DUAL DRIVE TRACKING SERVOMECHANISM

by

RONNIE CHI NANG LEUNG

Department of Mechanical Engineering
Faculty of Engineering and Applied Science

September 1983

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Faculty of Engineering and Applied Science

Doctor of Philosophy

"DUAL DRIVE TRACKING SERVOMECHANISM"

by Ronnie Chi-nang Leung

The proposal is to put on top of a conventional slew motor in a piggy-back fashion a tracking motor that can develop high torques over a limited range. Such a system will satisfy the required high tracking performance but it presents a new control action problem. A solution to the problem is proposed involving a microprocessor-based monitor to realign the slow motor occasionally. When the tracking system is designed to operate on board ship, then two signals need consideration: The absolute target motion in space and the disturbance from ship rolling. The monitor is mainly to model these motions. The target motion and ship rolling are modelled with a quadratic polynomial and an autoregressive model, respectively. The parameters of the models are estimated in real-time using a Finite Memory Kalman Filter. The theory of the filter is that the parameters are estimated based on a finite number of past measurements such that the expected estimation error squares is minimum.

Anticipated target motion and ship rolling can be predicted using the estimated models. An optimal control strategy using the estimated models is then calculated. The control action is optimal in the sense that the probability of target loss weighted with the frequency of realigning the slew motor is minimal. The tracking motor responds directly to the target motion and disturbance. However, certain parameters of the tracking motor could be up-dated by the monitor to improve tracking performance.

ACKNOWLEDGEMENTS

The Author would like to express his gratitude to the Admiralty Surface Weapon Establishment for the award of this research. Without it this work would not be possible. The Author would also like to express his appreciation to Dr P J Lawrence and Dr R D Dawson for their helpful guidance, suggestions and encouragements throughout the project. Members of the Mechanical Workshop and the Electronic Workshop of the Mechanical Engineering Department have contributed a significance of effort in building the hardware. The Author would like to especially thank Mr C R Peach and Mr P W Wilkes for their exceptional assistance. Finally the Author would like to thank Karen Kirby for doing the typing.

Ronnie Chi-nang Leung

CONTENTS

SYMBOLS

Included in this list are some of the symbols used throughout the thesis with an indication in parentheses of the chapter in which they appear.

A	Transition matrix (2)(3)(4)
B	Delay operator z^{-1} (2)
	Control matrix (3)
C	Controlled output (1)
	Observation matrix (2)
	Time shifted matrix (3)
D	Disturbance (1)
	Damping coefficient (1)
	Observation matrix (3)
G	Transfer function (1)
	Gain matrix (3)
I	Moment of inertia (1)
J	Cost function (5)
K	Gain matrix (3)
M	Augmented observation matrix (3)
P	Estimation error covariance matrix (3)
Q	Covariance matrix (2)
R	Reference input (1)
	Covariance matrix (2) (3)
S	Time shift matrix (2)
	Augmented measurement noise covariance matrix (3)
T	Sampling period (2)
U	Augmented measurement vector (3)

a_o, a_1, a_2	parameters (2)(3)(5)
d	observation vector (3)
ℓ	altitude (2)
	window length (3)(4)
p	Probability density function (2) (5)
	Lead time (4)
r	Weight ratio (5)(6)
t	time in general (2)(4)(5)
t_o	current time (5)

t_p	Prediction period (5)
u	Control vector (3)
	Velocity (2)
	White noise (2)(3)(4)(6)
v	Measurement noise (2)(3)(4)
w	Weight (5)
x	State vector (2)(3)(4)
y	Noise free measurement (3)
z	Measurement (2)(3)(4)(5)

β	Parameter vector (2)(3)
γ	Autocovariance (2)
ε	Error (1)(3)(6)
θ	Angular position (2)
	Moving-average parameters (2)(3)(4)
θ_o	Initial slew motor position (5)
θ_s	Slew motor position (5)
θ_ℓ	Limit of tracking motor (5)
θ_T	Target position (5)
ρ	Correlation coefficient (2)
σ^2	Variance (2)(5)
Ø	Autoregressive parameters (2)(3)(4)(6)
ψ	Autoregressive parameters (6)
ω	Plant noise (2)

Superscribe

^	Estimate
~	Estimation error
*	Optimal
T	Transpose

CHAPTER 1

INTRODUCTION

Radar is an acronym for Radio Detection and Ranging. It is an active device that operates by radiating electromagnetic energy, and detecting the presence and character of the echo returned from reflecting objects. Radars can be categorised as; search radars and tracking radars. As the name implies, a search radar is planned and designed with emphasis on detecting target presence. It may also provide some coarse information regarding the target. A tracking radar emphasises information extraction, that is, parameter estimation, from the target echo. It tries to estimate with sufficient accuracy the significant target parameters such as range, range rate, angle, and angular velocity once the target presence is established. Since present day radars can perform both the functions adequately, such a binary classification of radars is not always possible.

There are two phases of present day operation, acquisition phase and tracking phase. In the acquisition phase, the antenna is being rotated continuously to search for targets. Once the target presence is established the radar will be switched to the tracking mode which will keep the antenna in line with the target. Between the two phases, the tracking task is harder to achieve in terms of positioning the antenna accurately. The accuracy of the antenna is generally obtained by means of electric, or hydraulic, servomechanism which keeps the antenna in track with targets.

1.1 Conventional Tracking Servo

The simplest model of an angular positional servo used in tracking systems is as shown in Figure 1.1. It is designed to follow an input command. The system consists of a controller, and a motor, electric or electro-hydraulic, to position the load.

In real life, disturbances such as winds, variable friction levels do exist affecting the performance of the servo. The situation can be visualised as in Figure 1.2.

Suppose the transfer function of the controller is $G_1(s)$, and the motor and load dynamics is $G_2(s)$, the transfer function between the output position of the load, C, and the input command, R, is:

