must now be found. A block diagram of the antiwindup loop is presented in fig.5.2. From this block diagram it may be concluded that the value of k_{sat} should be of the same order of b_p. After some trial and error, a value of 30×10^6 rad/m.s^3 was chosen for both axes controllers.

At this point the adaptive inner loop velocity controller is completely defined. The next step is to define the outer loop position controller. The position controller will be designed supposing perfect model following of the velocity loop. The simplest type of controller that can be used is a proportional controller. This was the controller type chosen for this work, because it leads to an easier interpretation of the performance results obtained, although any other more complex controller (as a PID or state feedback) could be used. Nevertheless the use of this simple controller enables the controller designer to do full pole placement by combining the choices of the velocity model with the proportional gain acting on position error.

So, the input signal for the adaptive velocity controller (u_m) will be given by:

$$u_m = k_p(\theta_r - \theta) \tag{5.4}$$

where θ_r is the axis reference position and θ is the actual axis position. A block diagram of the position control loop is presented in fig.5.3.

In this particular system it was decided to use a proportional gain as high as possible, subject to the restriction of maintaining the closed loop poles of the axes transfer functions real. This choice avoids any undesirable overshoots of end effector position trajectory. As the velocity models are equal in both axes, the same proportional gain was chosen for both axes. The value of this proportional gain, k_p, was chosen as 4.0 rad.s^{-1}/rad. The obtained closed loop axis transfer function is given by:

$$\frac{\mathcal{L}(\theta)}{\mathcal{L}(\theta_r)} = \frac{3948}{(s + 41.2)(s^2 + 21.6s + 95.8)} \tag{5.5}$$

where the second order term has a natural frequency of 1.6Hz and a damping ratio of 1.1.

5.3 Simulation Results

A set of simulation experiments was made in order to evaluate the performance of the proposed controller. Most of these experiments involved moving axis 1 by ±0.2 rad around its middle position, that is θ_1 = 1.309 rad, and axis 2 by ±0.2 rad around its in-line position with axis 1, that is θ_2 = 0 rad, where θ_1 and θ_2 are the links angular positions defined in fig.1.1 and link working range is given by: $\theta_{1max}=2\pi/3$rad, $\theta_{1min}=\pi/6$rad, $\theta_{2max}=\pi/2$rad and $\theta_{2min}=-2\pi/3$rad. These simulations were done with several payloads (0, 60, 90 and an overload test with 120Kg) and various axes position time references (square waves in phase, square waves in quadrature, sine waves and triangular waves). Simulations were also done in the presence of noise and servo-valve saturation. Finally, and in order to better evaluate the advantages that result from the use of the proposed adaptive controller, some simulation experiments were performed with the adaptive controller substituted by a state feedback controller.

In a first subset, two simulations were performed using square waves in phase as the axes position references, but with different manipulator payloads. The first simulation was done with null payload and in the second test maximum design payload (90Kg) was used. The position results obtained are presented in fig.5.4 for the first case and fig.5.9 for the second case. In these plots the manipulator behaviour is compared with the ideal behaviour that would result from perfect model following of equ.5.5. The obtained results lead to the conclusion that the manipulator follows the model of desired behaviour with very small errors (although the plant saturates for a very short time in the 90Kg case) and that the adaptive controller is nearly insensitive to the payload value, as expected. This small payload sensitivity can also be appreciated in the velocity signals presented in figs.5.5 (null payload) and 5.10 (90Kg) where the actual axes velocities are compared to the desired ones (the velocities that would result from PMF of equ.5.5). The adaptation signal v_a is presented in figs.5.6 and 5.11 for both axes. These signals are used to generate the disturbance cancellation signals h and the k_a gains and may be interpreted as a measure of model following error (they have the units of an acceleration). The signals presented are smooth, although oscillatory in the 90Kg case. This smoothness translates into k_a

gains, presented in figs.5.7 and 5.12, that are free of discontinuities (unless u_m is crossing zero). Consequently the generated control action signals u_p (figs.6.8 and 6.13) are both smooth and free of high frequency components that could induce a chattering behaviour.

As the gain adaptation laws are memory-less, the input gains k_u only change during transients. This behaviour is in agreement with the plant gain behaviour. One of the strongest factors influencing axis gain is the servo-valve flow gain. When axis acceleration increases the same must happen to axis pressure drop (assuming an actuator torque increment). Given that total pressure drop is always equal to the supply pressure, the servo-valve pressure drop must become smaller. This implies a decrease of servo-valve flow gain because servo-valve flow gain is proportional to the square root of its pressure drop (Appendix A). When the axis comes to rest, servo-valve flow gain raises again, returning to its initial value. The results presented in figs.5.7 and 5.12 show that the behaviour of the k_u gains are in agreement with the servo-valve flow gain behaviour, as the adaptive controller tries to maintain their product constant, during the transients, so that the gain of the plant-plus-controller system is kept equal to b_m.

With a null payload the controller finds the initial k_u gains too high and accordingly decreases their values very rapidly. On the other hand, with a payload of 90Kg the initial k_u gains are considered too low. As the axes accelerate and decelerate the values of the k_u gains must be increased and then decreased. When the axes come to rest, the k_u gains return to their initial average values.

The oscillatory behaviour observed in the v_c signals when the payload is made equal to 90Kg is due to the fact that the axes are being forced to behave as systems with natural frequencies and damping ratios higher than their own. This can only be achieved by driving the plant with hard control actions (actually, saturation occurs for a short time). As the dominant plant poles are only very lightly damped, an oscillatory control action results when the model behaviour is being superimposed to the natural plant behaviour.

A second subset of simulations was made with a payload of 60Kg and different input time reference functions like sine waves, triangular waves and square waves in quadrature:

- The position and velocity results obtained with sine (figs.5.14 and 5.15) and triangular waves (fig.5.17 and 5.18) show how accurately the manipulator axes are capable of following their reference models with these classes of input signals. The typical behaviour of a type one system is obtained. It should be noticed that a sine function is a difficult signal for an adaptive controller to follow, given the poorness of its spectrum (just one line), that causes in many cases identifiability problems (drift of adaptive gains) as discussed in Chapter 3. The behaviour of the adaptive gains k_a is presented in figs.5.16 and 5.19. In these cases, as the demanded axes movements are smooth the obtained results show that the input gains nearly stabilise around certain values. The small observed changes are broadly coherent with the axes accelerations and hence with servo-valve flow gain variations as discussed in the first subset of simulation results.

- Two other simulation studies were performed using square waves in quadrature as input references with the same ±0.2 rad amplitude but around two different manipulator positions: the first one is similar to the other tests presented up to now, but the second one was performed with link 1 moving around vertical ($\theta_1 = \pi/2$rad) and link 2 moving around horizontal ($\theta_2 = -\pi/2$rad). These simulations enable an evaluation of the disturbance rejection and decoupling capabilities achieved by the proposed adaptive control algorithm. The position and velocity results obtained are presented in figs. 5.20 and 5.21 for the first case and 5.22 and 5.23 for the second case. The simulation results presented lead to the conclusion that the proposed adaptive controller achieves a high degree of disturbance rejection (and, consequently, of decoupling) and that model following properties are virtually independent of manipulator working position.

A third subset of simulation studies was performed in order to evaluate the behaviour of the adaptive controller under plant overload and saturation, in the first case, and additive noise in the position, velocity and acceleration feedback signals, in the second case:

- In the first case the manipulator was made to move from an initial position given by $\theta_1=\pi/6$ rad and $\theta_2 = -2\pi/3$ rad to a final position where $\theta_1 = \pi/2$ rad and $\theta_2 = 0$ rad (inverted pendulum) with a payload of 120Kg as quickly as possible (step input). The obtained position results (fig.5.24) show that although the system is under severe perturbation it behaves correctly, given the physical limits imposed by the plant. This can be better appreciated in the velocity behaviour presented in fig.5.25 where some strong coupling effects are readily observed. Here it may be clearly seen that the adaptive velocity loops try to follow their reference models (equ.5.2) as best as possible, under the saturation constraints, resulting in an end of trajectory (when the system comes out of saturation) similar to the one desired (equ.5.5) although delayed in time. The effectiveness of the antiwindup loops can be appreciated in fig.5.26 where the servo-valves input signals are presented. The control actions are kept just above the saturation level, in order to provide the right initial condition to v_c when the plant desaturates. This test shows that the proposed adaptive controller works well, subjected to the physical constraints of the plant, even under severe manipulator overloading and actuator saturation.

- In the second case the reference signals were square waves with an amplitude of ± 0.2 rad. Axis 1 was moved around $\pi/2$ rad (vertical) and axis 2 around 0 rad (inverted pendulum). The manipulator had a payload of 60Kg and noise was added to the feedback signals. The criteria used to chose the RMS value of the noise amplitude (that is, the standard deviation of the noise signal) was:

position signals - RMS noise amplitude of $1 / 2^{12}$ of the full range (enabling quantization by an A/D converter with 10 ± 1LSB stable bits, assuming a Gaussian distribution and using the 3σ criteria). Given a transducer range of $\pm\pi$ rad a RMS value of 1.53×10^{-3} rad results.

velocity signals - RMS noise amplitude of $1 / 2^8$ of the full range (enabling a 6 ± 1LSB stable bits quantization). Given transducer ranges of ± 3 rad/s in axis 1 and ± 6 rad/s in axis 2, RMS noise amplitudes of 23.44×10^{-3} rad/s and 46.88×10^{-3} rad/s result for axes 1 and 2 respectively.

acceleration signals - using the same criteria of the velocity signals and supposing transducers with a range of ± 50 rad/s^2 results in a RMS noise amplitude of 390.6×10^{-3} rad/s^2.

The bandwidth of the noise sources was limited to 1kHz. This bandwidth was thought to be sufficiently broad and representative of the actual conditions. Nevertheless, as the plant is not expected to provide significant information above this frequency, filters can always be used to limit the noise bandwidth to this value without incurring information losses or large phase lags.

The obtained position and velocity results are presented in figs.5.27 and 5.28. In these figures the actual (noiseless) positions and velocities are compared with the desired ones. The feedback signals actually used in the simulation are presented in figs.5.29 (axes positions), 5.30 (velocities) and 5.31 (accelerations). The servo-valves input signals are shown in fig.5.32. The obtained results lead to the conclusion that the adaptive controller is not significantly perturbed by feedback noise, that is, its model following capabilities are maintained even under severe noise conditions.

The set of simulation results obtained lead to the conclusion that the proposed adaptive control algorithm when used in the manipulator control case results into controllers that are able to maintain the desired performance, expressed in the reference model, independent of manipulator working position, trajectory and payload. Its disturbance rejection capabilities, that result from the synthesis of a dedicated disturbance rejection signal (h), are very good and enable the use of decentralized controllers. In this way, the computing task grows only linearly with the number of manipulator axes. Only very small model following errors result, and a good manipulator performance is achieved.

At the end of Chapter 1 the possibility of using filters acting on the axes reference signals in order to improve the manipulator tracking performance was briefly mentioned. These filters must use some sort of reference time extrapolation in order to avoid differentiation of the reference signals. This may be achieved by the use of either reference preview or reference derivatives provided by the tactical controller. Examples of feedforward discrete time filters using reference preview are presented by Tomizuka [48] and Haack and Tomizuka [49]. The use of reference derivatives in order to improve the tracking performance of a manipulator may be seen in Seraji [7]. In order to be trackable, the axis reference signal must be continuous up to the plant order and saturation cannot occur. Several solutions of this tactical level control problem can be found in the literature, like the ones proposed by Coiffet [50], Fu *et al.* [1], Vukobratovic *et al.* [5] and Butler *et al.* [51]. As a profound assessment of the problems associated with the tactical control level of robot control is out of the scope of this work, a simple but effective solution was used for demonstration purposes. The desired position trajectory of each axis was computed using a fifth order spline. This choice enables the specification of initial and final axis position, velocity and acceleration and ensures the necessary trajectory continuity. This fifth order interpolator polynomial can be easily differentiated algebraically twice in order to provide the expressions needed to compute the reference velocity and acceleration signals. These velocity and acceleration reference signals were fedforward through proportional gains directly to the input of the adaptive velocity loop as presented in the block diagram of fig.5.33. Gains of 1 and 0.06 were used in both axes as respectively velocity and acceleration feedforward gains. A simulation test was made, using a payload of 60Kg, with a movement amplitude of 1 rad for axis 1 and 2 rad for axis 2. The end position was the vertical for both axes (inverted pendulum) and the initial positions were ($\pi/2$ - 1)rad for axis 1 and -2 rad for axis 2. Initial and final velocities and accelerations were zero. The obtained position and velocity simulation results are presented in figs.5.34 and 5.35. These results show that a very good tracking performance is achievable with the use of a feedforward filter even in the absence of complete plant transfer function cancellation (in the present case that would involve the feedforward of acceleration derivatives).

This simulation experiment shows how a model reference adaptive controller with good model following performance, by offering a predictable manipulator dynamic behaviour, may be associated with a reference signal feedforward filter, using reference time extrapolation, in order to achieve a high degree of tracking performance. Naturally the reference signal must be followable by the plant; that is: it must be continuous up to the order of the plant, its frequency content must be inside plant bandwidth and plant saturation must not occur.

In order to provide a controller performance reference, a last subset of simulation experiments was made. In these simulations the adaptive controller was substituted by a state feedback controller. As the state feedback controller is a well known type of fixed gain controller, the obtained results allow for an easy and direct evaluation of the improvements that can be achieved when the proposed adaptive controller is used. The gains of the state feedback controllers were chosen in a way that stability is ensured over all the manipulator working space and payload range. They were also tuned in order to achieve a single axis dynamic behaviour (when the axes are moving one at a time) equal to the axes reference model (equ.5.5), around the nominal test positions and with a null payload. The position results obtained, with test conditions equal to the ones of the first simulations subset, are presented in figs.5.36 (null payload) and 5.38 (90Kg). These results show how the obtained performance varies and becomes unacceptable when the payload is increased. The velocity results presented in figs.5.37 (null payload) and 5.39 (90Kg) just confirm how the state feedback controller is unable to compensate the highly oscillatory behaviour that emerges as an outcome of the higher payload. Strong coupling effects are also present, mainly in axis 2 that always starts moving in the wrong direction due to the reaction torques that result from the simultaneous movement of axis 1.

The poor decoupling properties of the state feedback controller are easily revealed by the simulation results presented in figs.5.40 (position) and 5.41 (velocity). This simulation test was performed with the same test conditions used in the second simulations subset, that is, with square

waves in quadrature and a 60Kg payload. These simulation results show that the state feedback controller fails to provide an acceptable performance in the presence of external disturbances, coupling effects being a particular but important case.

When these results, using the fixed gain state feedback controller, are compared with the ones that were obtained with the adaptive controller, it can only be concluded that the proposed adaptive controller offers a much improved performance, achieving a dynamic behaviour (model following) that is nearly independent of the manipulator payload and a high insensitivity to external disturbances, and so, a high degree of axes decoupling.

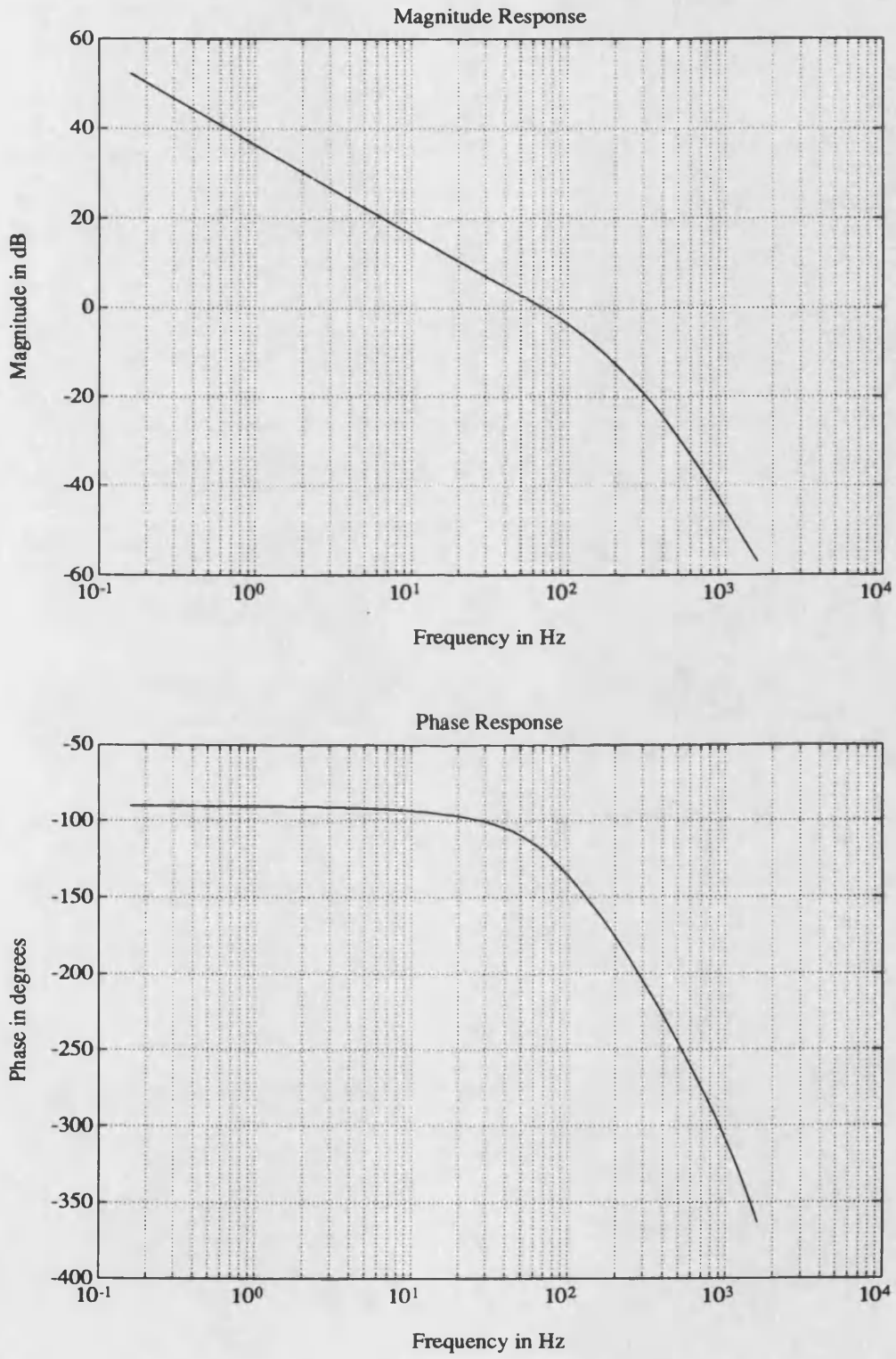


Figure 5.1 - Bode diagram of the open-loop frequency response of the v_e loop

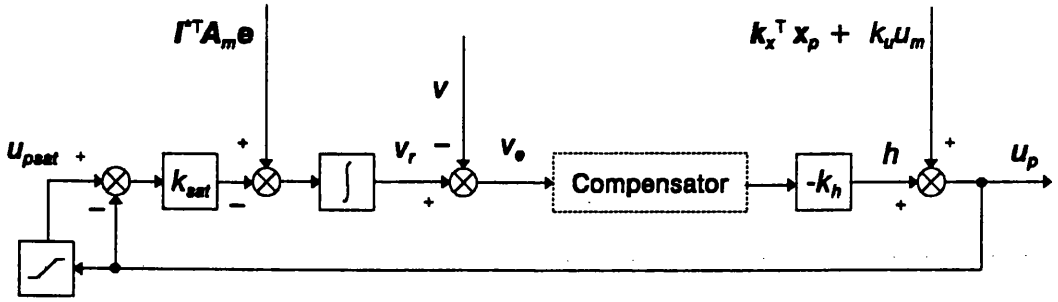


Figure 5.2 - Block diagram of the antiwindup loop

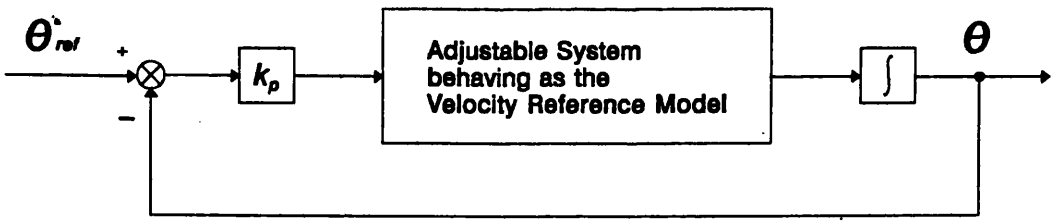


Figure 5.3 - Position control loop

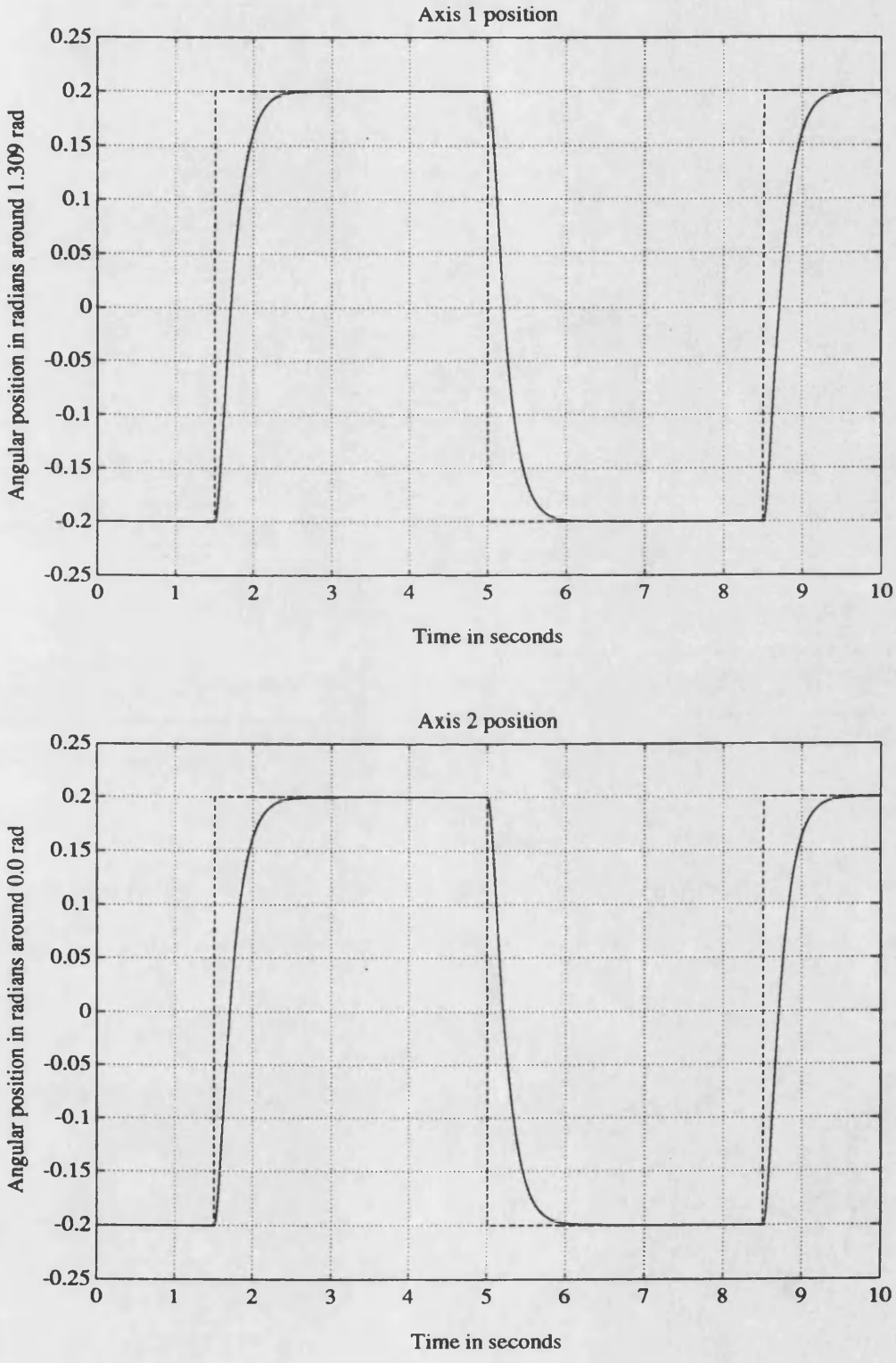


Figure 5.4 - Axes 1 and 2 position results (square wave, null payload)
---- reference ···· model —— axis

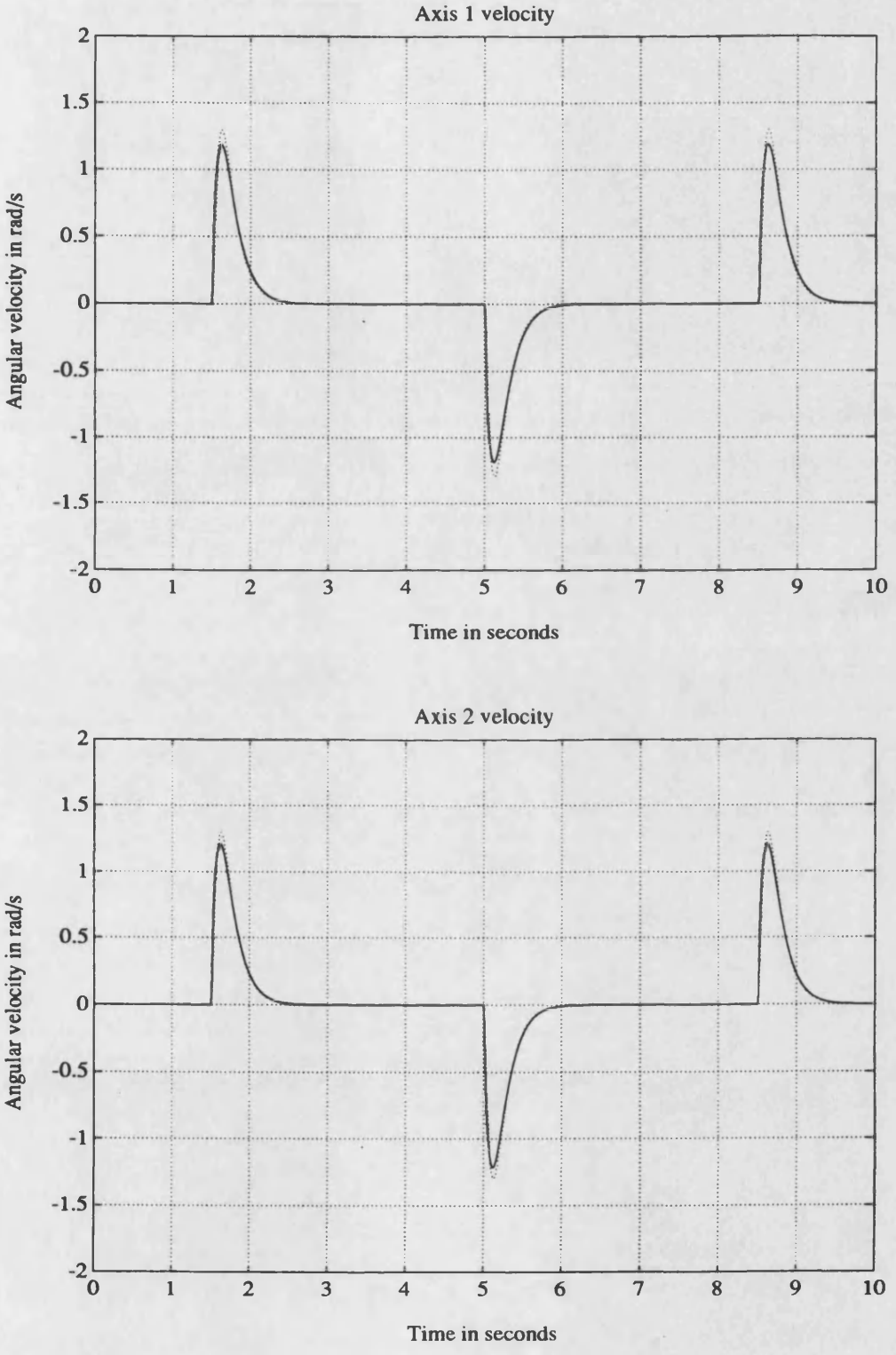


Figure 5.5 - Axes 1 and 2 velocity results (square wave, null payload)
···· model —— axis

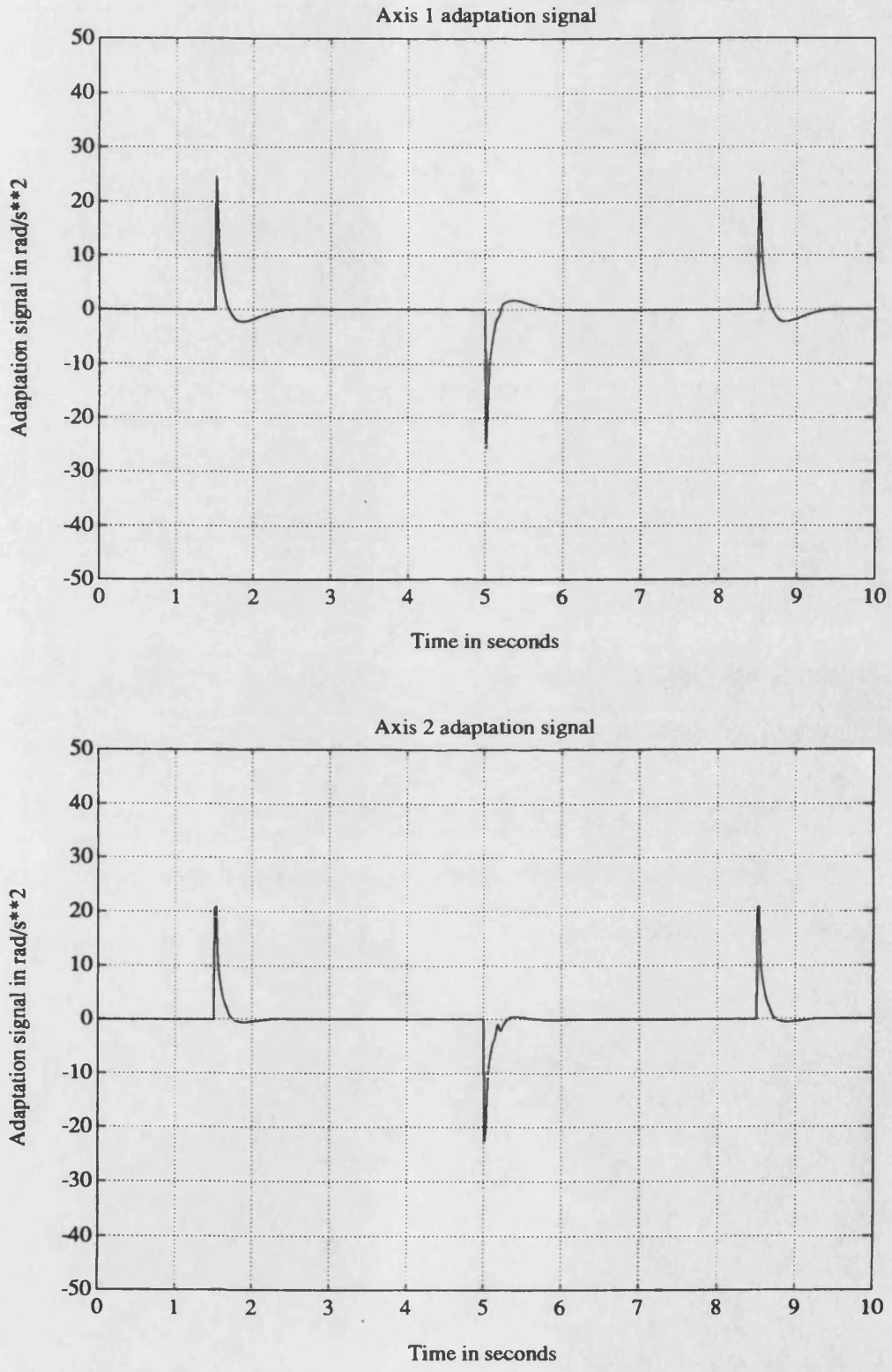
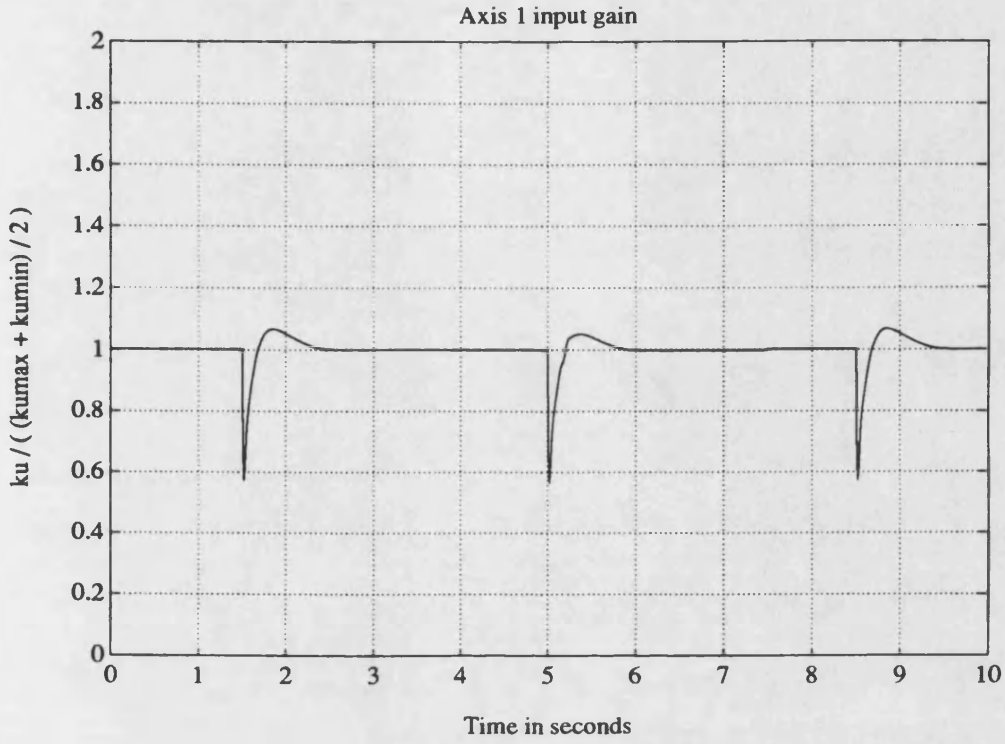


Figure 5.6 - Axes 1 and 2 adaptation signals, v_a (square wave, null payload)



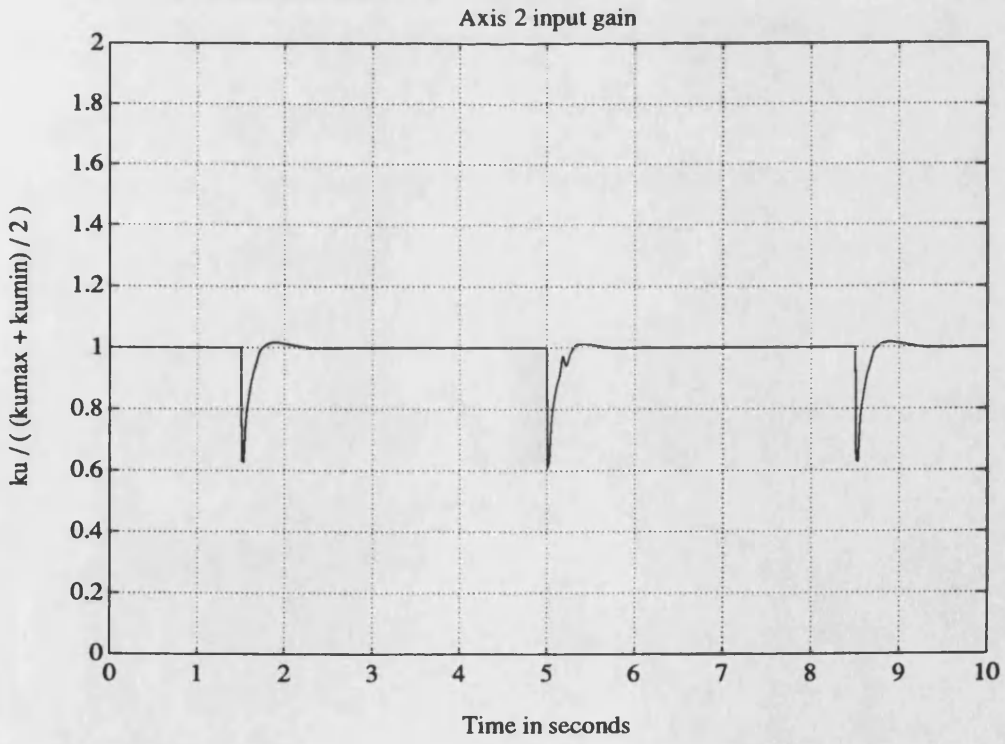


Figure 5.7 - Axes 1 and 2 adaptive input gains, k_u (square wave, null payload)

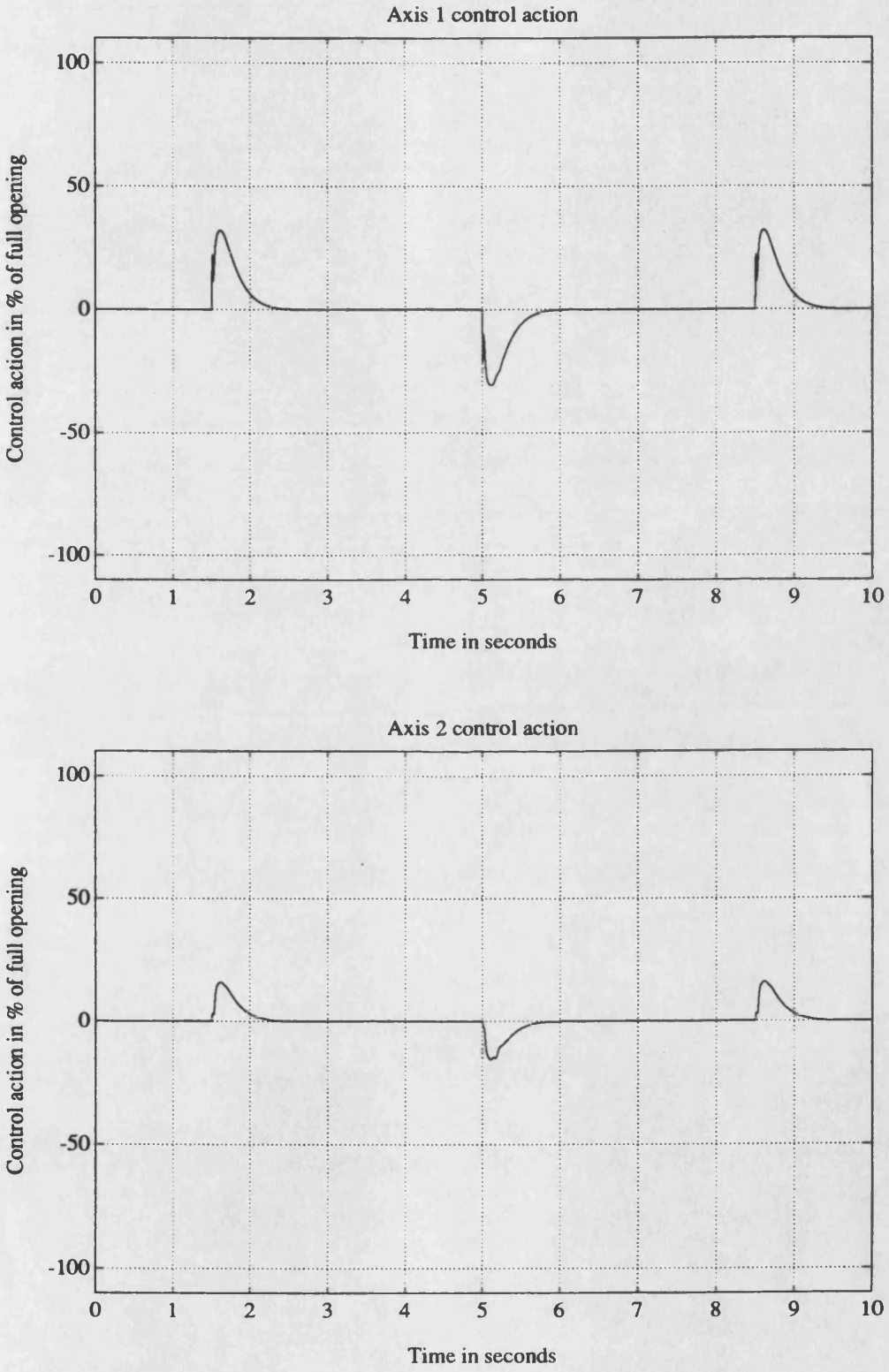


Figure 5.8 - Axes 1 and 2 control actions, u_p (square wave, null payload)

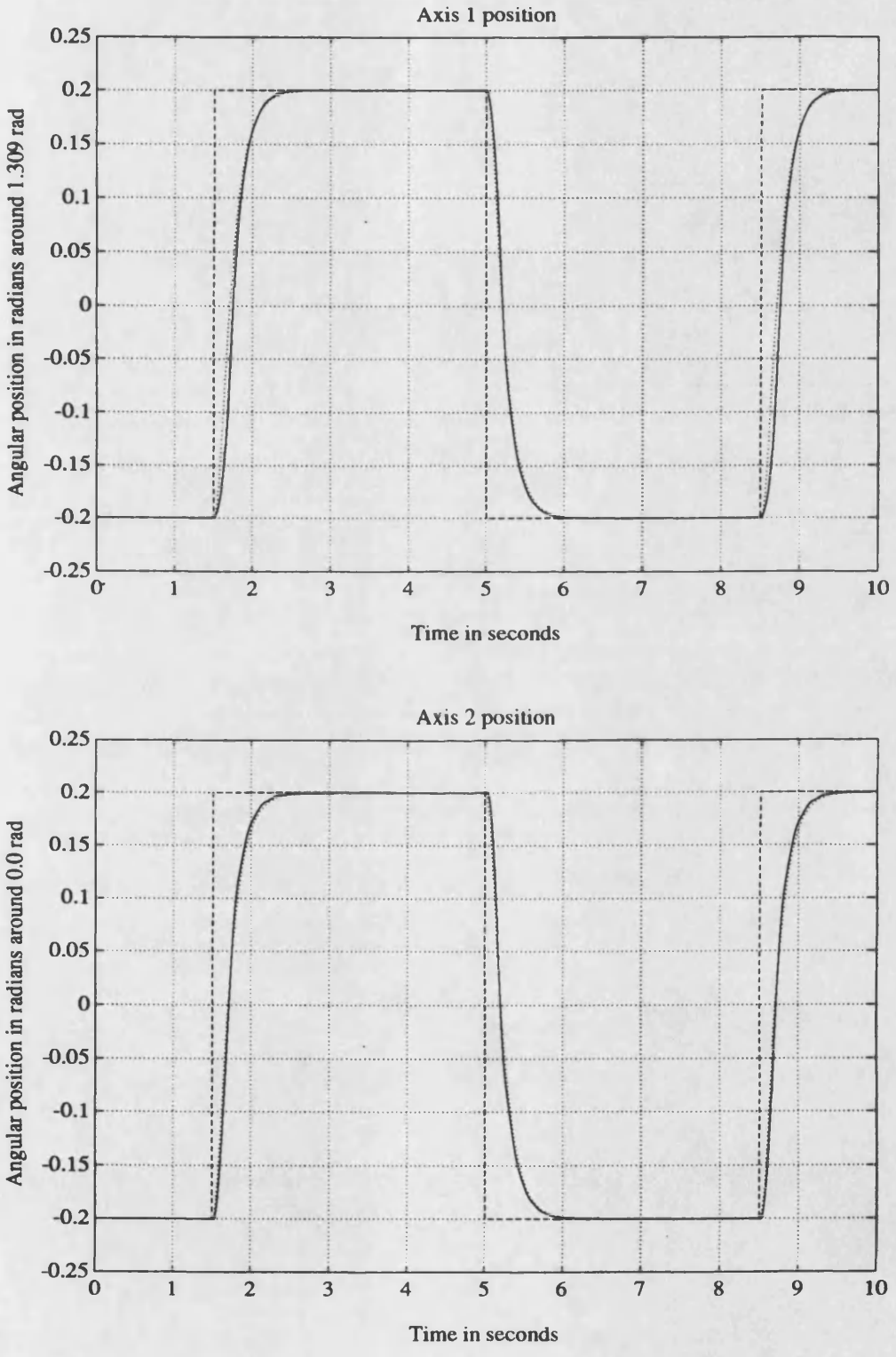


Figure 5.9 - Axes 1 and 2 position results (square wave, 90Kg payload)
---- reference ···· model —— axis

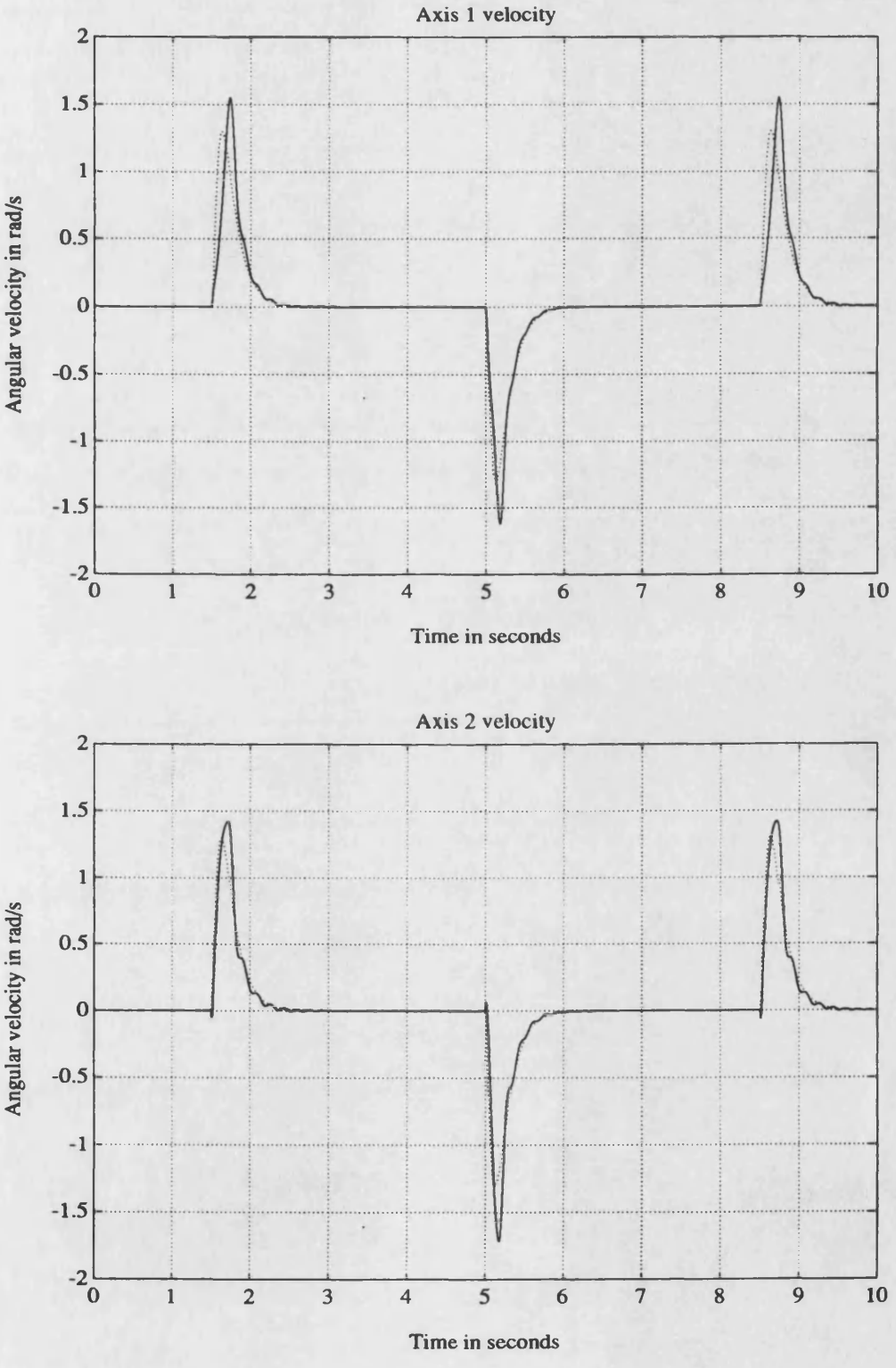


Figure 5.10 - Axes 1 and 2 velocity results (square wave, 90Kg payload)
···· model —— axis

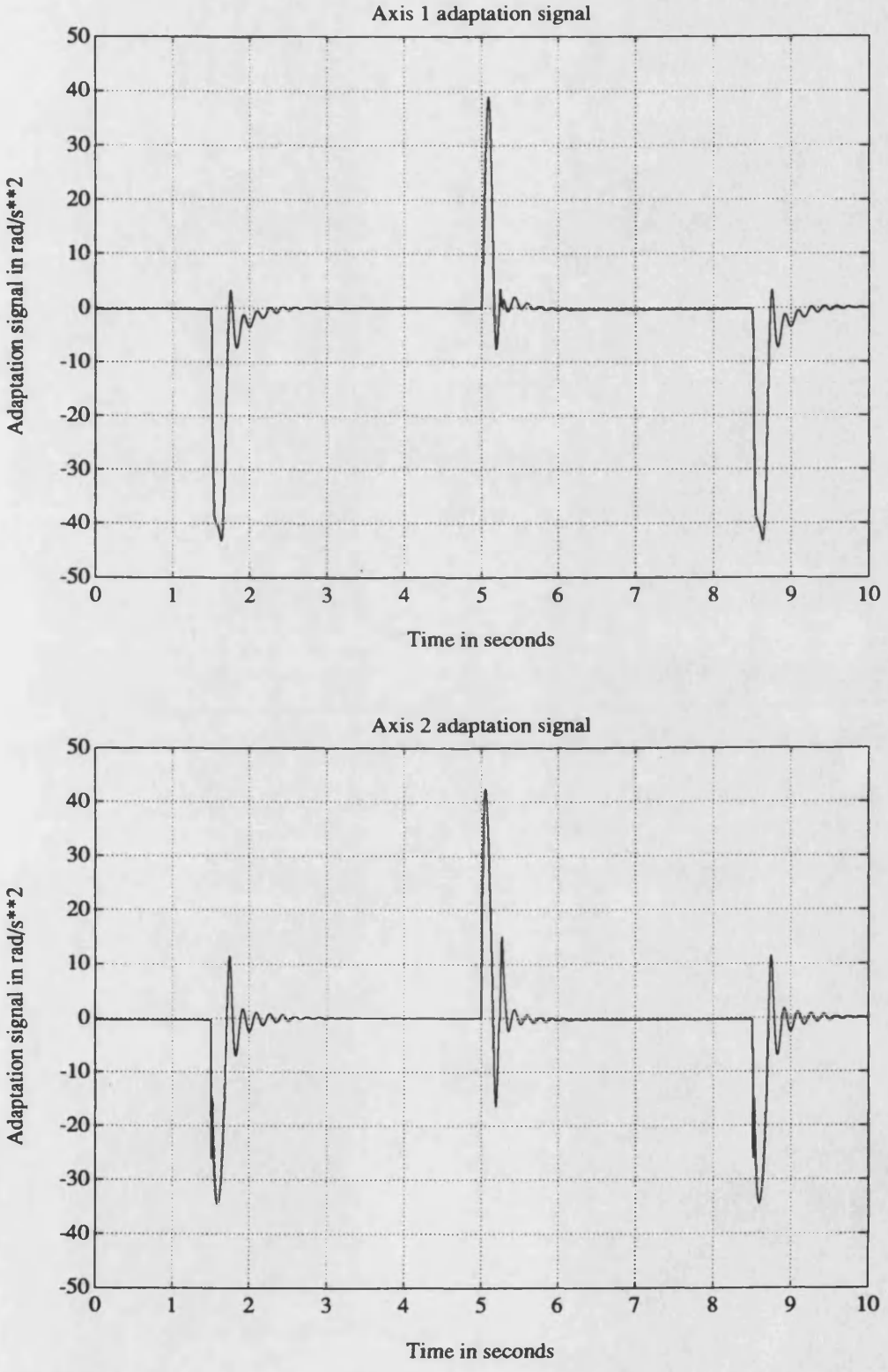
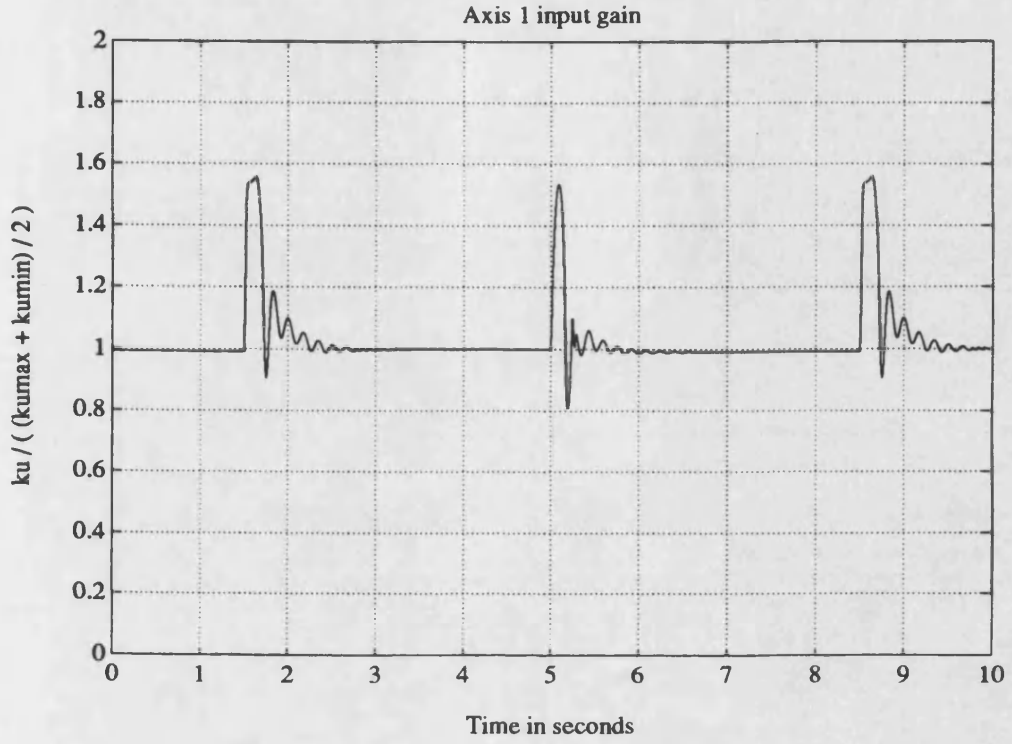


Figure 5.11 - Axes 1 and 2 adaptation signals, v_e (square wave, 90Kg payload)



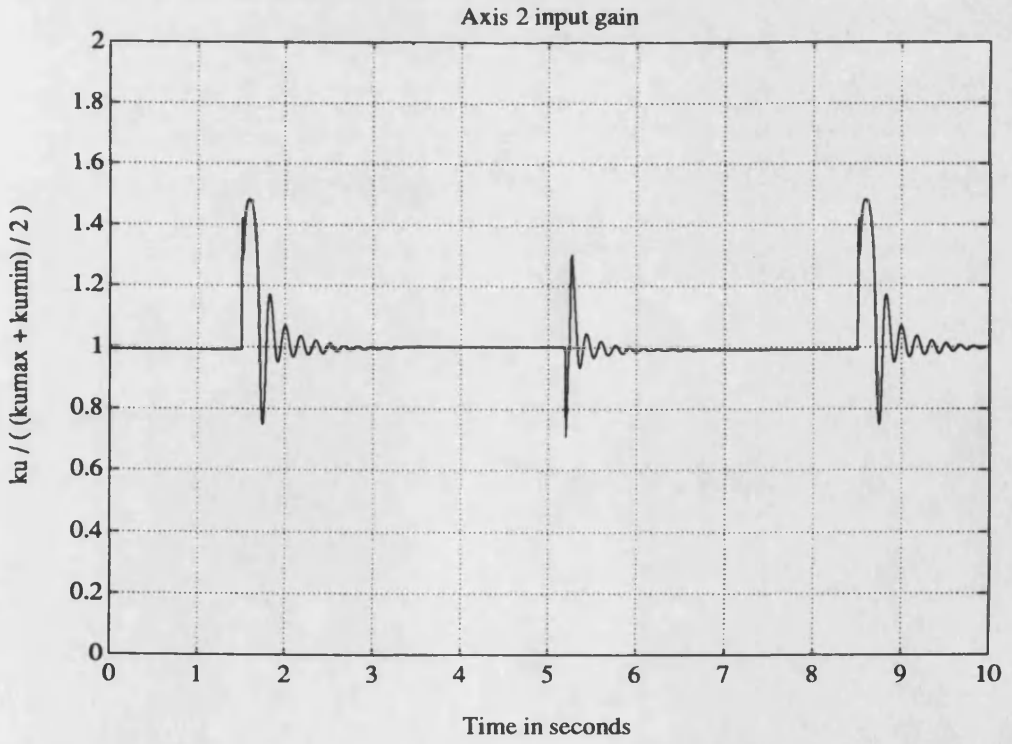


Figure 5.12 - Axes 1 and 2 adaptive input gains, k_u (square wave, 90Kg payload)

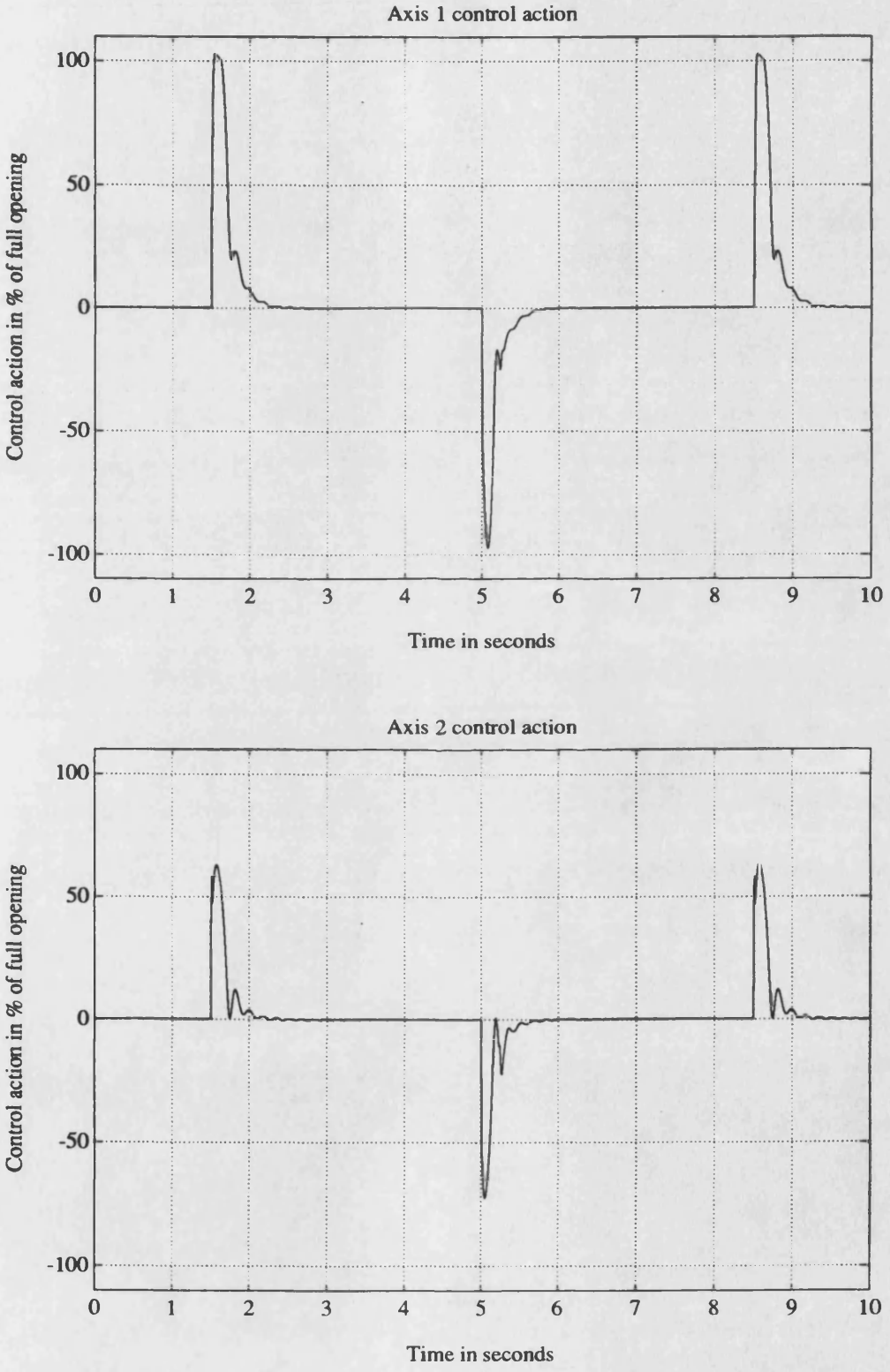


Figure 5.13 - Axes 1 and 2 control actions, u_p (square wave, 90Kg payload)

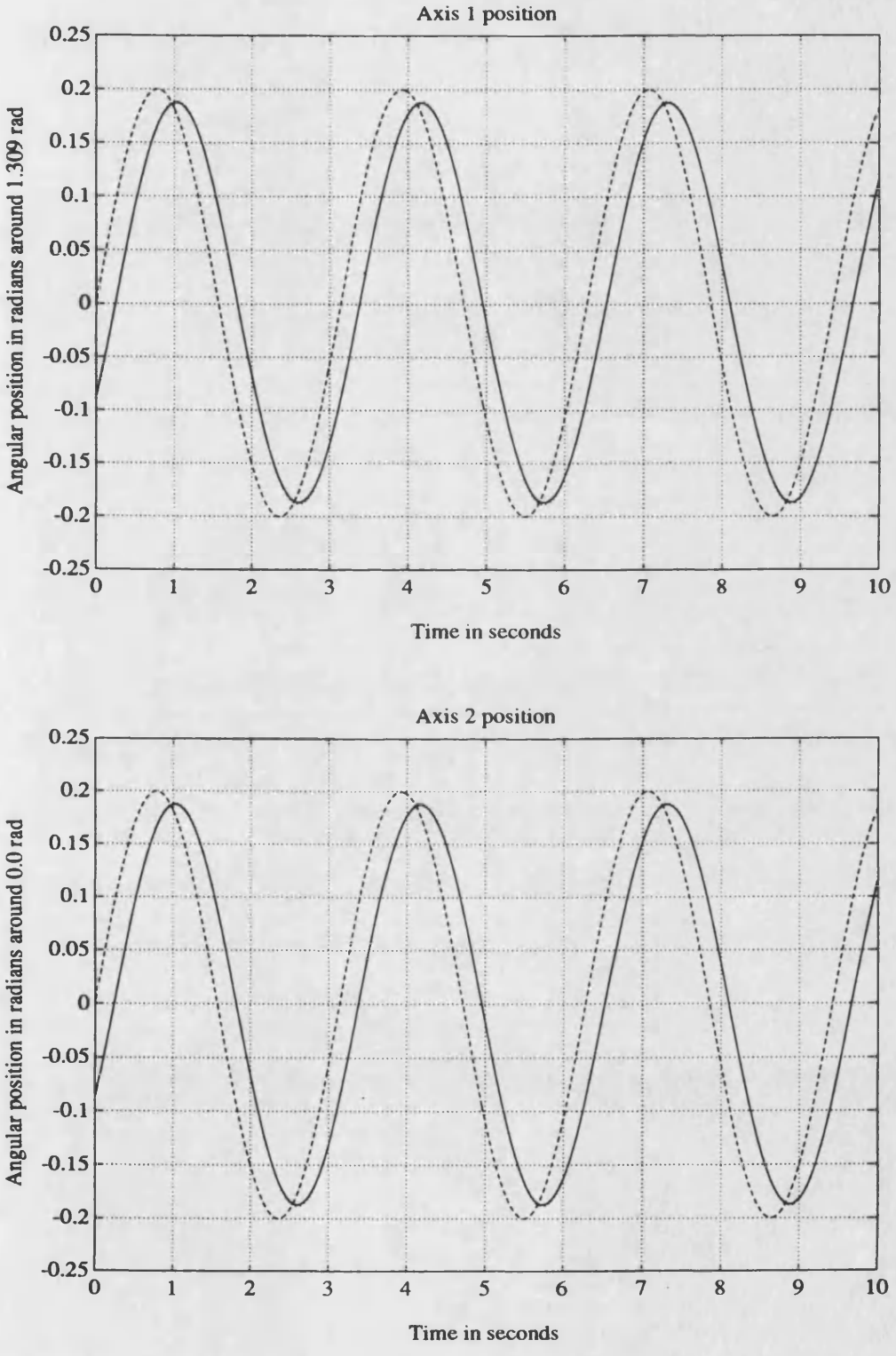


Figure 5.14 - Axes 1 and 2 position results (sine wave, 60Kg payload)
---- reference ···· model —— axis

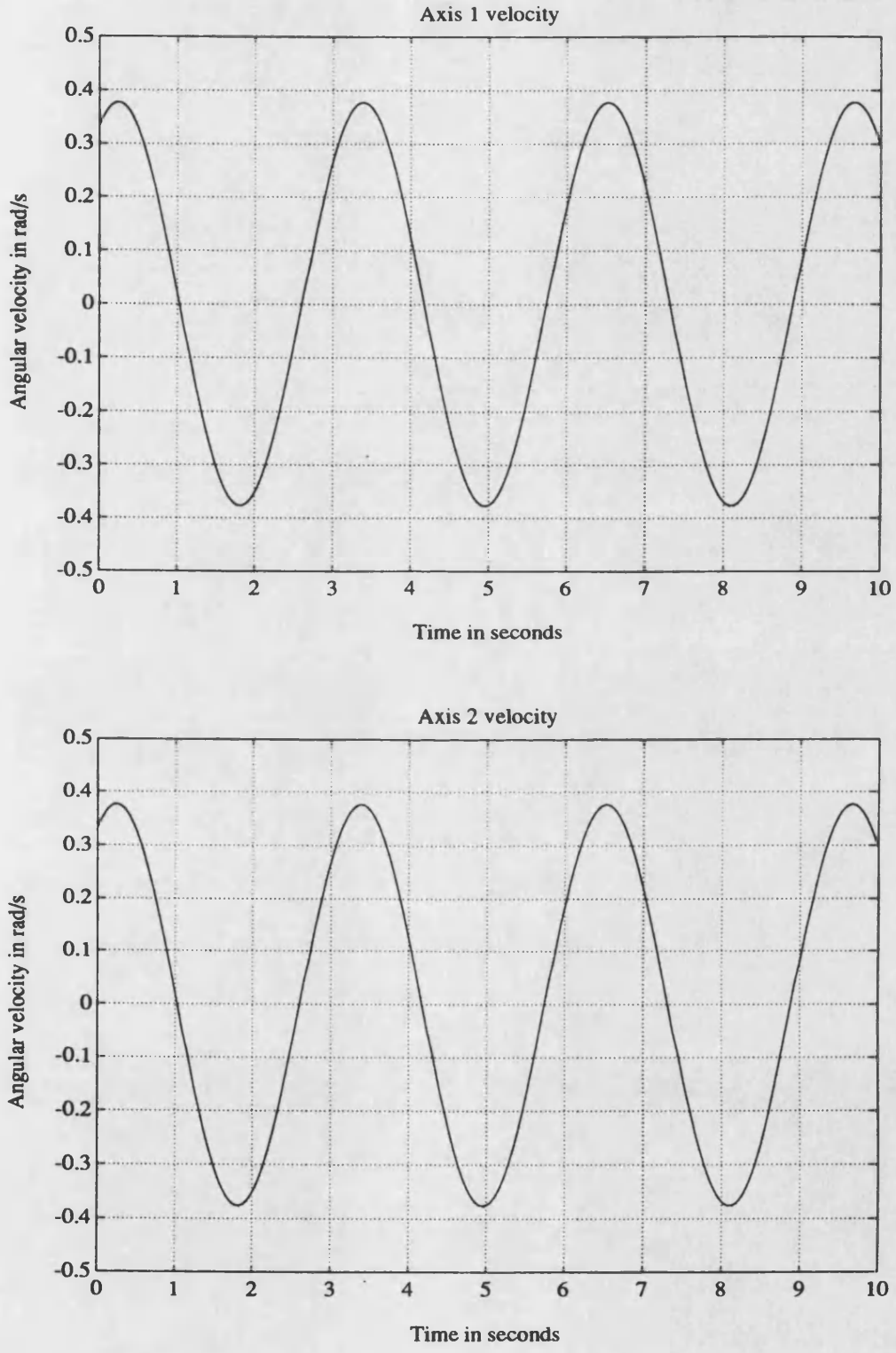
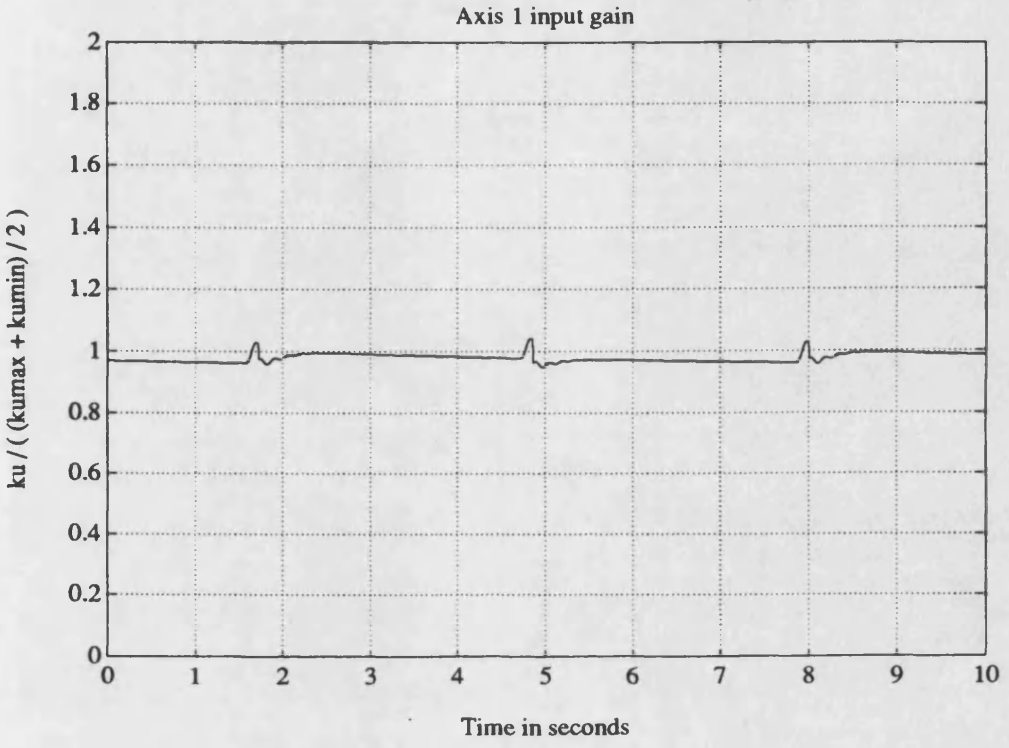


Figure 5.15 - Axes 1 and 2 velocity results (sine wave, 60Kg payload)
···· model —— axis



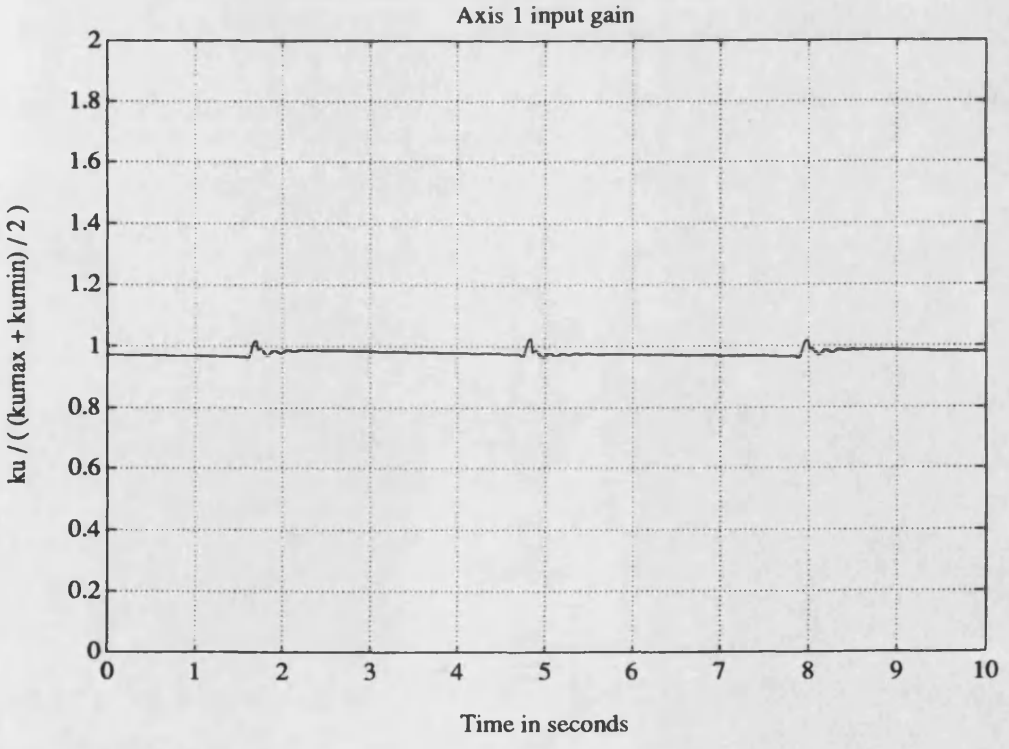


Figure 5.16 - Axes 1 and 2 adaptive input gains, k_u (sine wave, 60Kg payload)

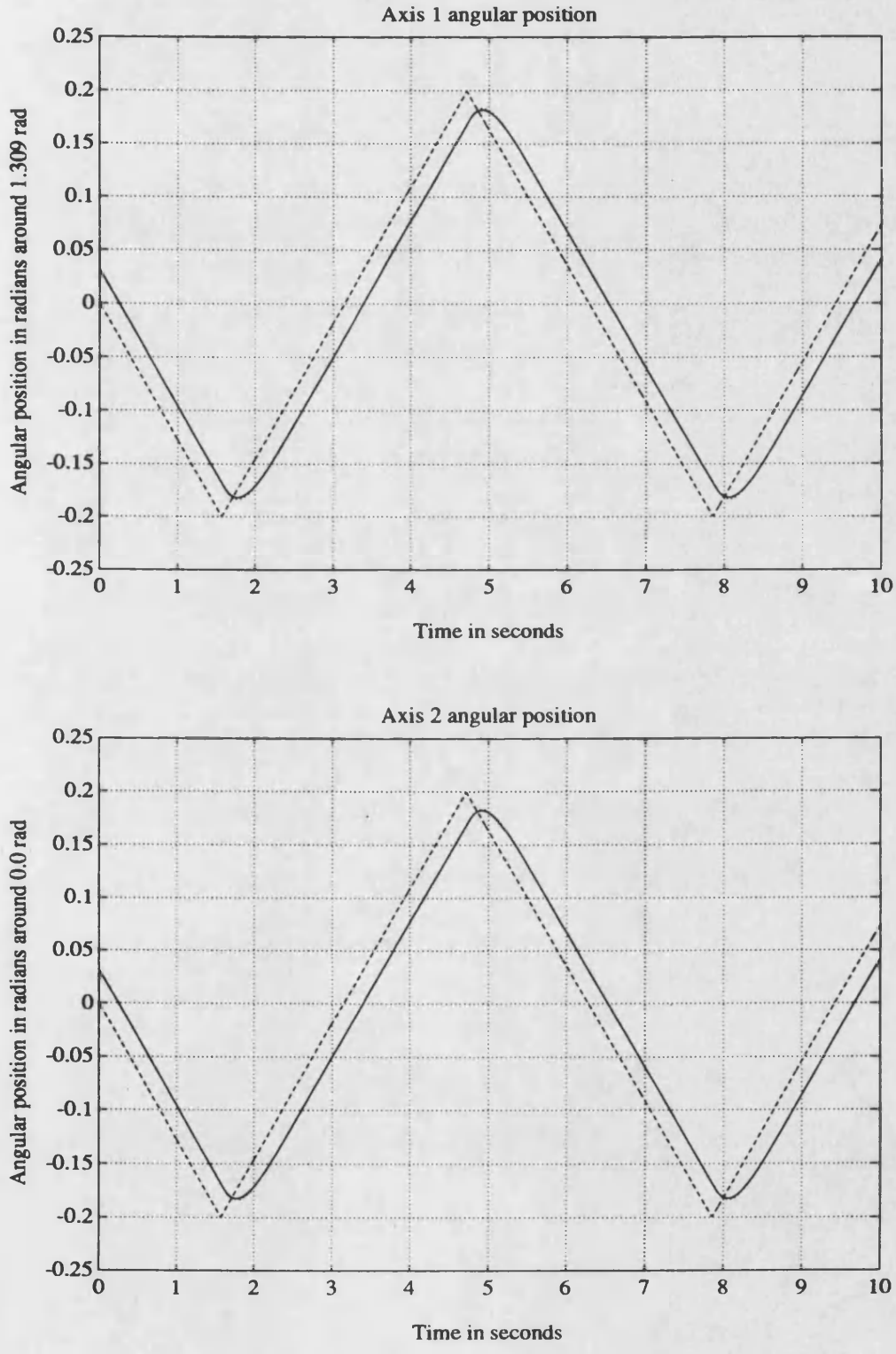


Figure 5.17 - Axes 1 and 2 position results (triangular wave, 60Kg payload)
---- reference ···· model —— axis

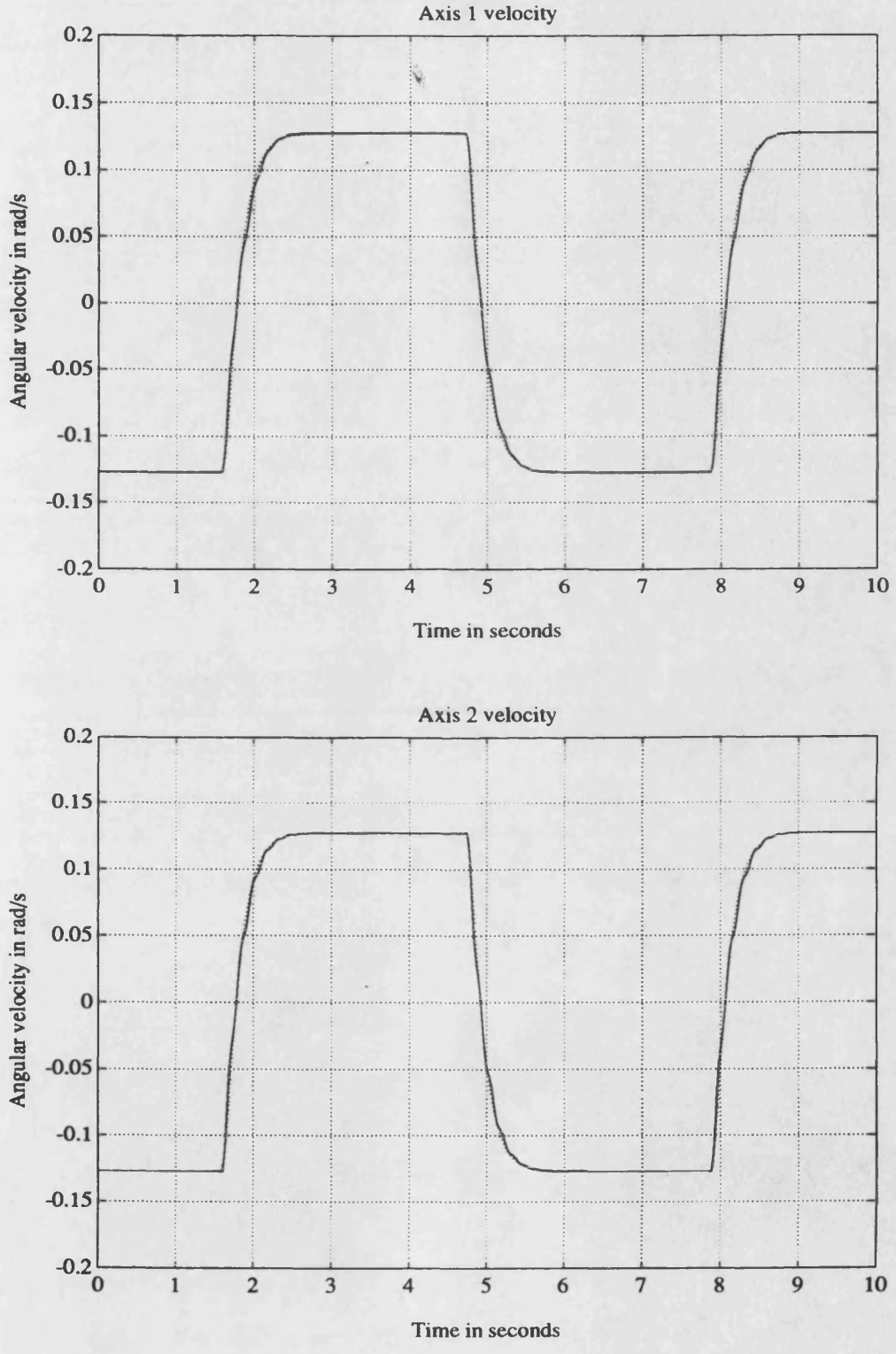
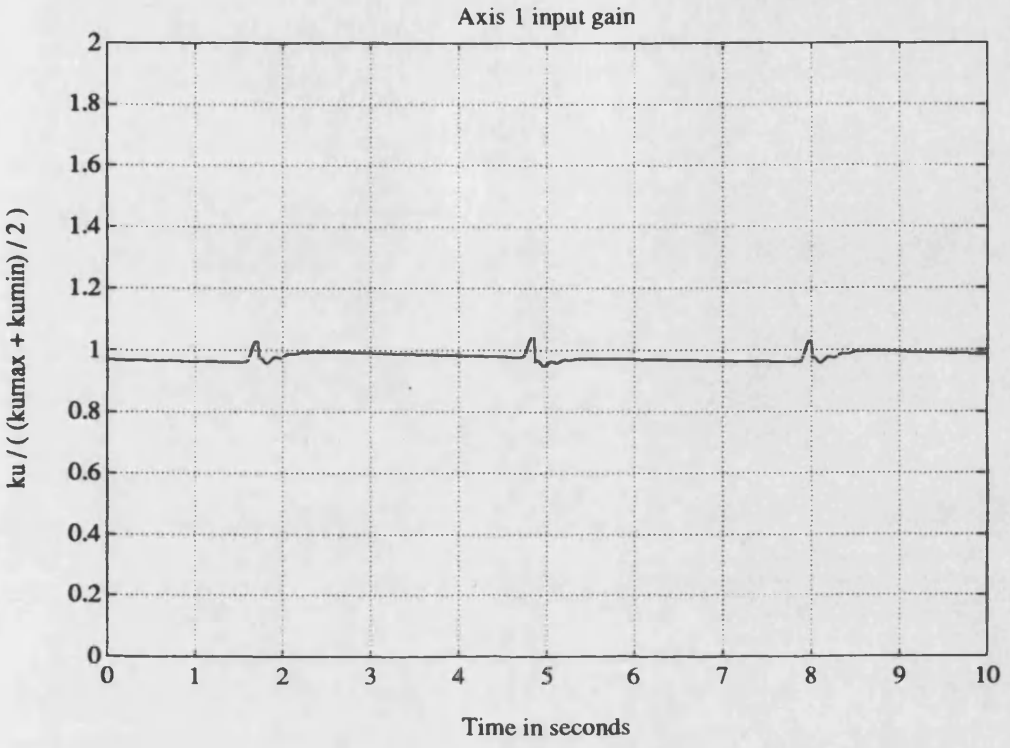


Figure 5.18 - Axes 1 and 2 velocity results (triangular wave, 60Kg payload)
···· model —— axis



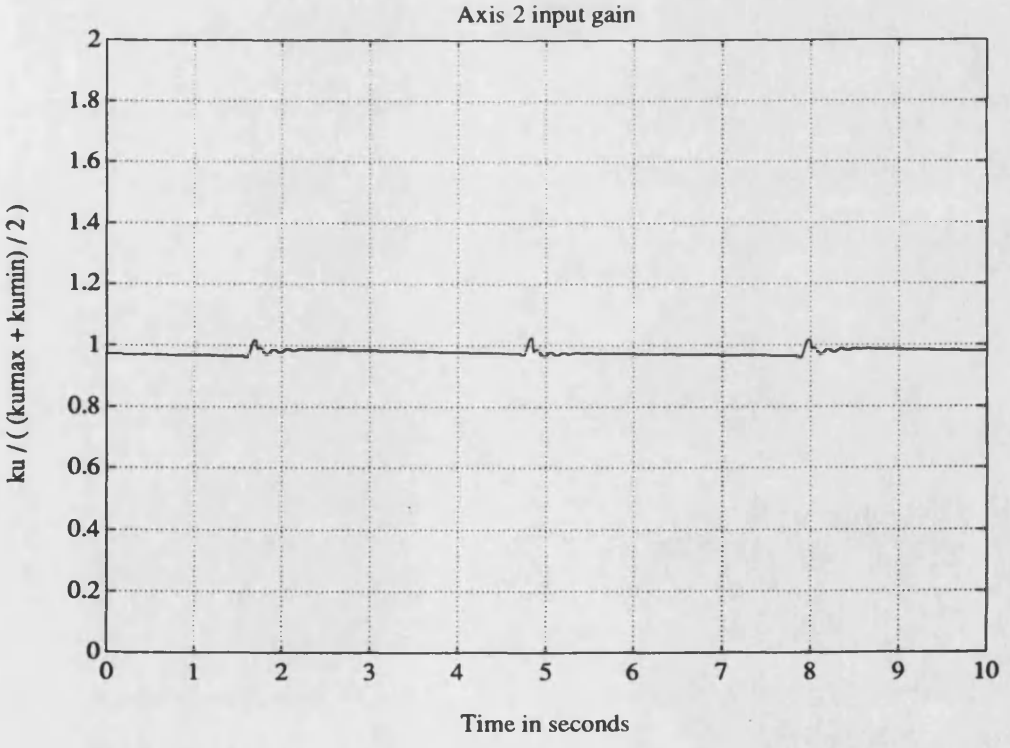
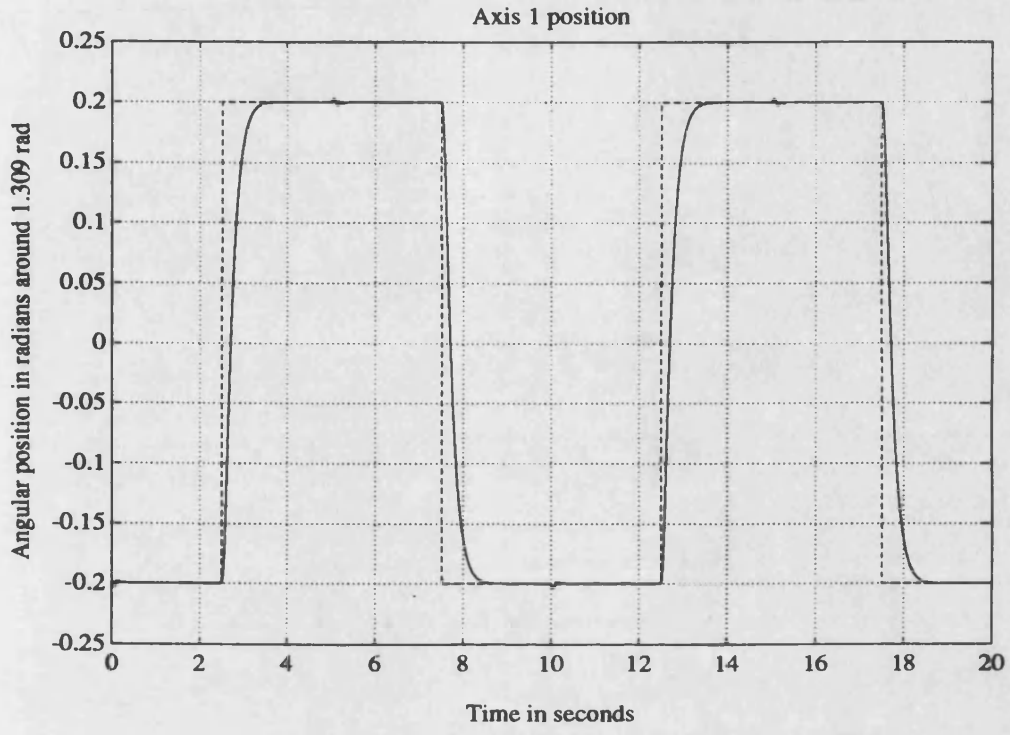


Figure 5.19 - Axes 1 and 2 adaptive input gains, k_u (triangular wave, 60Kg payload)