$$\frac{C}{R}(s) = \frac{1}{\frac{1}{G_1(s)G_2(s)} + 1} \qquad (1.1)$$

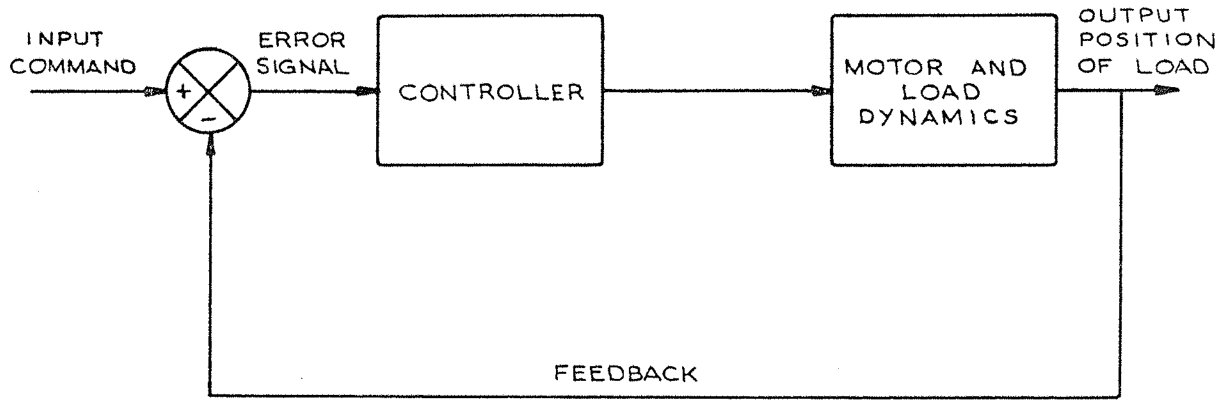


Figure 1.1 Simple positional servo

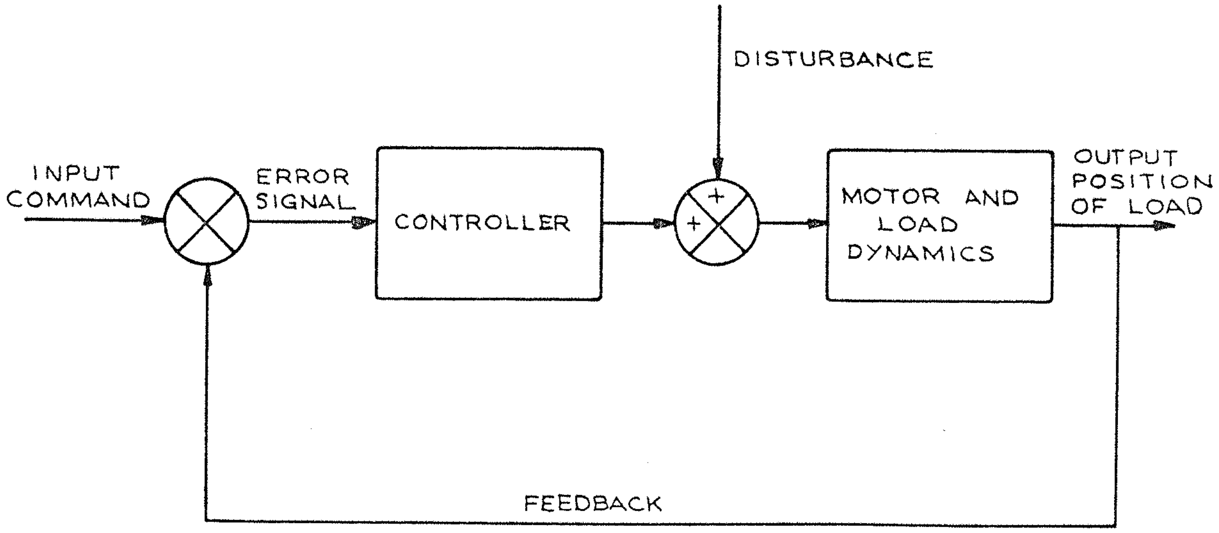


Figure 1.2 Servo with disturbances

It tends to unity as $G_1(s)$ $G_2(s)$ tends to infinity. In other words, a good closed-loop system asks for an open loop gain as high as possible. Now consider the effect of the disturbance, D. Ideally the value of the output, C, due to D should be zero. In this instance the transfer function between the output and the disturbance is:

$$\frac{C}{D}(s) = \frac{G_2(s)}{1 + G_1(s)G_2(s)} \tag{1.2}$$

The effect of the disturbance, D, on the output tends to zero if $G_1(s)$ tends to infinity. Therefore, it is seen that the requirements to follow an input, C, and reject a disturbance, D, in general do not conflict; they both require $G_1(s)$ to be as large as possible. However, such is not the case if a noise, N, originates at the same point as the true input, R. The transfer function between the output, C, and the noise, N, is exactly the same as that for the output, C, and the input, R, given in equation 1.1. For the output, C, tends to zero for any given noise, N, $G_1(s)G_2(s)$ must also tend to zero. This is clearly totally incompatible with the prime task of following the target accurately.

From classical linear control theory the effect of disturbances, D, (e.g. wind, etc.) can be made small either by using high system gain or by increasing the type number of the controller. Besides the ability of rejecting disturbances, a high system gain also contributes to faster response. At present, the general design practise is to design the system transfer function (or filter) with the highest gain that the noise, N, will have minimal effect on the output performance.

1.2 Dual Drive Tracking Servo

A novel D.C. electric motor has recently been proposed [12,42]. It has a gain characteristic much higher than conventional drives. Higher precision and faster response for tracking radar seems possible. Unfortunately, this motor has a disadvantage: It is capable of delivering a tremendously high torque only over a limited angle. Outside this working range the generated torque decreases drastically. One solution to compensate for this is to couple the high torque motor with a conventional motor in a piggy-back fashion. Both motors complement each other to give a superior overall performance on accuracy. The new high torque motor is used as a tracking motor responding to targets whereas the conventional drives acts as a slew motor positioning the tracking motor to favourable position in target tracking. Such novel configuration will have a high system gain much higher than previously obtained without gearing. However, conventional control strategy is no longer applicable to this novel dual drive configuration. A new strategy is required to explore the optimal benefit from the novel drive.

The simplest way of implementing the dual drive tracking servomechanism is as shown in Figure 1.3. Each drive is designed as a tracking servo with feedbacks forming a control loop. The tracking motor will have a higher bandwidth than the slew motor loop. The slew motor responds directly to the command signal. The tracking loop responds only to the error signal between the output from the slew motor and the input. The overall system output is the sum of both control loops. The system then tracks targets with the coarse adjustment from the slew motor in conjunction with the fine adjustment of the tracking motor. This strategy is simple, but the complexity of the system has increased. The physical construction of the newly proposed high torque motor imposes it to be mounted on top of the slew motor. Thus, all torques generated by the tracking motor must be transmitted via the slew motor which itself does not possess a bandwidth as high as the tracking motor. This coupling effect between the motors is difficult to eliminate. Hence, unwanted noise is inevitably induced within the whole system which has adverse consequences possibly the loss of target. The slew motor is then the weakest link within the whole system.

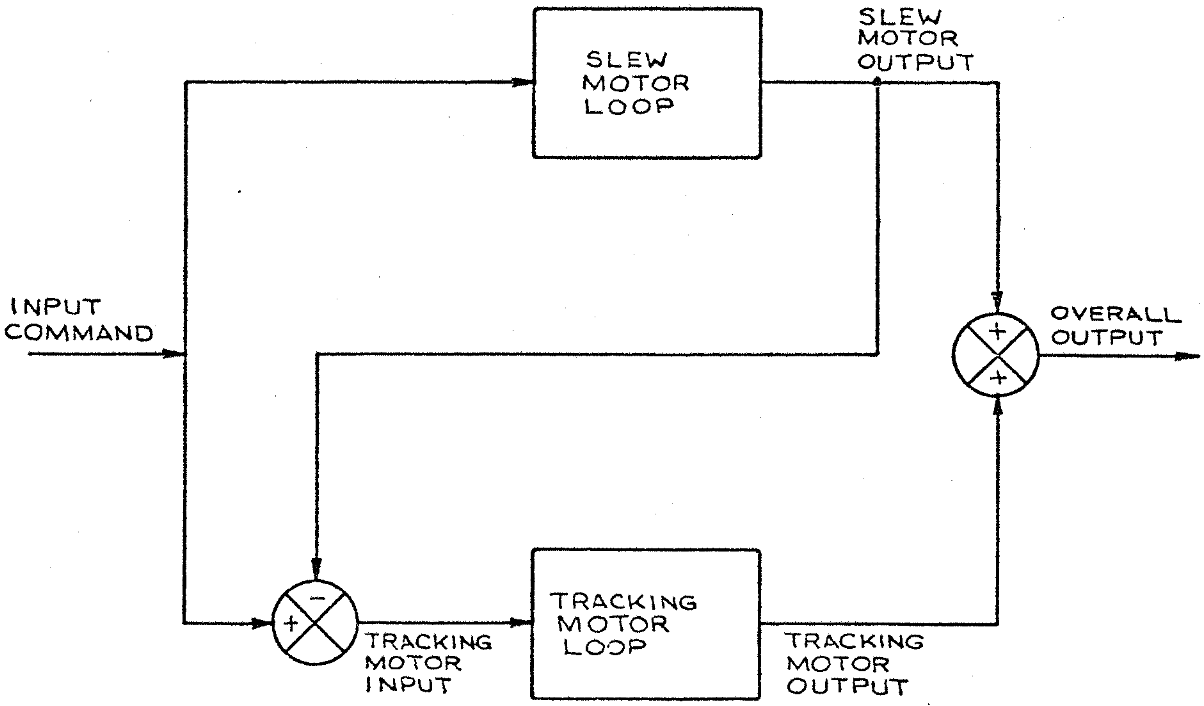


Figure 1.3 Simple Dual Drive Tracking Servo.

With the advance in today's microelectronics, a more direct approach can be considered. A microprocessor-based monitor is being integrated into the system as in Figure 1.4. The tracking loop will still respond directly to the error signal between the output of the slew motor and the input command. Its parameters will be adaptively adjusted by the monitor to accommodate the varying incoming signals. On the other hand, the slew motor is now relegated to a supporting role. It will remain inactive most of the time. Its main task is to provide a rigid base which the tracking motor can react upon. Occasionally it will be activated when the tracking motor will probably lose its target by reaching its limits.