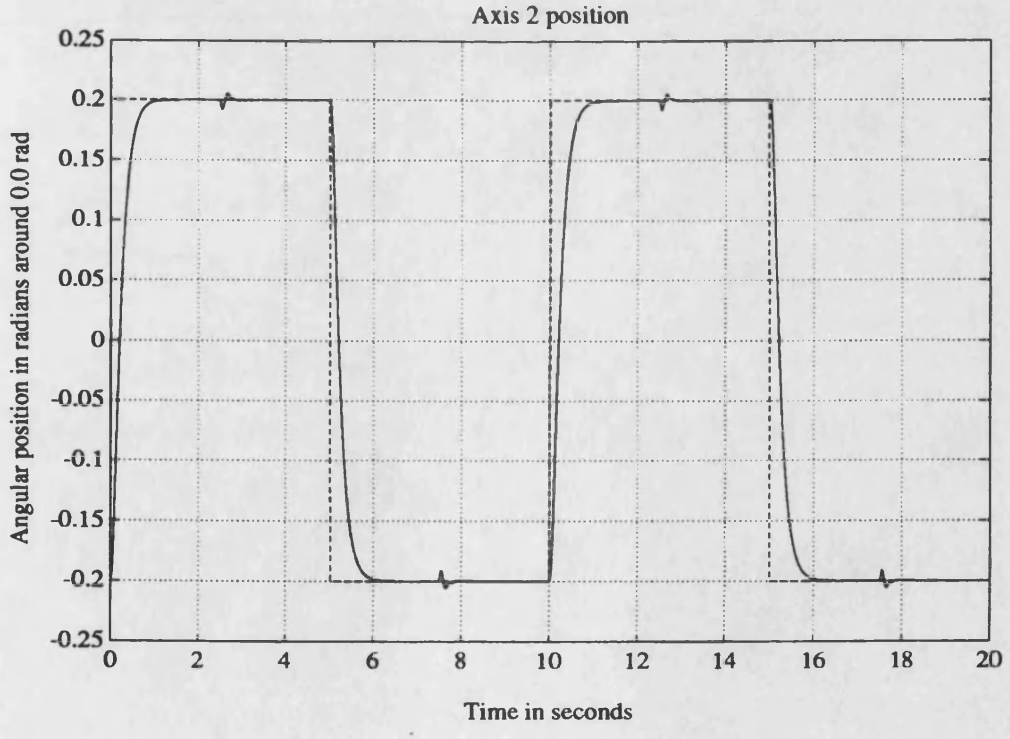


Figure 5.20 - Axes 1 and 2 position results (sq. waves in quadrature - 1st case, 60Kg)
---- reference ···· model —— axis

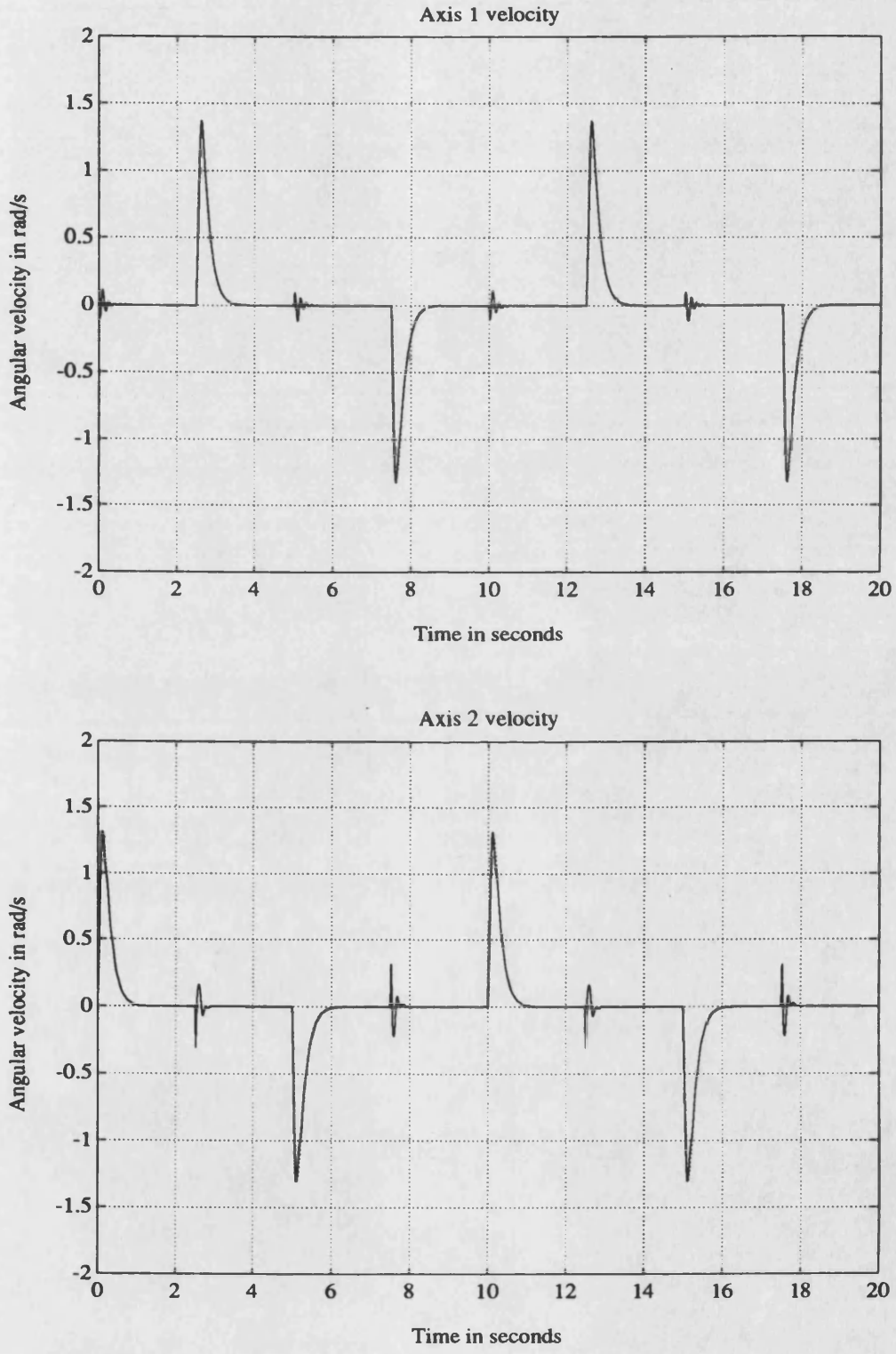


Figure 5.21 - Axes 1 and 2 velocity results (sq. waves in quadrature - 1st case, 60Kg)
···· model —— axis

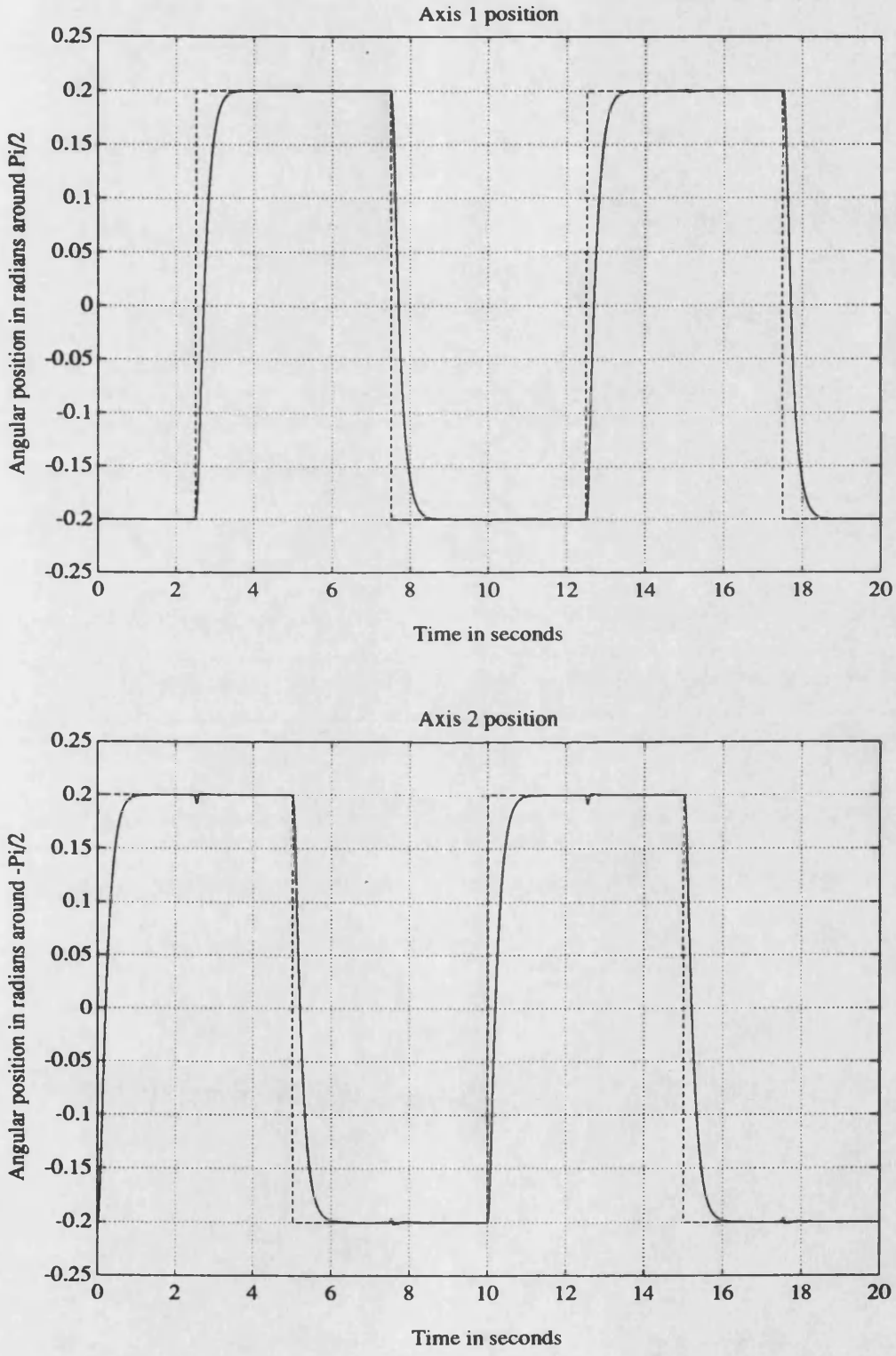


Figure 5.22 - Axes 1 and 2 position results (sq. waves in quadrature - 2nd case, 60Kg)
---- reference ···· model —— axis

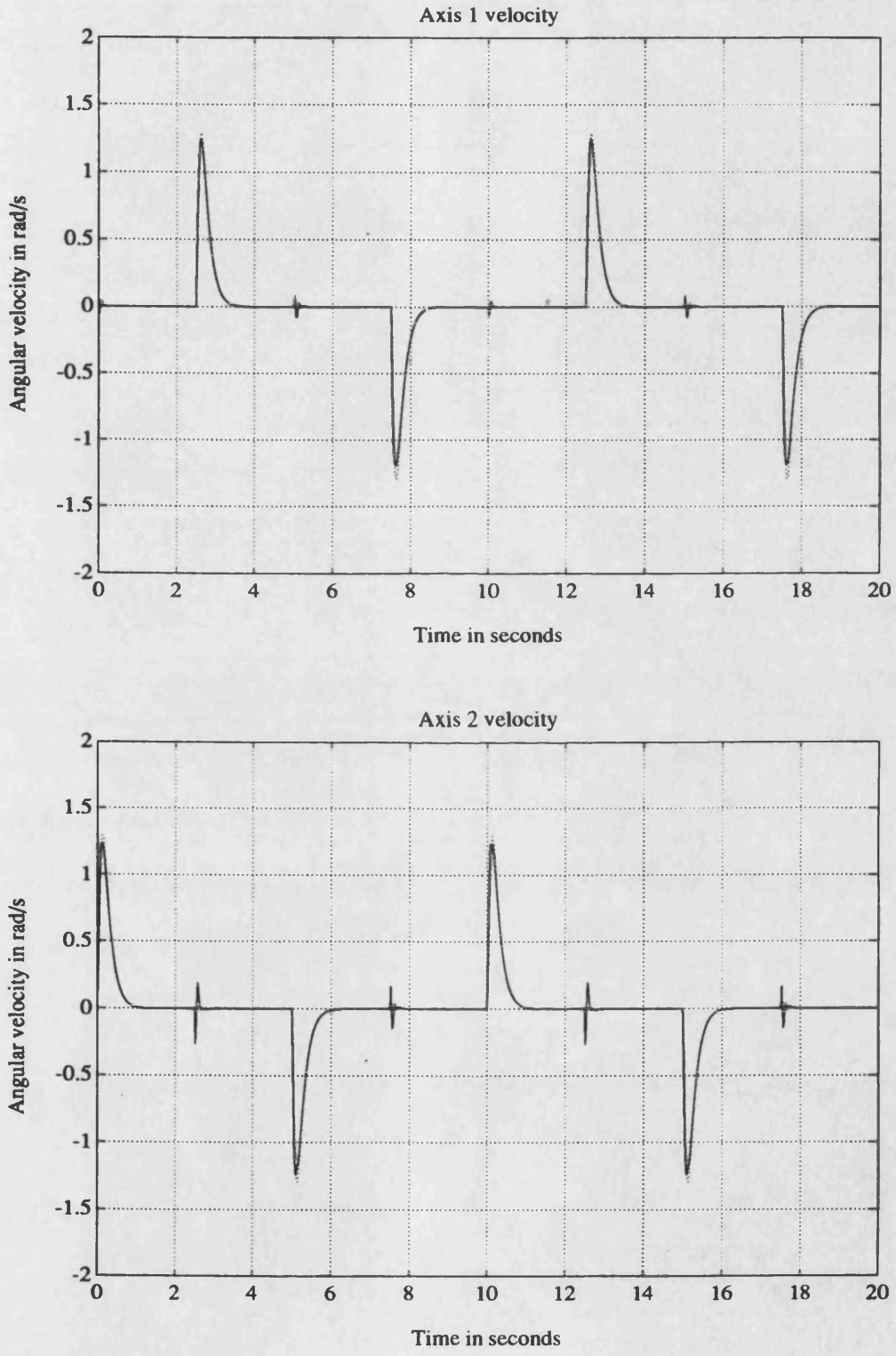


Figure 5.23 - Axes 1 and 2 velocity results (sq. waves in quadrature - 2nd case, 60Kg)
···· model —— axis

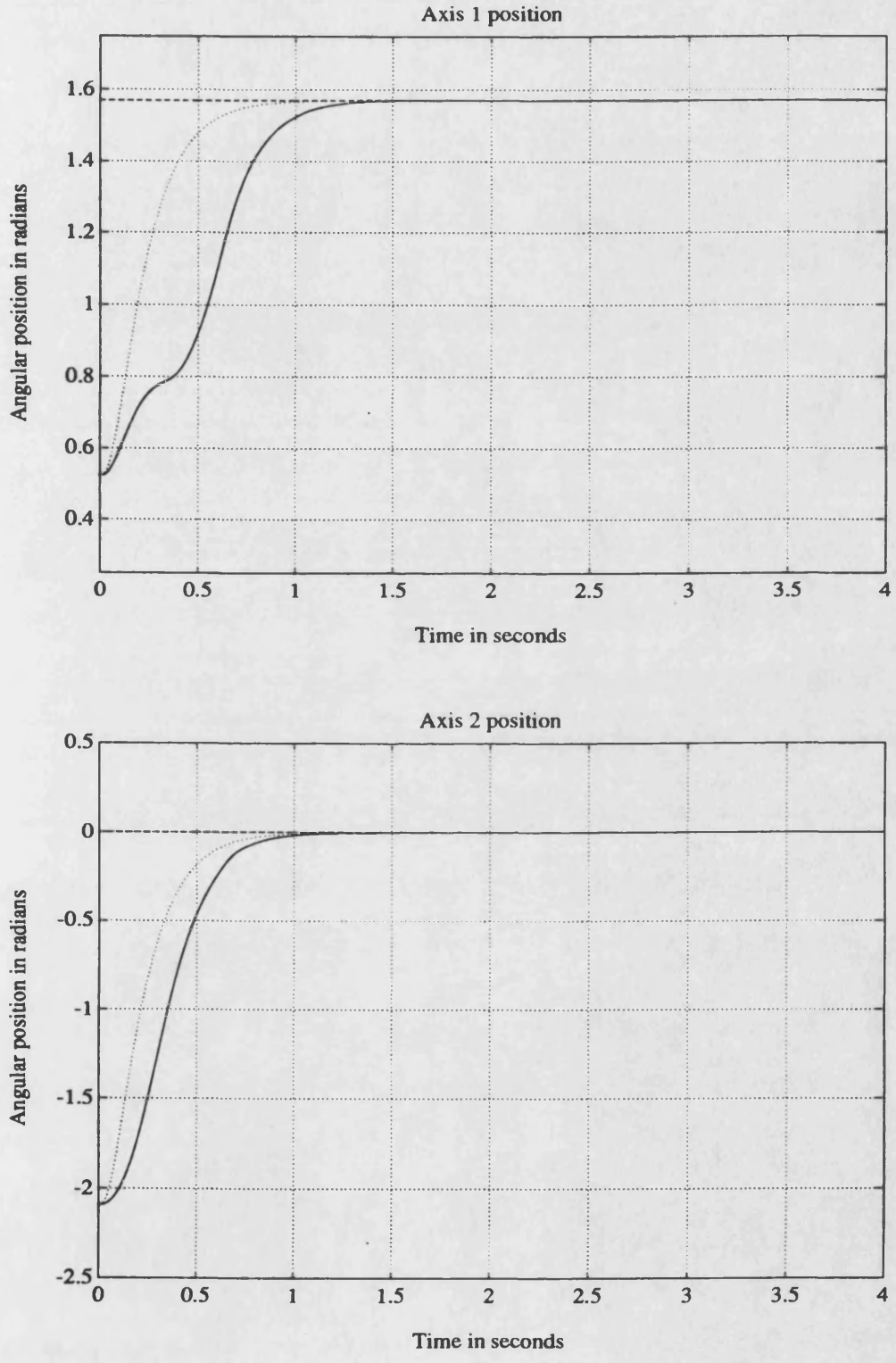


Figure 5.24 - Axes 1 and 2 position results (large step, 120Kg payload)
---- reference ···· model —— axis

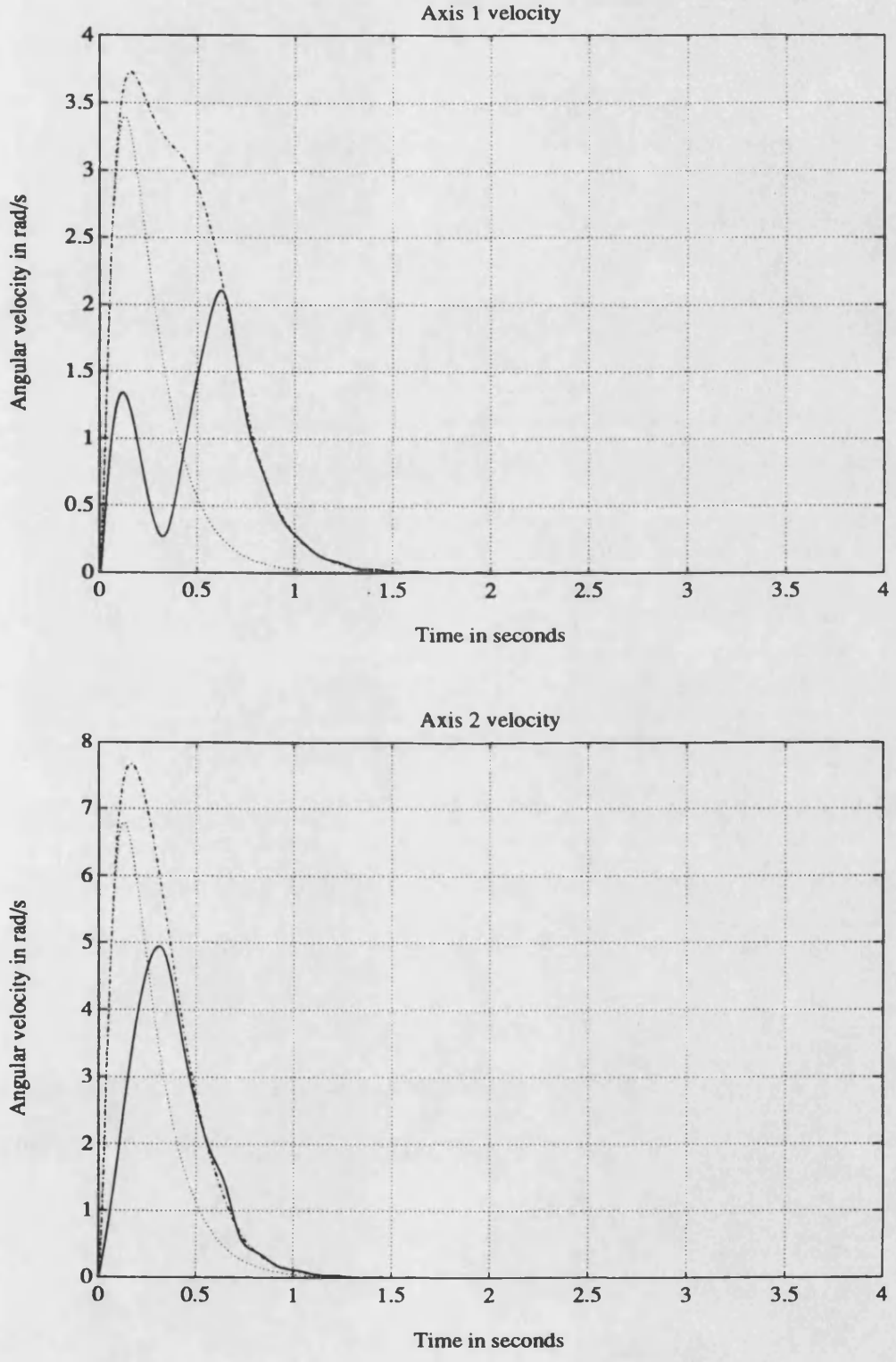


Figure 5.25 - Axes 1 and 2 velocity results (large step, 120Kg payload)
···· model — · — · vel. model —— axis

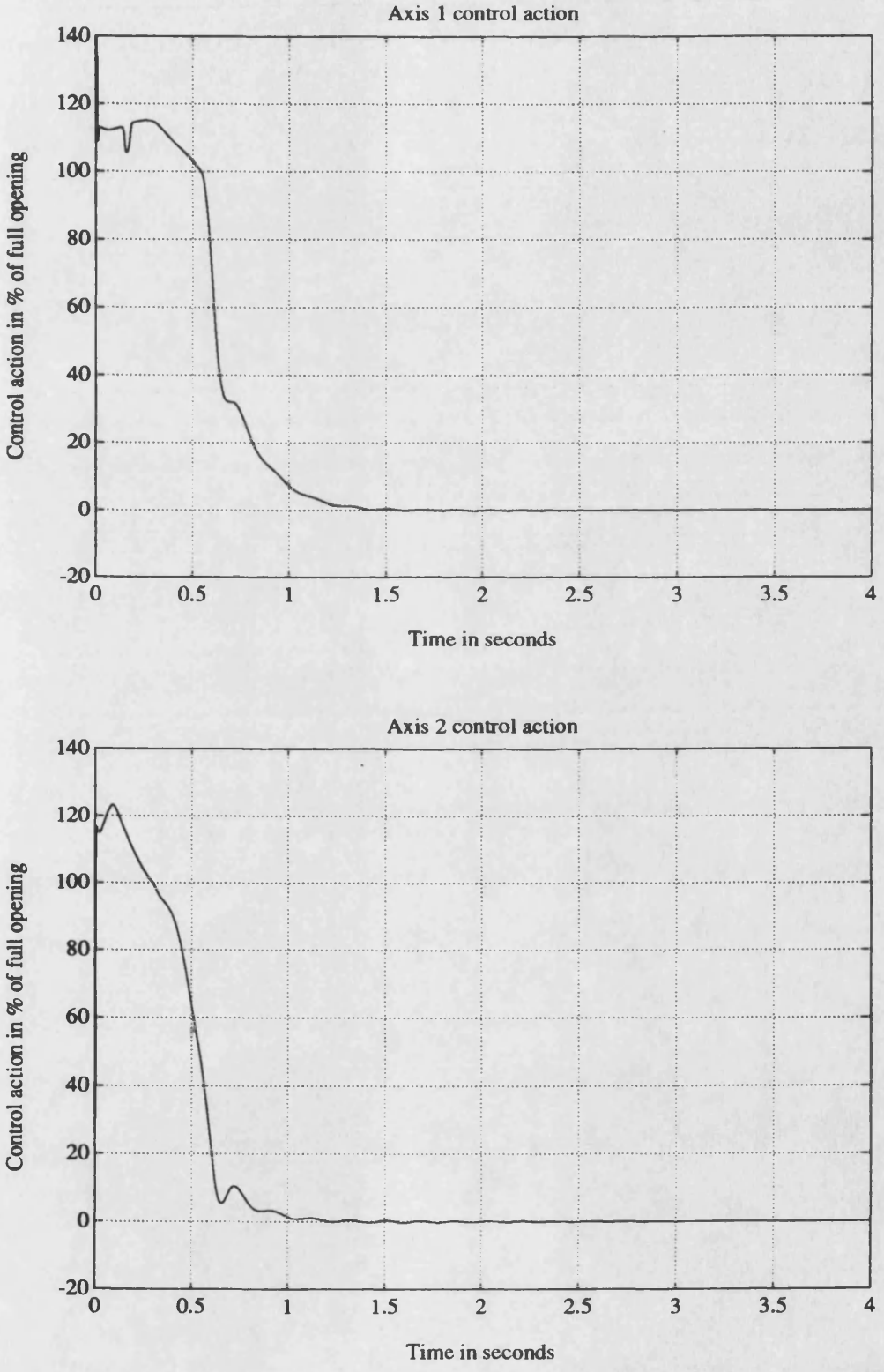


Figure 5.26 - Axes 1 and 2 control actions, u_p (large step, 120Kg payload)

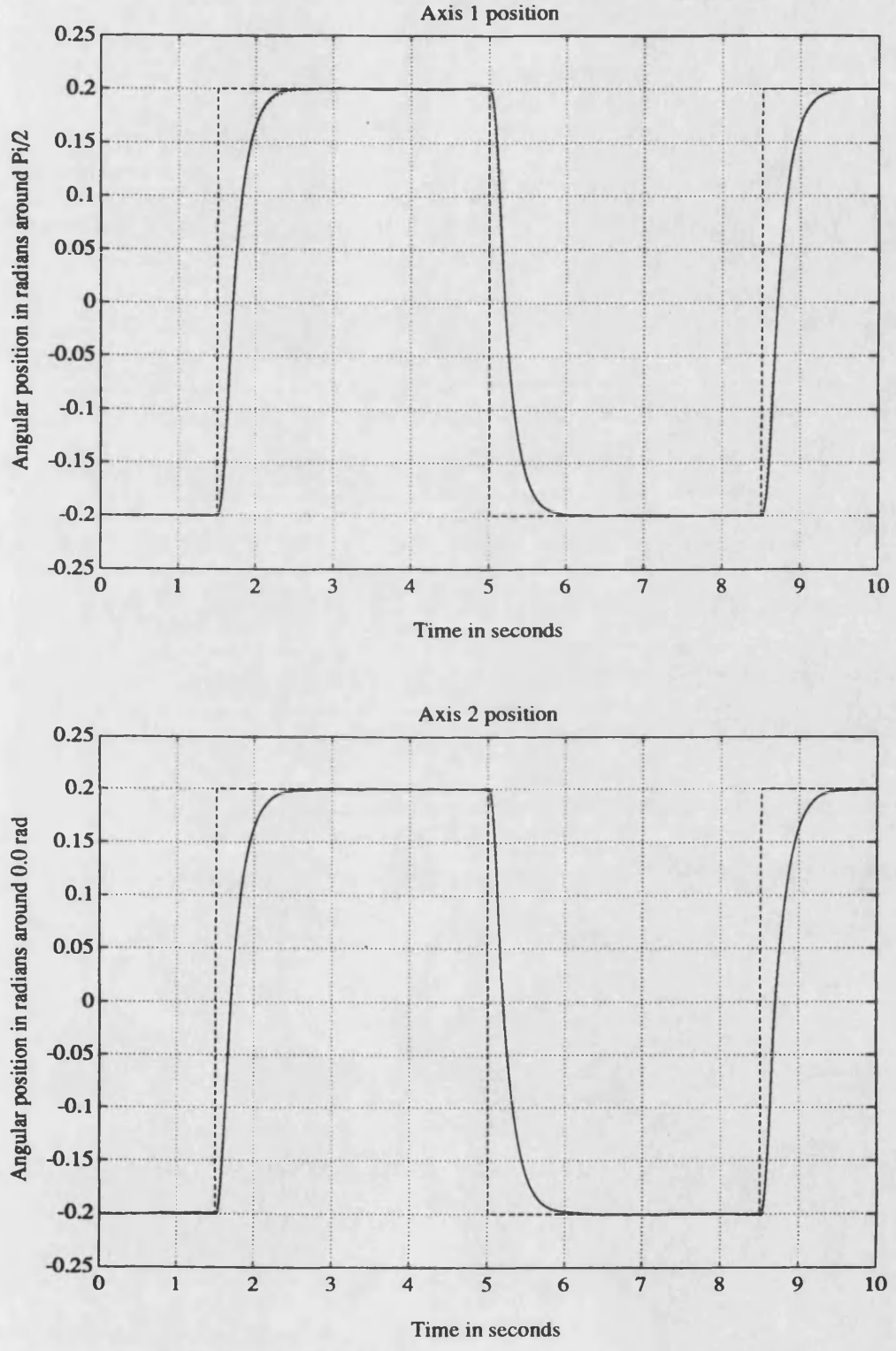


Figure 5.27 - Axes 1 and 2 position results (sq. wave, 60Kg, feedback noise)
---- reference ···· model —— axis

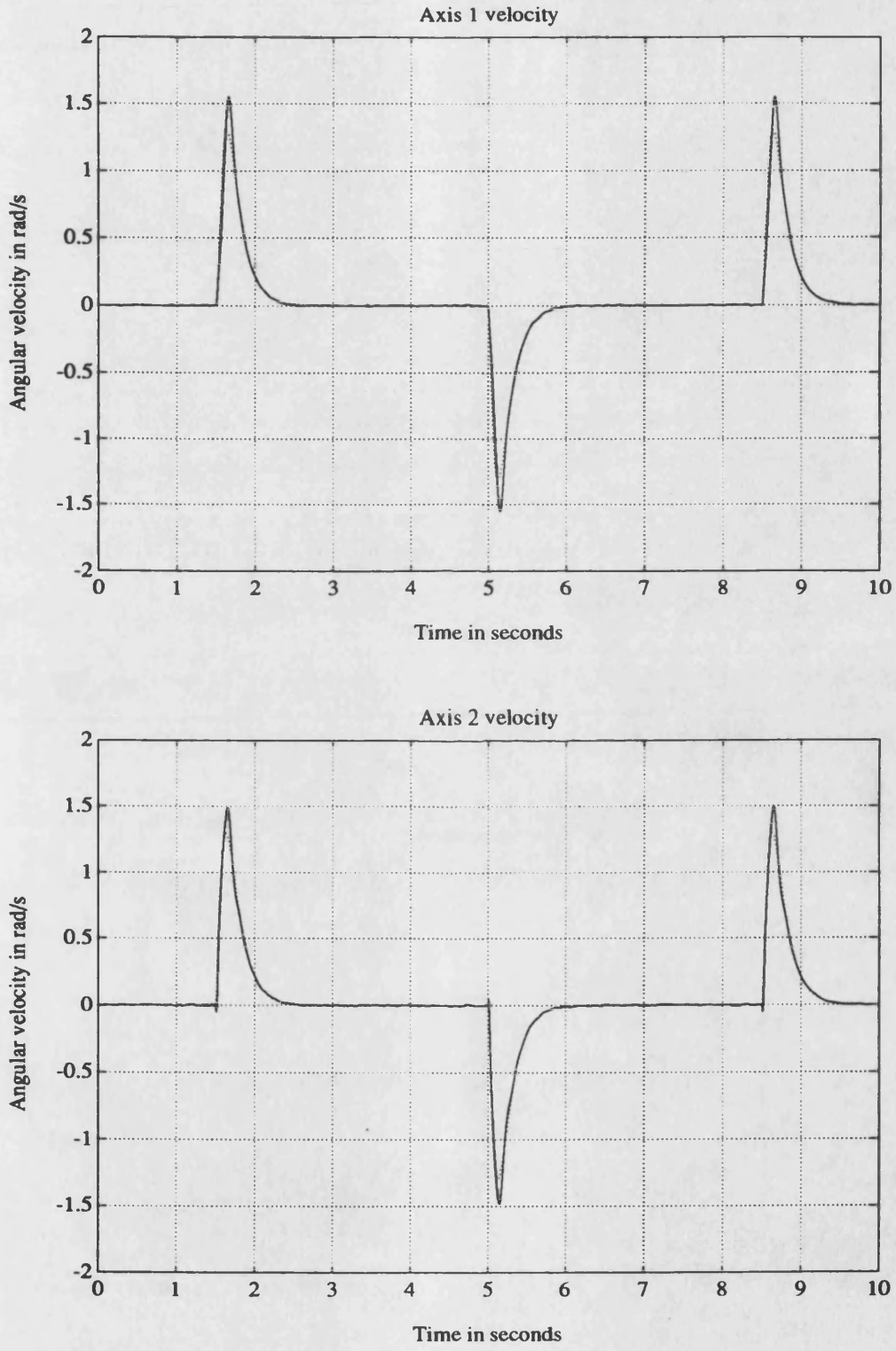
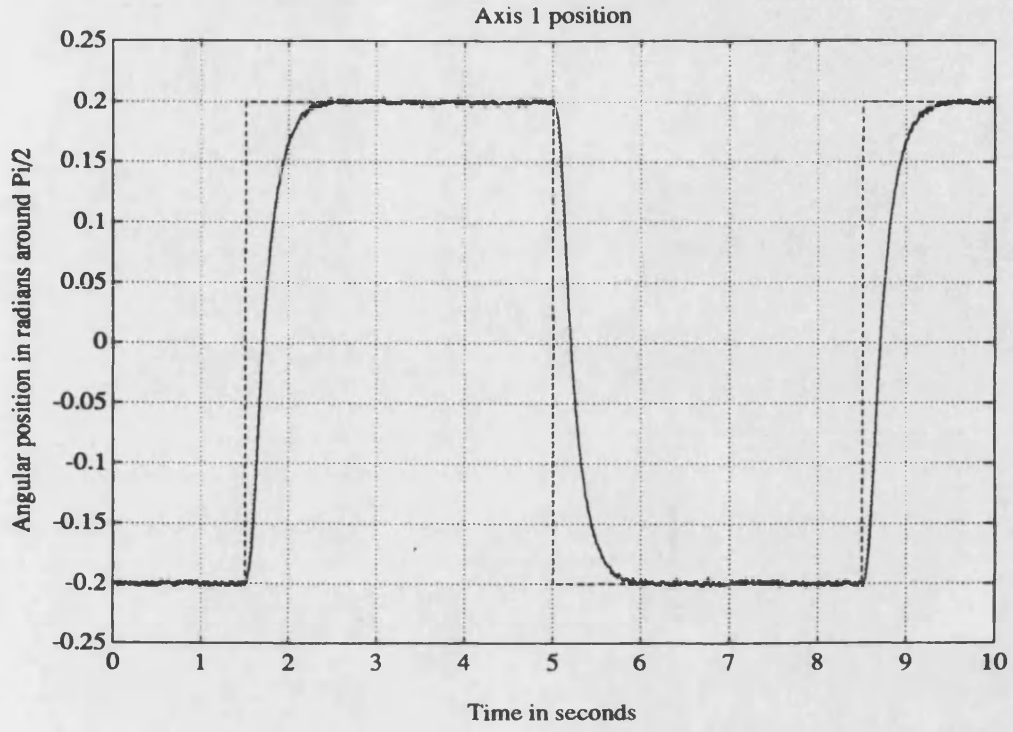


Figure 5.28 - Axes 1 and 2 velocity results (sq. wave, 60Kg, feedback noise)
···· model —— axis



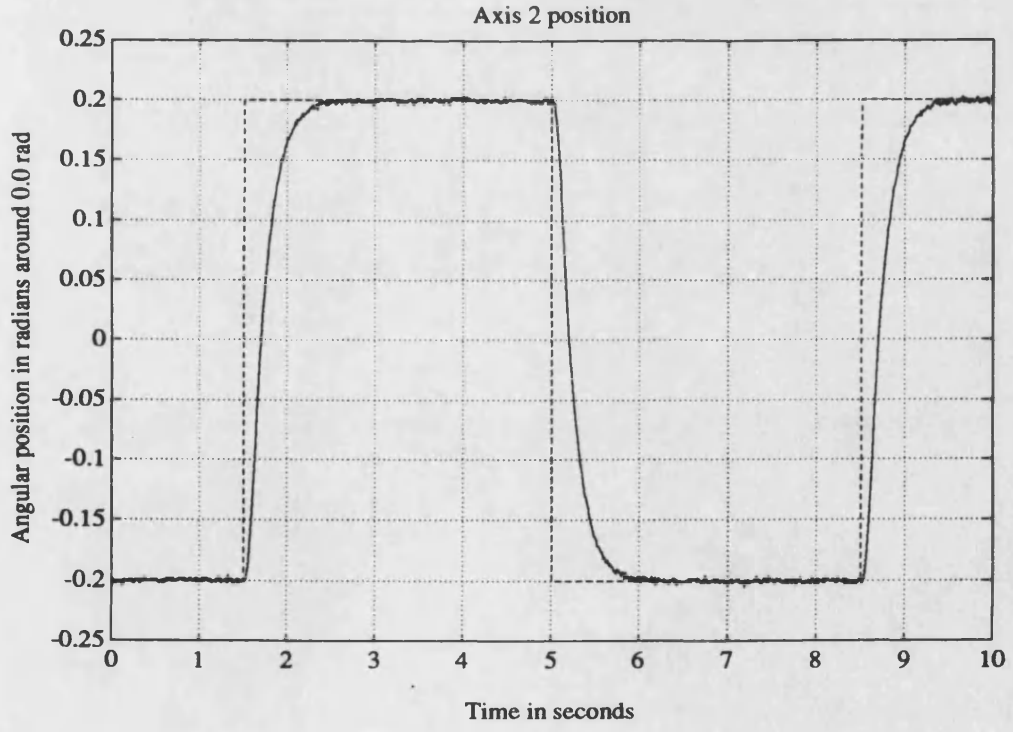


Figure 5.29 - Axes 1 and 2 position feedback signals (sq. wave, 60Kg, feedback noise)
---- reference ···· model —— axis feedback

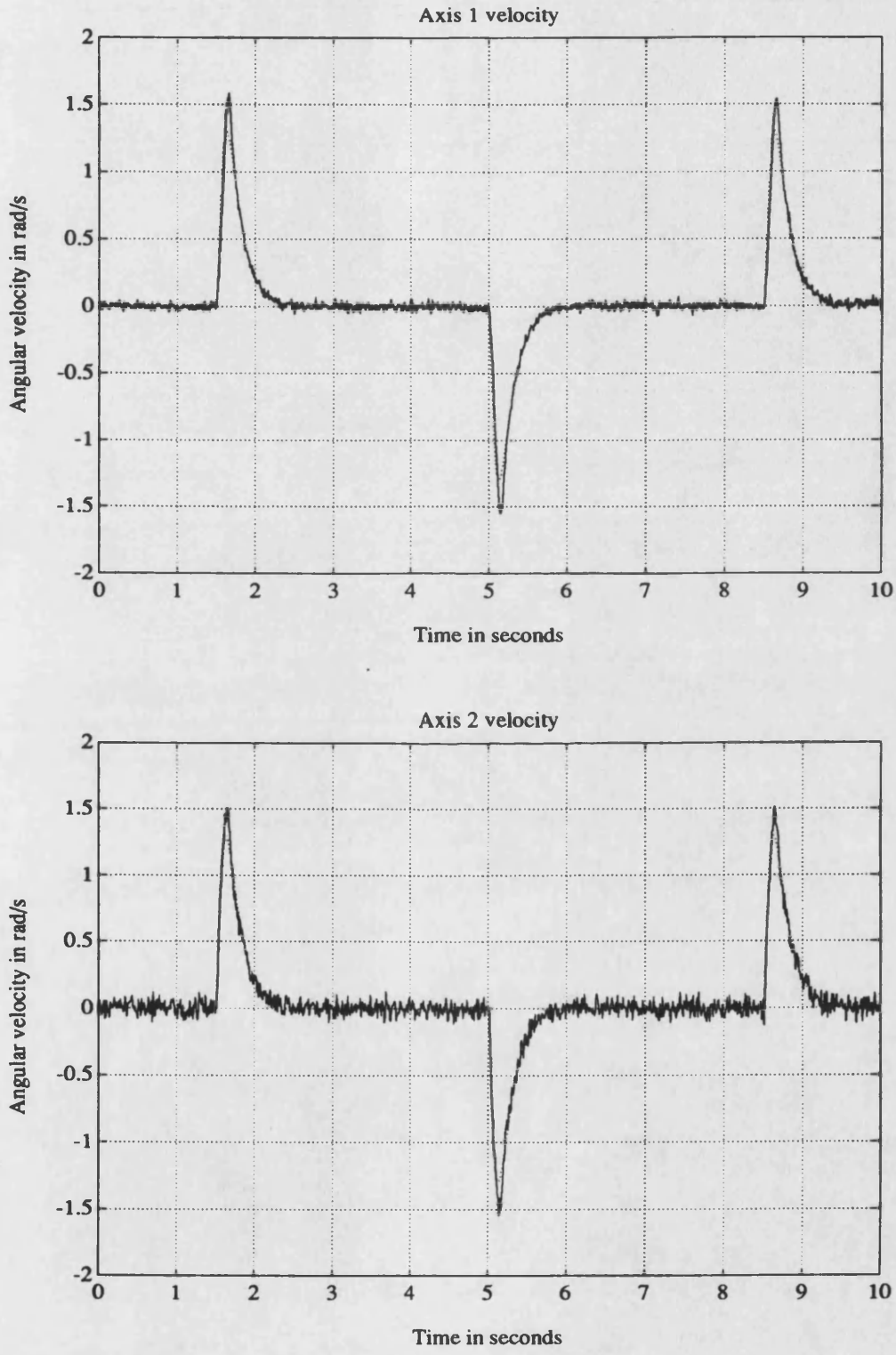


Figure 5.30 - Axes 1 and 2 velocity feedback signals (sq. wave, 60Kg, feedback noise)
···· model —— axis feedback

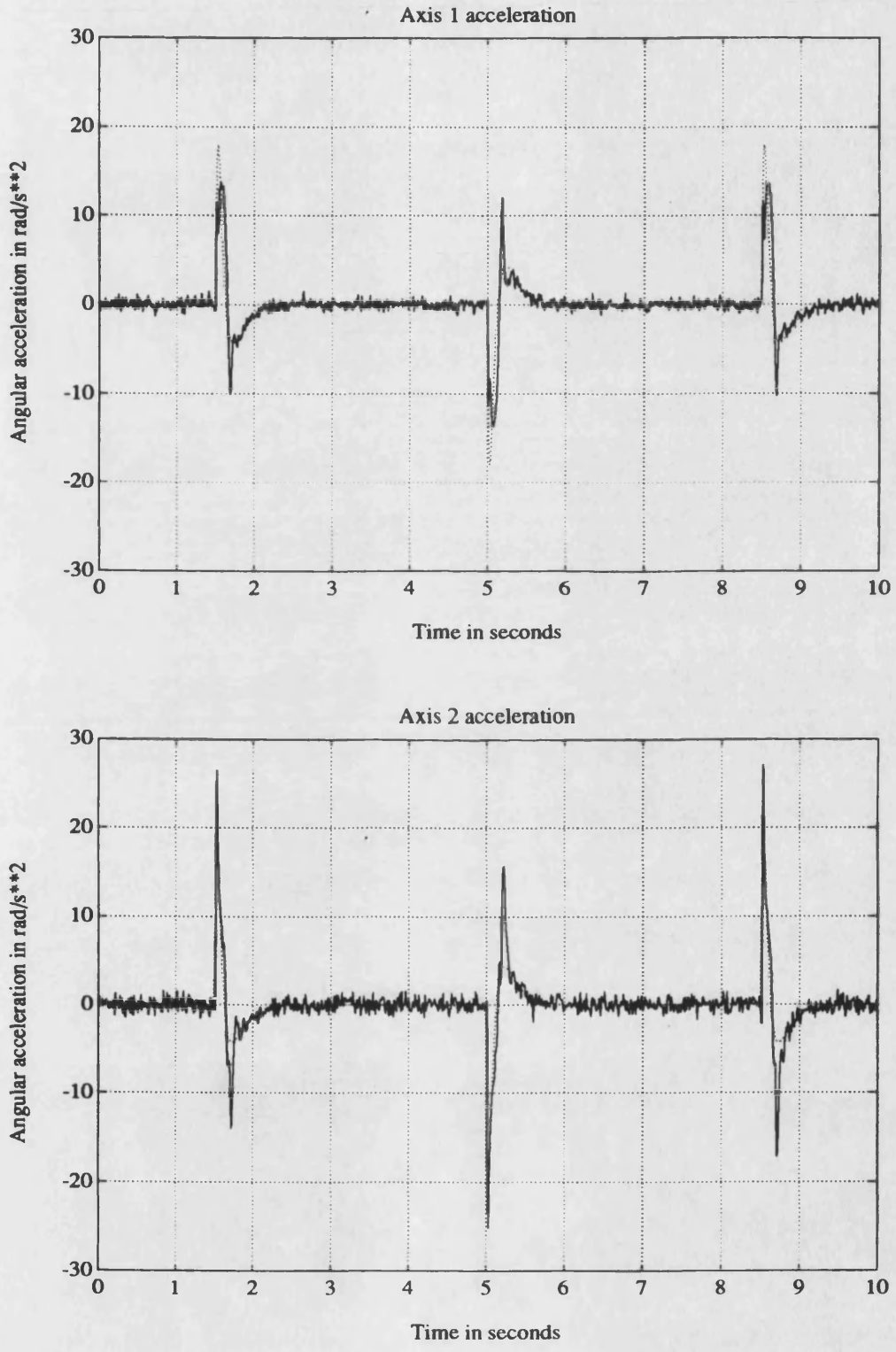


Figure 5.31 - Axes 1 and 2 accel. feedback signals (sq. wave, 60Kg, feedback noise)
···· model —— axis feedback