The philosophy behind the proposed control strategy is to tackle the weakest link of the system, the slew motor, directly. It has already been pointed out that the slew motor itself cannot transmit all the torques developed by the tracking motor. The only solution to it is to clamp the output of the slew motor by an external device, such as a brake. Once the slew motor is clamped, the tracking motor has a rigid base to react upon. It can then track the target with its high accuracy. Since the slew motor has been isolated from the system the coupling effect is virtually completely eliminated. However, the slew motor cannot be clamped forever because of the limited working range of the tracking motor. A way to get round this is to realign the tracking motor occasionally by the slew motor. The whole policy of realignment is to keep the frequency of putting the slew motor in action to a minimum that the high performance of the tracking motor will not be deteriorated. Even when realignment takes place the duration will be minimal.

To implement the control strategy, a microprocessor-based monitor is called for. The monitor samples the input signal. With a crude model of the input signal already stored in the memory of the monitor the parameters of the model are estimated in real-time based on a finite past sampled data. A crude model is used because it is intended to work with all sorts of targets. Thus, exact model is not possible. Furthermore, the input signal may consist of non-stationary disturbances. Such disturbances may, however, be approximated by a stationary model over a finite length of the time history. Therefore, discarding the very old sampled data in the estimation process is very important. A finite memory Kalman filtering algorithm is then developed to estimate the parameters based on the most recently sampled data recursively. The input signal within a prediction period ahead is forecasted by extrapolating the estimated model. The variance of the predicted trajectory is supplied from the parameter estimation algorithm. Assuming the

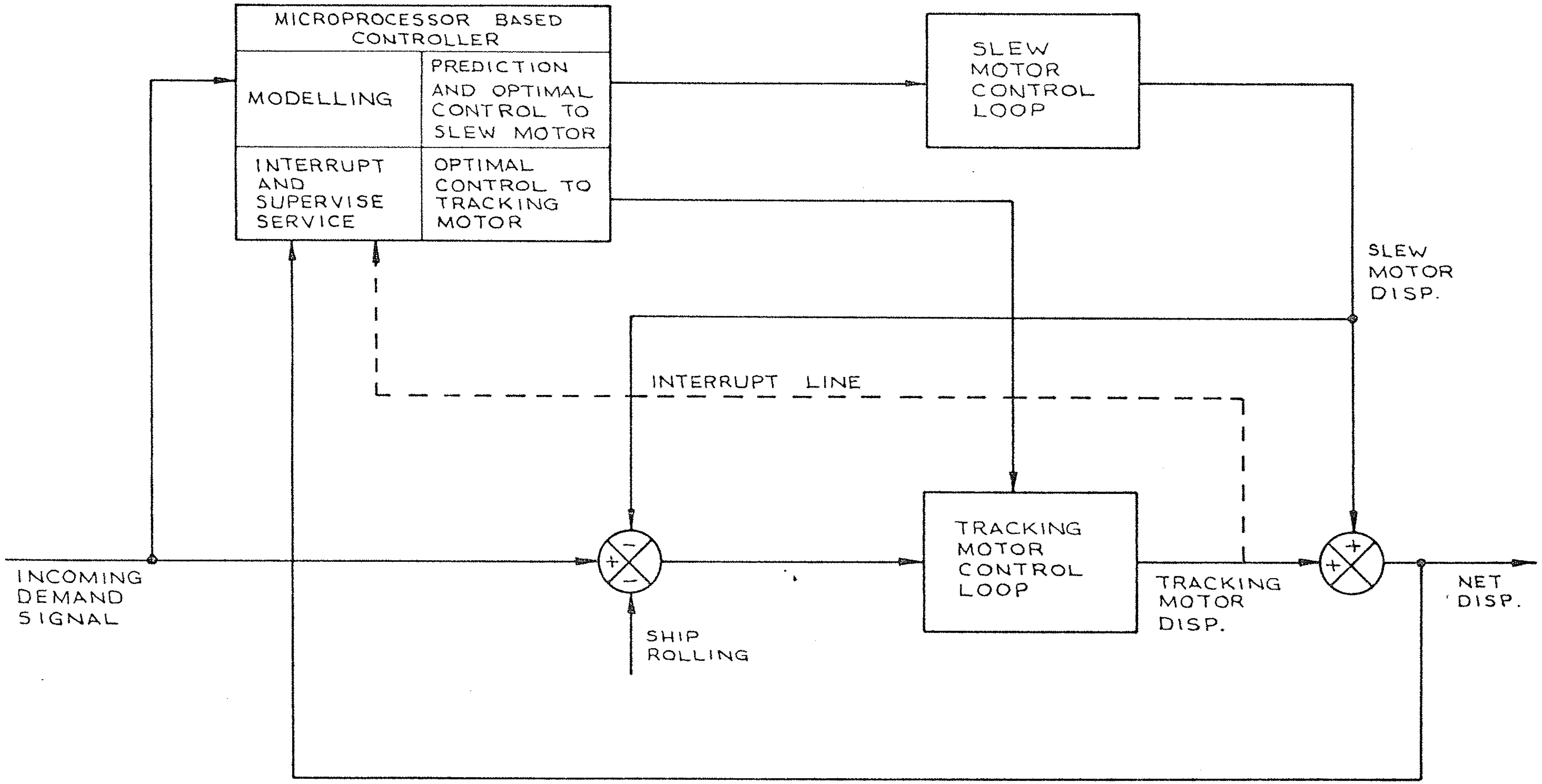

MICROPROCESSOR BASED CONTROLLER
MODELLING
PREDICTION AND OPTIMAL CONTROL TO SLEW MOTOR
INTERRUPT AND SUPERVISE SERVICE
OPTIMAL CONTROL TO TRACKING MOTOR
SLEW MOTOR CONTROL LOOP
SLEW MOTOR DISP.
INTERRUPT LINE
INCOMING DEMAND SIGNAL
+
−
−
SHIP ROLLING
TRACKING MOTOR CONTROL LOOP
TRACKING MOTOR DISP.
+
+
NET DISP.

Figure 1.4 Schematic diagram of proposed dual-drive tracking servomechanism

predicted trajectory as gaussian, the probability of target loss at any time within the prediction period can be evaluated. The monitor makes decisions when and where the slew motor will be realigned within the prediction period. The decisions are made by weighing the probability of target loss and the frequency of realignments. On the whole, modelling is the first step to superior performance in this strategy. The input signal that the servomechanism is supposed to work with in this thesis is described in the next section.

1.3 Working Environment

The tracking servomechanism is aimed to be working on board, and to be driving an antenna. The anticipated working situation is visualised as in Figure 1.5. The only signal obtained from the radar is the angular error, ε, between the target and the electrical null axis of the antenna. The position of the antenna, ϕ, relative to the ship is the output from the tracking servomechanism. The axis of the ship is, however, not stationary in space. The wave causes the ship to roll with an angle, θ . Normally, the target will be far away, the centres of ship rolling and the antenna can be treated concentric. The absolute position of the target at a particular time can be re-established by the relationship:

$$\Psi = \phi + \theta + \varepsilon \qquad (1.3)$$

It is proposed that the absolute position of the target will be modelled rather than its relative position to the ship. This is because the absolute measurements can be modelled by deterministic model. Whereas the relative measurements consist of the rolling motion, θ, which is stochastic in nature.

$$\Psi_r = \Psi + \theta \qquad (1.4)$$

Measurement noise, such as thermal noise, is always associated with all types of sensors. The measured target position will be corrupted with noise as:

$$\Psi_m = \Psi + n \qquad (1.5)$$

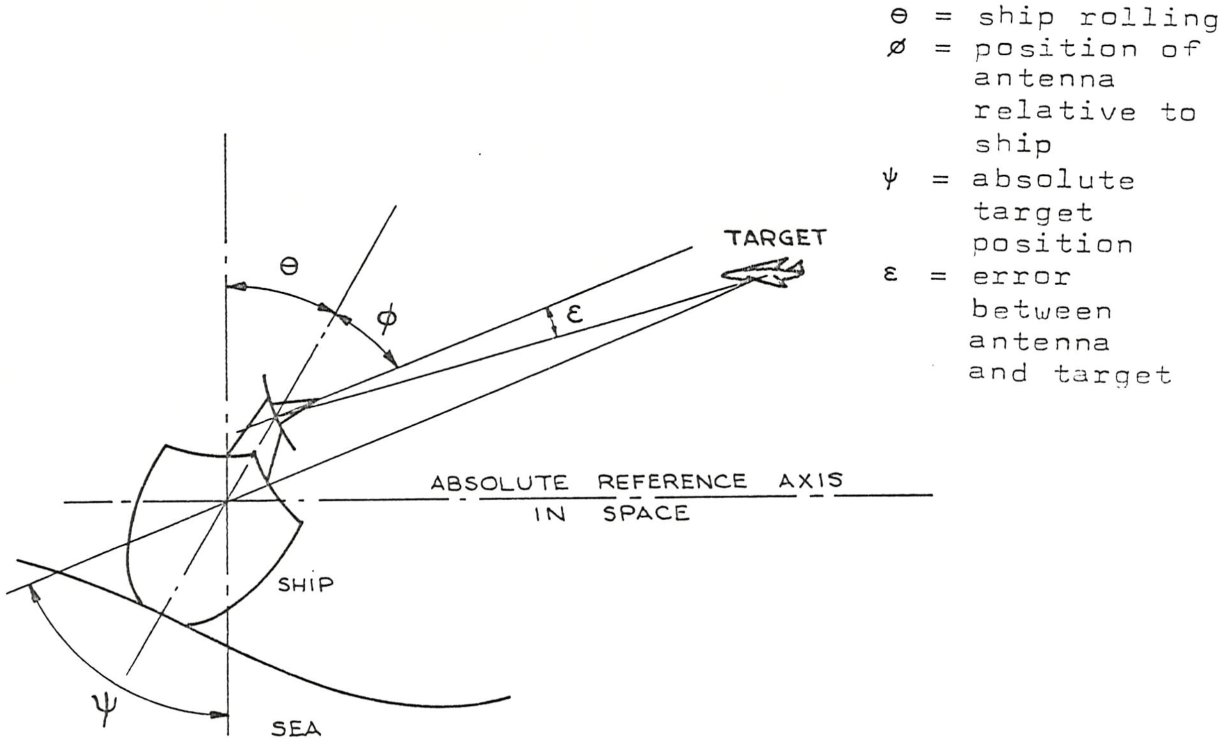

θ = ship rolling
ϕ = position of antenna relative to ship
ψ = absolute target position
ε = error between antenna and target
TARGET
θ
ϕ
ε
ABSOLUTE REFERENCE AXIS IN SPACE
SHIP
SEA
ψ

Figure 1.5 Angular relationship of target and servomechanism

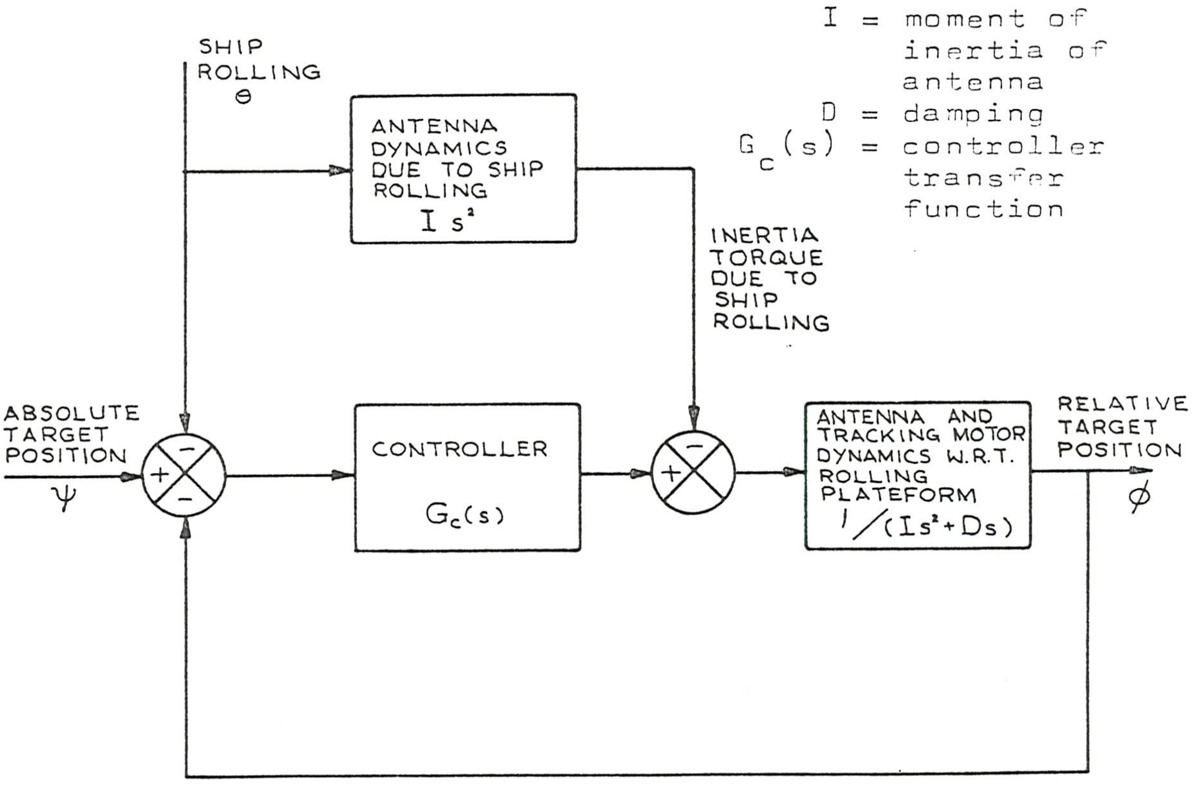

I = moment of inertia of antenna
D = damping
$G_c(s)$ = controller transfer function
SHIP ROLLING θ
ANTENNA DYNAMICS DUE TO SHIP ROLLING $I s^2$
INERTIA TORQUE DUE TO SHIP ROLLING
ABSOLUTE TARGET POSITION ψ
−
+
−
CONTROLLER $G_c(s)$
−
+
ANTENNA AND TRACKING MOTOR DYNAMICS W.R.T. ROLLING PLATEFORM $1/(Is^2+Ds)$
RELATIVE TARGET POSITION ϕ

Figure 1.6 Tracking servomechanism with slew motor immobilized

In practice, wind loading is the main disturbance on the antenna. This could be eliminated by employing a radome. The complete block diagram of the servomechanism with the slew motor out of action is shown in Figure 1.6. The output position of the servomechanism, ϕ, is related to the ship rolling and the demanded absolute position as:

$$\phi = \frac{G_c(s)}{Is^2 + Ds + G_c(s)} \Psi - \frac{G_c(s) + Is^2}{Is^2 + Ds + G_c(s)} \theta \qquad (1.6)$$

1.4 Conclusions

A novel dual drive tracking servomechanism is proposed. This configuration is largely imposed by the non-linearity of a newly proposed high gain motor. This motor is superior to conventional motors without gearing, but it has a limited working range. In the proposed configuration a conventional slew motor is employed in realigning the tracking motor from time to time.

In the working environment of the servomechanism, two motions need consideration. They are the target and the ship rolling motions. Realignments are decided on the predictions of these two motions.

In equation 1.6, neglecting the term associating with the absolute target position, Ψ, it was seen that the ship rolling motion will be counter-balanced passively if the system is very lightly damped, i.e. D tends to zero giving:

$$\phi = -\theta \qquad (1.7)$$

It is then tempting to suggest that if the system is to respond to a low bandwidth target, for instance a satellite, there will be no need to have a stabilising platform. However, it is not really the case in practice as there is friction between mechanical parts.

CHAPTER 2

SIGNAL MODELLING

The proposed strategy is described in the last chapter. It's main task is to determine when and where the slew motor should be realigned in order to minimise the possibility of losing the target due to the non-linearity of the tracking motor. If a realignment is called for, the tracking system is more concerned with the possible position of the target in the future than the target in the past, i.e. a prediction on the target trejectory is essential in the process of decision making. Mathematical modelling is thus the essence of the work of the project. The incoming demand signal is modelled by mathematical models on a microprocessor-based monitor. The optimal control law is derived based on the prediction models. Hence the models for the incoming demand signal must possess the appropriate characteristics of the actual signal. Unfortunately, it is sometimes impossible, or impracticable, to fit a model with the exact characteristics of the incoming signal. Normally, a highly accurate model can only be used at the expense of complexity, which in turn increases the computation time. Furthermore, the target can be of any type. No single mathematical model can cover all types of targets exactly. A general low order model is a good compromise between complexity and exactness. This, however, imposes a restriction that the motions can only be modelled within a short time interval since parameters of this model must vary with time.