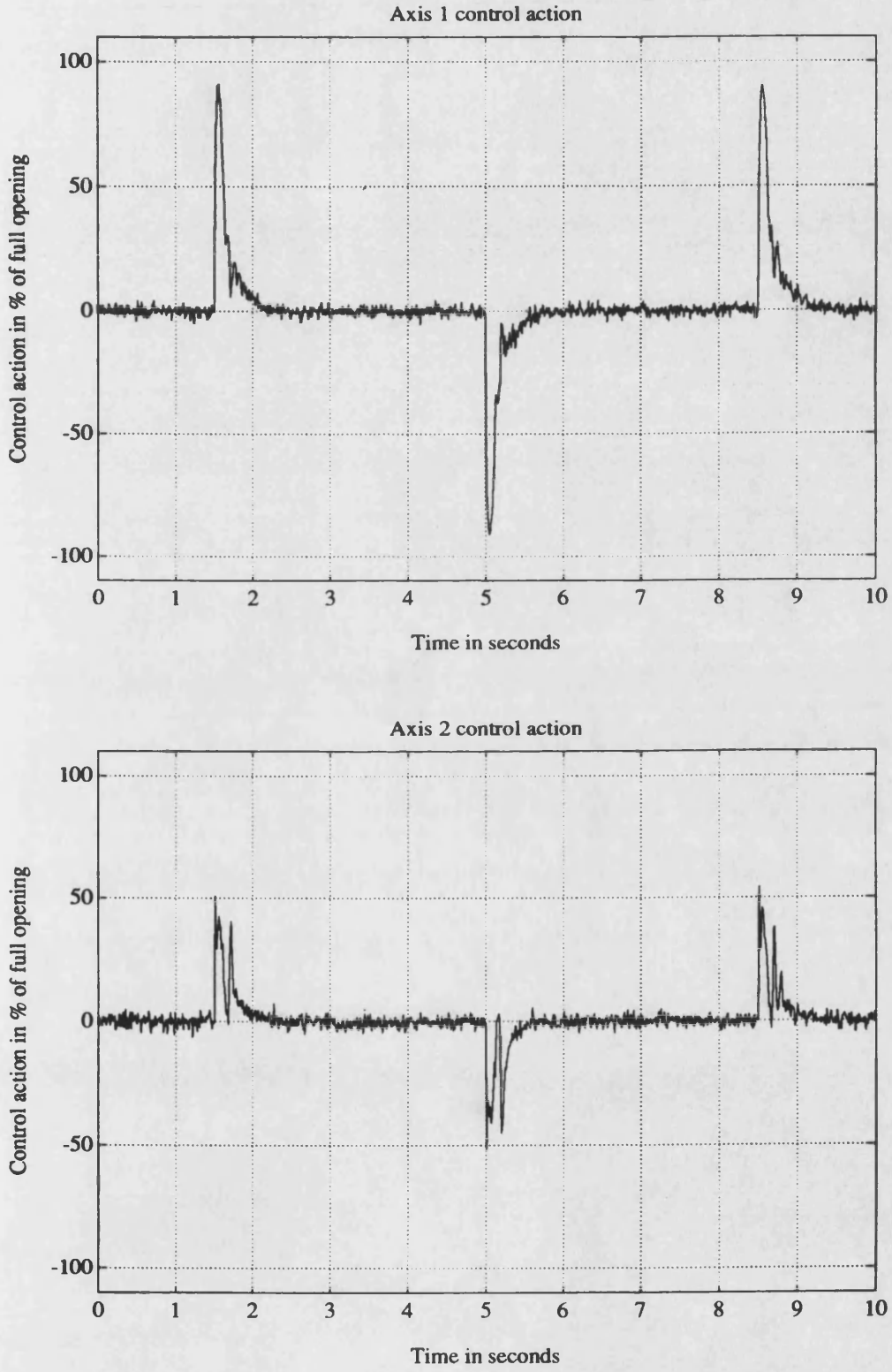


Figure 5.32 - Axes 1 and 2 control actions, u_p (sq. wave, 60Kg, feedback noise)

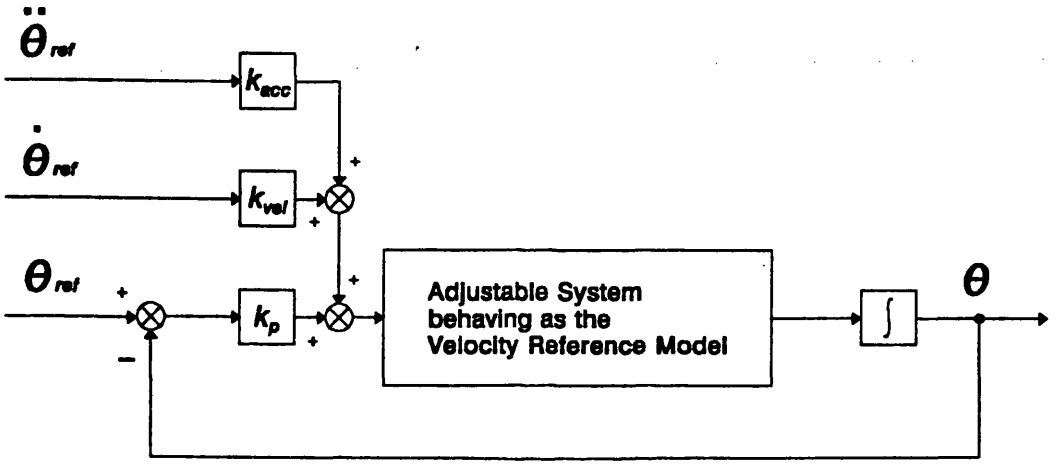


Figure 5.33 - Block diagram of the position control loop, showing the use of velocity and acceleration feedforward

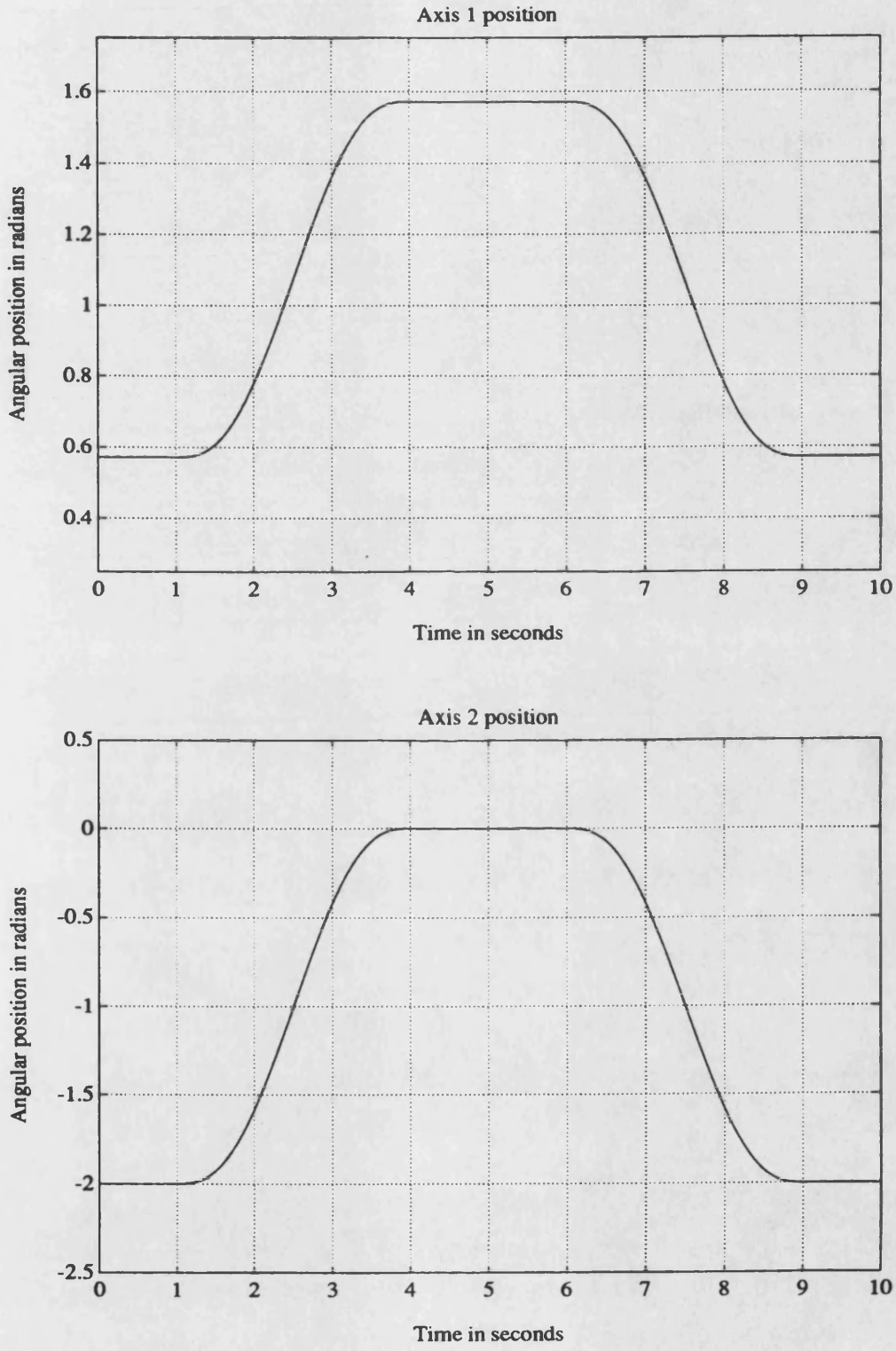


Figure 5.34 - Axes 1 and 2 position results (5th order polynomial reference, 60Kg)
---- reference ···· model —— axis

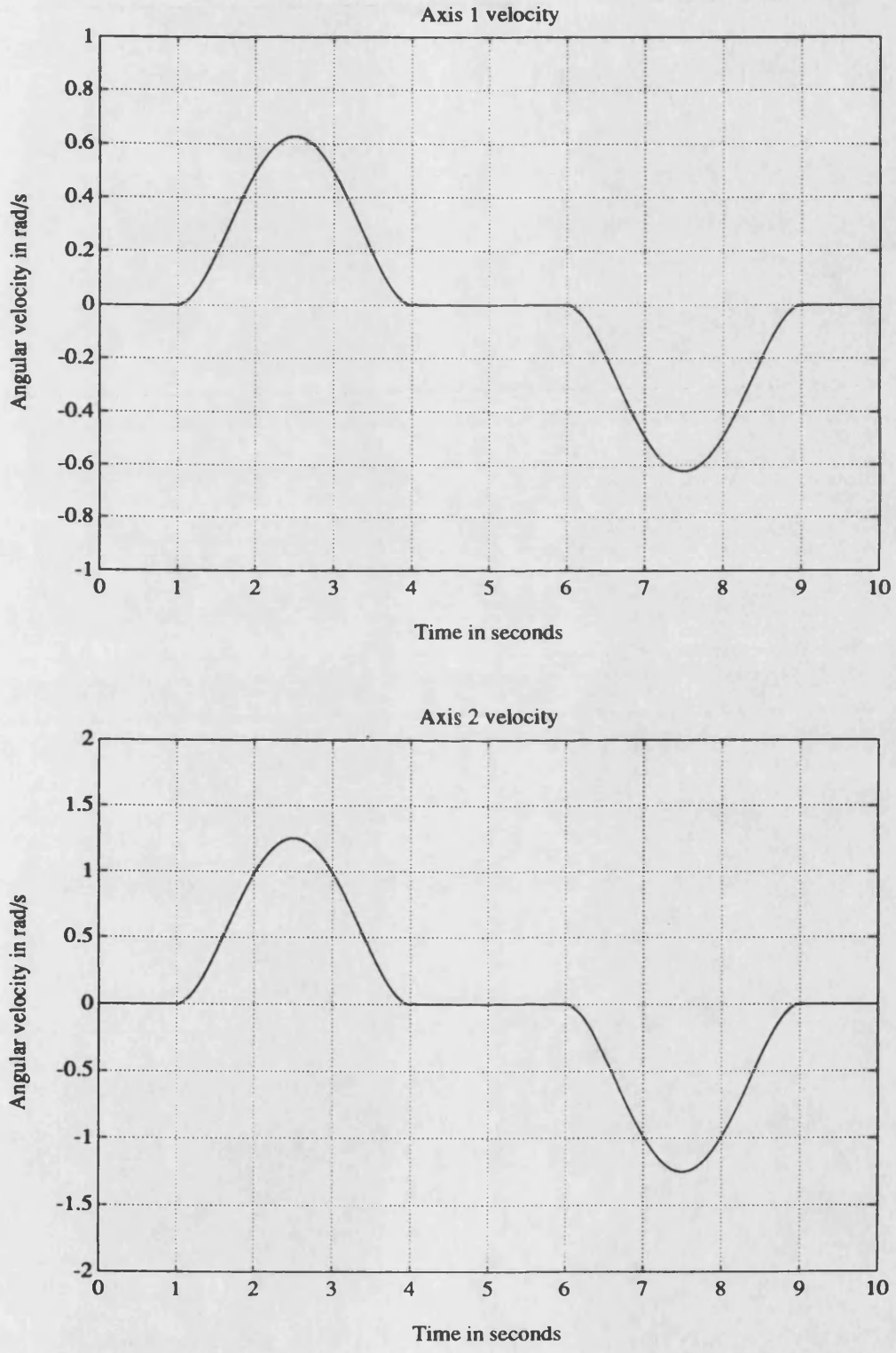


Figure 5.35 - Axes 1 and 2 velocity results (5th order polynomial reference, 60Kg)
···· model —— axis feedback

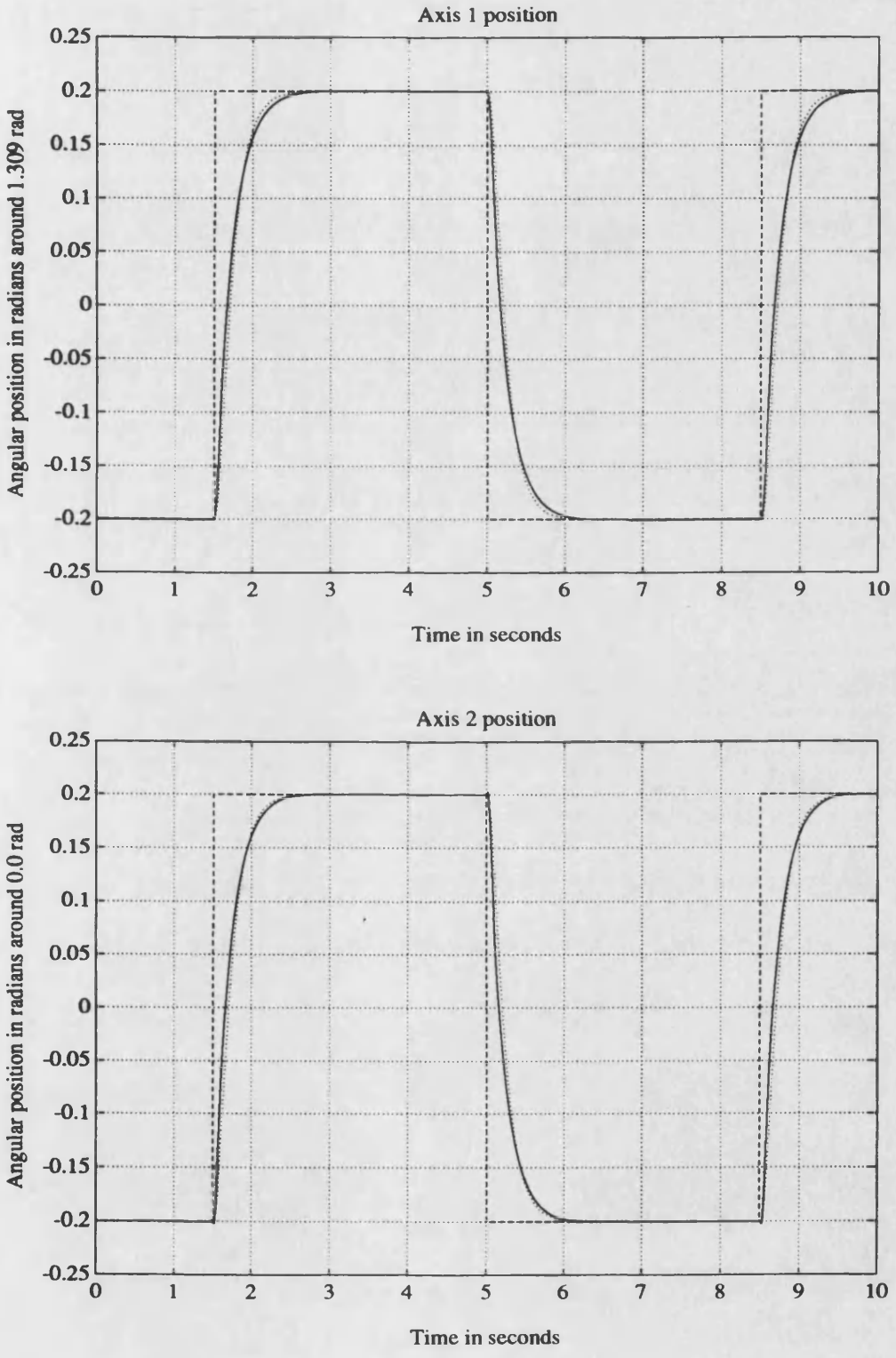


Figure 5.36 - Axes 1 and 2 position results using a state feedback cont. (sq. wave, 0Kg)
---- reference ···· model —— axis

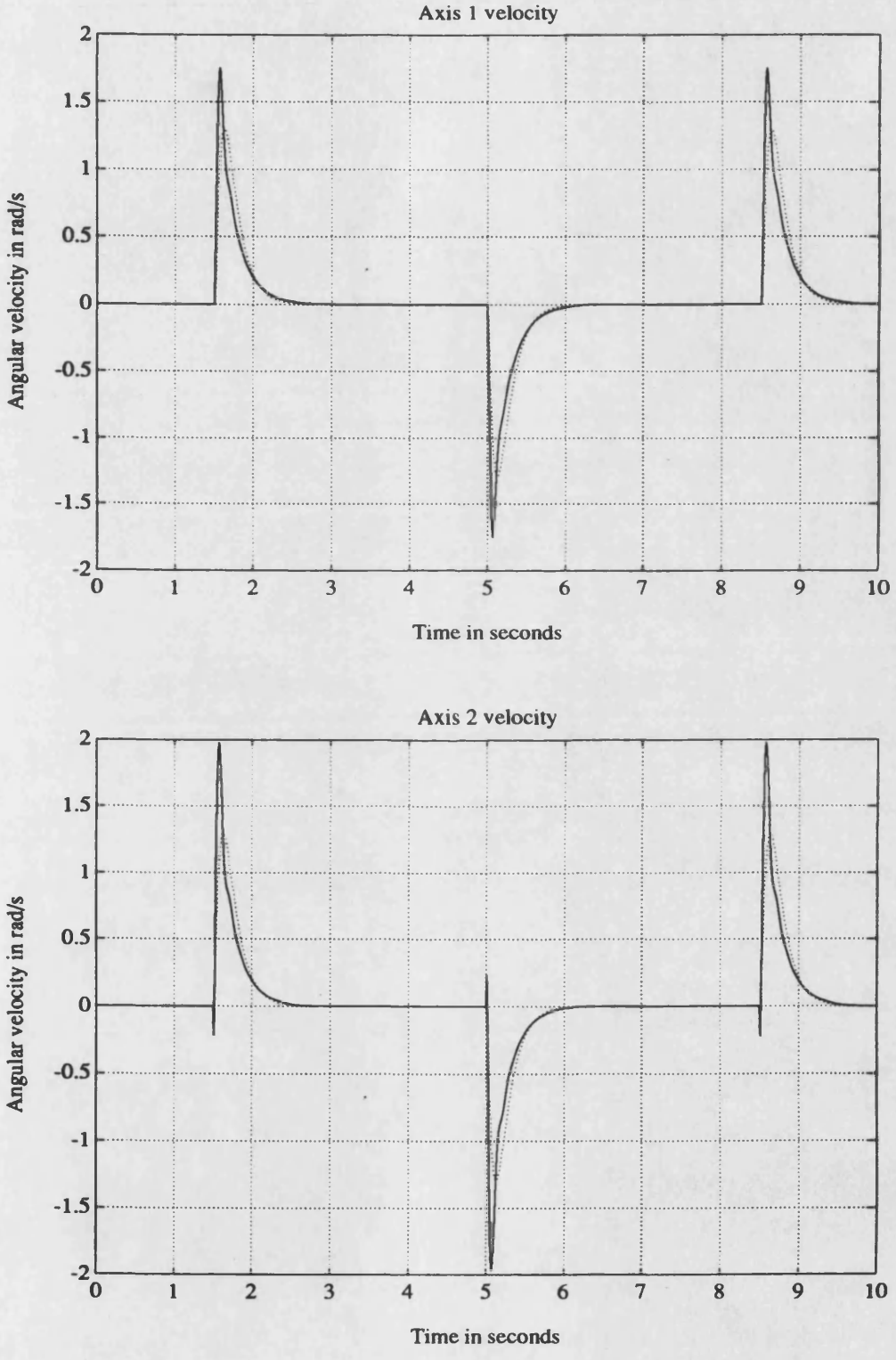


Figure 5.37 - Axes 1 and 2 velocity results using a state feedback cont. (sq. wave, 0Kg)
···· model —— axis

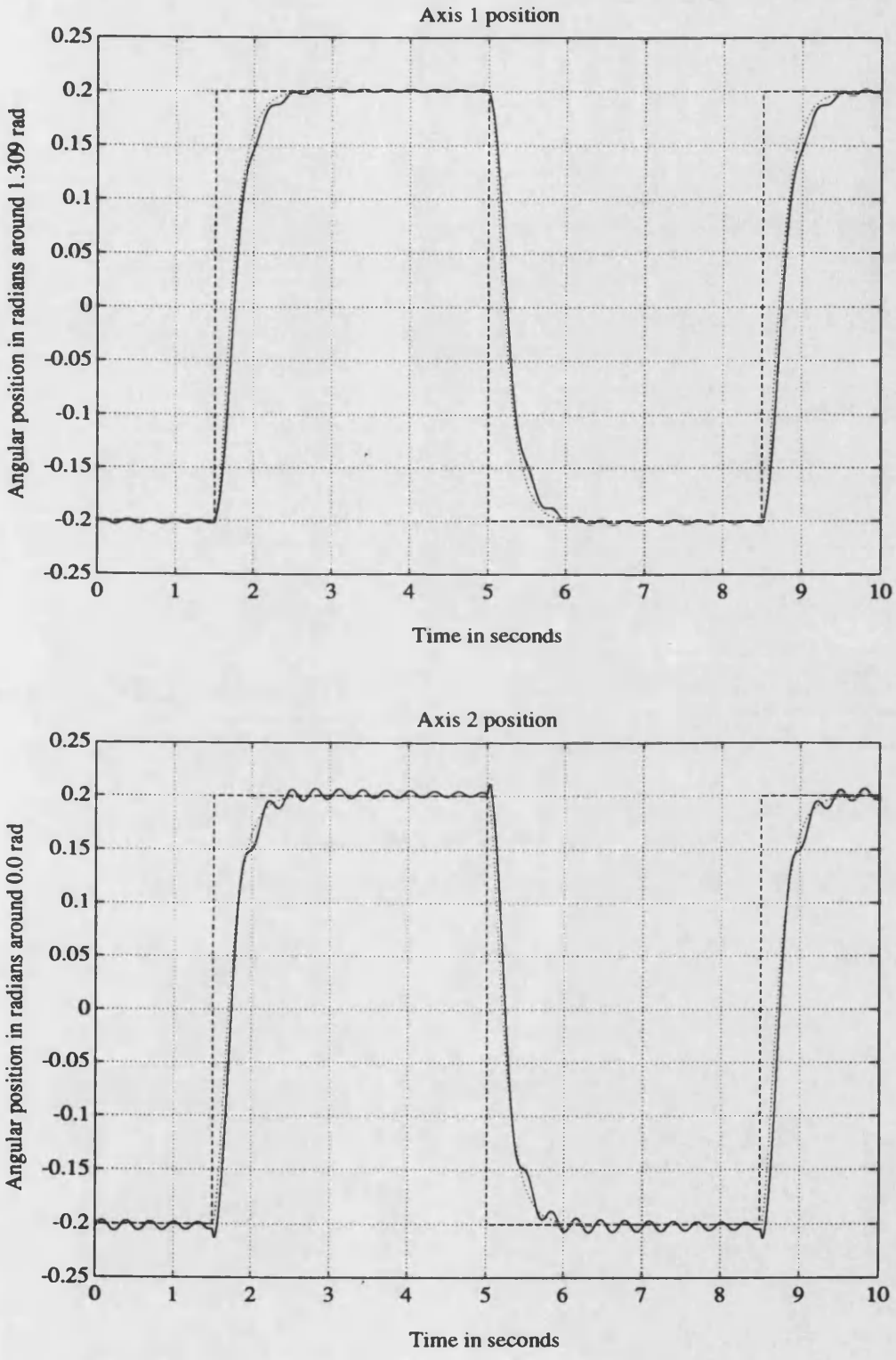


Figure 5.38 - Axes 1 and 2 position results using a state feedback cont. (sq. wave, 90Kg)
---- reference ···· model —— axis

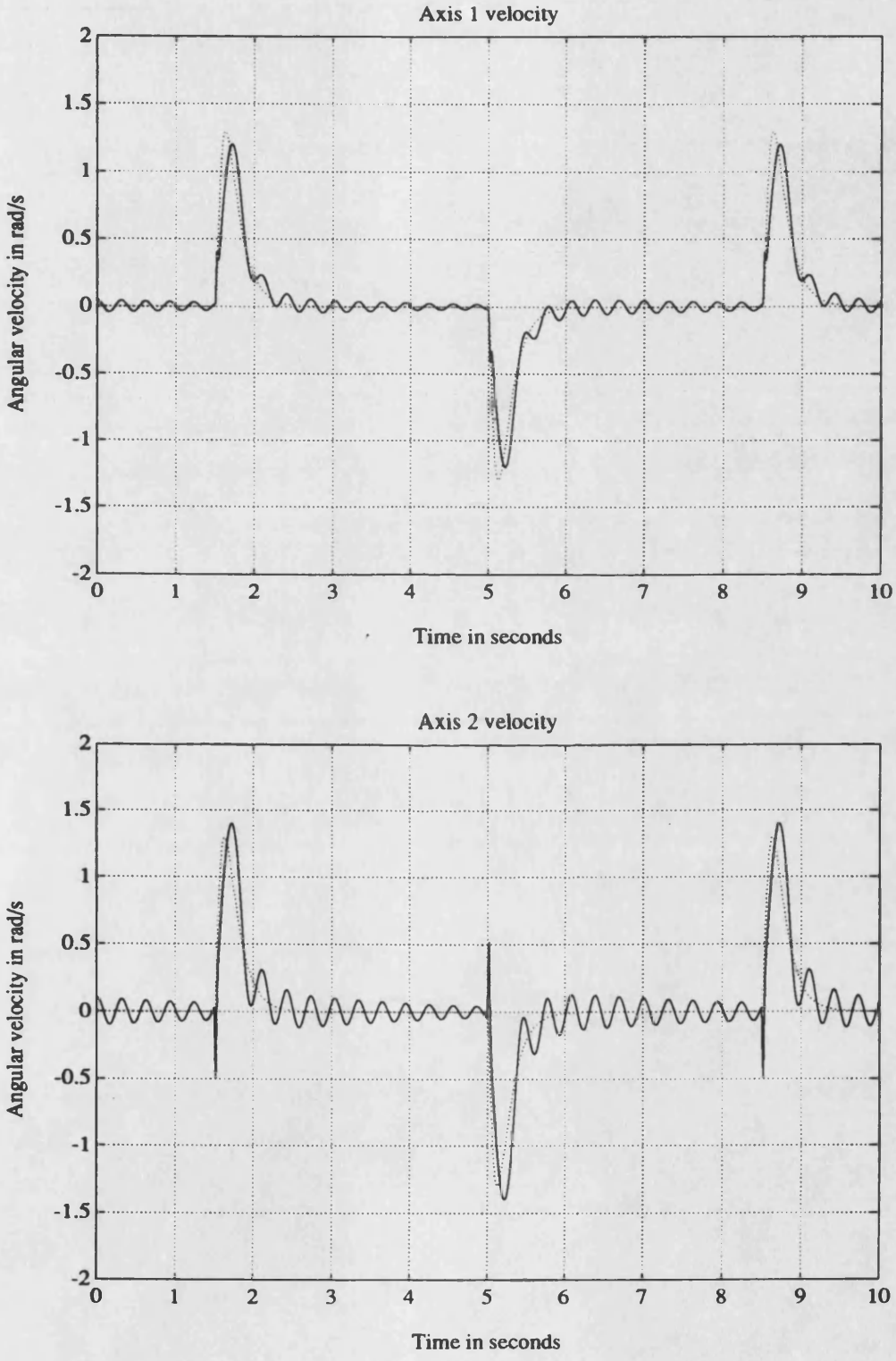


Figure 5.39 - Axes 1 and 2 velocity results using a state feedback cont. (sq. wave, 90Kg)
···· model —— axis

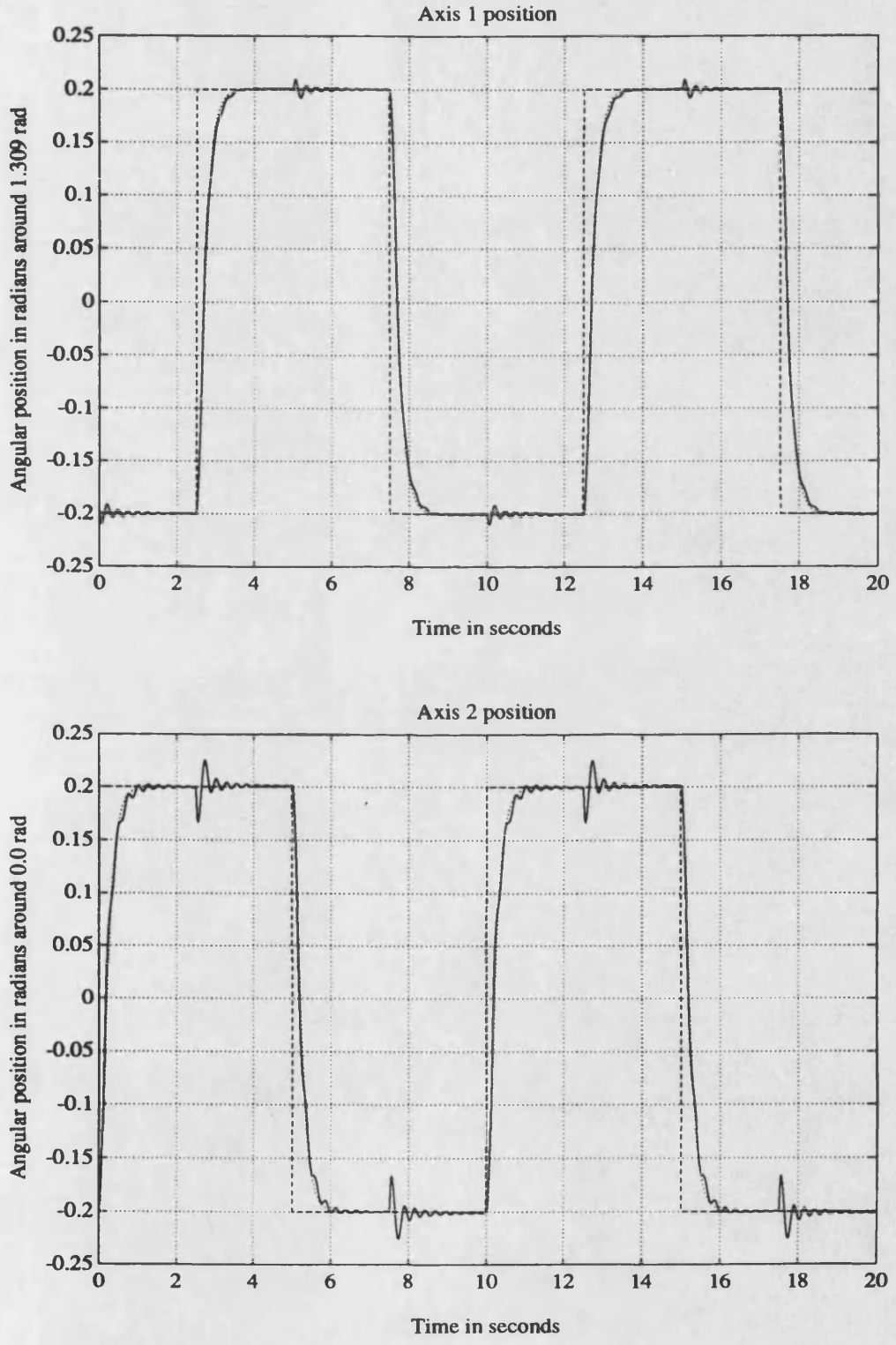


Figure 5.40 - Axes 1 and 2 pos. res. using a state feedb. cont. (sq. waves in quad., 60Kg)
---- reference ···· model —— axis

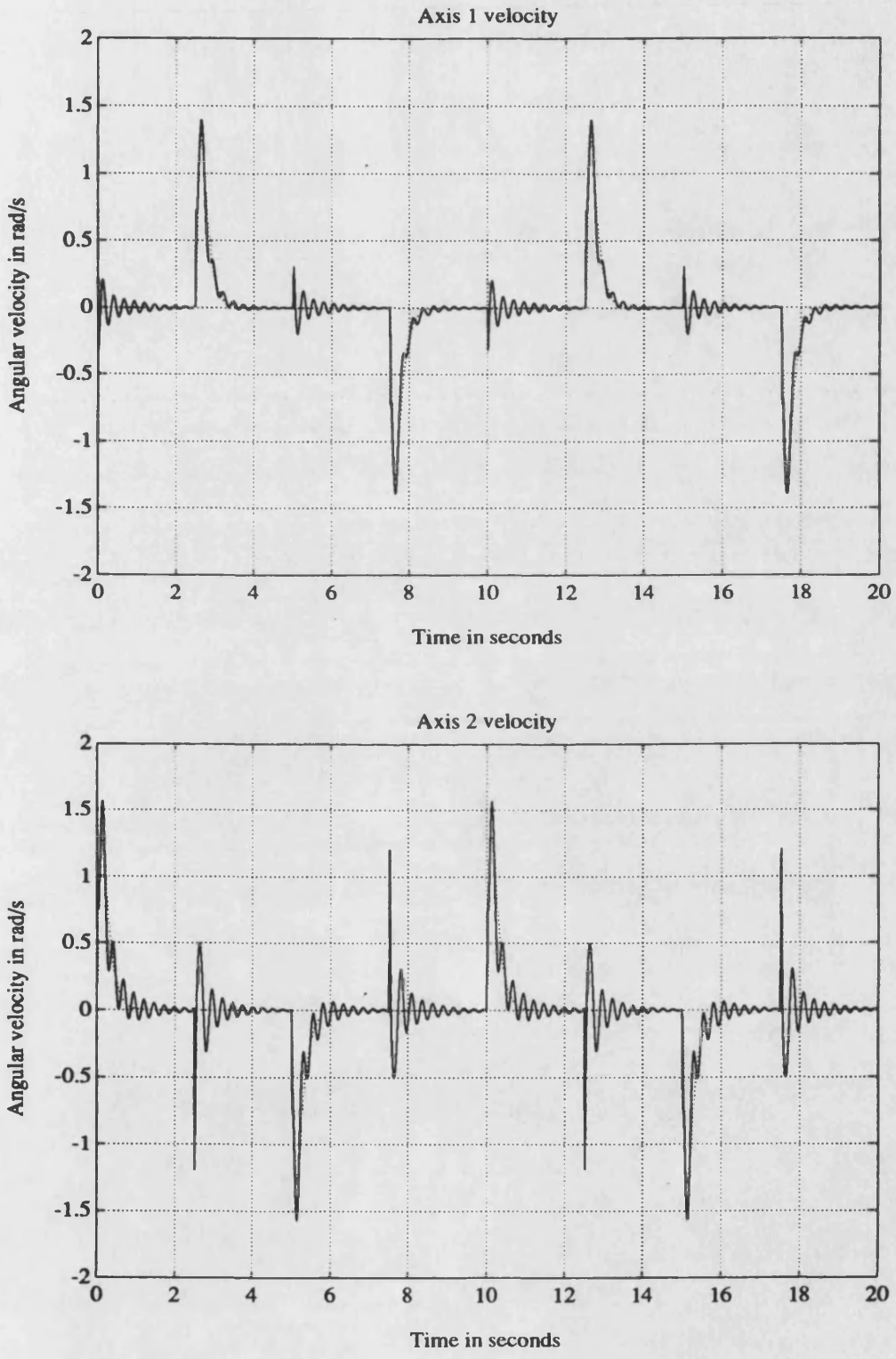


Figure 5.41 - Axes 1 and 2 vel. res. using a state feedb. cont. (sq. waves in quad., 60Kg)
···· model —— axis

Chapter 6 | Adaptive Controller Implementation and Experimental Results

The new adaptive control algorithm was implemented as a discrete time digital controller of a two axes hydraulic manipulator. This purpose built manipulator is intended to be used as a test apparatus for robot control algorithms. A photograph of the manipulator is presented in fig.6.1. The mechanical properties of this manipulator, as well as the hydraulic actuators used, are equal to the ones presented in the previous chapter. The servo-valves, however, are different. Due to ease of availability it was decided to use servo-valves that already existed in the Fluid Power Laboratory at Bath. These valves are industrial grade Dowty servo-valves model 4683 with a rated flow of 38 l/min at 70 bar. In these servo-valves bandwidth is somewhat traded off by contamination tolerance, a very important characteristic in the industrial environments hydraulic robots are typically used. A second order plus delay transfer function was fitted to the servo-valve frequency response data made available by the manufacturer with the following results: natural frequency, 50Hz; damping ratio, 2; delay, 0.2ms. As in the previous chapter, payload was allowed to vary between 0 and 90Kg. With these servo-valves, as they have a higher rated flow, the velocity loop input gain b_p is, in both axes, greater than presented in the last chapter. The actual values are: between 117.5×10^6 and 3.2×10^6 rad^3/ms^3 in axis 1 and between 116.9×10^6 and 5.9×10^6 rad^3/ms^3 in axis 2.

6.1 Controller Design

All the steps and considerations presented in Chapter 5 were followed in the same manner. The differences between the controller used in the last chapter simulation studies and the one used in the experimental manipulator relate to the differences between servo-valves and the existence of feedback transducer dynamics. These differences lead to the use of other reference models, k_h gains and v_e compensators. A simulation program of the actual manipulator was used as an assisting tool in this controller design stage.

The use of slower servo-valves led to the use of velocity reference models with a smaller natural frequency. As the slowest servo-valve pole is located at 13Hz the reference models of the velocity loop adaptive controllers cannot be expected to have natural frequencies much higher than 1.3Hz, in order to ensure that the parasitic dynamics induced by the servo-valves remains small. As the axes present higher hydraulic natural frequencies, the dynamic properties of the manipulator will be restricted by servo-valves bandwidth. After some preliminary simulation tests a second order model with a natural frequency of 2Hz and a damping ratio of 1 was chosen for both axes. A unitary DC gain was used so that the output of the position outer loop controller is a reference of desired axis velocity. The continuous-time state space representation of the reference model is represented by the following equation:

$$\begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -157.9 & -25.13 \end{bmatrix} \cdot \begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix} + \begin{bmatrix} 0 \\ 157.9 \end{bmatrix} \cdot u_m \qquad (6.1)$$

The value of k_h as well as the compensators acting on v_e must now be chosen. The simulated case presented in the last chapter only had servo-valve-induced unmodelled dynamics. In the present case unmodelled dynamics are also introduced by the feedback transducers used. Ideally, measurements

of position, velocity and acceleration should be available for control purposes. However, as relative axis acceleration is a difficult quantity to measure, the feedback acceleration signal was obtained by numerical differentiation of the velocity signal using backwards differencing.

Position and velocity signals were obtained using resolvers as feedback transducers. The signal conditioning of the resolvers was made using resolver-to-digital converters from Analog Devices (AD2S80A). These converters have two outputs: a digital position output and an analog velocity output as shown in the functional block diagram of fig.6.2. This diagram shows how the position and velocity signals are obtained: the converter uses a type-two tracking loop that enables zero position output error with constant acceleration inputs. With general inputs, the converter position output is a filtered version of its input. The velocity signal is obtained from the input of the VCO (Voltage Controlled Oscillator), that is, the input of the second integrator in the tracking loop (the up/down counter). This signal is a filtered version of the resolver velocity and, being obtained by what is essentially an analog differentiation of the position input, it is prone to have a high noise content. A block diagram of the tracking loop is presented in fig.6.3. The noise is mainly due to the 10kHz carrier frequency used in the resolvers and several other minor sources of input signal noise. The tracking loop transfer function may be adjusted by the choice of some external passive components that determine the k_a gain (converter acceleration constant) and the time-constants τ_z and τ_p of the lead compensator. With this freedom, the designer has the opportunity to chose a compromise between bandwidth and signal-to-noise ratio.

In order to achieve a high adaptation bandwidth, the phase lag introduced by the resolver-to-digital converter must be kept small at frequencies bellow 100Hz. This constraint lead to the choice of a 400Hz converter bandwidth. With the help of an Analog Devices software tool, the passive converter components were chosen in a way to achieve that bandwidth. The following values of k_a, τ_z and τ_p resulted: $k_a=1.329\times10^6 s^2$, $\tau_z=1.50\times10^{-3}s$, and $\tau_p=270.5\times10^{-6}s$. A Bode plot of the obtained closed loop transfer function of the resolver-to-digital converter is presented in fig.6.4.

During the calibration tests of the resolvers and resolver-to-digital converters, it was found that the velocity signals had a significant content of 10kHz carrier noise. In order to decrease the magnitude of this noise signal, a low-pass third order Butterworth filter, with a break frequency of 1kHz, was used.

After some preliminary simulations it was decided to use two compensators: the first one included only in the v_e loop, similar to the compensator derived in the last chapter, and a second one, a servo-valve pre-compensator, acting on the servo-valve input signal. A block diagram of the adaptive velocity controller including these two compensators is presented in fig.6.5. The servo-valve pre-compensator was used as a way to virtually increase its bandwidth. The transfer function of this pre-compensator was chosen as:

$$G_{svc}(s) = \frac{(5.0\times10^{-3}s + 1)}{(0.8\times10^{-3}s + 1)} \tag{6.2}$$

As axis 1 and axis 2 have very similar maximum velocity loop input gains (b_p), the compensators used on the v_e signals as well as the k_h gains were chosen equal in both axes. Using classical control techniques, with the help of Matlab, the value of k_h and a lag-lead compensator were chosen as:

$$k_h \frac{(\tau_{z1}s + 1)(\tau_{z2}s + 1)}{(\tau_{p1}s + 1)(\tau_{p2}s + 1)} = 7.0\times10^{-6} \frac{(50.0\times10^{-3}s + 1)(2.0\times10^{-3}s + 1)}{(200.0\times10^{-3}s + 1)(0.4\times10^{-3}s + 1)} \tag{6.3}$$

At the same time, and because the controller was to be implemented digitally, the value of the sampling time period had to be chosen. The choice of the k_h gains is related to the value of the sampling period, because the discretization of the control algorithm introduces some time delays, as computing delays and D/A zero-order-hold delays. These delays have the effect of decreasing gain margin and phase margin. Taking into account the achievable v_e loop bandwidth, the frequency characteristics of the compensators and the need for a fast reaction time to external disturbances, after

some simulation studies a sampling period of 1ms was found to be adequate. This choice of sampling period is fast enough for all of the controller needs, and easily achieved with current micro-processor technology.

A Bode diagram of the open-loop frequency response of the v_e loop, supposing $b_p = 118\times10^6$ rad^3/ms^3, is presented in fig.6.6. In the phase plot of this Bode diagram several extra delays, apart from the servo-valve delay, are included. The first one, of 0.1ms, was used as an estimate of the control algorithm computing time delay (10% of a 1ms sampling period). A second one, of 0.5ms, is due to the half sample time delay introduced by the backwards difference algorithm used to compute axis acceleration. A third one is due to the half sample time delay introduced by the D/A converter zero-order-hold. From this Bode diagram it can be concluded that, with the choices made, the v_e loop has a minimum phase margin of 46°, a minimum gain margin of 4.5 and a maximum bandwidth of approximately 22Hz. This bandwidth cannot be further extended, the main reason being the servo-valve would not respond to higher frequencies.

The preliminary simulation work carried out in the controller design stage lead to the same conclusions presented in the previous chapter about the choices of k_u and k_x gains, that is:

- h is the dominant feedback signal. It was found, as in the previous chapter, that adaptation time behaviour (the time behaviour of v_e) did not change when the k_x gains are omitted. So, these feedback gains were not implemented, in order to decrease the controller computing time.

- The k_u gain is fundamental for a good model following performance. Model following performance becomes very sluggish if this gain is omitted. As in the previous chapter simulations, making k_u^+ smaller than its maximum value ($\min(b_m / b_p)$) or k_u^- greater than its minimum value ($\max(b_m / b_p)$) does not improve model following performance.

- The simulations revealed a value of μ equal to 30 rad/s^2 to be a good compromise between fast adaptation with small model following errors and smooth control signals.

- The antiwindup loop proportional gain k_{sat} was found by trial and error (its value must be of the same order of b_p). A value of 70×10^6 rad/m.s^3 was chosen for both axes controllers.

At this point the inner loop adaptive velocity controller is completely defined. The next design step is the choice of a outer loop position controller. A proportional controller was selected for this purpose. As inner loop velocity models are equal in the two manipulator axes, the position controllers proportional gains were also chosen equal in order to obtain the same transfer function for both axes dynamic behaviour.

Following the same design criteria used in the previous chapter, it was decided to use a proportional gain which is as high as possible, subject to the restriction of maintaining all closed loop poles of the axes transfer functions real. Using this choice undesirable overshoots of the end effector position trajectory are avoided. The value of this proportional gain, k_p, was chosen as 1.5 rad.s^{-1}/rad. The obtained closed loop axis transfer function is given by:

$$\frac{\mathcal{L}(\theta)}{\mathcal{L}(\theta_r)} = \frac{237}{(s + 16.4)(s^2 + 8.8s + 14.5)} \qquad (6.4)$$

where the second order term has a natural frequency of 0.6Hz and a damping ratio of 1.2.

6.2 Controller Implementation

The controller design proceeded entirely in the continuous time domain. Nevertheless, to implement the proposed controller using a digital processor, this one must be discretized. The discretization process involves two different but closely related problems: time discretization, that is, to find discrete time controller functions that are functionally equivalent to the continuous time ones; and the discretization of the controller coefficients and variables, that is, the problems associated with the use of finite (limited) precision arithmetic and input/output in the digital computer.

Two methods were used for controller time discretization:

- The state space velocity reference model, as well as the series-parallel model of the state error behaviour, were discretized using the zero-order-hold equivalence method. As the plant is driven by the outputs of D/A converters with a zero-order-hold, this discretization method enables the discrete models to provide state and state error reference values that have the same time behaviour (at the sampling instants) as the sampled continuous time models.

- The compensators were discretized using the Tustin approximation rule. With this method the frequency response of the discretized filters is preserved, especially the phase characteristics. Frequency prewarping was not used because as the crossover frequency is more than one decade below Nyquist frequency any benefits would be marginal.

In order to avoid the problems associated with the use of finite precision controller coefficients and arithmetic, the discrete time controllers were implemented using the delta operator (δ) instead of the more usual forward shift operator (q). The relationship between δ and q is a simple linear function,

$$\delta \triangleq \frac{q - 1}{T} \tag{6.5}$$

and thus δ offers the same flexibility in the representation of discrete time systems as does q [52]. The use of q generally leads to simpler expressions that highlight the sequential nature of the sampled signals. On the other hand, δ can be seen as an approximation to d/dt:

$$\delta x_k = \frac{x_{k+1} - x_k}{T} \tag{6.6}$$

Thus it leads to models that are more like continuous time models, enabling continuous time insights to be used in discrete time design. Delta transforms (the transform associated with the delta operator) have an interesting property that results from this approximation to a time derivative: the Laplace transform can be obtained from the delta transform as the limit when $T \rightarrow 0$. This enables the use of a unified transform theory, covering both discrete and continuous cases at the same time [52].

Nevertheless the main reason that lead to the use of the δ operator in this work was its superior numerical properties. Usually the sampling frequency used in the implementation of digital controllers is significantly in excess of the dominant frequencies of its filters. Other considerations as fast disturbance rejection and the influence of sampling period on phase margin frequently dictate the use of fast sampling frequencies. In this situation the use of the z transform results in the controller poles being clustered around the $z = 1$ point in the z-plane. This leads to a high sensitivity to coefficient quantization and roundoff noise. Agarwal and Burrus [53] and Orlandi and Martinelli [54] proposed filter structures using a transform of the origin of the z-plane to the point $z = 1$ through a linear change of coordinates that are less sensitive to this problem. A more formal approach is the use of the delta transform as proposed by Middleton and Goodwin [55]. These authors compare the control performance

obtained with the use of δ and q in the implementation of a given control algorithm. With 6 significant bits, the differences between the delta controller and one with an infinite word length are nearly unnoticeable. On the other hand, the performance of the shift controller is unacceptable. In a comparative study presented by Forsythe [56], where z direct and canonical forms are compared with delta form, the use of the delta operator emerges as the preferred filter structure. Goodall [57] presents a delta structure that enables a minimization of digital controller computation with the benefit of a simpler assessment of the size (word-length) of the filter internal variables (state variables) needed in order to ensure the proper behaviour of the filter. The same author comes to the point of questioning the appropriateness of using the delay operator for recursive digital filters [58], particulary in the implementation of digital controllers.

The digital processor interacts with the manipulator using digital inputs, analog inputs and analog outputs, in order to read the axes' positions, axes' velocities and position references, and to output the control actions. The D/A converters used to output the control action have a 12 bit resolution. This control action voltage signal is then converted into a proportional current signal in order to drive the servo-valves. The digital processor must read the positions and velocities of the axes, as well as the axes position references. The resolver-to-digital converters output the position values in digital form, with a 16 bit resolution, that is directly read by the digital processor. They also output the axes' velocities as analog voltages. These are read by the DSP using a multiplexed 12 bit A/D converter. The input axes' position references are also in analog form and are sampled using another multiplexed 12 bit A/D converter. Using this input arrangement, simultaneous sampling of each axis position, velocity and position reference is possible. Two 12 bit D/A converters were used to translate the 12 most significant bits of the resolver-to-digital converters digital position outputs into two position analog signals. These signals were used for monitoring and data acquisition purposes only.

A Digital Signal Processor (DSP) was used in the implementation of the proposed adaptive controller. This type of processor offers, at a reasonable cost, a high computing performance. This high

performance is achieved through the use of a reduced instruction set architecture that is tailored for the software implementation of digital filters. As a digital controller is mainly composed by a set of such filters, this type of digital processor is also very efficient when used to implement complex digital control algorithms, enabling the use of fast sampling frequencies. The particular DSP used was a Motorola DSP56001. This is a fixed point DSP with a 24 bit word-length. It has an Harvard memory architecture with one program and two data memory spaces. This enables the completion of one complex instruction that loads two data values, multiplies them together (generating a 48 bit result) and accumulates the result in a 56 bit accumulator in just one instruction cycle (100ns in the present case). This sequence of events is the one needed to implement a digital filter tap, and may be seen as an example of the efficiency presented by a DSP in this type of task.

The two axes' controllers were implemented in a single DSP as two completely independent assembly program routines. A DSP internal timer is used to generate an interrupt with a frequency of 2kHz, and the interrupt service routine acts as a simple scheduler that alternates between the two independent controllers. This way each controller operates with a sampling period of one millisecond and has an available time of 500 microseconds. The chosen DSP engine is implemented on a PC expansion card (Loughborough Sound Images Ltd. DSP56001 PC Processor Board) and is able to communicate with the host PC bus via a dedicated I/O port. Although this facility enables a tight coupling between the PC and the DSP (it may be used as a signal processing coprocessor), it was only used in this work for development purposes. The DSP software was developed in the host PC, using a cross-assembler, and a small monitor program is used to download the resulting object code and start it running on the DSP. The DSP card also has an independent dedicated 16 bit I/O bus (DSPLink) that was used to connect the A/D, D/A and resolver-to-digital converters. With this architecture the host PC is free to run other software and preform other tasks, possibly of other control levels, as trajectory generation, security checks and interactions with other machines. As none of these tasks were implemented in this work, the PC was idle (running the monitor program) during the adaptive controller experiments. A general diagram of the controller hardware architecture is presented in fig.6.7.

The computing time of each axis controller was found to be 30 microseconds. As a computing time of 100 microseconds was used in the v_e loop shaping, the obtained phase margin is slightly bigger than predicted. With the use of decentralized controllers there is a linear growth of the computing time with the number of controlled axes. Consequently a single DSP would be more than sufficient to implement the controllers needed for a six axes manipulator. The unused computing time could be used in the computation of inverted transfer functions of the axes' models, in order to improve reference signal tracking performance, as presented in the last chapter. A flowchart of the controller software is presented in figs.6.8 and 6.9.

6.3 Experimental Results

Following the line of the work presented in the previous chapter, a set of experimental tests was made in order to evaluate the performance of the proposed controller. Most of these experiments involved moving axis 1 by ±0.2 rad around its middle position (nominally 1.309 rad) and axis 2 by ±0.2 rad around its in-line position with axis 1 (nominally 0.0 rad). These tests were done with several payloads (0, 60, 90Kg) and various demand signals (square waves in phase, square waves in quadrature, sine waves and triangular waves). In the earliest tests preformed, it was observed that excessive noise present on the feedback acceleration signals was degrading the controller performance in an unacceptable way. The cause of this could be traced to the way relative link accelerations are obtained: first, an analog position signal is differentiated by the resolver-to-digital converter originating an analog velocity signal; then, this analog velocity signal is sampled, quantized and numerically differentiated, using a backwards difference. This means that the acceleration signal is obtained by double differentiation of the position signal, and throughout the process there are several noise sources present. This problem was acceptably solved by clipping and strongly quantizing this signal to a low resolution. In this way, the signal noise was masked out by a well known and less harmful quantization noise and a satisfactory behaviour was obtained.

The collection of manipulator and controller data was performed by a PC based data logging system. This system was completely independent of the controller in terms of software and hardware. The recorded signals were:

axes' positions - The outputs of the D/A converters, that convert the outputs of the resolver-to-digital converters into analog form, used expressly for this purpose.

axes' velocities - Exactly the same signals as used by the axes' controllers.

axes' position references - Exactly the same signals as used by the axes' controllers.

controllers outputs - The outputs of the DSP D/A converters, prior the voltage-to-current conversion.

time - Data-logger internally-generated time information.

The ASCII files containing the recorded signals were afterwards imported into Matlab. Matlab was used to perform scaling and plotting. It was also used to generate the time functions of the desired behaviour of the position and velocity signals, for plotting purposes, using the model expressed by equ.6.4 (the same that was indirectly used by the axes' controllers).

In a first subset of tests, the manipulator was excited using square waves in phase, and with different manipulator payloads. The first test was realized with null payload and in the second one maximum design payload (90Kg) was used. The position results obtained are presented in fig.6.10 for the first case and fig.6.13 for the second case. In these plots the manipulator behaviour is compared with the ideal behaviour that would result from perfect model following of equ.6.4. From this results it can be concluded that the manipulator follows the model of desired behaviour with small deviations and that there is only a small sensitivity to the payload value. The effects of the use of a reduced v_e loop bandwidth, limited by the dynamic capabilities of the servo-valves utilized, may be appreciated in the velocity signals presented in figs.6.11 (null payload) and 6.14 (90Kg) where the actual axes velocities are compared to the desired ones (the velocities that would result from PMF of equ.6.4). These plots also show the large amount of noise present in the velocity feedback signals. The obtained control actions are presented in figs.6.12 (null payload) and 6.15 (90Kg). These are both smooth and free of high frequency components that could induce a chattering behaviour. The offsets present in the control action signals are due to the gravity induced torques. The servo-valves used have a less-than-ideal pressure gain, meaning that a finite control action is needed to compensate for the weight of the manipulator links. The servo-valve model used in the previous chapter simulations has a pressure gain that is much nearer the ideal one (infinity), although presenting a finite value. Other sources of offset that are surely present are servo-valve and servo-valve driver offsets.

A second subset of tests was made with a payload of 60Kg and different input time reference functions such as sine waves, triangular waves and square waves in quadrature:

- The position and velocity results obtained with sine (figs.6.16 and 6.17) and triangular waves (fig.6.18 and 6.19) show how the manipulator axes perform when following their reference models with these classes of input signals, presenting the expected type-one system dynamic behaviour. The model following overshoots observed are due to the forced reduction on the information content of the acceleration signals, that contributes to the low bandwidth of the v_e loop. These show up precisely when the acceleration information is most needed leading to the conclusion that although the control algorithm is performing as expected, a better way of obtaining relative acceleration information would result in a superior overall performance.

- Following the steps of the simulation studies presented in the last chapter, two other tests using square waves in quadrature as input references were performed using the same ±0.2 rad amplitude but around two different manipulator positions. These tests enable an evaluation of the disturbance rejection and decoupling capabilities achieved by the proposed adaptive control algorithm, and its relation with the working position. The first test is similar to the other tests presented up to now, but the second one was performed with link 1 moving nominally around vertical ($\theta_1 = \pi/2$ rad) and link 2 moving nominally around horizontal ($\theta_2 = -\pi/2$ rad). The obtained position and velocity results are presented in figs. 6.20 and 6.21 for the first case and 6.22 and 6.23 for the second case. A good degree of disturbance rejection (and, consequently, of decoupling) is achieved by the adaptive controller and its model following properties were found to be insensitive to manipulator working position.

In a third subset of experiments, two more tests were executed in order to better evaluate the degree of decoupling achieved by the adaptive controller. In the first of these tests, axis 1 is driven by a square wave, wile axis 2 is kept stationary by the controller. In the second test the situations are reversed. The payload used was 60Kg and the mean positions and input amplitudes are the same as those used for the first tests subset. The position and velocity results obtained are presented in figs. 6.24 and 6.25 for the first test, and 6.26 and 6.27 for the second one. These tests show that the controller is capable of immobilizing one axis while the other is moving, and illustrates its ability to decouple the reaction forces that each axis induces on the other.

From the set of experimental results obtained it can be concluded that the proposed decentralized adaptive axes' controllers are able to maintain the desired model following performance, with a large independence of manipulator working position, trajectory and payload. The achieved disturbance rejection capabilities, that result from the synthesis of a dedicated disturbance rejection signal (h) are very good. In this way, the use of decentralized controllers does not impair the manipulator performance. It has the advantage of not using any complex manipulator model, presenting a simpler computing task that grows only linearly with the number of manipulator axes. Only small model following errors result, and a good manipulator performance is achieved.

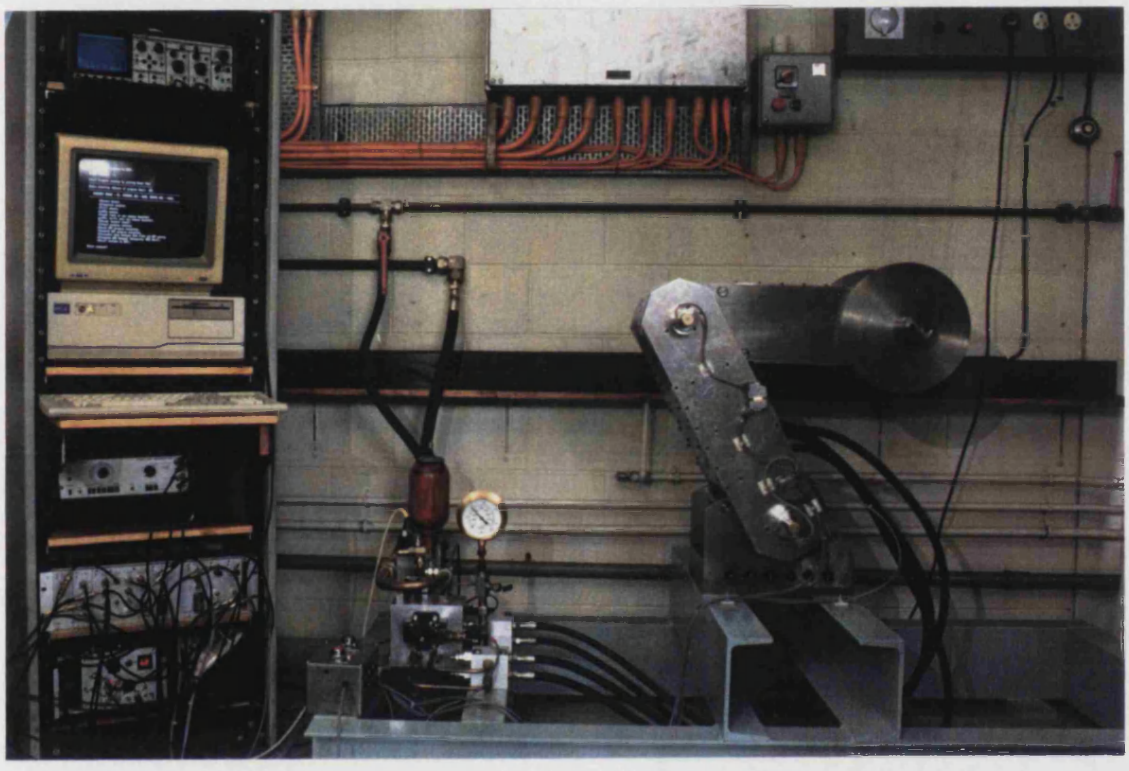


Figure 6.1 - Picture of the two-axes hydraulic manipulator

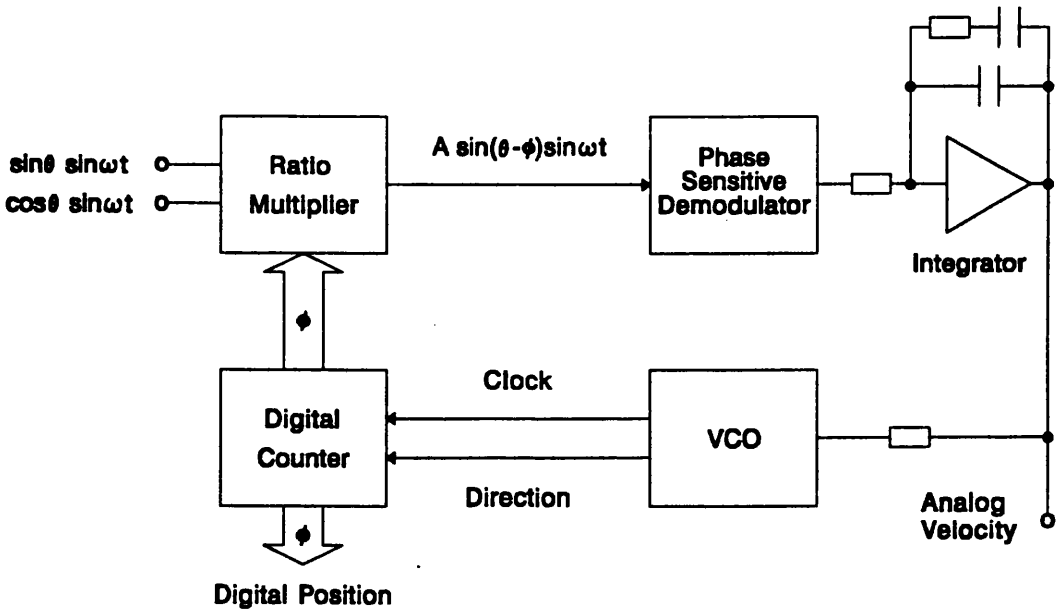


Figure 6.2 - Functional block diagram of the R/D converter

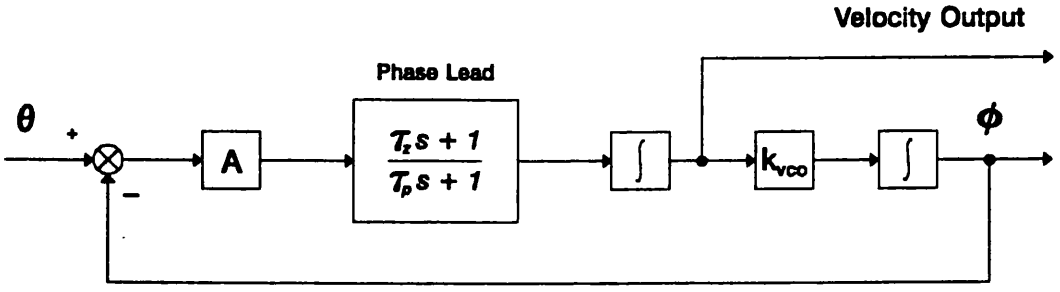


Figure 6.3 - R/D converter tracking loop

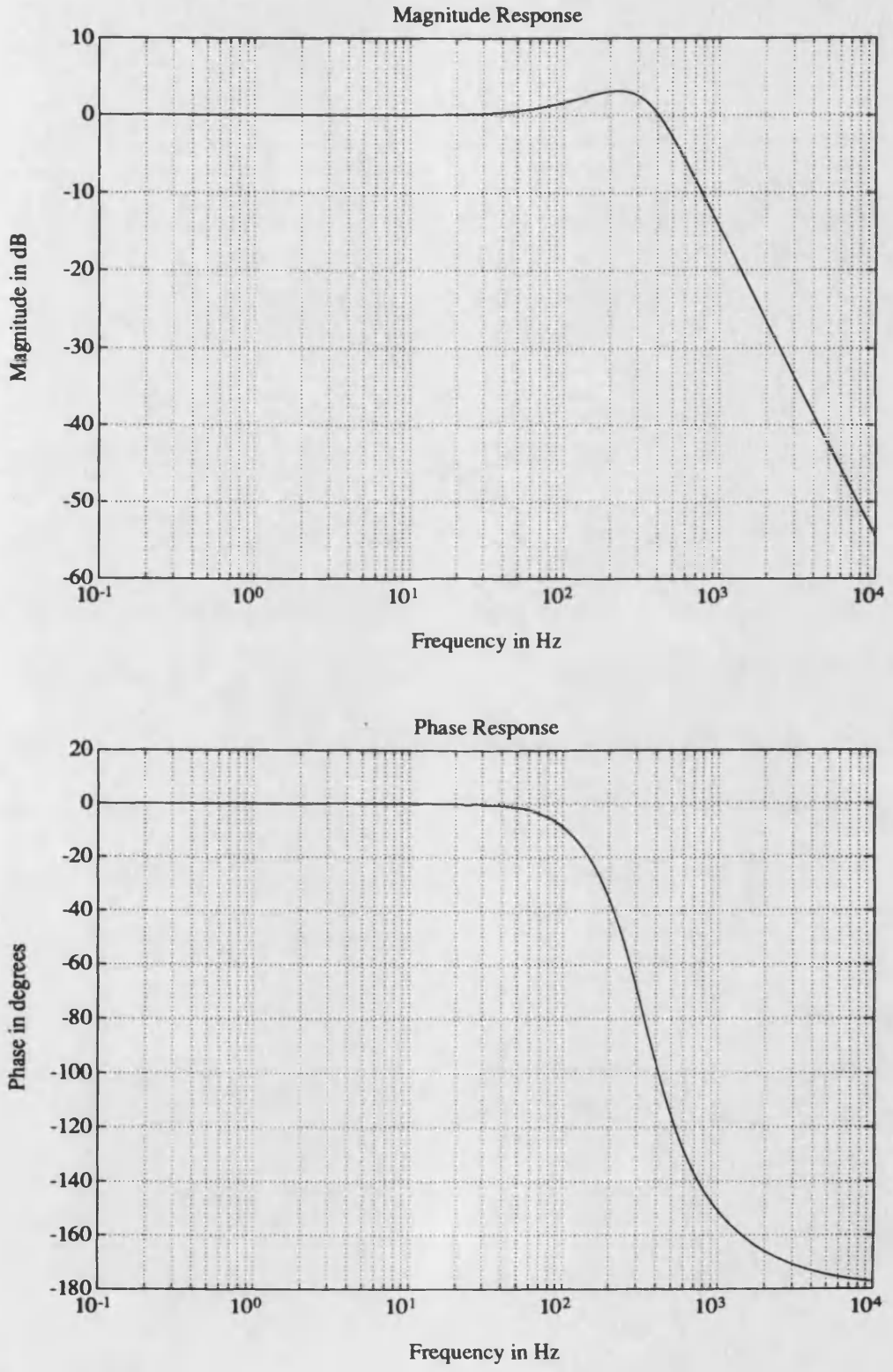


Figure 6.4 - Bode plot of the R/D converter closed loop T.F.

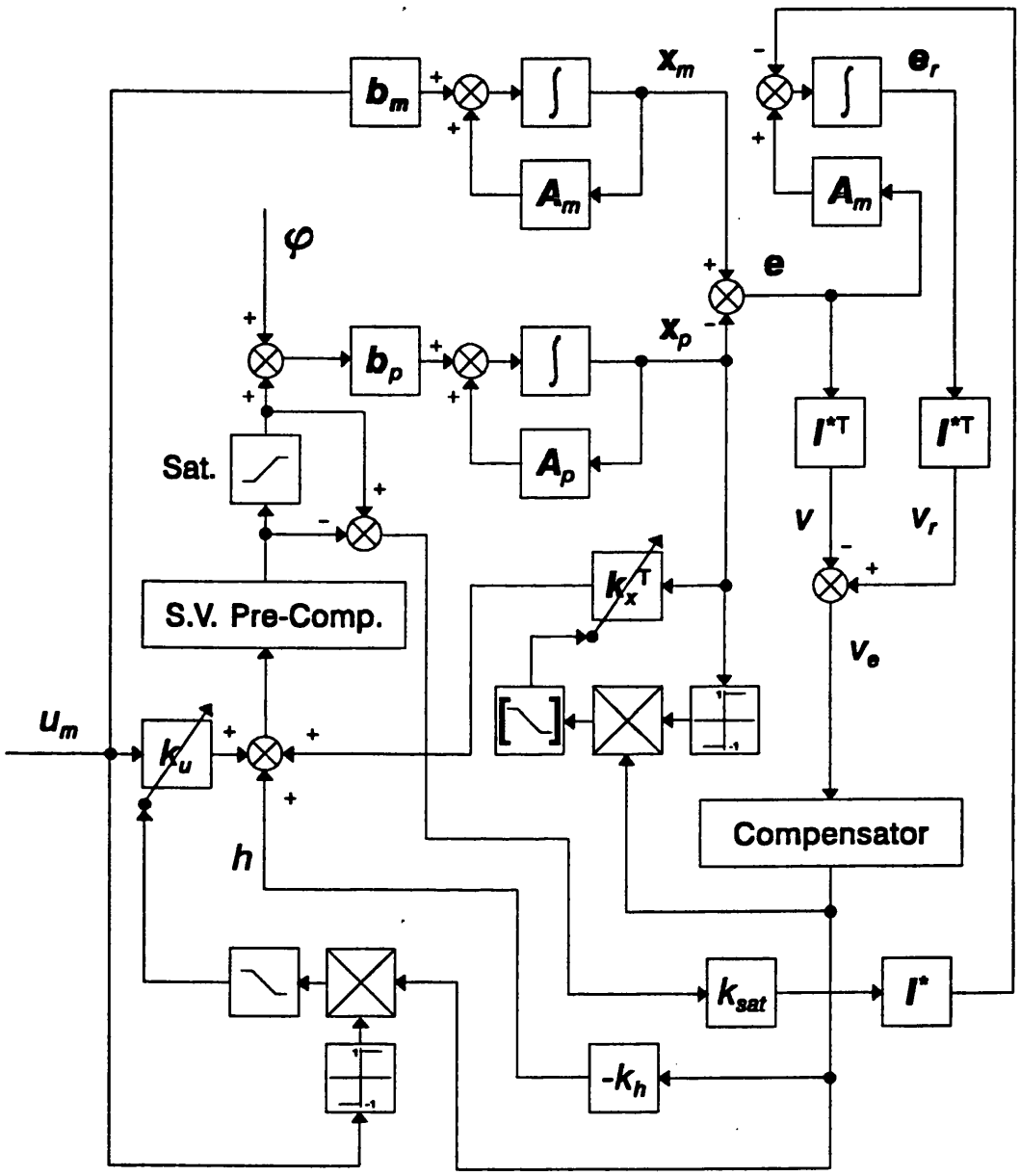


Figure 6.5 - Block diagram of the adaptive velocity controller showing the servo-valve pre-compensator

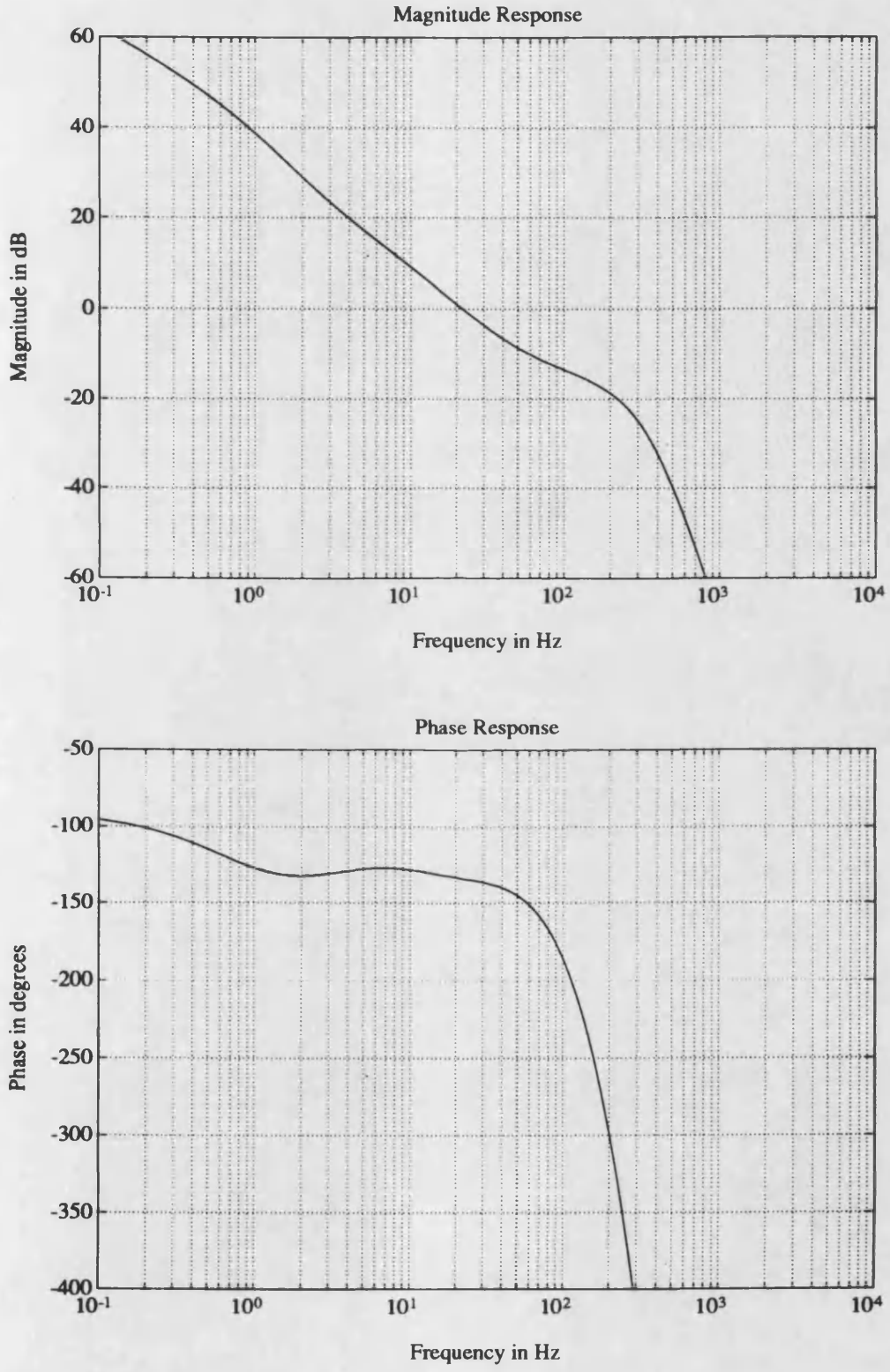


Figure 6.6 - Bode diagram of the open-loop frequency response of the v_e loop

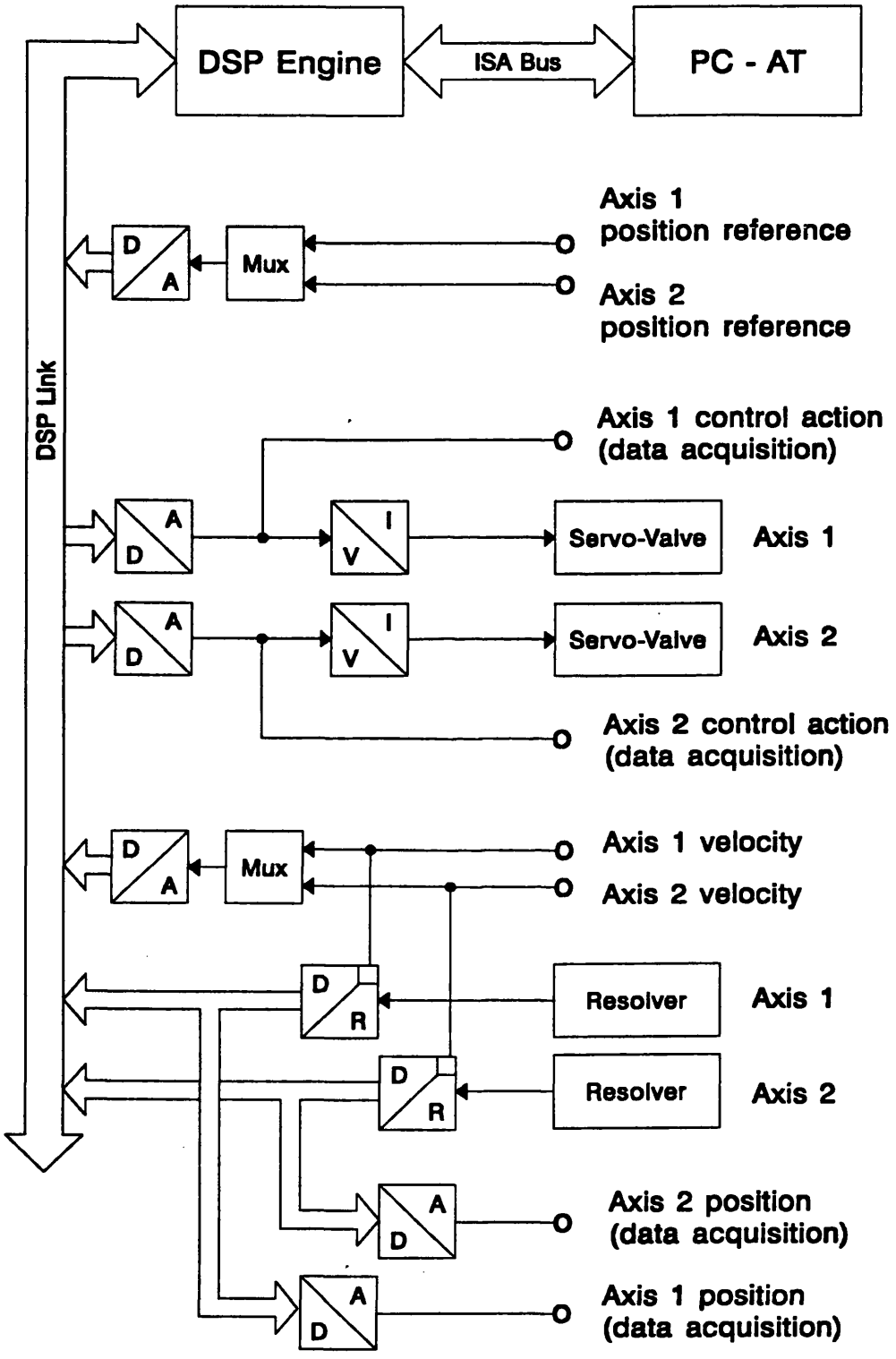


Figure 6.7 - General diagram of the controller hardware architecture

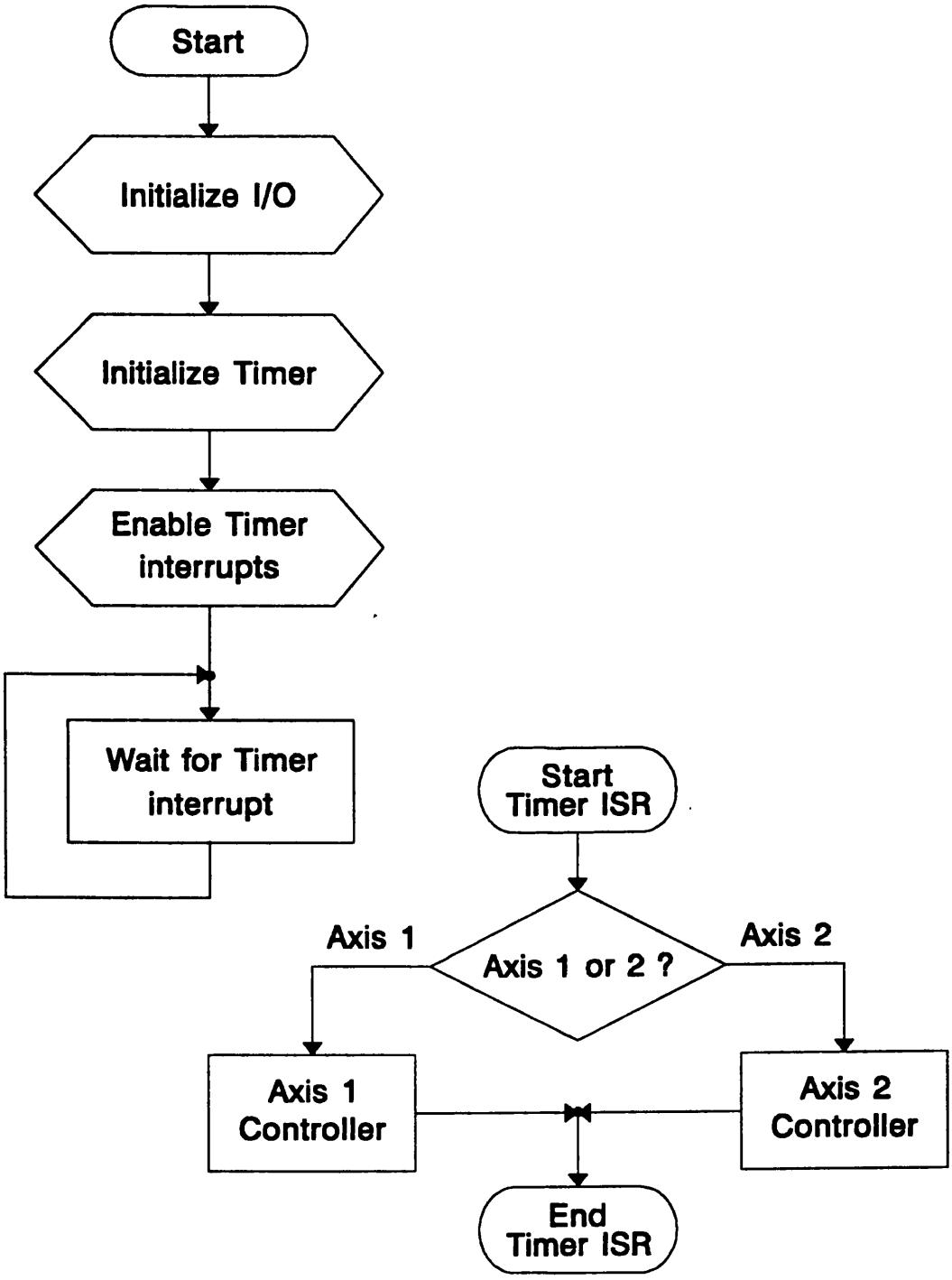


Figure 6.8 - Flow chart of the two axes controller software

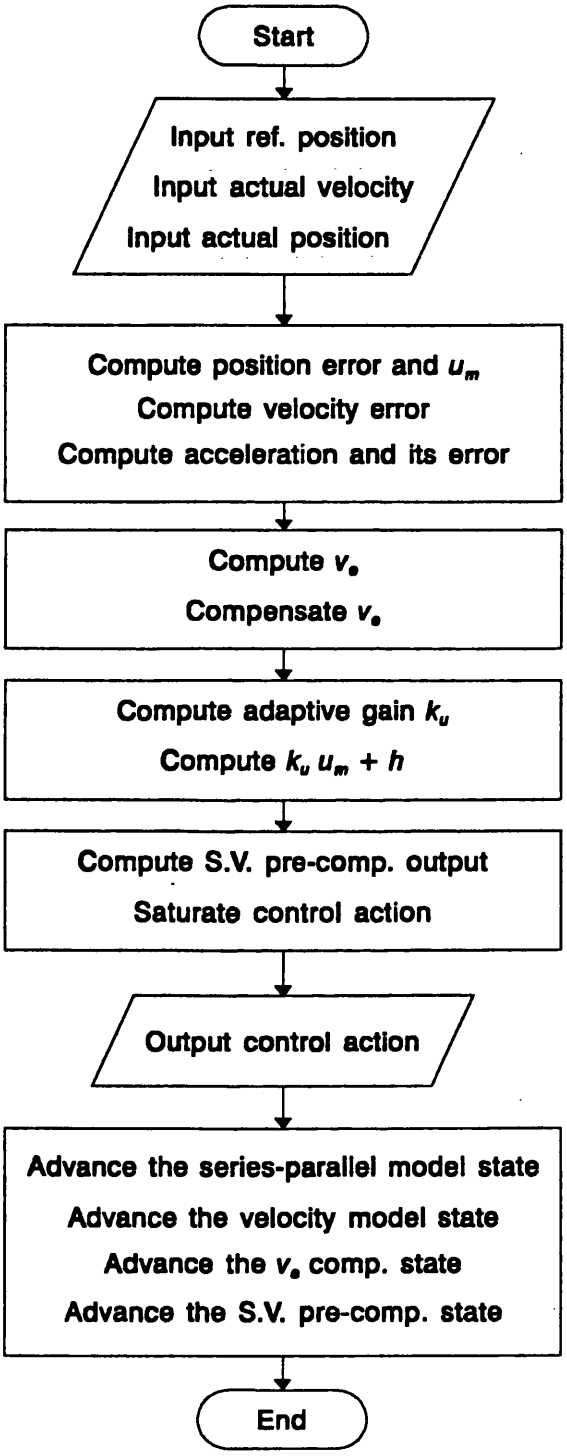
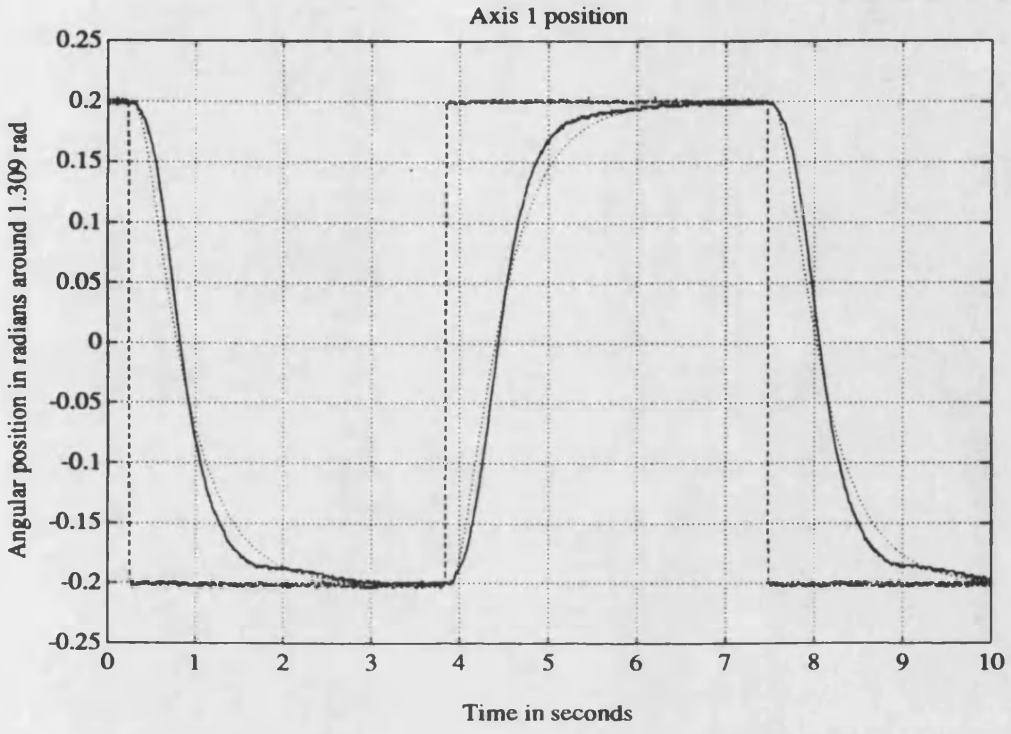