An analysis of the incoming signal is necessary before proceeding to model building. As it has been already discussed in Chapter 1, besides the target motion, a model of the ship rolling motion is also required. In real life the target will be in a three-dimensional space. The tracking system will then have two degrees of freedom: The bearing and the azimuth. In order to avoid the coupling effect between the two degrees of freedom, only the azimuth of the target on a two-dimensional plane is considered. The performance of this dual-drive tracking servomechanism is evaluated upon the angular error on this two-dimensional plane.

2.1 Target Flying at Constant Speed, Constant Velocity

In general, a target flies at a constant speed and at a constant altitude for most of the time to conserve fuel. Manoeuvres will only take place when it is necessary. Referring to Figure 2.1, a target is flying at constant speed, u, at constant altitude, ℓ. The angular profile of the tracker is:

$$\theta(t) = \frac{\pi}{2} + \tan^{-1}\left(\frac{u}{\ell}t - \frac{x_o}{\ell}\right) \qquad (2.1)$$

The angular velocity is:

$$\dot{\theta}(t) = \frac{u}{R} \sin \theta(t)$$

$$= \frac{u}{\ell} \sin^2 \theta(t) \qquad (2.2)$$

The acceleration of the tracker can be written as:

$$\ddot{\theta}(t) = \frac{d\dot{\theta}(t)}{dt}$$

$$= \frac{2u^2}{\ell^2} \cos \theta(t) \quad \sin \theta(t) \quad \sin^2 \theta(t)$$

$$= \frac{u^2}{\ell^2} \sin 2\,\theta(t) \quad \sin^2 \theta(t)$$

$$= \frac{u^2}{R^2} \sin 2\,\theta(t) \qquad (2.3)$$

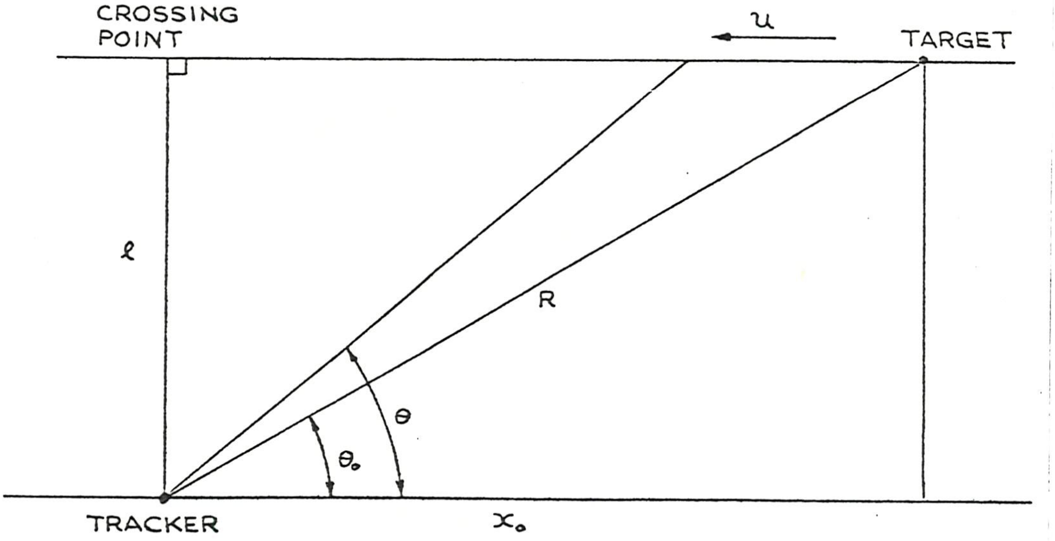


Figure 2.1 Angles associated with tracking a target flying at constant altitude.

It is seen therefore, that even for apparently straightforward cases of a constant speed target flying at constant altitude, the angular rate and the angular acceleration of the tracker is not constant.

2.2 Constant Velocity Model

Most tactical weapons' guidance systems require accurate tracking of manoeuvrable targets such as aircraft, ships and submarines. In the process of modelling the targets, one assumption is generally used: It is commonly assumed that targets, without manoeuvring, follow straight line constant velocity trajectories. In the 1970's, Singer [43, 44] first presented a model that accounted for the manoeuvre capability in a simple way. Basically, the model is still based on the consideration that targets normally move at constant velocity, either linear or angular. Turns, evasive manoeuvres and accelerations due to atmospheric turbulence may be viewed as perturbations upon the constant velocity trajectory. These perturbations are assumed to have a probability density as shown in Figure 2.2. The quantity A_{max} is the maximum acceleration which the target can withstand.

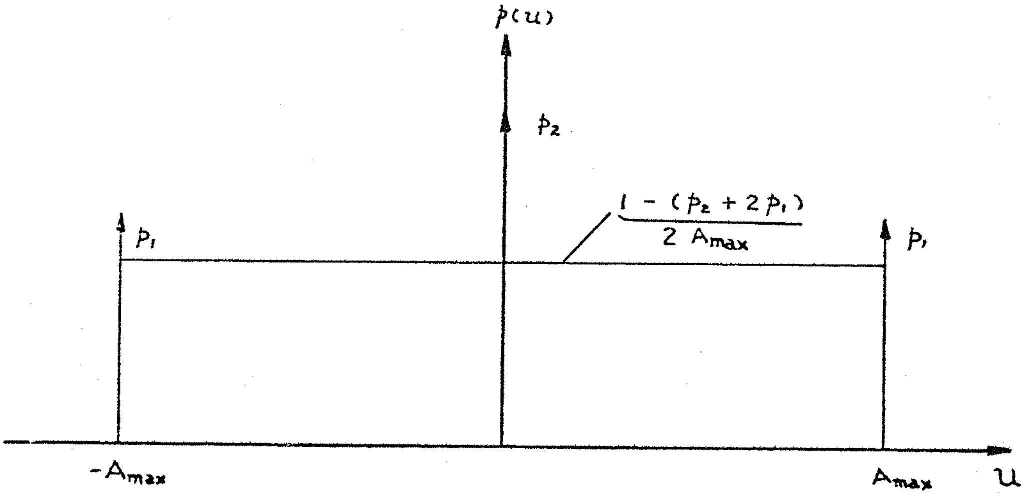


Figure 2.2 Probability density of Target Manoeuvre.

After the whitening of the perturbations, the model (for one dimension) is given as:

$$\underline{x}(k+1) = A(k+1,\ k)\ \underline{x}(k) + G\ \omega(k) \qquad (2.4)$$

$$z(k+1) = C\ \underline{x}(k+1) + v\ (k) \qquad (2.5)$$

where;

$$\underline{x}^T(k) = (x(k) \quad \dot{x}(k) \quad u(k)) \tag{2.6}$$

$$G^T = (0 \quad 0 \quad 1) \tag{2.7}$$

$$C = (1 \quad 0 \quad 0) \tag{2.8}$$

$$A(k+1,k) = \begin{pmatrix} 1 & T & 0 \\ 0 & 1 & 1 \\ 0 & 0 & \rho \end{pmatrix} \tag{2.9}$$

The term u(k) is the perturbation (manoeuvre) $\omega(k)$ and v(k) are gaussian white noise with

$$Q(k) = E\left[\omega(k) \quad \omega^T(k) \right] = \sigma_m^2 (1 - \rho^2) \tag{2.10}$$

and

$$R(k) = E\left[v(k) \quad v^T(k) \right] = \sigma_r^2 \tag{2.11}$$

The values for the manoeuvre variance, σ_m^2, and the correlation coefficient, ρ, depend on the manoeuvre characteristics of the targets being tracked. The correlation coefficient assures that as time between manoeuvre samples increases, the correlation between these samples decreases.

The model presented by Singer is a good model combining simplicity with reliable representation of the modelled phenomena. However, the model is by no means without flaws. First of all, the implementation of the model is restricted to one class of targets once the quantity of maximum acceleration, the manoeuvre variance and the correlation coefficient are chosen a priori. The model is thus not a general one. Furthermore, as pointed out in the last section, the bearing rate and acceleration between the target and the tracker will not be constant even with the target flying at constant velocity, and constant altitude. The assumption of constant angular velocity is only justified if the target is far away. This can be seen from (2.1) and (2.2). Therefore, if the target is at close range, the model becomes insufficient. Nowadays, conventional tracking systems have to switch between models, close range and long range models.