Figure 6.9 - Flow chart of the axis controller routine



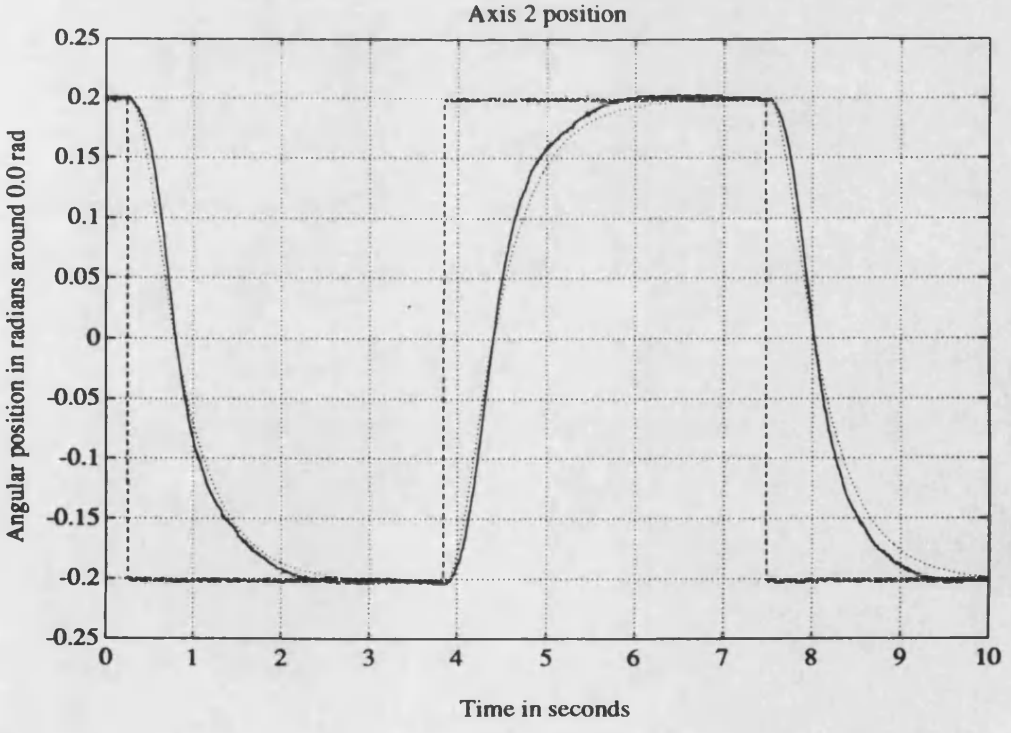


Figure 6.10 - Axes 1 and 2 position results (square wave, null payload)
---- reference ···· model —— axis

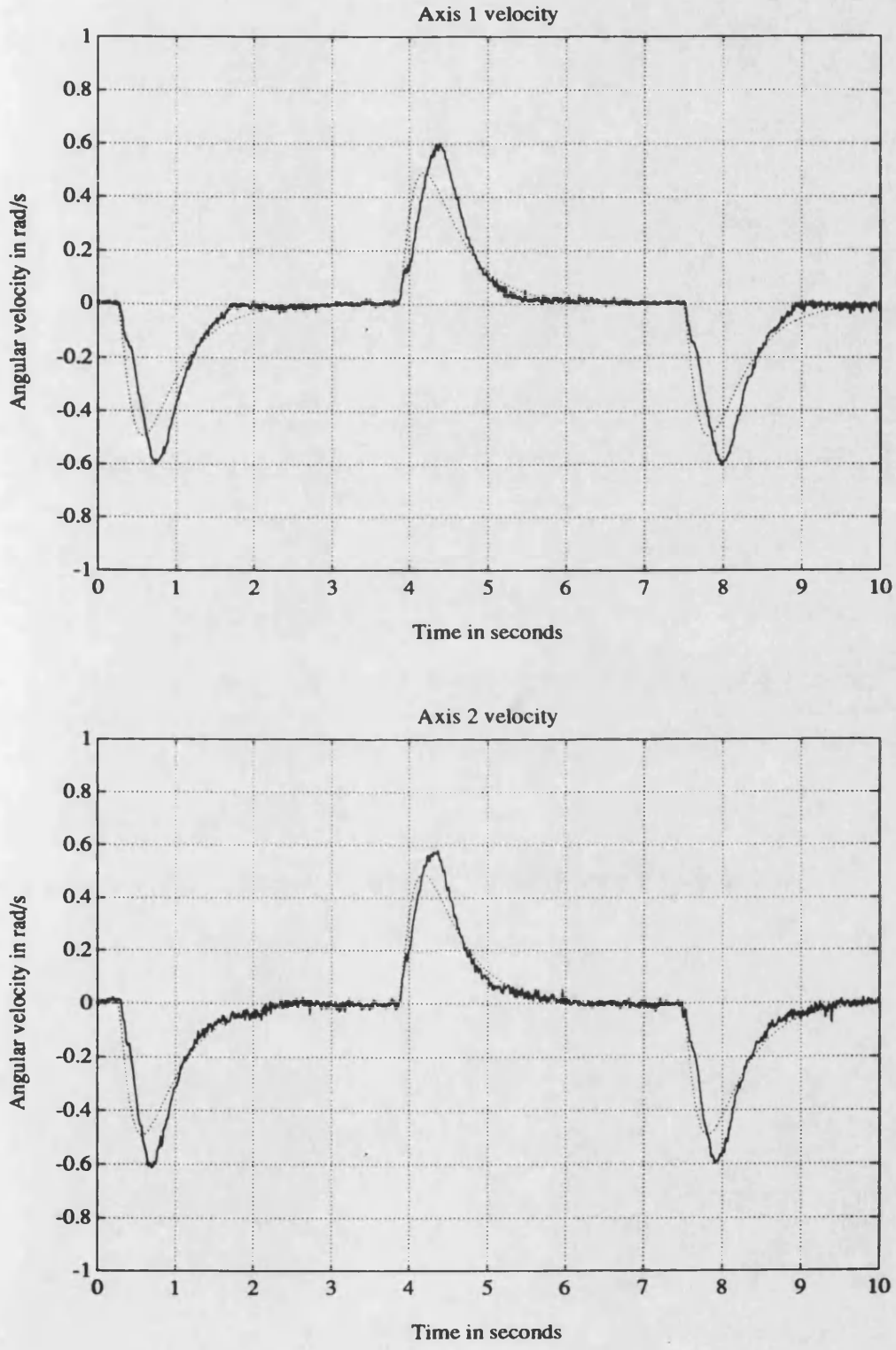
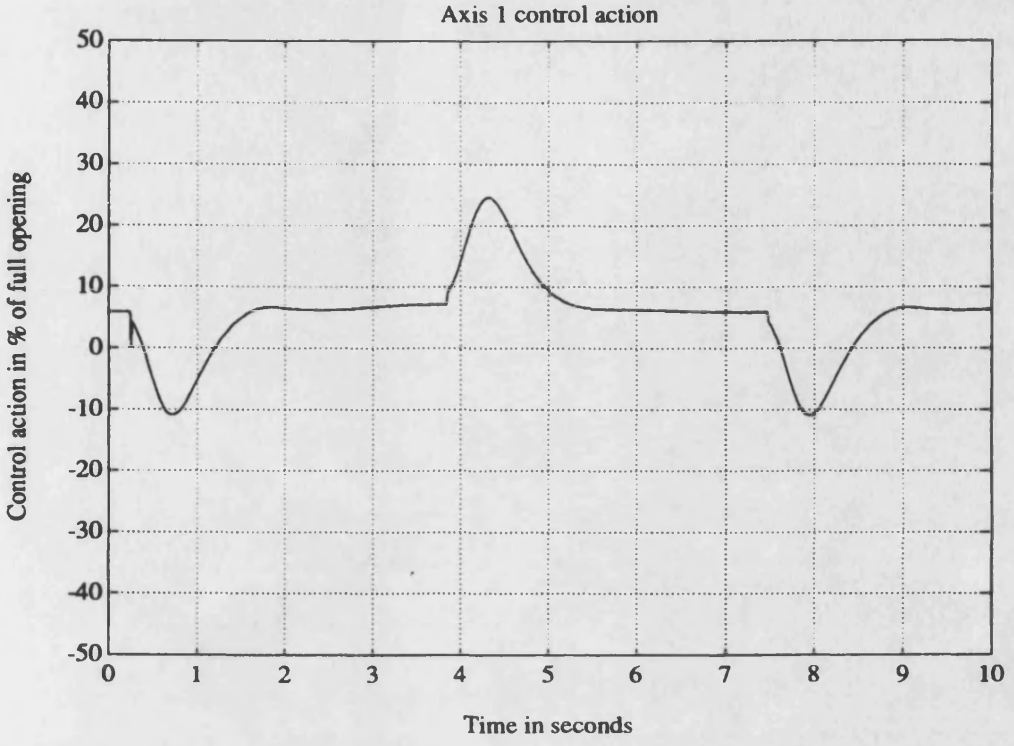


Figure 6.11 - Axes 1 and 2 velocity results (square wave, null payload)
···· model —— axis



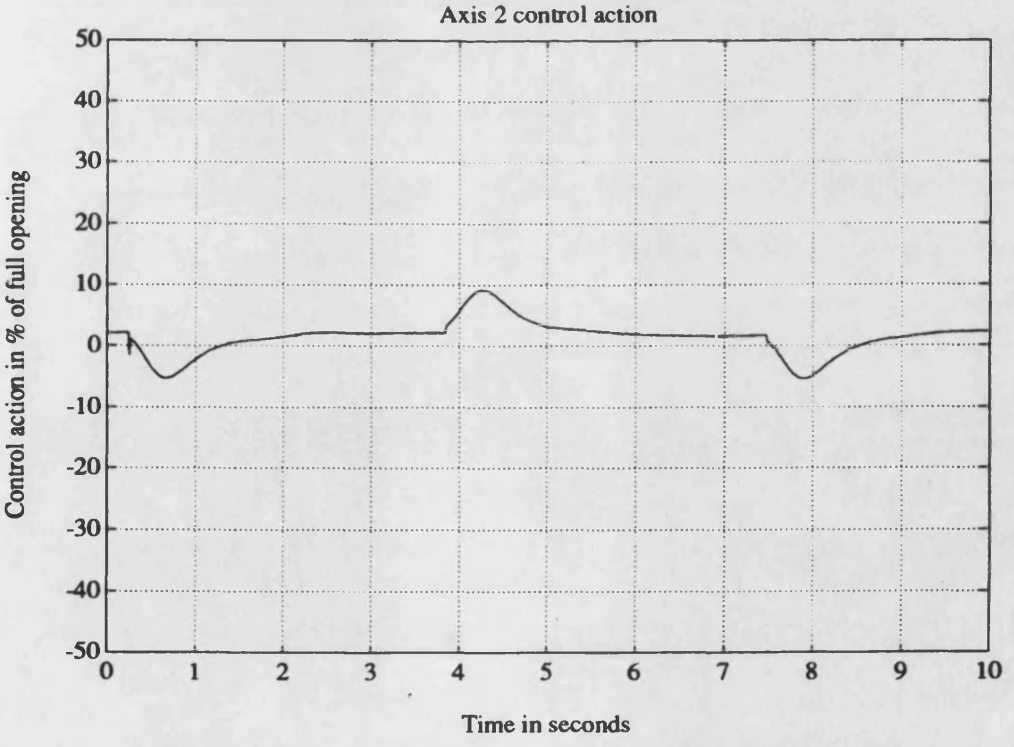


Figure 6.12 - Axes 1 and 2 control actions, u_p (square wave, null payload)

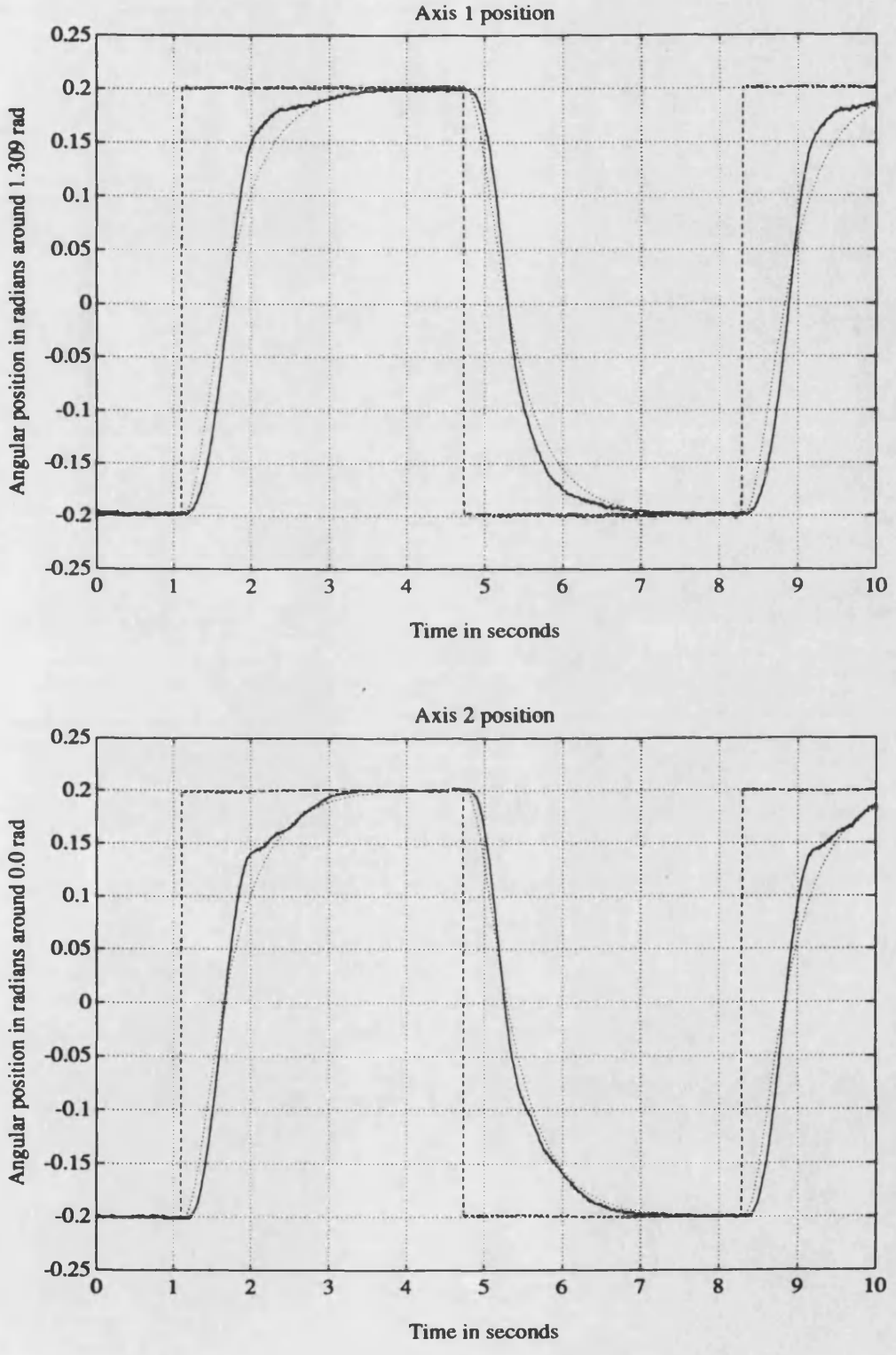


Figure 6.13 - Axes 1 and 2 position results (square wave, 90Kg payload)
---- reference ···· model —— axis

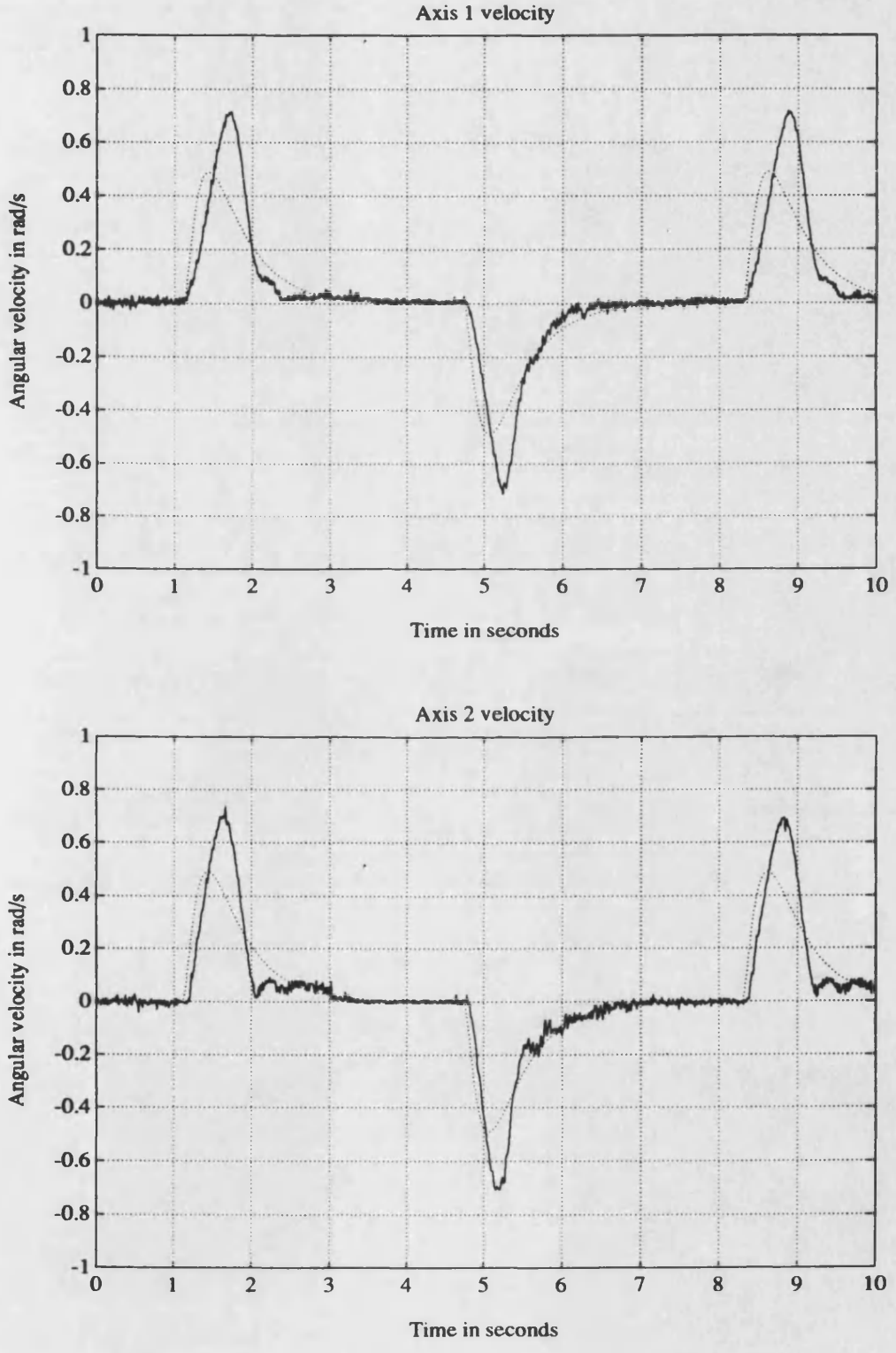


Figure 6.14 - Axes 1 and 2 velocity results (square wave, 90Kg payload)
···· model —— axis

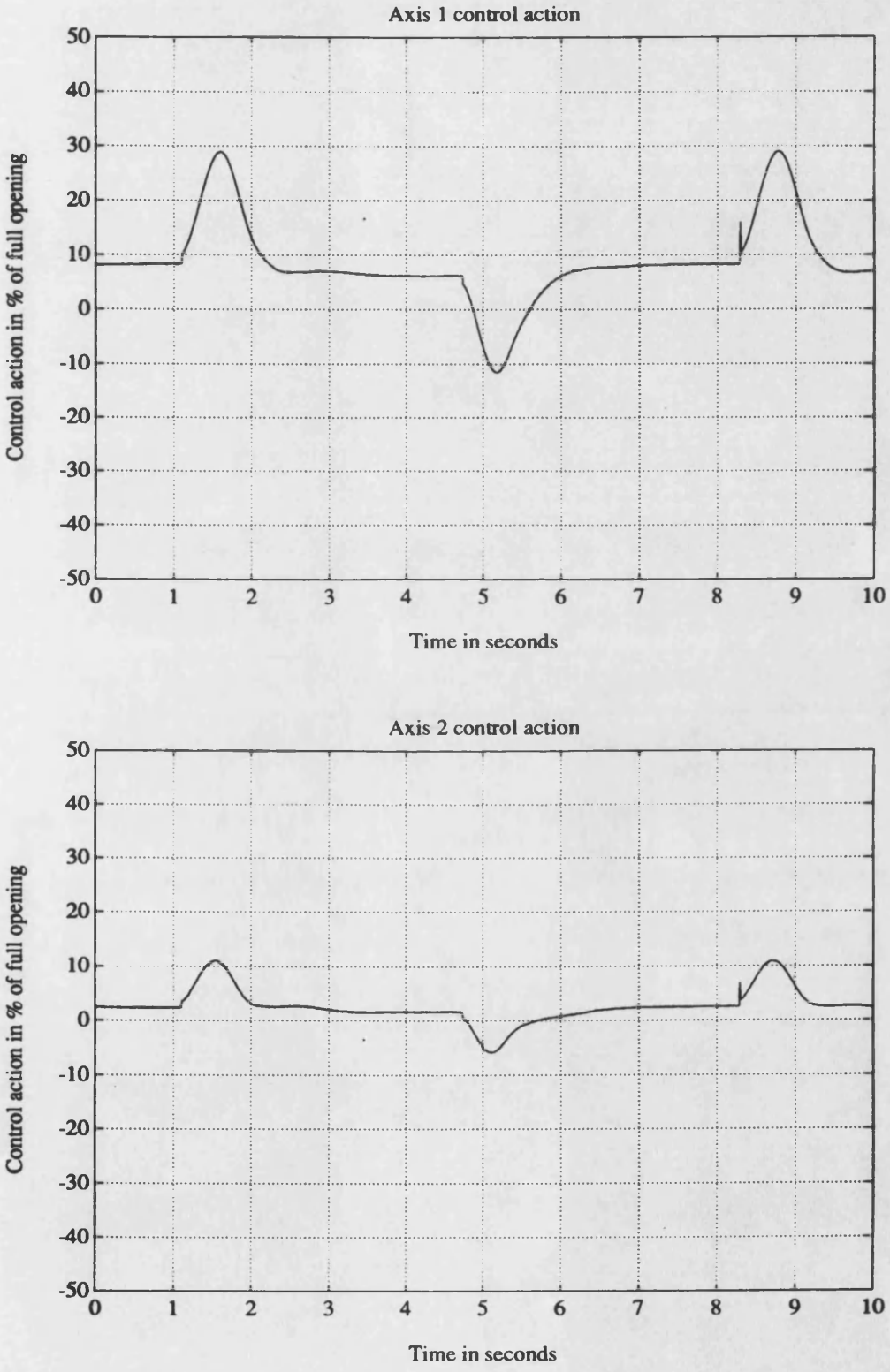


Figure 6.15 - Axes 1 and 2 control actions, u_p (square wave, 90Kg payload)

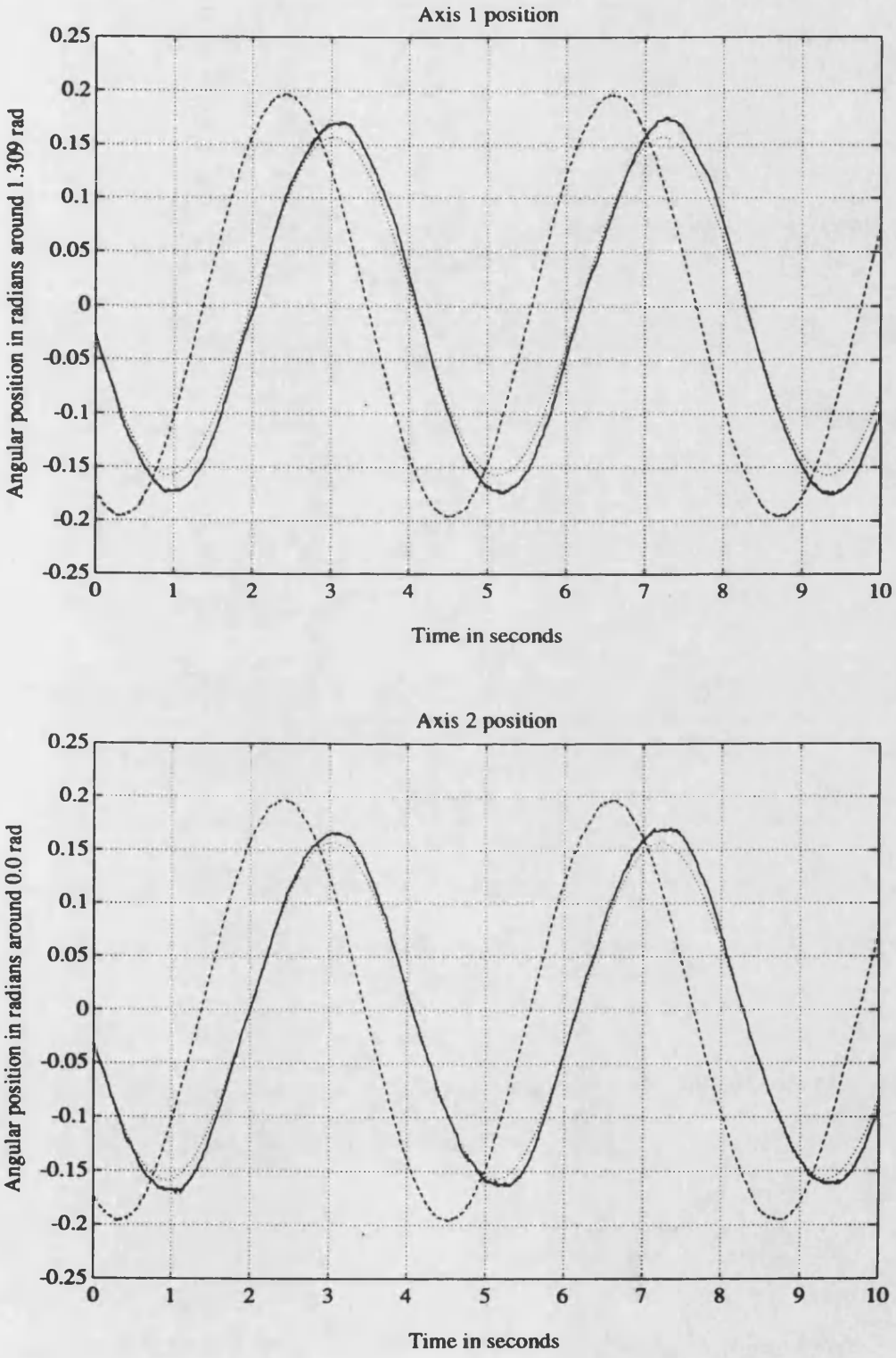


Figure 6.16 - Axes 1 and 2 position results (sine wave, 60Kg payload)
---- reference ···· model —— axis

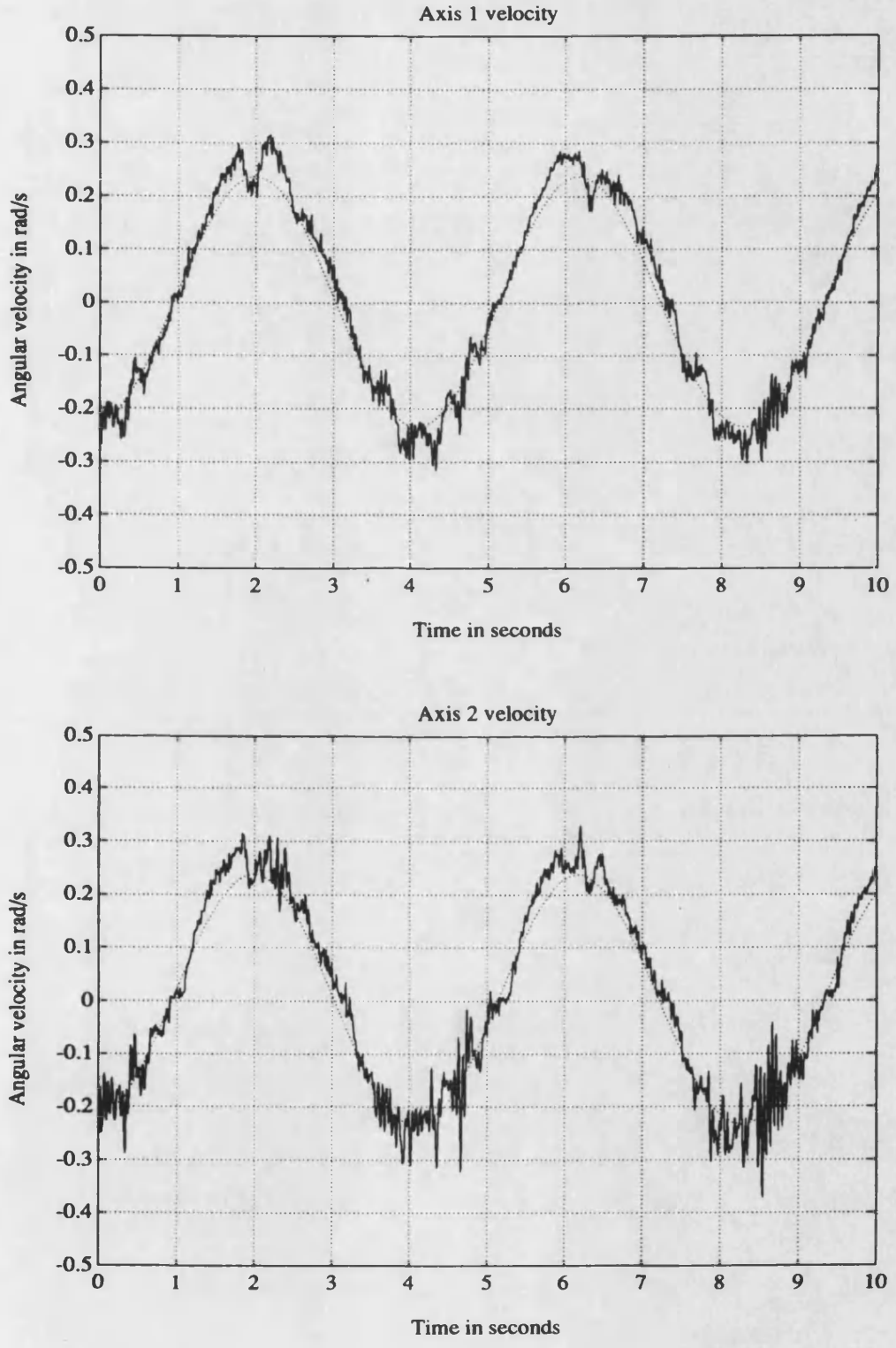


Figure 6.17 - Axes 1 and 2 velocity results (sine wave, 60Kg payload)
···· model —— axis

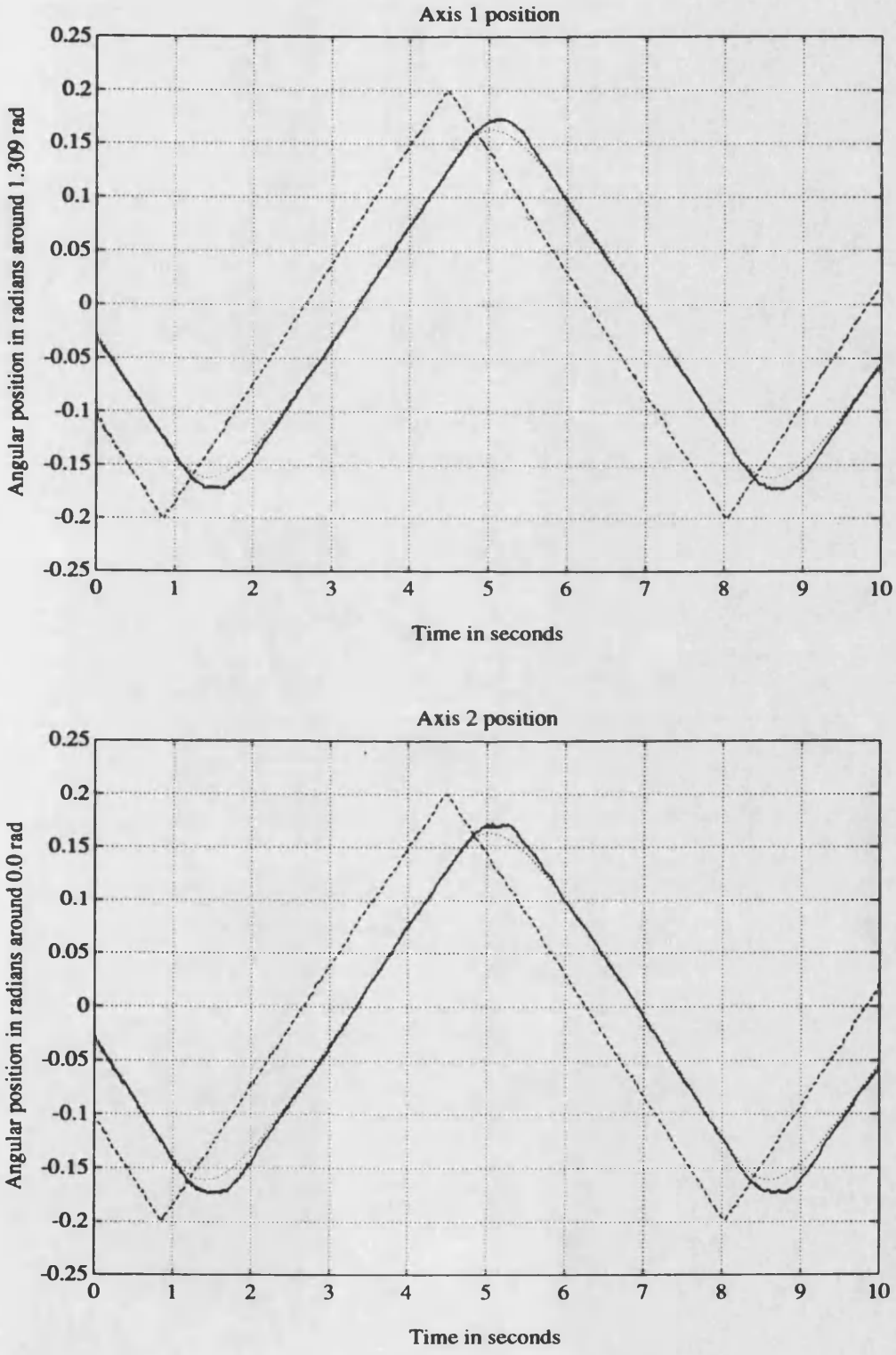


Figure 6.18 - Axes 1 and 2 position results (triangular wave, 60Kg payload)
---- reference ···· model —— axis

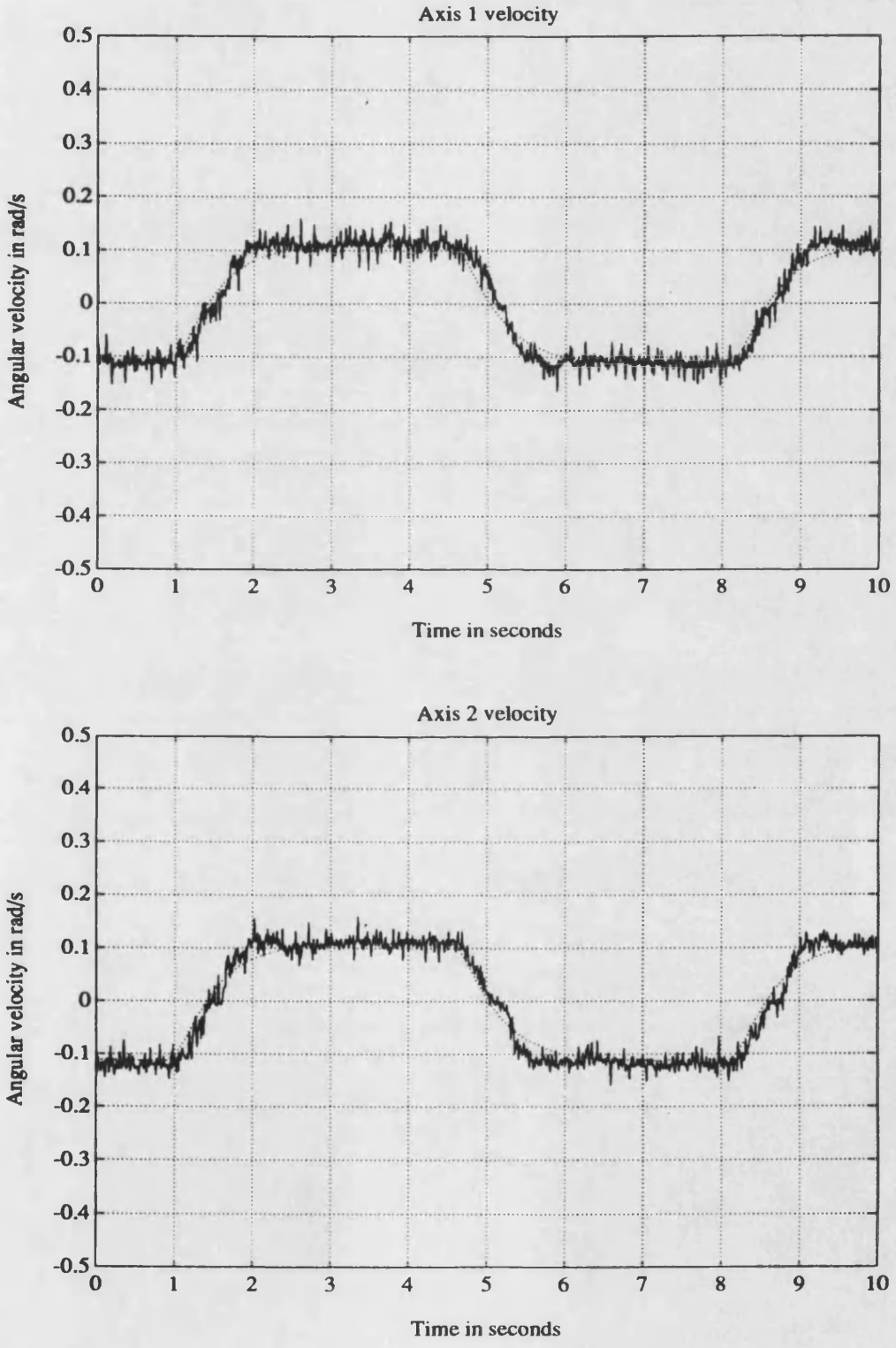


Figure 6.19 - Axes 1 and 2 velocity results (triangular wave, 60Kg payload)
···· model —— axis

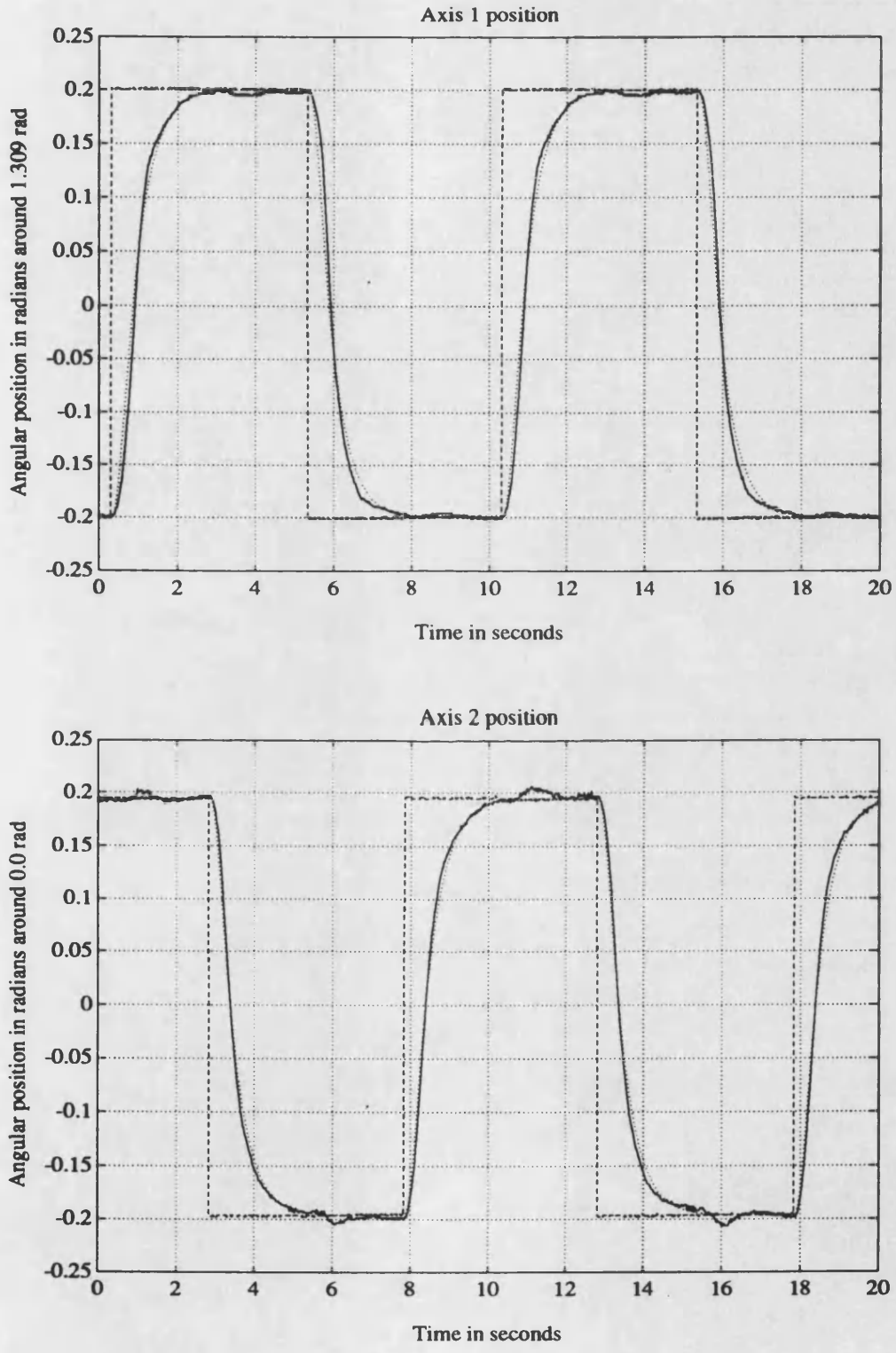


Figure 6.20 - Axes 1 and 2 position results (sq. waves in quadrature - 1st case, 60Kg)
---- reference ···· model —— axis