2.3 Quadratic Polynomial Model

Restating (2.1) here

$$\theta(t) = \frac{\pi}{2} + \tan^{-1}\left(\frac{u}{\ell}t - \frac{x_o}{\ell}\right) \qquad (2.12)$$

Considering a short time interval only, the term $(\frac{u}{\ell}t - \frac{x_o}{\ell})$ will have an absolute value of less than unity, hence:

$$\theta(t) \approx \frac{\pi}{2} + \left(\frac{u}{\ell}t - \frac{x_o}{\ell}\right) - \frac{1}{3}\left(\frac{u}{\ell}t - \frac{x_o}{\ell}\right)^3 + \ldots$$

$$\theta(t) \approx \frac{\pi}{2} - \frac{x_o}{\ell} + \frac{1}{3}\left(\frac{x_o}{\ell}\right)^3 - \left(\frac{x_o}{\ell}\right)^2 \frac{u}{\ell}t + \frac{u}{\ell}t +$$

$$\left(\frac{x_o}{\ell}\right)\left(\frac{u}{\ell}\right)^2 t^2 - \frac{1}{3}\left(\frac{u}{\ell}\right)^3 t^3 \ldots \qquad (2.13)$$

In view of (2.13) and the drawbacks of the constant velocity model, a quadratic polynomial may also be used to model the target trajectory. The model is:

$$z(t) = a_o + a_1 t + a_2 t^2 \qquad (2.14)$$

where,

$z(t)$ = angular position of target at time (t)

a_o, a_1, a_2 = parameters

t = time

This quadratic polynomial model is very simple in concept. A curve is being fitted to the measurements. It is also capable of coping with the changes of the target characteristics that occurs with the target range. On top of this, this model has no restriction on the type of target being tracked. However, the model is not good enough to accommodate a manoeuvring target. One of the possible ways to get round this problem is to apply a window to the measurements so that the estimated parameters, $\hat{a}_o$, $\hat{a}_1$ and $\hat{a}_2$ depend only on the measurements inside the window. The use of the model also imposes an assumption that the parameters are time-invariant within the estimating period but can vary with time over long periods.

It is shown in Section 2.4 that this quadratic polynomial is closely related to the constant velocity model presented by Singer. If the parameters of the quadratic polynomial model is assumed to be random, the quadratic polynomial is identical to the constant velocity model with a very fast sampling rate.

The polynomial model needs to be rewritten in state-space form in order to facilitate the parameter estimation. In state-space form, the model (2.14) can be written as:

$$\dot{\underline{x}}(t) = 0 \tag{2.15}$$

$$z(t) = \underline{d}^T(t) \; x(t) + v(t) \tag{2.16}$$

where

$$\underline{x}^T(t) = (a_o \quad a_1 \quad a_2) \; ; \tag{2.17}$$

$$\underline{d}^T(t) = (1 \quad t \quad t^2) \; ; \tag{2.18}$$

$z(t)$ = measured signal at time, t ;

and $v(t)$ = measurement noise at time, t.

In discrete time form:

$$\underline{x}(k+1) = \underline{x}(k) \tag{2.19}$$

$$z(k+1) = \underline{d}^T(k+1) \; \underline{x}(k+1) + v(k+1) \tag{2.20}$$

where

$$d^T(k) = (1 \quad kT \quad k^2T^2) \tag{2.21}$$

with sampling time T.

There is one problem in association with this model. Since the vector $\underline{d}(k)$ involves time it will become unacceptably large as time progresses. Eventually the values of the vector $\underline{d}(k)$ will be too large for a digital computer to handle. One method to overcome this is the shifting of the time axis forward after each estimation interval, i.e. the vector $\underline{d}(t)$ becomes constant in each parameter estimation process. For instance if the model is time shifted one step backwards,

$$\underline{x}'(k+1) = S \; \underline{x}(k+1) \tag{2.22}$$

where

$\underline{x}'(k+1)$ = time-shifted state vector, and

$$S = \begin{pmatrix} 1 & T & T^2 \\ 0 & 1 & 2T \\ 0 & 0 & 1 \end{pmatrix} \tag{2.23}$$

2.4 Constant Acceleration Model

Re-defining the state vector in (2.17) as

$$\begin{aligned} \underline{x}^T(k) &= (a_o \quad a_1 \quad 2a_2) \\ &= (b_o \quad b_1 \quad b_2) \end{aligned} \tag{2.24}$$

The polynomial model becomes:

$$\underline{x}(k+1) = \underline{x}(k) \tag{2.25}$$

$$z(k+1) = \underline{d}^T(k+1) \; \underline{x}(k+1) + v(k+1) \tag{2.26}$$

where

$$\underline{d}^T(k) = (1 \quad kT \quad \tfrac{1}{2}(kT)^2) \tag{2.27}$$

and the time shift matrix S becomes:

$$S = \begin{pmatrix} 1 & T & \tfrac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \tag{2.28}$$

Combining the time shift operation, kT = (n+1)T, with the polynomial model gives:

$$\begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{(n+1)T} = \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{kT} \qquad (2.29)$$

and

$$z(kT) = z((n+1)T)$$

$$= \begin{pmatrix} 1 & (n+1)T & \frac{1}{2}\left[(n+1)T\right]^2 \end{pmatrix} \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{kT} + v(kT)$$

$$z(kT) = \begin{pmatrix} 1 & nT & \frac{1}{2}(nT)^2 \end{pmatrix} \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{kT} + v(kT)$$

$$= \begin{pmatrix} 1 & nT & \frac{1}{2}(nT)^2 \end{pmatrix} \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{(n+1)T} + v(kT) \qquad (2.30)$$

Let $\begin{pmatrix} \alpha_o \\ \alpha_1 \\ \alpha_2 \end{pmatrix}_{kT} = \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{(n+1)T} = \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{kT}$ (2.31)

Therefore

$$z(k) = \begin{pmatrix} 1 & nT & (nT)^2 \end{pmatrix} \begin{pmatrix} \alpha_o \\ \alpha_1 \\ \alpha_2 \end{pmatrix}_{k} + v(k) \qquad (2.32)$$

$$\text{and} \begin{pmatrix} \alpha_o \\ \alpha_1 \\ \alpha_2 \end{pmatrix}_{kT} = \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{kT}$$

$$= \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{(k-1)T}$$

$$= \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_o \\ \alpha_1 \\ \alpha_2 \end{pmatrix}_{(k-1)T} \qquad (2.33)$$

The whole polynomial model can be reduced with k setting to zero

$$\underline{x}\,(k+1) = S\ \underline{x}(k) \qquad (2.34)$$

$$z(k+1) = \underline{d}^T\ \underline{x}(k+1) + v(k+1) \qquad (2.35)$$

where

$$\underline{x}^T(k) = (\alpha_o \quad \alpha_1 \quad \alpha_2) \qquad (2.36)$$

$$\text{and} \quad \underline{d}^T = (\ 1 \quad 0 \quad 0\ ) \qquad (2.37)$$

The model (2.34) and (2.35) is now a Newtonian one. The elements of the state vector, $\underline{x}$, are the position (α_o), velocity (α_1) and acceleration (α_2). From the matrix S, which is now the transition matrix, it imposes that the acceleration (α_2) to be constant.

It is shown in [43] that the constant velocity model can be reduced to:

$$\underline{x}(k+1) = S\ \underline{x}(k) + \underline{\omega}(k) \qquad (2.38)$$

(2.38) is identical to (2.34) except that a white noise sequence term $\underline{\omega}(k)$ is introduced. This in effect allows the parameters to be random variables. (2.38) is derived by discretizing a continuous time counterpart, with an assumption that the product of the sampling period and the reciprocal of the acceleration time constant of the target is small.

In this section, the constant acceleration model is actually derived from the quadratic model in the previous section. That is, the constant acceleration model can actually be treated as fitting a quadratic curve to the data. If the parameters of the quadratic model be treated as random variables, the quadratic model becomes a constant velocity model as presented by Singer.