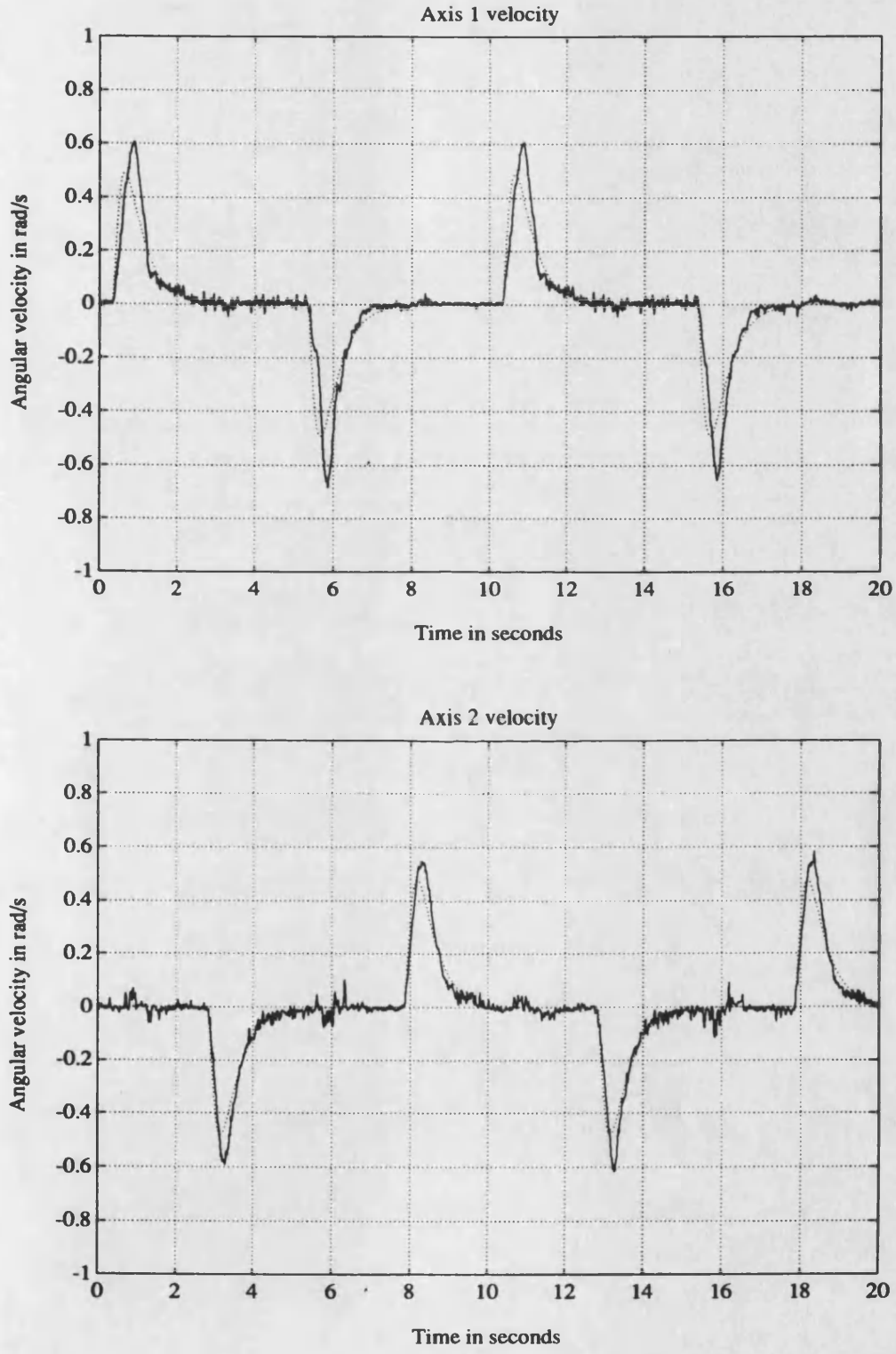
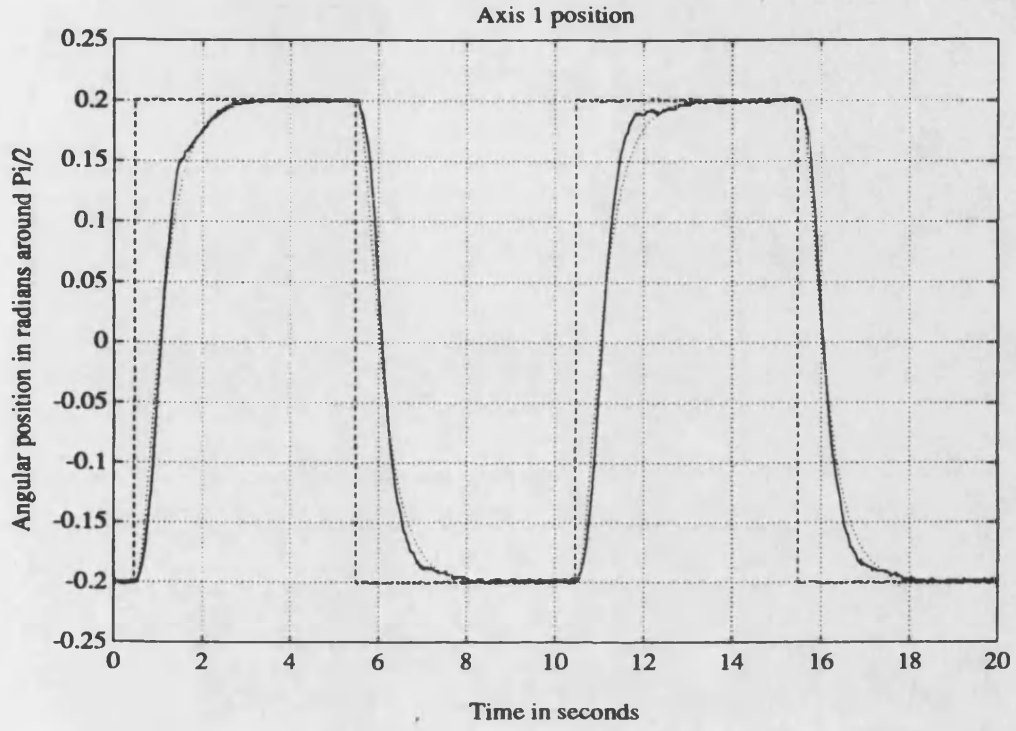


Figure 6.21 - Axes 1 and 2 velocity results (sq. waves in quadrature - 1st case, 60Kg)
···· model —— axis



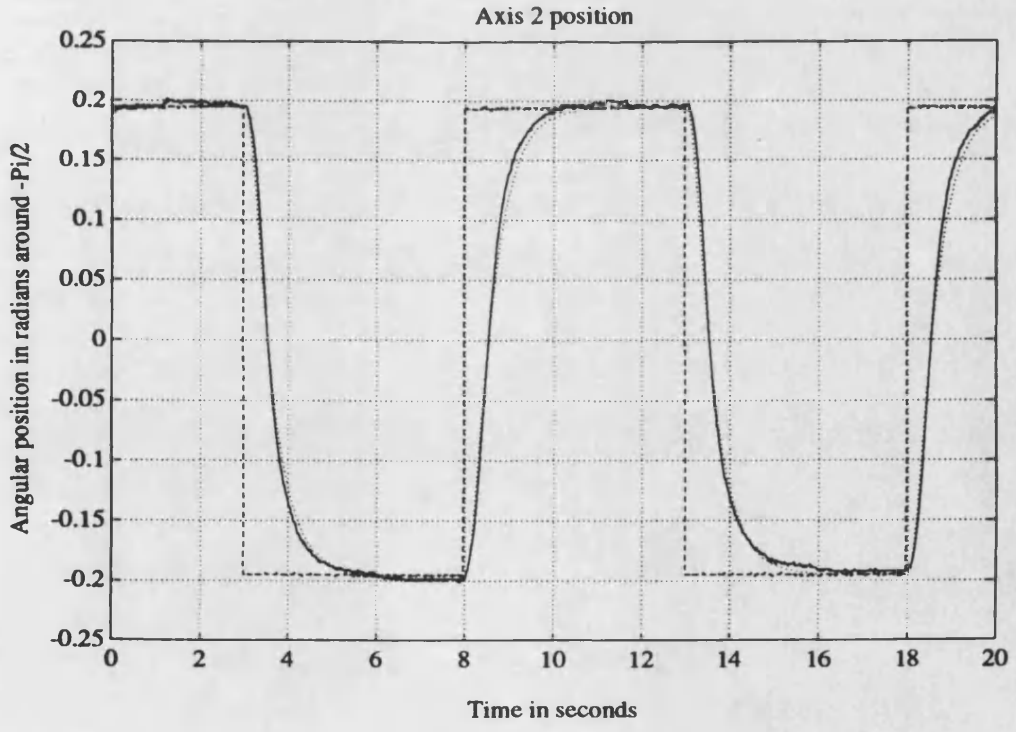


Figure 6.22 - Axes 1 and 2 position results (sq. waves in quadrature - 2nd case, 60Kg)
---- reference ···· model —— axis

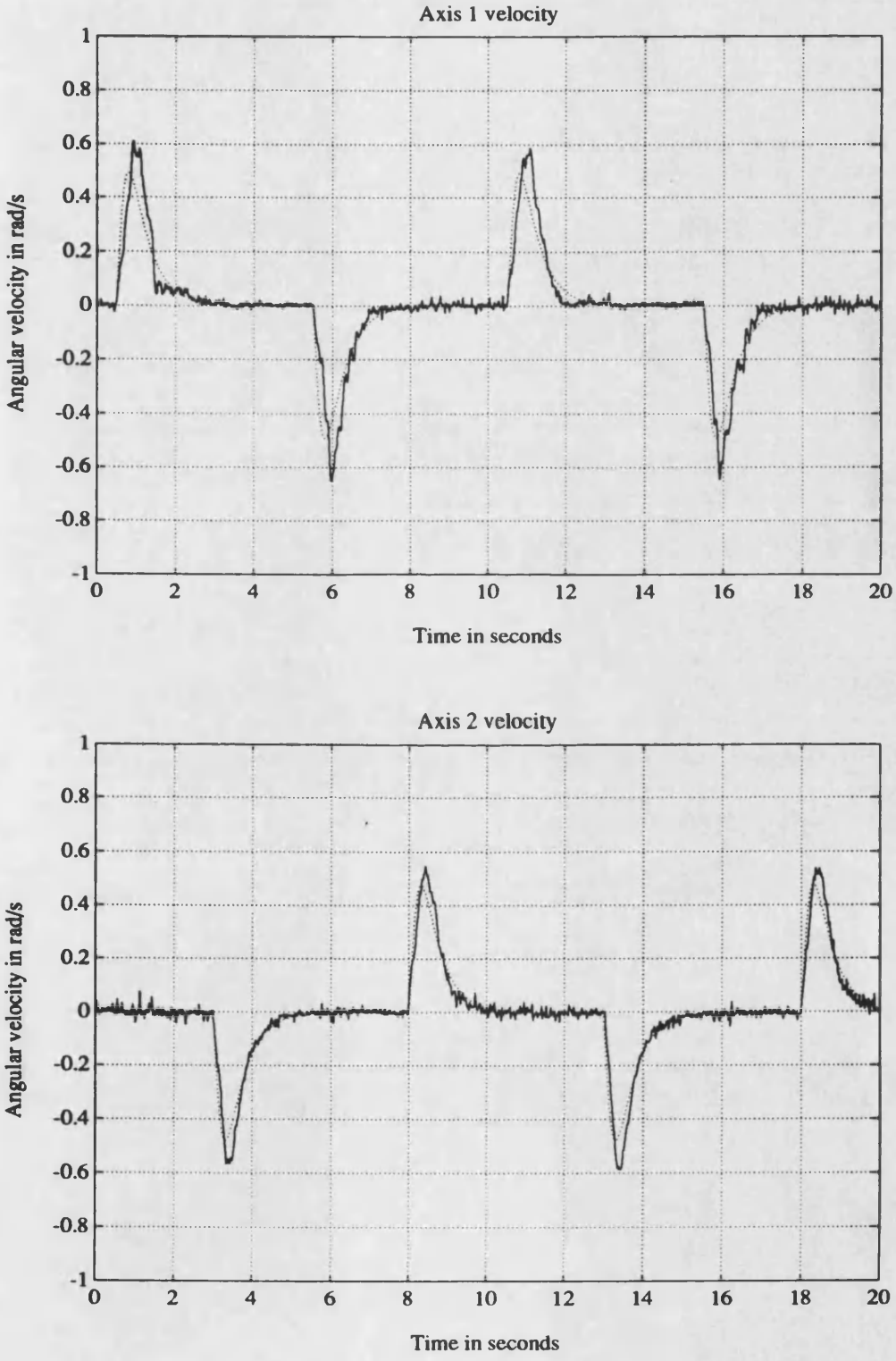


Figure 6.23 - Axes 1 and 2 velocity results (sq. waves in quadrature - 2nd case, 60Kg)
···· model —— axis

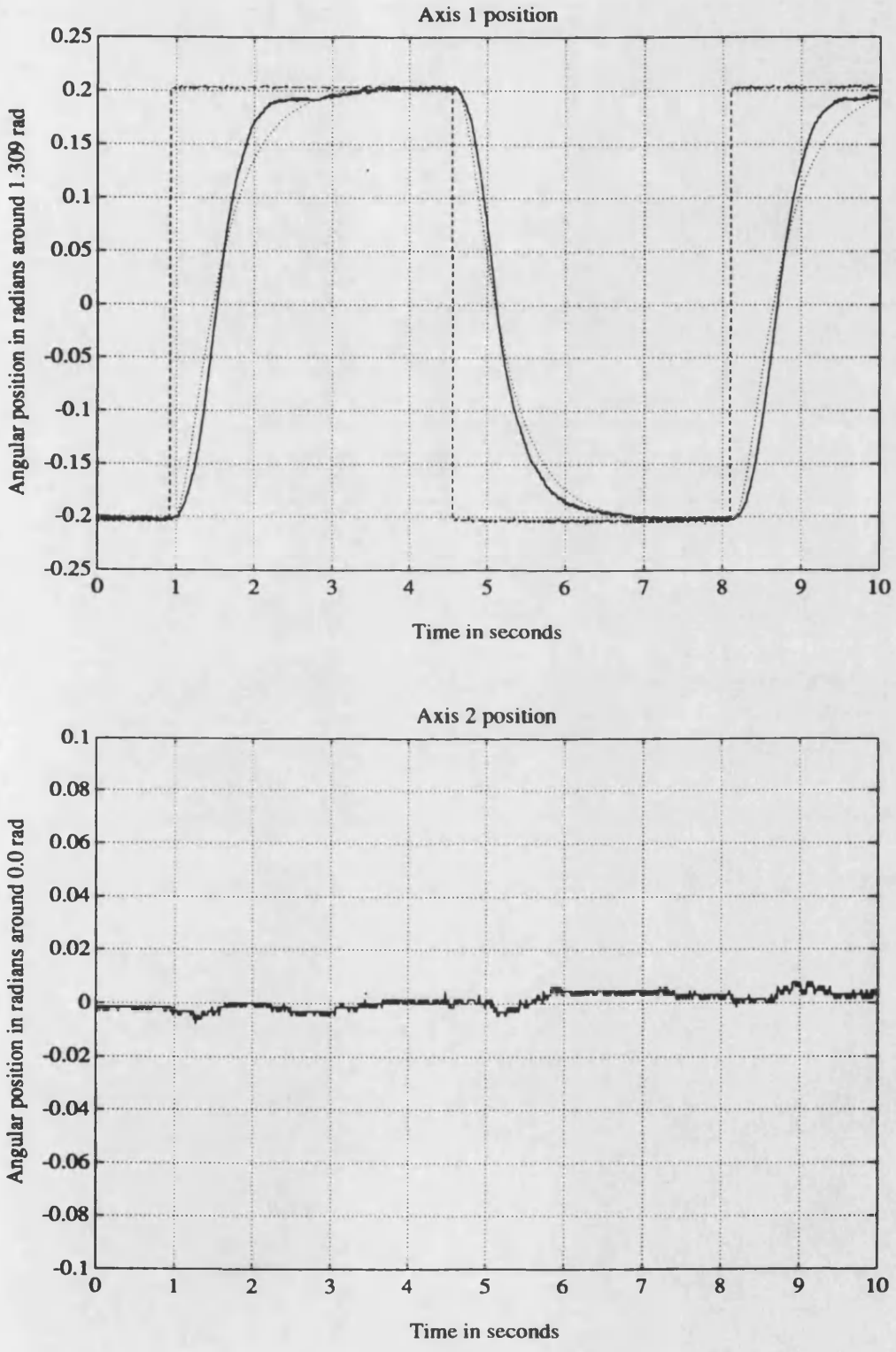


Figure 6.24 - Axes 1 and 2 position results (square wave, 60Kg payload, axis 2 stopped)
---- reference ···· model —— axis

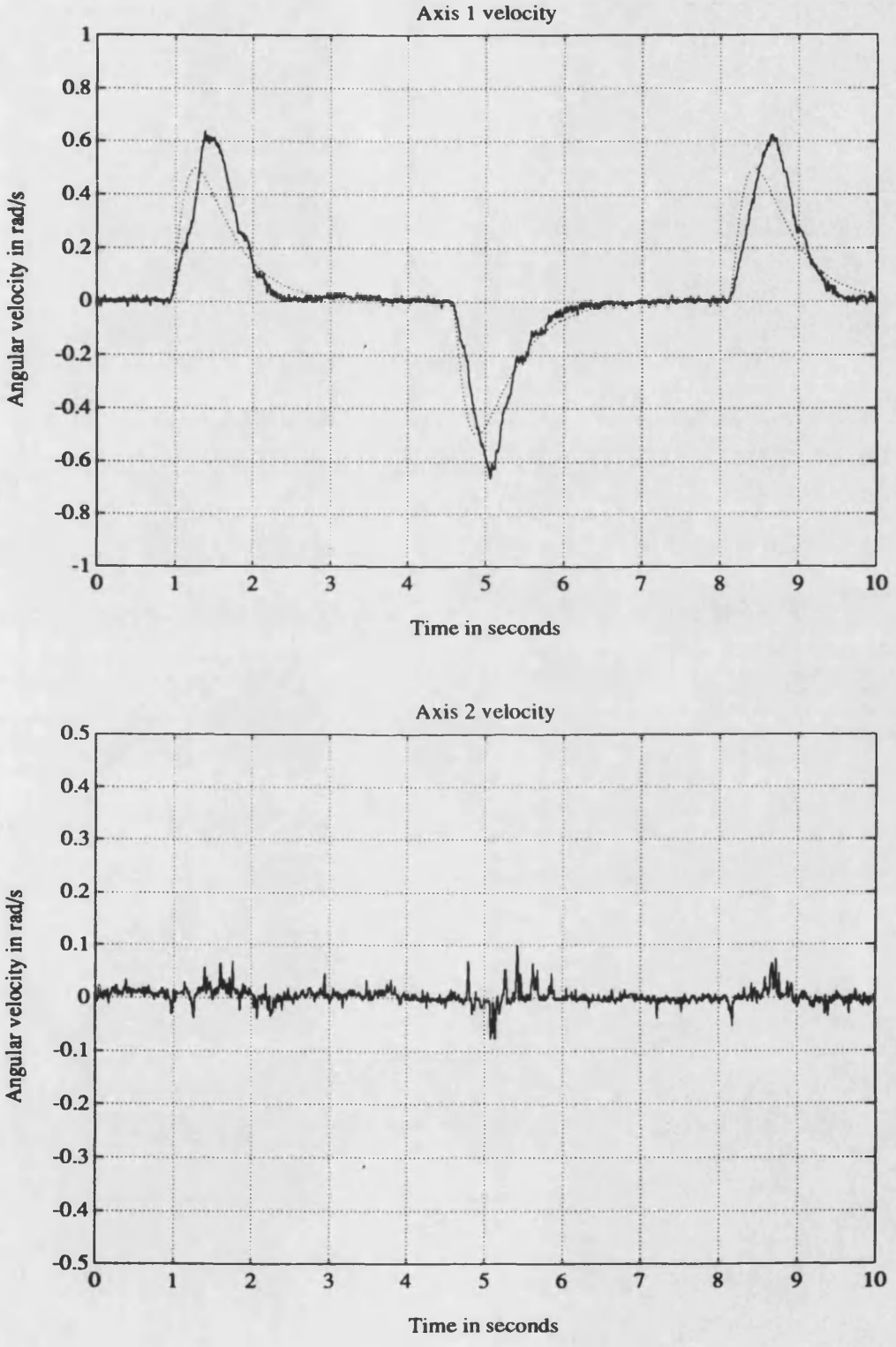


Figure 6.25 - Axes 1 and 2 velocity results (square wave, 60Kg payload, axis 2 stopped)
···· model —— axis

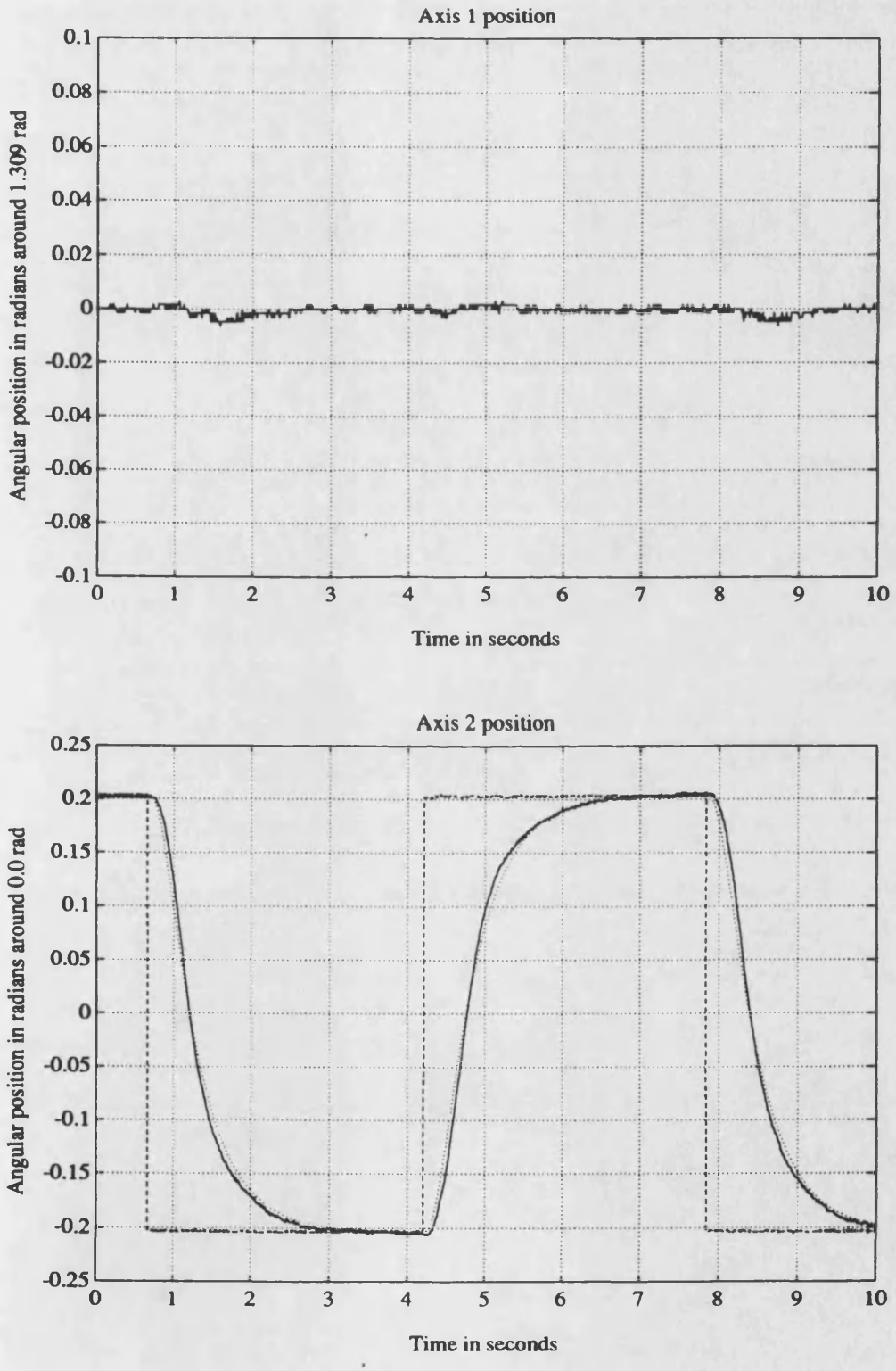


Figure 6.26 - Axes 1 and 2 position results (square wave, 60Kg payload, axis 1 stopped)
---- reference ···· model —— axis

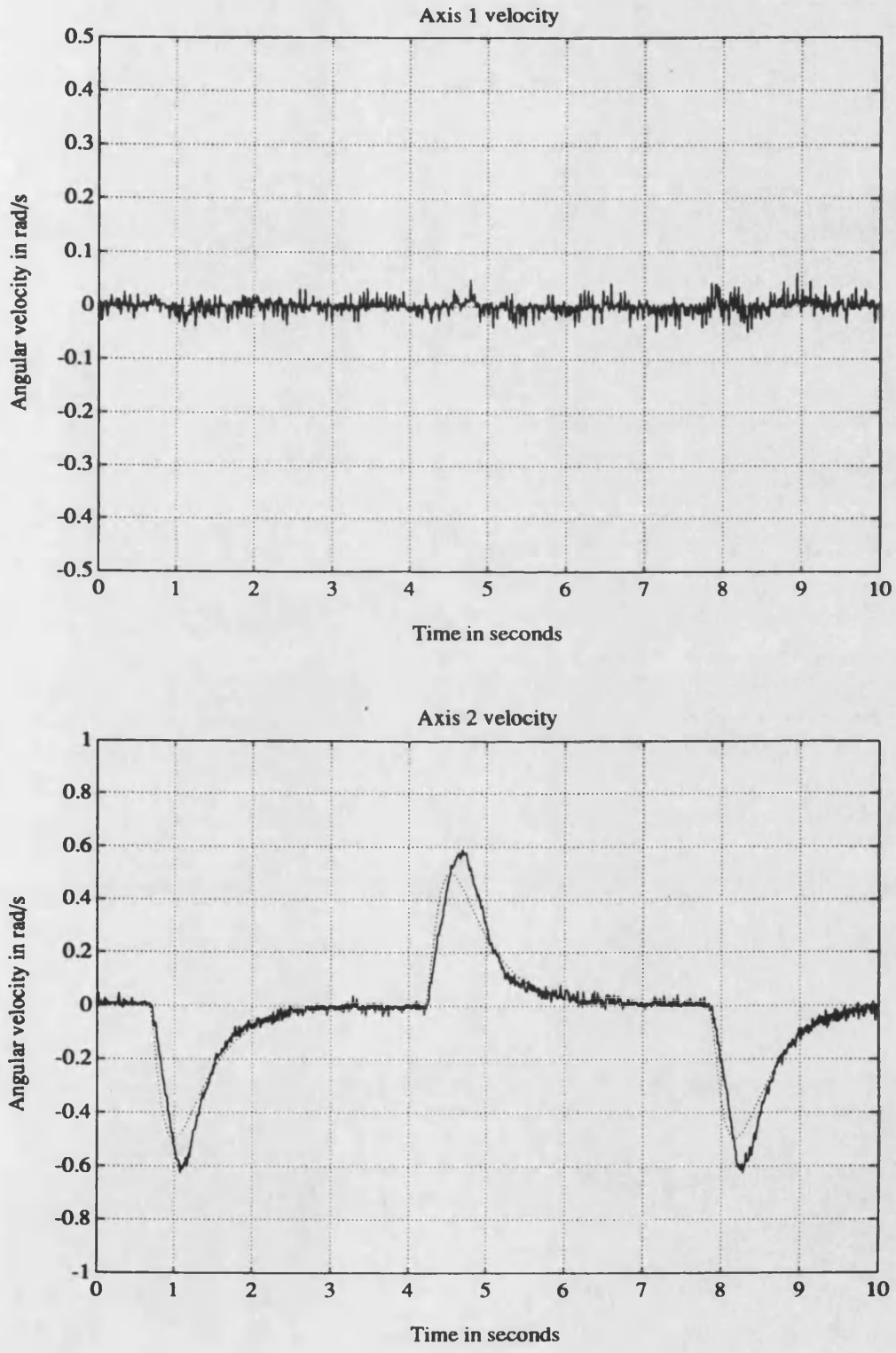


Figure 6.27 - Axes 1 and 2 velocity results (square wave, 60Kg payload, axis 1 stopped)
···· model —— axis

Chapter 7 | Conclusions

Robots are increasedly being used in industrial manipulation tasks. As the field of application of these general purpose machines broadens the need for faster action and smaller work cycle times increases. In order to achieve these goals the manipulator link acceleration capabilities must be improved. The recent trend for the utilization of direct drive actuators is a consequence. Direct drive actuation has the potential of offering, for the same link actuating torque, greater accelerations than those provided by a motor-reducer combination. Mechanical transmissions may disappear and, with the inherent simplification advantages, backlash is avoided. Nevertheless, the use of direct drive actuation has two important drawbacks:

- the actuator must be able to provide a high torque at low speed, which usually leads to heavier actuators. Hydraulic actuators are specially suited to this type of application, mainly in heavy-duty situations, given their high torque-to-weight ratio.

- as the reducer had the effect of decreasing the impact of the manipulator non-linear dynamics on the actuators, the control task for direct-drive systems is more difficult and the usual fixed gain joint controllers are no longer capable of offering an acceptable performance. A more complex control algorithm is needed. Given the highly non-linear dynamic behaviour presented by robotic manipulators, there is scope for the use of adaptive control.

Precise and fast robot control requires the use of adaptive algorithms. Some results showing this need are presented by Stoten [6].

A very attractive way of implementing robot adaptive control is to decentralize the controller, actually using n local controllers (one per axis). The alternative approach, using one global multivariable controller, often leads to a high numerical burden.

Model Reference Adaptive Control (MRAC) is an appropriate control technique for the manipulator control task. It provides an effective way of expressing the desired axis dynamic behaviour, by means of a reference model, and it does not require a manipulator model. It also has the capability of high speed adaptation, as its index of performance is readily available from the error between reference model state and system state. Model Reference Adaptive Control can be seen as an extension of Linear Model Following Control (LMFC), and as such, must obey the same structural conditions, the Prefect Model Following (PMF) conditions. In the classic literature the PMF conditions were established by taking into account the structural relation between the model and plant dynamics only, disregarding the effects of external disturbances on the system. As in the present case external disturbances are as significant as dynamic mismatches, an additional condition was taken into account in this work, relating the way disturbances and control action act on the system (it is a load controllability property). This extra PMF condition (the invariance with load condition) leads to the use of phase variables as state variables.

Using a decentralized control approach, dynamic interactions between axes are treated as external disturbances. This implies that the local controllers must be able to cope effectively not only with the time-varying properties of the axes, but with time-varying disturbances as well. A novel disturbance rejection signal is introduced in this work. The use of this signal, which increases the external disturbance rejection capabilities of the controller, enables the utilization of a decentralized

control architecture where each manipulator axis is independently controlled, with its own dedicated controller. The use of decentralized control has the advantage of providing a linear scaling between the computing time and the number of controlled axes.

Continuous and discrete time MRACs were developed, using Landau's hyperstability approach, with the modifications that occur from the introduction of the extra PMF condition, in order to improve the system behaviour in the presence of external disturbances. In the discrete time case, another modification was made to the Landau's method in the way the calculation delay is dealt with. Instead of "a priori" and "a posteriori" plant state vectors, a delay precompensation was used in the model.

Although the classical MRAC synthesis technique is based on hyperstability theory, the recognized robustness problems in the presence of unmodelled dynamics and external disturbances, preclude its use without modifications. Hyperstability is based upon a positivity condition. As any physical system has parasitic dynamics, the positivity condition is sure to be violated above a certain frequency, leading to the drift of the adaptive gains and so, system instability. Because these controllers were developed on the basis of an unknown but linear time-invariant plant, then, instability may occur when the actual plant is a time-varying system.

In a first step to address these problems, a MRAC was developed, still using hyperstability theory, without integral adaptive gains and taking into account the time-varying nature of the plant. The resultant controller has a VSS structure, and as such, uses discontinuous gains as well as preknowledge of plant parameter and external disturbances bounds. The possible occurrence of chattering, due to the discontinuous gains, and the potential lack of stability on under or overloaded conditions, pointed the need of modifications to the control algorithm. The obtained controller structure also lead to the conclusion that the desired controller properties would be easier to achieve if VSS theory was used for its synthesis.

A novel MRAC synthesis technique that avoids the aforementioned problems was developed. The utilization of this new technique, which is based upon Variable Structure Systems (VSS) theory, results in the use of memory-less adaptation gains, so they cannot drift. This enables the controller designer to establish a limit for the adaptation bandwidth, taking into account the presence of unmodelled dynamics above a certain frequency. The new design also supports the use of time-varying models under the restriction of fulfilment of the PMF conditions.

Usually the utilization of VSS theory results in controllers that, although very robust, have the inconvenience of using switching gains and so, are very prone to induce system chatter. This problem is solved, in the new proposed synthesis technique, with the use of a series-parallel reference model of the error behaviour, restraining the system from presenting a dynamic behaviour faster than the model behaviour. Also, the discontinuous gains are smoothed with a continuous interpolating function.

The new MRAC algorithm was used in the controller design of a two axis hydraulically-driven manipulator. Simulation and experimental results were obtained. In the simulation studies, high bandwidth servo-valves were used, allowing the utilization of fast reference models. As the main unmodelled dynamics are due to servo-valve dynamics, it was also possible to use high bandwidth adaptation loops, enabling fast adaptation capabilities. The simulation results obtained show that the controller is able to achieve very good model following independent of the payload used (0 to 90Kg) and manipulator working position. The control actions are always smooth and free of high frequency components. The simulation studies also show that the proposed adaptive controller behaves well in the face of feedback transducer noise and servo-valve saturation. Through the use of velocity and acceleration feedforward signals, as approximations to inverted reference model transfer functions input filters, very good input-output tracking performance was achieved. This could be achieved only because the adaptive controller effectively compels the manipulator axes to behave in the same manner as their reference models. For comparison purposes, simulations were also performed using a state feedback

controller. As expected, the large variations observed in the manipulator behaviour, when the payload is changed, are totally unacceptable.

The adaptive controller was then applied to a physical system, a purposely built two axes manipulator, directly driven by hydraulic rotary actuators. This manipulator has similar characteristics to the simulated one. A single Digital Signal Processor (DSP) was used for the discrete implementation of the controllers for both axes. Using this type of processor, tailored for the execution of digital filters, small computing times are achieved. After implementation it was found that with the sampling frequency used (1kHz) a single DSP would be sufficient for the control of a six axis manipulator. The models and compensator filters used in the controllers were implemented using the δ operator instead of the more usual delay operator. The δ operator was chosen because of its superior numerical properties, the delta filter structure presenting smaller discretisation errors. As the servo-valves used actually have smaller bandwidths than the simulated ones, slower reference models and reduced adaptation loop bandwidths had to be used. Excessive noise present in the acceleration feedback signal, due to the fact that this signal was effectively obtained by a double differentiation of the position signal, lead to a loss of acceleration information. The experimental work revealed that there is place for further work in the field of acceleration measurement. A relative acceleration transducer or a lower noise velocity transducer would certainly have improved the manipulator performance. The experimental results obtained are nevertheless a confirmation of the simulation behaviour: the new adaptive controller is capable of good model following performance and axis decoupling irrespectively of manipulator payload and working position.

References

1 - Fu, K.S.; Gonzalez, R.C. and Lee, C.S.G.; "Robotics: Control, Sensing, Vision and Intelligence", McGraw-Hill International Edition, 1987.

2 - Klafter, R.A.; Chmielewski, T.A. and Negin, M.; "Robotic Engineering, an Integrated Approach", Prentice-Hall International Editions, 1989.

3 - Merritt, Herbert E.; "Hydraulic Control Systems", John Wiley & Sons, 1967.

4 - Asada, H. and Slotine, J.-J. E.; "Robot Analysis and Control", John Wiley & Sons, 1986.

5 - Vukobratovic, M.; Stokic, D. and Kircanski, N.; "Kinematics and Trajectory Synthesis of Manipulation Robots", Scientific Fundamentals of Robotics 3, Springer-Verlag, 1986.

6 - Stoten, D.P.; "Model Reference Adaptive Control of Manipulators", Research Studies Press Ltd., John Wiley, 1990.

7 - Seraji, H.; "A New Approach to Adaptive Control of Manipulators", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.109, pp.109-202, September 1987.

8 - Landau, I.D.; "A Survey of Model Reference Adaptive Techniques — Theory and Applications", Automatica, Vol.10, pp.353-379, 1974.

9 - Landau, I.D.; "Model Reference Adaptive Controllers and Stochastic Self-Tuning Regulators — A Unified Approach", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.103, pp.404-416, December 1981.

10 - Åström, Karl J. and Wittenmark, Björn; "Computer Controlled Systems", second edition, Prentice-Hall International Editions, 1990.

11 - Åström, K.J. and Wittenmark, B.; "Self-Tuning Controllers Based on Pole-Zero Placement", IEE Proceedings, Vol.127, Pt.D, No.3, pp.120-130, May 1980.

12 - Vaughan, N.D. and Plummer, A.R.; "Robust Adaptive Control for Hydraulic Servosystems", ASME Winter Annual Meeting, Dallas, Texas, November 25-30, 1990.

13 - Landau, Yoan D.; "Adaptive Control — The Model Reference Approach", Marcel Dekker, New York, 1979.

14 - Seraji, H.; "Simple Method for Model Reference Adaptive Control", Int. J. of Control, Vol.49, No.1, pp.367-371, 1989.

15 - Courtiol, B. and Landau, I.D.; "High Speed Adaptation System for Controlled Electrical Drives", Automatica, Vol.11, pp.119-127, 1975.

16 - Edge, K.A. and Figueredo, K.R.A.; "An Adaptively Controlled Electrohydraulic Servo-Mechanism, Part 1: Adaptive Controller Design", Proc. Instn. Mech. Engrs., Vol.201, No.B3, pp.175-180, 1987.

17 - Edge, K.A. and Figueredo, K.R.A.; "An Adaptively Controlled Electrohydraulic Servo-Mechanism, Part 2: Implementation", Proc. Instn. Mech. Engrs., Vol.201, No.B3, pp.181-189, 1987.

18 - Dubowsky, S. and DesForges, D.T.; "The Application of Model-Referenced Adaptive Control to Robotic Manipulators", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.101, pp.193-200, September 1979.

19 - Nicosia, S. and Tomei, P.; "Model Reference Adaptive Control Algorithms for Industrial Robots", Automatica, Vol.20, No.5, pp.635-644, 1984.

20 - Erzberger, H.; "Analysis and Design of Model Following Control Systems by State Space Techniques", Proceedings of the Joint Automatic Control Conference, pp.572-581, Ann Arbor, 1968.

21 - Wang, S.H. and Desoer, C.A.; "The Exact Model Matching of Linear Multivariable Systems", IEEE Transactions on Automatic Control, pp.347-349, June 1972.

22 - Morse, A.S.; "Structure and Design of Linear Model Following Systems", IEEE Transactions on Automatic Control, Vol.AC-18, No.4, pp.346-354, August 1973.

23 - Drazenovic, B.; "The Invariance Conditions in Variable Structure Systems", Automatica, Vol.5, pp.287-295, 1969.

24 - Utkin, V.I.; "Discontinuous Control System: State of the Art in Theory and Applications", IFAC 10th Triennial World Congress, pp.25-44, 1987.

25 - Landau, Y.D. and Courtiol, B.; "Design of Multivariable Adaptive Model Following Control Systems", Automatica, Vol.10, pp.483-494, 1974.

26 - Stoten, D.P. and Benchoubane, H.; "Robustness of a Minimal Controller Synthesis Algorithm", Int. J. of Control, Vol.51, No.4, pp.851-861, 1990.

27 - Rohrs, Charles E.; Valavani, Lena; Athans, Michael and Stein, Gunter; "Robustness of Continuous-Time Adaptive Control Algorithms in the Presence of Unmodeled Dynamics", IEEE Transactions on Automatic Control, Vol. AC-30, No.9, pp.881-889, September 1985.

28 - Åström, Karl Johan; "A Commentary on the C.E. Rohrs et al. Paper 'Robustness of Continuous-Time Adaptive Control Algorithms in the Presence of Unmodeled Dynamics' ", IEEE Transactions on Automatic Control, Vol. AC-30, No.9, p.889, September 1985.

29 - Kidd, P.T.; "Comparison of the Performance of a Model-Reference Adaptive System and a Classical Linear Control System Under Non-Ideal Conditions", Int. J. of Control, Vol.42, No.3, pp.671-694, 1985.

30 - Narendra, K.S. and Annaswamy, A.M.; "Robust Adaptive Control", Adaptive and Learning Systems — Proceedings of the Fourth Workshop in Adaptive Systems Control Theory held at Yale University in May 1985, Edited by K.S. Narendra, pp.3-31, Plenum Press, New York, 1986.

31 - Ortega, R. and Yu, T.; "Theoretical Results on Robustness of Direct Adaptive Controllers: A Survey", IFAC 10th Triennial World Congress, pp.1-15, 1987.

32 - Peterson, Benjamin B. and Narendra, Kumpati S.; "Bounded Error Adaptive Control", IEEE Transactions on Automatic Control, Vol. AC-27, No.6, pp.1161-1168, December 1982.

33 - Kreisselmeier, Gerhard and Narendra, Kumpati S.; "Stable Model Reference Adaptive Control in the Presence of Bounded Disturbances", IEEE Transactions on Automatic Control, Vol. AC-27, No.6, pp.1169-1175, December 1982.

34 - Ioannou, Petros A. and Kokotovic, Petar V.; "Robust Redesign of Adaptive Control", IEEE Transactions on Automatic Control, Vol. AC-29, No.3, pp.202-211, March 1984.

35 - Narendra, Kumpati S. and Annaswamy, Anuradha M.; "A New Adaptive Law for Robust Adaptation Without Persistent Excitation", IEEE Transactions on Automatic Control, Vol. AC-32, No.2, pp.134-145, February 1987.

36 - Ortega, R.; Prayly, L. and Landau, I.D.; "Robustness of Discrete-Time Direct Adaptive Controllers", IEEE Transactions on Automatic Control, Vol. AC-30, No.12, pp.1179-1187, December 1985.

37 - Prayly, L.; "Global Stability of a Direct Adaptive Control Scheme With Respect to a Graph Topology", Adaptive and Learning Systems — Proceedings of the Fourth Workshop in Adaptive Systems Control Theory held at Yale University in May 1985, Edited by K.S. Narendra, pp.57-72, Plenum Press, New York, 1986.

38 - Ioannou, Petros and Tsakalis, Kostas; "Robust Discrete-Time Adaptive Control", Adaptive and Learning Systems — Proceedings of the Fourth Workshop in Adaptive Systems Control Theory held at Yale University in May 1985, Edited by K.S. Narendra, pp.73-85, Plenum Press, New York, 1986.

39 - Kreisselmeier, Gerhard and Anderson, Brian D.O.; "Robust Model Reference Adaptive Control", IEEE Transactions on Automatic Control, Vol. AC-31, No.2, pp.127-133, February 1986.

40 - Utkin, V.I.; "Sliding Modes and Their Application in Variable Structure Systems", MIR, 1978.

41 - Balestrino, A.; De Maria, G. and Sciavicco, L.; "An Adaptive Model Following Control for Robotic Manipulators", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.105, pp.143-151, September 1983.

42 - Utkin, Vadim I.; "Variable Structure Systems with Sliding Modes", IEEE Transactions on Automatic Control, Vol. AC-22, No.2, pp.212-222, April 1977.

43 - Slotine, Jean-Jacques E.; "Sliding Controller Design for Non-Linear Systems", Int. J. Control, Vol.40, No.2, pp.421-434, 1984.

44 - Ambrosino, G.; Celentano, G. and Gaofalo, F.; "Variable Structure Model Reference Adaptive Control Systems", Int. J. Control, Vol.39, No.6, pp.1339-1349, 1984.

45 - Rivin, Eugene I.; "Mechanical Design of Robots", McGraw-Hill, U.S.A., 1988.

46 - de Pennington, A.; Mannetje, J.J.'t and Bell, R.; "The Modelling of Electrohydraulic Control Valves and Its Influence on the Design of Electrohydraulic Drives", IMechE Journal of Mechanical Engineering Science, Vol.16, No.3, pp.196-204, 1974.

47 - Martin, D.J. and Burrows, C.R.; "The Dynamic Characteristics of an Electro-Hydraulic Servovalve", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, December 1976, pp.395-406.

48 - Tomizuka, Masayoshi; "Zero Phase Error Tracking Algorithm for Digital Control", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.109, pp.65-68, March 1987.

49 - Haack, B. and Tomizuka, M.; "The Effect of Adding Zeroes to Feedforward Controllers", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.113, pp.6-10, March 1991.

50 - Coiffet, Philippe; "La Robotique - principes et applications", Hermes, Paris 1986.

51 - Butler, J.; Haack, B. and Tomizuka, M.; "Reference Input Generation for High Speed Coordinated Motion of a Two Axis System", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.113, pp.67-74, March 1991.

52 - Middleton, Richard H. and Goodwin, Graham C.; "Digital Control and Estimation — A Unified Approach", Prentice-Hall, 1990.

53 - Agarwal, Ramesh C. and Burrus, C. Sidney; "New Recursive Digital Filter Structures Having Very Low Sensitivity and Roundoff Noise", IEEE Transactions on Circuits and Systems, Vol. CAS-22, No.12, pp.921-927, December 1975.

54 - Orlandi, G. and Martinelli, G.; "Low-Sensitivity Recursive Digital Filters Obtained Via the Delay Replacement", IEEE Transactions on Circuits and Systems, Vol. CAS-31, No.7, pp.654-657, July 1984.

55 - Middleton, Richard H. and Goodwin, Graham C.; "Improved Finite Word Length Characteristics in Digital Control Using Delta Operators", IEEE Transactions on Automatic Control, Vol. AC-31, No.11, pp.1015-1021, November 1986.

56 - Forsythe, W.; "Quantisation Errors in Digital Filters and Their Dependence on Sampling Frequency", Trans. Inst. M.C., Vol.11, No.1, pp.48-56, January-March 1989.

57 - Goodall, R.M.; "Minimisation of Computation for Digital Controllers", Trans. Inst. M.C., Vol.11, No.5, pp.218-224, October-December 1989.

58 - Goodall, R.M.; "The Delay Operator z^{-1} — Inappropriate For Use in Recursive Digital Filters ?", Trans. Inst. M.C., Vol.12, No.5, pp.246-250, 1990.

Appendix A | Continuous Time Models of a Servo-Valve Driven Hydraulic Rotary Actuator

Linear and non-linear models of a servo-valve driven hydraulic rotary actuator are developed in this appendix. The models are produced following a two-step approach: initially a non-linear model is developed; the non-linear model is afterwards linearized, and linear models in state-space and Laplace operator are derived.