2.5 Deterministic Model for Target Motion

The tracking system is supposed to work with all sorts of targets. When the system is tracking a target, it has no clear idea of what class and type of targets it is tracking. In view of this, the model for the target motion must be general and be capable of handling every class of target. Fortunately, nearly all types of targets follow a straight or nearly straight trajectory with constant velocity. Manoeuvres only occur occasionally. If a reasonable model of the target is known, its whole trajectory can be determined with sufficient confidence once its initial position is known. Thus, a deterministic model is considered to be appropriate in modelling the target motion.

It was shown in the last three sections that the quadratic polynomial model, the constant velocity model and the constant acceleration model are very similar to each other. There is actually nothing to distinguish between the quadratic polynomial model and the constant acceleration model as one can be derived from another. However, from the point of parameter estimation using Kalman filtering, the constant acceleration model is easier to implement than the quadratic polynomial model. The constant acceleration model was finally proposed for modelling the target motion. Although the constant velocity model is commonly used. It was not chosen because the constant velocity model is not truely a general model. It will not cover all classes of targets without modification. It needs the knowledge of the class of target being tracked. For generality this information was not assumed in this work. The constant acceleration model is not without problem. It imposes an assumption that the acceleration is constant. This is clearly not the case as pointed out in Section 2.1. There are two possible solutions to it. The parameters of the model may be treated as random variables as in (2.38) and are estimated based on the infinite past measurements. Alternatively, the parameters of the constant acceleration model may be treated as invariable over a finite window of the measurements and are estimated based only on the measurements within the window. The former solution, however, still requires some knowledge about the target, which is assumed unavailable, so as to establish the variance of the introduced plant noise term in (2.38). It is then the second solution which is adopted for this work.

It seems reasonable to treat the parameters of the constant acceleration model to be random variables and to be estimated based on a finite window of the measurements. This is not chosen because of the difficulty in establishing a stable filtering algorithm for such conditions. In the derivation of such an algorithm, the correlation between the past plant noise and the estimation error is required. It is the inability of establishing this correlation that hinders the development of the algorithm (Appendix B). In the case of Kalman filtering algorithm, the introduction of a plant noise term to the model is in effect preventing the covariance matrix becoming zero [32, 35, 47] . It is pointed out in Chapter 3 that the introduction of a window stops the covariance matrix from decreasing, too. Thus, if both plant noise and windowing be introduced simultaneously, it is suspected that the covariance matrix might be forced to increase, which in turn increases the gain, and thus would cause instability to the algorithm.

It is interesting to mention here that the polynomial model can be formulated as a stochastic process: An integrated moving-average process (Section 2.9).

2.6 Autoregressive - Moving - Average (ARMA) Model

ARMA models are stochastic models based on the idea of Yule [49] that a stationary time series in which successive values are highly dependent can be usefully regarded as generated from a series of independent inputs. These inputs are usually assumed to be white noise. The concept can be visualised as a white noise sequence passing through a linear filter. The shaped output from the filter will then resemble the properties of the stochastic process, (Figure 2.3). The mentioned linear filter may be an ARMA model.

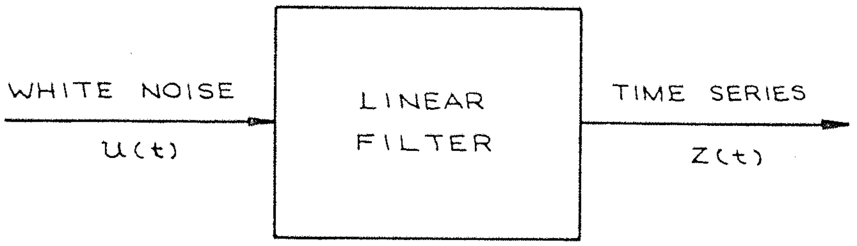


Figure 2.3 Representation of a time series as output from a linear filter

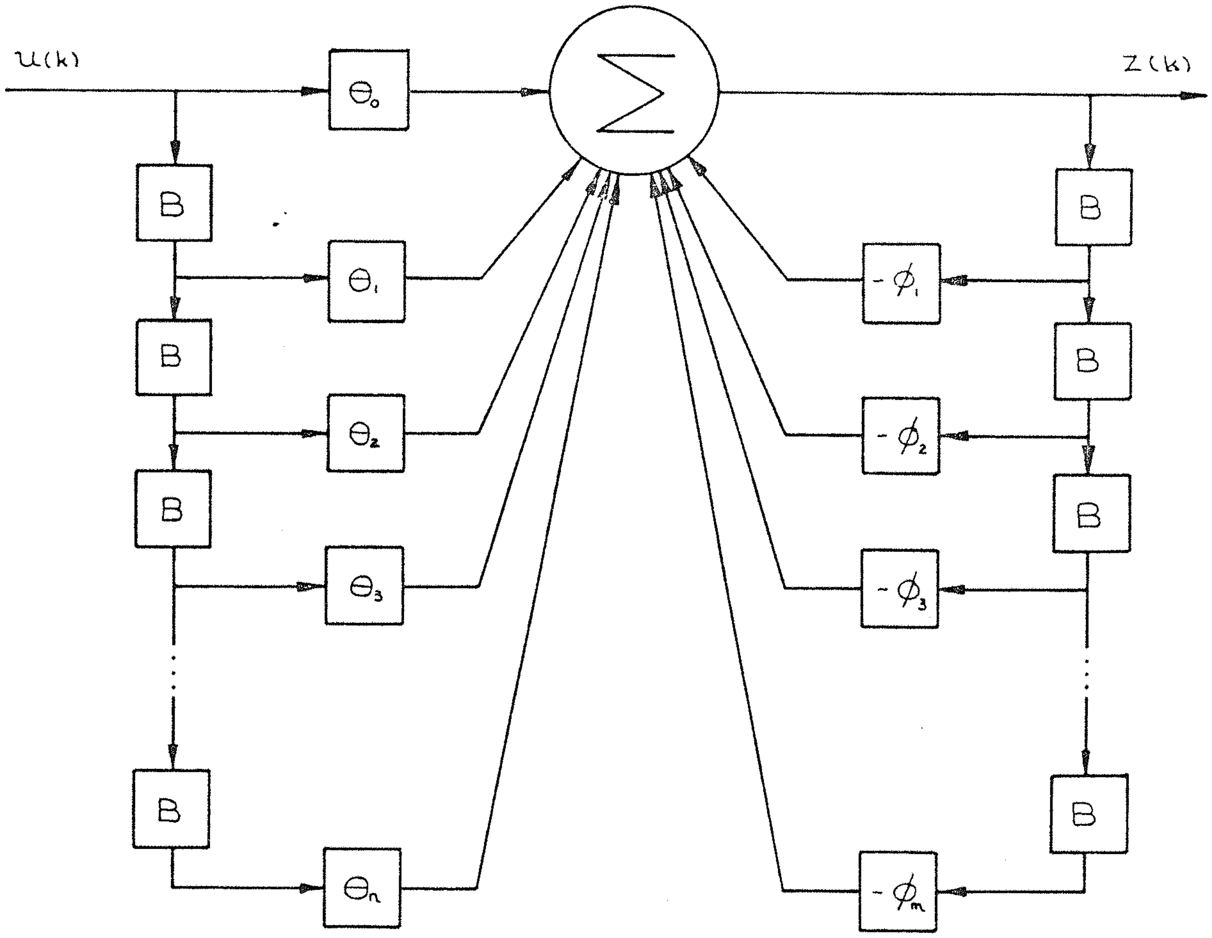


Figure 2.4 ARMA model

Mathematically, Figure 2.4, an ARMA model is of the form:

$$\phi(B) \quad z(t) \quad = \quad \theta(B) \quad u(t) \tag{2.39}$$

where

$$\phi(B) \quad = \quad 1 \; + \; \phi_1 B \; + \; \phi_2 B^2 \; + \; . \; . \; . \; + \phi_m B^m \quad ;$$

$$\theta(B) \quad = \quad \theta_o \; + \; \theta_1 B \; + \; \theta_2 B^2 \; + \; . \; . \; . \; + \theta_n B^n \quad ;$$

$z(t)$ = measured signal at time, t

$u(t)$ = gaussian inaccessible input at time, t

B = delay operator (z^{-1})

ϕ's = autoregressive coefficients

and θ's = moving - average coefficients

The model is normally referred to as an ARMA(m,n) where m is the order of the autoregressive part and n is the order of the moving-average part. The process may be thought of in two ways. Namely:

i) As a mth order autoregressive (AR(m)) process

$$\phi(B) \quad z(t) \quad = \quad e(t) \tag{2.40}$$

with e(t) following the nth order moving average (MA(n)) process

$$e(t) \quad = \quad \theta(B) \quad u(t) \tag{2.41}$$

ii) As a nth order moving-average process

$$z(t) \quad = \quad \theta(B) \quad b(t) \tag{2.42}$$

with b(t) following the mth order autoregressive process

$$\phi(B) \quad b(t) \quad = \quad u(t) \tag{2.43}$$

So that,

$$\phi(B) \quad z(t) \quad = \quad \theta(B) \quad \phi(B) \quad b(t)$$

$$= \quad \theta(B) \quad u(t) \qquad (2.44)$$

The ARMA(m,n) process is defined as a stationary process only if the characteristic equation $\phi(B) = 0$ has all its roots lying outside the unit circle in the Z^{-1} - plane [8] . Similarly, the process is said to be invertible if the roots of $\theta(B) = 0$ lie outside the unit circle in the Z^{-1} plane.