A.1 Notation

A_1, A_2, A_3, A_4 - throat areas

C_d - discharge coefficient

C_l - leakage coefficient

c_r - radial clearance

D_{act} - actuator displacement

J - moment of inertia

K_c, K_{c0} - flow-pressure coefficient

K_q, K_{q0} - flow gain

P_A, P_B - pressures in sides A and B of the actuator

P_S - supply pressure

P_T - return (tank) pressure

Q - flow

Q_A, Q_B - flows between valve and actuator

Q_L - load flow

Q_1, Q_2, Q_3, Q_4 - flows through the valve openings

R_1, R_2, R_3, R_4 - hydraulic resistors

t - time

u - underlap

V_A, V_B - fluid volumes of sides A and B of the actuator

V_t - total volume of fluid

w - valve openings area gain

x_v - valve spool position

x_1, x_2, x_3, x_4 - throat openings

β_e - effective bulk modulus of the fluid

θ - angular position of the hydraulic rotary actuator

ρ - fluid density

τ - actuator torque

τ_L - disturbances (load) torque

ω_n - natural frequency

ζ - damping ratio

A.2 Non-Linear Model of a Servo-Valve Driven Hydraulic Rotary Actuator

A diagram of the servo-valve spool/sleeve arrangement is presented in fig.A.1. The flow through the valve is defined by the areas of the four flow passages A_1, A_2, A_3 and A_4. These areas depend on the spool position, x_v, and in a valve with rectangular or circumferential ports are proportional to the lengths x_1, x_2, x_3 and x_4. The four "orifices" will be supposed matched and symmetrical, that is:

$$\text{matched} \Rightarrow \left| \begin{array}{l} A_1(x_v) = A_4(x_v) \\ A_2(x_v) = A_3(x_v) \end{array} \right. \tag{A.1}$$

$$\text{symmetrical} \Rightarrow \left| \begin{array}{l} A_1(x_v) = A_2(-x_v) \\ A_4(x_v) = A_3(-x_v) \end{array} \right. \tag{A.2}$$

This means that the orifice areas may be computed as:

$$A_i = x_i w \tag{A.3}$$

where w is the area gain of the orifice (in a servo-valve with fully circumferential ports w is equal to the average spool and sleeve circumferences), and:

$$x_1 = x_4 \quad , \quad x_2 = x_3 \tag{A.4}$$

From the geometry of the valve it can be said that:

$$x_1(x_v) = \sqrt{c_r^2 + \frac{((u + x_v) + |u + x_v|)^2}{4}} \tag{A.5}$$

$$x_2(x_v) = \sqrt{c_r^2 + \frac{((u - x_v) + |u - x_v|)^2}{4}} \tag{A.6}$$

where c_r is the radial clearance between the valve spool and sleeve and u is the length of the spool underlap.

Assuming fully turbulent flow, the flow trough an orifice, Q, may be computed as [A.1]:

$$Q = A C_d \sqrt{\frac{2\Delta P}{\rho}} \tag{A.7}$$

where C_d is the discharge coefficient of the orifice (typically 0.67 for a servo-valve orifice [A.2]), ΔP is the pressure differential across the orifice and ρ is the fluid density.

So, the four flows Q_1, Q_2, Q_3 and Q_4 will be given by:

$$Q_1 = x_1 w C_d \frac{(P_S - P_A)}{|P_S - P_A|} \sqrt{\frac{2|P_S - P_A|}{\rho}} \tag{A.8}$$

$$Q_2 = x_2 w C_d \frac{(P_A - P_T)}{|P_A - P_T|} \sqrt{\frac{2|P_A - P_T|}{\rho}} \tag{A.9}$$

$$Q_3 = x_3 w C_d \frac{(P_S - P_B)}{|P_S - P_B|} \sqrt{\frac{2|P_S - P_B|}{\rho}} \tag{A.10}$$

$$Q_4 = x_4 w C_d \frac{(P_B - P_T)}{|P_B - P_T|} \sqrt{\frac{2|P_B - P_T|}{\rho}} \tag{A.11}$$

where P_S and P_T are the supply and return pressures and P_A and P_B are the ports A and B pressures.

The flows at the servo-valve ports, Q_A and Q_B will be:

$$Q_A = Q_1 - Q_2 \quad , \quad Q_B = Q_4 - Q_3 \tag{A.12}$$

A diagram of a hydraulic rotary actuator is presented in fig.A.2. The actuator output torque is given by:

$$\tau = D_{act}(P_A - P_B) \tag{A.13}$$

where D_{act} is the rotary actuator volumetric displacement.

Pressures P_A and P_B may be computed from the following differential equations, where the presence of an internal flow leakage is taken into account:

$$\frac{dP_A}{dt}\frac{V_A}{\beta_e} + \frac{dV_A}{dt} = Q_A - C_l(P_A - P_B) \tag{A.14}$$

$$\frac{dP_B}{dt}\frac{V_B}{\beta_e} + \frac{dV_B}{dt} = C_l(P_A - P_B) - Q_B \tag{A.15}$$

The internal flow leakage between the two actuator chambers is assumed to be laminar, with a leakage coefficient C_l ; β_e is the effective bulk modulus of the fluid in the circuit and V_A and V_B are the total volumes of fluid in sides A and B of the actuator, including hose volumes, etc.

From the geometry of the actuator, it can be said that:

$$\frac{dV_A}{dt} = -\frac{dV_B}{dt} = D_{act}\frac{d\theta}{dt} \tag{A.16}$$

If the actuator is moving an essentially inertial load, its output torque is given by:

$$\tau = D_{act}(P_A - P_B) = J\frac{d^2\theta}{dt^2} + \tau_L \qquad \text{(A.17)}$$

where J is the moment of inertia seen by the actuator and τ_L is an external torque disturbance.

Equations A.8 to A.17 define a non-linear model of a servo-valve driven hydraulic rotary actuator, relating its output position, θ, with the valve spool displacement, x_v. In the manipulator case, the manipulator dynamic model equations (equ.1.4 or 1.5) must be used in the place of equ.A.17.

A.3 Linearized Servo-Valve Model

It is convenient to reduce the number of pressure and flow variables, in order to linearize the servo-valve model.

In steady state (or if V_A is equal to V_B) and assuming no external leakages in the actuator,

$$Q_A = Q_B = Q_L \tag{A.18}$$

where Q_L is the load flow.

The pressure droop in the actuator, P_L (load pressure), is given by:

$$P_L = P_A - P_B \tag{A.19}$$

The tank pressure, P_T, will, without loss of generality, be made equal to zero (or P_T may be considered as a datum for all the other pressures).

Fig.A.3 shows a bridge representation of the valve plus actuator system. The valve restrictions may be seen as hydraulic resistors, leading to the following identities:

$$\begin{aligned} \Delta P_1 &= R_1 Q_1^2 \\ \Delta P_2 &= R_2 Q_2^2 \\ \Delta P_3 &= R_3 Q_3^2 \\ \Delta P_4 &= R_4 Q_4^2 \end{aligned} \tag{A.20}$$

$$P_S = \Delta P_1 + \Delta P_2 = \Delta P_3 + \Delta P_4 \Rightarrow R_1 Q_1^2 + R_2 Q_2^2 = R_3 Q_3^2 + R_4 Q_4^2 \tag{A.21}$$

As the valve orifices are matched,

$$R_1 = R_4 \quad , \quad R_2 = R_3 \tag{A.22}$$

leading to:

$$R_1(Q_1^2 - Q_4^2) = R_2(Q_3^2 - Q_2^2) \tag{A.23}$$

From the definition of Q_L, Q_A and Q_B, it can be said that:

$$Q_L = Q_A = Q_B \Rightarrow Q_1 - Q_2 = Q_4 - Q_3 \tag{A.24}$$

leading to:

$$Q_1 - Q_4 = Q_2 - Q_3 \tag{A.25}$$

The simultaneous solution of equ.A.23 and A.25 implies that:

$$Q_1 = Q_4 \quad , \quad Q_2 = Q_3 \tag{A.26}$$

leading to:

$$\Delta P_1 = \Delta P_4 \quad , \quad \Delta P_2 = \Delta P_3 \tag{A.27}$$

From fig.A.3 and using equ.A.27, it can be said that:

$$P_L = \Delta P_2 - \Delta P_4 \tag{A.28}$$

$$P_A = \Delta P_2 = (P_S - \Delta P_1) \Rightarrow 2P_A = P_S + (\Delta P_2 - \Delta P_4) \Rightarrow P_A = \frac{P_S + P_L}{2} \tag{A.29}$$

$$P_B = \Delta P_4 = (P_S - \Delta P_3) \Rightarrow 2P_B = P_S + (\Delta P_4 - \Delta P_2) \Rightarrow P_B = \frac{P_S - P_L}{2} \tag{A.30}$$

$$P_S = P_A + P_B \tag{A.31}$$

From these equations it may be concluded that when the actuator is at rest, and with no external toques applied,

$$P_L = 0 \Rightarrow P_A = P_B = \frac{P_S}{2} \tag{A.32}$$

Assuming that P_A and P_B are always less than P_S, an equation for the load flow, Q_L, may now be written:

$$Q_L = x_1 w C_d \sqrt{\frac{P_S - P_L}{\rho}} - x_2 w C_d \sqrt{\frac{P_S + P_L}{\rho}} \tag{A.33}$$

If $-u \le x_v \le u$:

$$Q_L = \sqrt{c_r^2 + (u + x_v)^2}\; w C_d \sqrt{\frac{P_S - P_L}{\rho}} - \sqrt{c_r^2 + (u - x_v)^2}\; w C_d \sqrt{\frac{P_S + P_L}{\rho}} \tag{A.34}$$

If $x_v > u$:

$$Q_L = \sqrt{c_r^2 + (u + x_v)^2}\; w C_d \sqrt{\frac{P_S - P_L}{\rho}} - c_r w C_d \sqrt{\frac{P_S + P_L}{\rho}} \tag{A.35}$$

If $x_v < -u$:

$$Q_L = c_r w C_d \sqrt{\frac{P_S - P_L}{\rho}} - \sqrt{c_r^2 + (u - x_v)^2}\; w C_d \sqrt{\frac{P_S + P_L}{\rho}} \tag{A.36}$$

The load flow, Q_L, may be linearized around an operating point defined by $Q_L = Q_L(x_v, P_L)$. So, for small changes in x_v and P_L:

$$Q_L = Q_L(x_v, P_L) + dQ_L \tag{A.37}$$

$$dQ_L = \left.\frac{\partial Q_L}{\partial x_v}\right|_{(x_v, P_L)} dx_v + \left.\frac{\partial Q_L}{\partial P_L}\right|_{(x_v, P_L)} dP_L \tag{A.38}$$

These partial derivatives define two important servo-valve parameters:

- the flow gain, $K_q \equiv \frac{\partial Q_L}{\partial x_v}$ (A.39)

- the flow-pressure coefficient, $K_c \equiv -\frac{\partial Q_L}{\partial P_L}$ (A.40)

leading to the linearized servo-valve model:

$$\Delta Q_L = K_q \Delta x_v - K_c \Delta P_L \tag{A.41}$$

If $-u \le x_v \le u$:

$$K_q = \frac{x_v + u}{\sqrt{c_r^2 + (u + x_v)^2}} w C_d \sqrt{\frac{P_S - P_L}{\rho}} - \frac{x_v - u}{\sqrt{c_r^2 + (u - x_v)^2}} w C_d \sqrt{\frac{P_S + P_L}{\rho}} \tag{A.42}$$

$$K_c = \sqrt{c_r^2 + (u + x_v)^2}\, \frac{w C_d \sqrt{(P_S - P_L)/\rho}}{2(P_S - P_L)} + \sqrt{c_r^2 + (u - x_v)^2}\, \frac{w C_d \sqrt{(P_S + P_L)/\rho}}{2(P_S + P_L)} \tag{A.43}$$

If $x_v > u$:

$$K_q = \frac{x_v + u}{\sqrt{c_r^2 + (u + x_v)^2}} w C_d \sqrt{\frac{P_S - P_L}{\rho}} \tag{A.44}$$

$$K_c = \sqrt{c_r^2 + (u + x_v)^2}\ \frac{w C_d \sqrt{(P_S - P_L)/\rho}}{2(P_S - P_L)} + c_r \frac{w C_d \sqrt{(P_S + P_L)/\rho}}{2(P_S + P_L)} \tag{A.45}$$

If $x_v < -u$:

$$K_q = \frac{u - x_v}{\sqrt{c_r^2 + (u - x_v)^2}} w C_d \sqrt{\frac{P_S + P_L}{\rho}} \tag{A.46}$$

$$K_c = c_r \frac{w C_d \sqrt{(P_S - P_L)/\rho}}{2(P_S - P_L)} + \sqrt{c_r^2 + (u - x_v)^2}\ \frac{w C_d \sqrt{(P_S + P_L)/\rho}}{2(P_S + P_L)} \tag{A.47}$$

The null coefficients (obtained with $Q_L = P_L = x_v = 0$), that result from a linearization around the rest condition with no external torques applied to the actuator, are:

$$K_{q0} = \frac{2u}{\sqrt{c_r^2 + u^2}} w C_d \sqrt{\frac{P_S}{\rho}} \tag{A.48}$$

$$K_{c0} = \sqrt{c_r^2 + u^2}\ \frac{w C_d \sqrt{P_S/\rho}}{P_S} \tag{A.49}$$

and the pressure gain, $\partial P_L / \partial x_v$, around null is:

$$\left.\frac{\partial P_L}{\partial x_v}\right|_{x_v = 0} = \frac{K_{q0}}{K_{c0}} = \frac{2u}{(c_r^2 + u^2)} P_S \tag{A.50}$$

Due to the existence of underlap and radial clearance, the flow gain is not constant with x_v. The maximum deviation from linearity occurs at $x_v = 0$. If, in order to obtain a maximum pressure gain [A.3], the radial clearance, c_r, is made equal to the underlap, u:

$$K_{q0} = \sqrt{2}\, w C_d \sqrt{\frac{P_S}{\rho}} \tag{A.51}$$

and at $x_v = u$, K_q is minimum and equal to:

$$K_{qu} = \frac{2}{\sqrt{5}} w C_d \sqrt{\frac{P_S}{\rho}} \tag{A.52}$$

Under the same conditions the pressure gain around null becomes:

$$\left.\frac{\partial P_L}{\partial x_v}\right|_{x_v = 0} = \frac{P_S}{u} \tag{A.53}$$

A.4 Linearized Model of a Servo-Valve Driven Hydraulic Rotary Actuator

The linearized servo-valve model (equ.A.41) will now be linked to the rotary actuator model. In the first place it must be noticed that:

$$P_A = \frac{1}{2}(P_S + P_L) \Rightarrow dP_A = \frac{1}{2}dP_L \tag{A.54}$$

$$P_B = \frac{1}{2}(P_S - P_L) \Rightarrow dP_B = -\frac{1}{2}dP_L \tag{A.55}$$

provided P_S is a constant.

So:

$$Q_A = K_q x_v - 2K_c P_A \tag{A.56}$$

$$Q_B = K_q x_v + 2K_c P_B \tag{A.57}$$

leading to:

$$Q_L = K_q x_v - K_c P_L \quad , \quad \text{with } Q_L = \frac{Q_A + Q_B}{2} \tag{A.58}$$

that is, the definition of Q_L is slightly changed from the input or output flow, to the average input-output flow.

Adding equ.A.14 and A.15, and taking into account equ.A.16, A.54 and A.55, results into:

$$Q_L = D_{act}\frac{d\theta}{dt} + \frac{V_t}{4\beta_e}\frac{dP_L}{dt} + C_t P_L \tag{A.59}$$

where $V_t = V_A + V_B$ (total volume).

If Q_L is substituted by the lincarized servo-valve model, the following equation results:

$$K_q x_v = D_{act}\frac{d\theta}{dt} + \frac{V_t}{4\beta_e}\frac{dP_L}{dt} + (K_c + C_l)P_L \qquad \text{(A.60)}$$

From equ.A.17 it can be said that:

$$P_L = \frac{J}{D_{act}}\frac{d^2\theta}{dt^2} + \frac{1}{D_{act}}\tau_L \qquad \text{(A.61)}$$

Substituting equ.A.61 into equ.A.60 results in the following differential equation:

$$K_q x_v = D_{act}\frac{d\theta}{dt} + \frac{(K_c + C_l)}{D_{act}}\left(J\frac{d^2\theta}{dt^2} + \tau_L\right) + \frac{V_t}{4\beta_e D_{act}}\left(J\frac{d^3\theta}{dt^3} + \frac{d\tau_L}{dt}\right) \qquad \text{(A.62)}$$

This differential equation leads to a third order transfer function in the Laplace operator from the servo-valve spool position and the torque disturbance to the output rotary actuator angular position:

$$\begin{aligned}\mathcal{L}(\theta) = {} & \frac{K_q/D_{act}}{s\left(1 + \frac{(K_c + C_l)J}{D_{act}^2}s + \frac{V_t J}{D_{act}^2 4\beta_e}s^2\right)}\mathcal{L}(x_v) \\ & + \frac{\frac{(K_c + C_l)}{D_{act}^2}\left(1 + \frac{V_t}{4\beta_e(K_c + C_l)}s\right)}{s\left(1 + \frac{(K_c + C_l)J}{D_{act}^2}s + \frac{V_t J}{D_{act}^2 4\beta_e}s^2\right)}\mathcal{L}(\tau_L)\end{aligned} \qquad \text{(A.63)}$$

From this transfer function, the natural frequency, ω_n, and the damping ratio, ζ, of the system may be computed as:

$$\omega_n = \sqrt{\frac{D_{act}^2 4\beta_e}{V_t J}} \tag{A.64}$$

$$\zeta = \frac{(K_c + C_l)}{D_{act}} \sqrt{\frac{J\beta_e}{V_t}} \tag{A.65}$$

A block diagram of this linearized model is presented in fig.A.4. From this diagram state space models may be easily extracted.

If $x = (\theta, \dot{\theta}, P_L)$:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{P}_L \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{D_{act}}{J} \\ 0 & -\frac{D_{act} 4\beta_e}{V_t} & -\frac{4\beta_e (K_c + C_l)}{V_t} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ P_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_q 4\beta_e}{V_t} \end{bmatrix} x_v + \begin{bmatrix} 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix} \tau_L \tag{A.66}$$

If $x = (\theta, \dot{\theta}, \ddot{\theta})$:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{D_{act}^2 4\beta_e}{V_t J} & -\frac{(K_c + C_l) 4\beta_e}{V_t} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_q D_{act} 4\beta_e}{V_t J} \end{bmatrix} x_v + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{(K_c + C_l) 4\beta_e}{V_t J} & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} \tau_L \\ \tau_L \end{bmatrix} \tag{A.67}$$

Due to the linearization process, the ω_n equation (equ.A.64) is valid only when the actuator is in its middle position. A better expression for the computation of the natural frequency of the actuator will now be presented.

If the actuator supply lines are supposed to be blocked, and an external torque is applied:

$$-\tau = D_{act}(P_A - P_B) \Rightarrow -d\tau = D_{act}(dP_A - dP_B) \tag{A.68}$$

Taking into account that:

$$dP_A = -\beta_e D_{act}\frac{d\theta}{V_A} \quad , \quad dP_B = \beta_e D_{act}\frac{d\theta}{V_B} \tag{A.69}$$

leads to:

$$d\tau = \beta_e D_{act}^2 \left(\frac{1}{V_A} + \frac{1}{V_B}\right) d\theta \tag{A.70}$$

The natural frequency of the actuator will now be:

$$\omega_n = \sqrt{\frac{d\tau/d\theta}{J}} = \sqrt{\frac{\beta_e D_{act}^2}{J}\left(\frac{1}{V_A} + \frac{1}{V_B}\right)} \tag{A.71}$$

Using this expression, the damping ratio becomes:

$$\zeta = \frac{(K_c + C_l)}{2D_{act}}\sqrt{J\beta_e\left(\frac{1}{V_A} + \frac{1}{V_B}\right)} \tag{A.72}$$

These results may be used to improve the system model:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{\beta_e D_{act}^2}{J}\left(\frac{1}{V_A}+\frac{1}{V_B}\right) & -(K_c+C_l)\beta_e\left(\frac{1}{V_A}+\frac{1}{V_B}\right) \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_q D_{act}\beta_e}{J}\left(\frac{1}{V_A}+\frac{1}{V_B}\right) \end{bmatrix} x_v + \begin{bmatrix} 0 & 0 \\ -\frac{(K_c+C_l)\beta_e}{J}\left(\frac{1}{V_A}+\frac{1}{V_B}\right) & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} \tau_L \\ \dot{\tau}_L \end{bmatrix} \tag{A.73}$$

with $V_A = V_A(\theta)$, $V_B = V_B(\theta)$, $K_q = K_q(x_v , P_L)$, $K_c = K_c(x_v , P_L)$.

A.5 References

A.1 - Merritt, Herbert E.; "Hydraulic Control Systems", John Wiley & Sons, 1967.

A.2 - McCloy, D. and Martin, H.R.; "Control of Fluid Power - Analysis and Design, 2nd (Revised) Edition", Ellis Horwood Limited (Halsted Press, John Wiley & Sons), 1980.

A.3 - LeQuoc, S.; "Design of Hydraulic Servovalve for Maximum Pressure Sensitivity", Transactions of the ASME, Journal of Dyn. Sys. Meas. and Control, Vol.106, pp.116-119, March 1984.

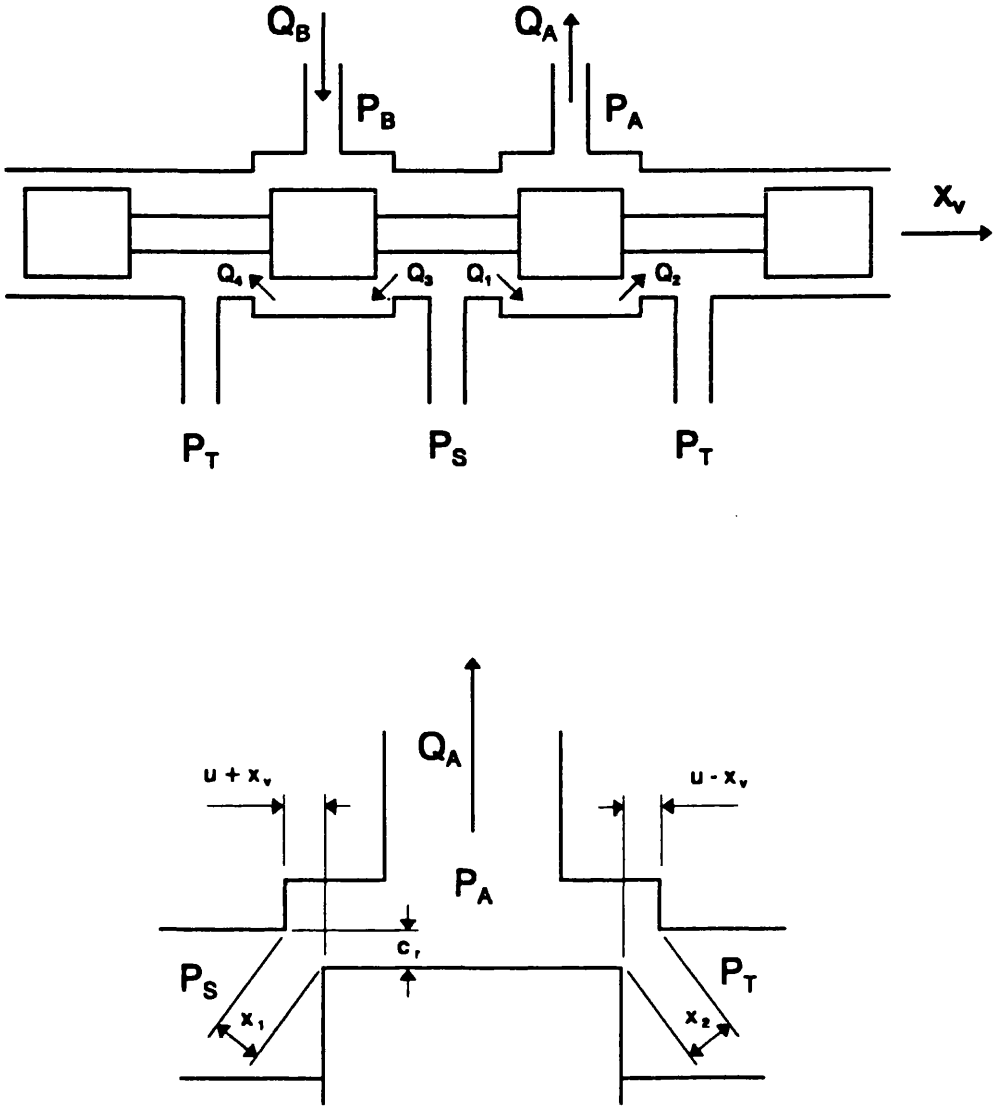


Figure A.1 - Diagram of the servo-valve spool/sleeve arrangement

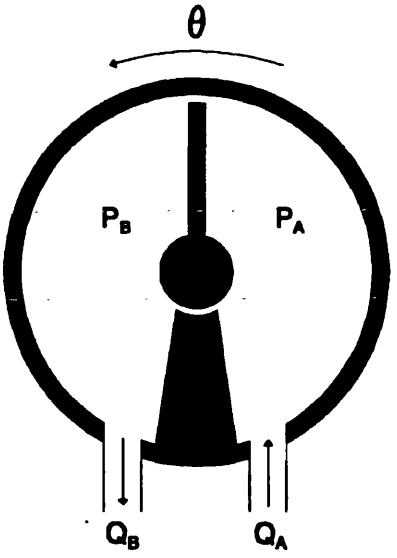


Figure A.2 - Diagram of the hydraulic actuator

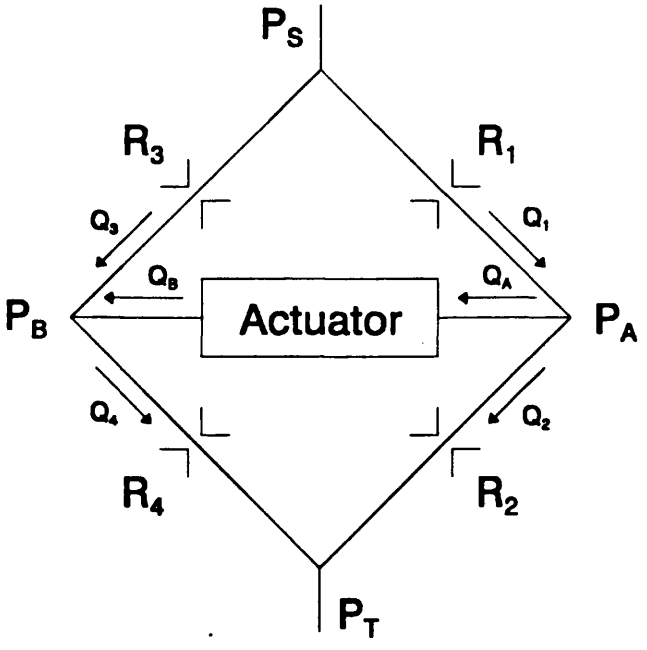


Figure A.3 - Bridge representation of the linearized model of the system

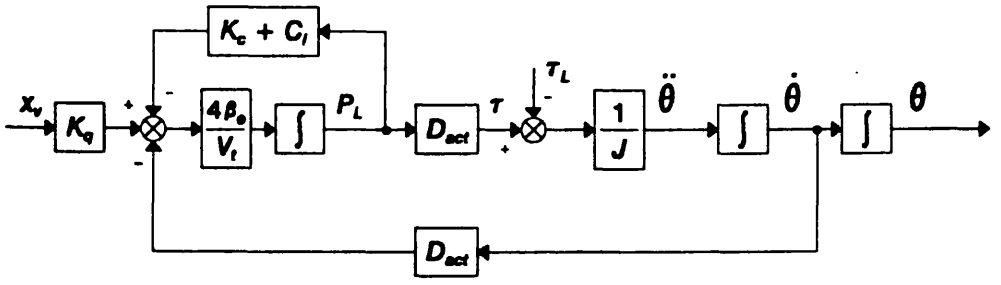


Figure A.4 - Block diagram of the linearized model of the system

Appendix B | Continuous Time Solution of the Popov Inequality

As shown in Chapter 4, the Popov integral inequality (equ.3.7) may be broken down into three integral inequalities which are repeated here for convenience:

$$\eta_1(0,t_1) \triangleq \int_0^{t_1} v^T(K_x - B_p^{\dagger}(A_m - A_p))x_p\,dt \geq -\gamma_1^2 \tag{3.8}$$

$$\eta_2(0,t_1) \triangleq \int_0^{t_1} v^T(K_u - B_p^{\dagger}B_m)u_m\,dt \geq -\gamma_2^2 \tag{3.9}$$

$$\eta_3(0,t_1) \triangleq \int_0^{t_1} v^T(\varphi + h)\,dt \geq -\gamma_3^2 \tag{3.10}$$

In the first inequality (equ.3.8), using integral plus proportional gain adaptation, the following identity can be made:

$$K_x(t) = K_{x0} + \int_0^t \Phi_{1I}(v,t,\tau)d\tau + \Phi_{1P}(v,t) \tag{B.1}$$

obtaining

$$\eta_1(0,t_1) = \int_0^{t_1} v^T\Big(\int_0^t \Phi_{1I}(v,t,\tau)d\tau + \Phi_{1P}(v,t) + K_{x0} - B_p^{\dagger}(A_m - A_p)\Big)x_p\,dt \geq -\gamma_1^2 \tag{B.2}$$

This inequality may be decomposed into:

$$\eta_{1I}(0,t_1) = \int_0^{t_1} v^T \left(\int_0^{t} \Phi_{1I}(v,t,\tau)\, d\tau + K_{x0} - B_p^{\dagger}(A_m - A_p) \right) x_p\, dt \geq -\gamma_{1I}^2 \tag{B.3}$$

and

$$\eta_{1P}(0,t_1) = \int_0^{t_1} v^T \Phi_{1P}(v,t) x_p\, dt \geq -\gamma_{1P}^2 \tag{B.4}$$

The proportional adaptation term results into:

$$\eta_{1P}(0,t_1) = \sum_{i=1}^{m} \sum_{j=1}^{n} \int_0^{t_1} v_i x_{pj} \phi_{1Pij}(v,t)\, dt \geq -\gamma_{1P}^2 \tag{B.5}$$

This inequality holds if each of the i, j inequalities hold. So,

$$\eta_{1Pij}(0,t_1) = \int_0^{t_1} v_i x_{pj} \phi_{1Pij}(v,t)\, dt \geq -\gamma_{1Pij}^2 \tag{B.6}$$

that holds if:

$$\phi_{1Pij}(v,t) = \alpha_{xPij}(t)\, v_i x_{pj}\ ,\ \alpha_{xPij}(t) \geq 0\ ,\ \forall t \geq 0 \tag{B.7}$$

The integral adaptation term (equ.B.3) results into:

$$\eta_{1I}(0,t_1) = \sum_{i=1}^{m} \sum_{j=1}^{n} \int_0^{t_1} v_i x_{pj} \left(\int_0^{t} \phi_{1Iij}(v,t,\tau)\, d\tau + \beta_{xij} - a_{ij} \right) dt \geq -\gamma_{1I}^2 \tag{B.8}$$

with $B_p^{\dagger}\,(A_m - A_p) = [a_{ij}]$ and $K_{x0} = [\beta_{xij}]$.

This inequality holds if each of the i, j inequalities hold. So,

$$\eta_{1Iij}(0,t_1) = \int_0^{t_1} v_i x_{pj} \left(\int_0^{t} \phi_{1Iij}(v,t,\tau)\, d\tau + \beta_{xij} - a_{ij} \right) dt \geq -\gamma_{1Iij}^2 \tag{B.9}$$

Using a known expression from calculus:

$$\int_0^{t_1} \dot{f}(t)\alpha_{xIij} f(t)\, dt = \frac{1}{2}\alpha_{xIij}\left(f^2(t_1) - f^2(0)\right) \geq -\frac{1}{2}\alpha_{xIij} f^2(0) \quad , \quad \alpha_{xIij} > 0 \tag{B.10}$$

it can be said that:

$$\dot{f}(t) = v_i x_{pj} \;,\; \alpha_{xIij} f(t) = \int_0^{t} \phi_{1Iij}(v,t,\tau)\, d\tau + \beta_{xij} - a_{ij} \tag{B.11}$$

$$\alpha_{xIij}\dot{f}(t) = \phi_{1Iij}(v,t,\tau) \cdot \Rightarrow$$

$$\phi_{1Iij}(v,t,\tau) = \alpha_{xIij} v_i x_{pj} \;,\; \alpha_{xIij} > 0 \tag{B.12}$$

The second integral inequality (equ.3.9), and also using an integral plus proportional gain adaptation law:

$$K_u(t) = K_{u0} + \int_0^{t} \Phi_{2I}(v,t,\tau)\, d\tau + \Phi_{2P}(v,t) \tag{B.13}$$

results into:

$$\eta_{2Pij}(0,t_1) = \int_0^{t_1} v_i u_{mj} \phi_{2Pij}(v,t)\, dt \geq -\gamma_{2Pij}^2 \tag{B.14}$$

and

$$\eta_{2Iij}(0,t_1) = \int_0^{t_1} v_i u_{mj} \left(\int_0^{t} \phi_{2Iij}(v,t,\tau)\, d\tau + \beta_{uij} - b_{ij} \right) dt \geq -\gamma_{2Iij}^2 \tag{B.15}$$

with $B_p^{\dagger} B_m = [b_{ij}]$ and $K_{u0} = [\beta_{uij}]$.

These inequalities are true if:

$$\phi_{2Iij}(v,t,\tau) = \alpha_{uIij} v_i u_{mj} \ , \ \alpha_{uIij} > 0 \tag{B.16}$$

$$\phi_{2Pij}(v,t) = \alpha_{uPij}(t) v_i u_{mj} \ , \ \alpha_{uPij}(t) \geq 0 \ , \ \forall t \geq 0 \tag{B.17}$$

Using a similar approach the following results are obtained from the third integral inequality (equ.3.10):

$$h(t) = h_0 + \int_0^t \Phi_{3I}(v,t,\tau)\, d\tau + \Phi_{3P}(v,t) \tag{B.18}$$

$$\eta_{3Pi}(0,t_1) = \int_0^{t_1} v_i \phi_{3Pi}(v,t)\, dt \geq -\gamma_{3Pi}^2 \tag{B.19}$$

$$\phi_{3Pi}(v,t) = \alpha_{hPi}(t) v_i \ , \ \alpha_{hPi}(t) \geq 0 \ , \ \forall t \geq 0 \tag{B.20}$$

$$\eta_{3Ii}(0,t_1) = \int_0^{t_1} v_i \left(\int_0^t \phi_{3Ii}(v,t,\tau)\, d\tau + \beta_{hi} + \varphi_i \right) dt \geq -\gamma_{3Ii}^2 \tag{B.21}$$

$$\phi_{3Ii}(v,t,\tau) = \alpha_{hIi} v_i \ , \ \alpha_{hIi} > 0 \tag{B.22}$$

with $h_0 = [\beta_{hi}]$.

Appendix C | Discrete Time Solution of the Popov Inequality

The discrete time Popov summation (equ.4.20) may be decomposed into three inequalities that are analogous to equations 3.8, 3.9 and 3.10:

$$\eta_1(0,k_1) \triangleq \sum_{k=0}^{k_1} v(k)^T(K_x(k) - B_p^+(A_m - A_p))x_p(k) \geq -\gamma_1^2 \tag{C.1}$$

$$\eta_2(0,k_1) \triangleq \sum_{k=0}^{k_1} v(k)^T(K_u(k) - B_p^+B_m)u_m(k) \geq -\gamma_2^2 \tag{C.2}$$

$$\eta_3(0,k_1) \triangleq \sum_{k=0}^{k_1} v(k)^T(\varphi + h(k)) \geq -\gamma_3^2 \tag{C.3}$$

If the calculation delays are sufficiently small to be discarded or an equal delay is added to the model, the model and plant states are synchronized. This enables the computation of the Popov summations up to the present sampling interval, k_1, as in the ideal case of zero computing time.

In the first inequality (equ.C.1), using integral plus proportional gain adaptation, the following identity can be made:

$$K_x(k) = K_{x0} + \sum_{l=0}^{k} \Phi_{1I}(v,k,l) + \Phi_{1P}(v,k) \tag{C.4}$$

obtaining

$$\eta_1(0,k_1) = \sum_{k=0}^{k_1} v(k)^T (\sum_{l=0}^{k} \Phi_{1l}(v,k,l) + \Phi_{1P}(v,k) + K_{x0} - B_p^{\dagger}(A_m - A_p)) x_p(k) \geq -\gamma_1^2 \quad \text{(C.5)}$$

This inequality may be decomposed into:

$$\eta_{1l}(0,k_1) = \sum_{k=0}^{k_1} v(k)^T (\sum_{l=0}^{k} \Phi_{1l}(v,k,l) + K_{x0} - B_p^{\dagger}(A_m - A_p)) x_p(k) \geq -\gamma_{1l}^2 \quad \text{(C.6)}$$

and

$$\eta_{1P}(0,k_1) = \sum_{k=0}^{k_1} v(k)^T \Phi_{1P}(v,k) x_p(k) \geq -\gamma_{1P}^2 \quad \text{(C.7)}$$

The proportional adaptation term results into:

$$\eta_{1P}(0,k_1) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=0}^{k_1} v_i(k) x_{pj}(k) \phi_{1Pij}(v,k) \geq -\gamma_{1P}^2 \quad \text{(C.8)}$$

This inequality holds if each of the i, j inequalities hold. So,

$$\eta_{1Pij}(0,k_1) = \sum_{k=0}^{k_1} v_i(k) x_{pj}(k) \phi_{1Pij}(v,k) \geq -\gamma_{1Pij}^2 \quad \text{(C.9)}$$

that holds if:

$$\phi_{1Pij}(v,k) = \alpha_{xPij}(k)\, v_i(k)\, x_{pj}(k) \ , \ \alpha_{xPij}(k) \geq 0 \ , \ \forall k \geq 0 \quad \text{(C.10)}$$

The integral adaptation term (equ.C.6) results into:

$$\eta_{1l}(0,k_1) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=0}^{k_1} v_i(k) x_{pj}(k) (\sum_{l=0}^{k} \phi_{1lij}(v,k,l) + \beta_{xij} - a_{ij}) \geq -\gamma_{1l}^2 \quad \text{(C.11)}$$

with $B_p^{\dagger}\,(A_m - A_p) = [a_{ij}]$ and $K_{x0} = [\beta_{xij}]$.

This inequality holds if each of the i, j inequalities hold. So,

$$\eta_{1Iij}(0,k_1) = \sum_{k=0}^{k_1} v_i(k)\, x_{pj}(k)\Big(\sum_{l=0}^{k} \phi_{1Iij}(v,k,l) + \beta_{xij} - a_{ij}\Big) \geq -\gamma_{1Iij}^2 \tag{C.12}$$

Using the mathematical results presented in the end of this appendix, the inequality can be made true by making:

$$\phi_{1Iij}(v,k,l) = \alpha_{xIij} v_i(k)\, x_{pj}(k)\ ,\ \alpha_{xIij} > 0 \tag{C.13}$$

The second integral inequality (equ.C.2), and also using an integral plus proportional gain adaptation law:

$$K_u(k) = K_{u0} + \sum_{l=0}^{k} \Phi_{2I}(v,k,l) + \Phi_{2P}(v,k) \tag{C.14}$$

results into:

$$\eta_{2Pij}(0,k_1) = \sum_{k=0}^{k_1} v_i(k)\, u_{mj}(k)\, \phi_{2Pij}(v,k) \geq -\gamma_{2Pij}^2 \tag{C.15}$$

and

$$\eta_{2Iij}(0,k_1) = \sum_{k=0}^{k_1} v_i(k)\, u_{mj}(k)\Big(\sum_{l=0}^{k} \phi_{2Iij}(v,k,l) + \beta_{uij} - b_{ij}\Big) \geq -\gamma_{2Iij}^2 \tag{C.16}$$

with $B_p^{\dagger} B_m = [b_{ij}]$ and $K_{u0} = [\beta_{uij}]$.

These inequalities are true if:

$$\phi_{2Iij}(v,k,l) = \alpha_{uIij} v_i(k) u_{mj}(k)\ ,\ \alpha_{uIij} > 0 \tag{C.17}$$

$$\phi_{2Pij}(v,k) = \alpha_{uPij}(k) v_i(k) u_{mj}(k)\ ,\ \alpha_{uPij}(k) \geq 0\ ,\ \forall k \geq 0 \tag{C.18}$$

Using a similar approach the following results are obtained from the third integral inequality (equ.C.3):

$$h(k) = h_0 + \sum_{l=0}^{k} \Phi_{3l}(v,k,l) + \Phi_{3P}(v,k) \tag{C.19}$$

$$\eta_{3Pi}(0,k_1) = \sum_{k=0}^{k_1} v_i(k)\,\phi_{3Pi}(v,k) \geq -\gamma_{3Pi}^2 \tag{C.20}$$

$$\phi_{3Pi}(v,k) = \alpha_{hPi}(k)\,v_i(k)\ ,\ \alpha_{hPi}(k) \geq 0\ ,\ \forall k \geq 0 \tag{C.21}$$

$$\eta_{3li}(0,k_1) = \sum_{k=0}^{k_1} v_i(k)\Big(\sum_{l=0}^{k} \phi_{3li}(v,k,l) + \beta_{hi} + \varphi_i\Big) \geq -\gamma_{3li}^2 \tag{C.22}$$

$$\phi_{3li}(v,k,l) = \alpha_{hli}\,v_i(k)\ ,\ \alpha_{hli} > 0 \tag{C.23}$$

with $h_0 = [\beta_{hi}]$.

C.1 Discrete Time Solution of the Integral Adaptation Term

In order to establish the final proof, the following preliminary result must be developed:

$$\sum_{k=0}^{k_1}(f(k)\alpha\sum_{l=0}^{k}f(l)) = \frac{\alpha}{2}((\sum_{k=0}^{k_1}f(k))^2 + \sum_{k=0}^{k_1}f^2(k)) \tag{C.24}$$

The first member of this identity may be developed as:

$$\sum_{k=0}^{k_1}(f(k)\alpha\sum_{l=0}^{k}f(l)) = \alpha(f(0)f(0) + f(1)(f(0)+f(1)) + f(2)(f(0)+f(1)+f(2)) + \ldots + f(k_1)(f(0)+f(1)+\ldots+f(k_1))) = \tag{C.25}$$

$$\begin{aligned}\alpha(&f(0)f(0)+f(0)f(1)+f(0)f(2)+\ldots+f(0)f(k_1)+\\ &f(1)f(1)+f(1)f(2)+\ldots+f(1)f(k_1)+\\ &f(2)f(2)+\ldots+f(2)f(k_1)+\\ &\vdots\\ &f(k_1)f(k_1)) =\end{aligned} \tag{C.26}$$

$$\begin{aligned}(\alpha/2)(&(f(0)f(0)+f(0)f(1)+f(0)f(2)+\ldots+f(0)f(k_1)+\\ &f(1)f(0)+f(1)f(1)+f(1)f(2)+\ldots+f(1)f(k_1)+\\ &f(2)f(0)+f(2)f(1)+f(2)f(2)+\ldots+f(2)f(k_1)+\\ &\vdots\\ &f(k_1)f(0)+f(k_1)f(1)+f(k_1)f(2)+\ldots+f(k_1)f(k_1)) +\\ &(f(0)f(0)+\\ &\quad f(1)f(1)+\\ &\quad\quad f(2)f(2)+\\ &\quad\quad\quad \ldots+f(k_1)f(k_1)))) =\end{aligned} \tag{C.27}$$

$$\frac{\alpha}{2}((\sum_{k=0}^{k_1}f(k))^2 + \sum_{k=0}^{k_1}f^2(k)) \tag{C.28}$$

The obtained result shows the truthfulness of the identity C.24.

Now, it is possible to use this result in order to prove that:

$$\sum_{k=0}^{k_1} f(k)\alpha\left(\sum_{l=0}^{k} f(l) + C\right) \geq -\frac{\alpha}{2}C^2 \quad , \quad \alpha > 0 \tag{C.29}$$

The first member of this identity may be developed as:

$$\sum_{k=0}^{k_1} f(k)\alpha\left(\sum_{l=0}^{k} f(l) + C\right) = \sum_{k=0}^{k_1} \left(f(k)\alpha\sum_{l=0}^{k} f(l)\right) + \sum_{k=0}^{k_1} f(k)\alpha C \tag{C.30}$$

Using the previously developed result (equ.C.24), it can be said this is equal to:

$$\begin{aligned} &\frac{\alpha}{2}\left(\sum_{k=0}^{k_1} f(k)\right)^2 + \frac{\alpha}{2}\sum_{k=0}^{k_1} f^2(k) + \alpha C\sum_{k=0}^{k_1} f(k) = \\ &\frac{\alpha}{2}\left(\sum_{k=0}^{k_1} f(k) + C\right)^2 - \frac{\alpha}{2}C^2 + \frac{\alpha}{2}\sum_{k=0}^{k_1} f^2(k) \geq -\frac{\alpha}{2}C^2 \quad , \quad \alpha > 0 \end{aligned} \tag{C.31}$$

This result may now be used to solve inequalities of the following type:

$$\eta_{ij}(0,k_1) = \sum_{k=0}^{k_1} v_i(k)\,x_j(k)\left(\sum_{l=0}^{k} \phi_{ij}(v,k,l) + \beta_{ij} - a_{ij}\right) \geq -\gamma_{ij}^2 \tag{C.32}$$

It may be said that:

$$\begin{aligned} \sum_{l=0}^{k} \phi_{ij}(v,k,l) &= \sum_{l=0}^{k} \alpha_{ij} f_{ij}(l) \ , \\ \beta_{ij} - a_{ij} &= C_{ij}\alpha_{ij} \ , \\ f_{ij}(k) &= v_i(k)\,x_j(k) \end{aligned} \tag{C.33}$$

leading to the following result:

$$\phi_{ij}(v,k,l) = \alpha_{ij}\,v_i(k)\,x_j(k) \quad , \quad \alpha_{ij} > 0 \tag{C.34}$$