2.7 Stochastic Model for Ship Rolling Motion

The ship rolling motion behaves differently from the target motion. The waves of an open sea is generated by random influences, such as winds, gust, and atmospheric changes, etc. The waves in turn roll the ship that is sitting on the sea. Therefore, the ship rolling motion can be treated stochastic. The idea of Yule [49] is then suitable in applying to the modelling of the ship rolling motion. It is known that the spectrum of the motion is dominated by three frequencies with periods of six, seven and eight seconds as shown in Figure 2.5. Another property of the motion is that it is not truely a stationary process. However, it may be treated stationary over a finite period of the motion.

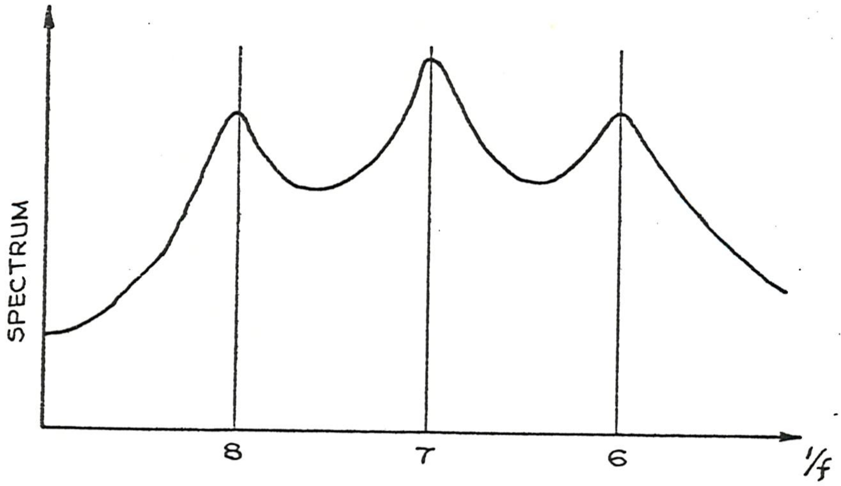


Figure 2.5 Spectrum of Ship Rolling Motion.

Clearly the model for the ship rolling motion must demonstrate a similar spectrum. Consider a transfer function, in S-domain:

$$H(s) = \frac{Ks}{s^2 + as + b} \tag{2.45}$$

It has a peaky frequency response at frequency f_r if the damping is light (Figure 2.6). The coefficient K scales the general shape of the frequency response. The coefficient a, which is the product of the damping ratio and the natural frequency, governs the peaky shape of the response. b is the square of the natural frequency. The position of the peak is determined by the damping ratio and the natural frequency. For a lightly damped filter, the peak will be very close to the natural frequency.

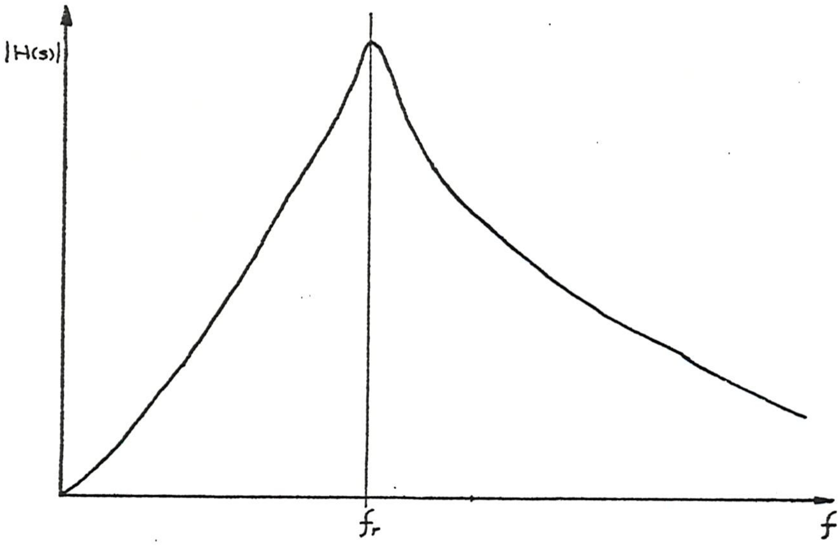
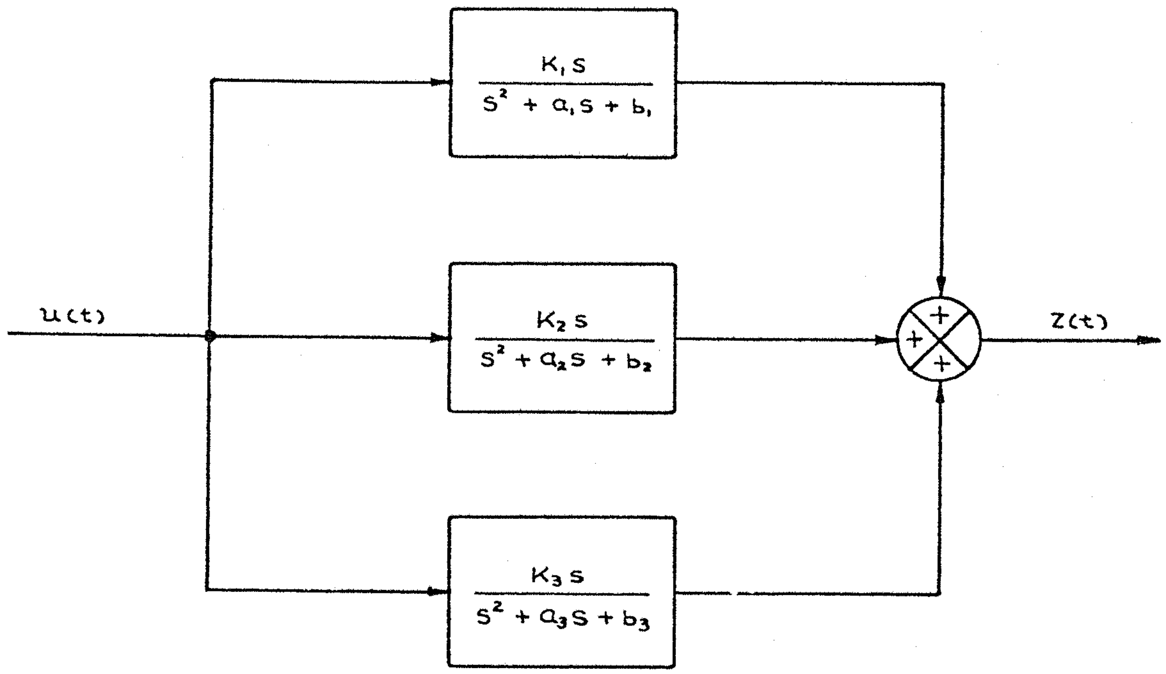


Figure 2.6 Typical Frequency Response of $H(s) = \dfrac{Ks}{s^2 + as + b}$

If three of these transfer functions, each having different coefficients, the ship rolling motion can be modelled by forming the three transfer functions in parallel (Figure 2.7). The result transfer function is:

$$\frac{z}{u}(s) = \frac{n_4 s^5 + n_3 s^4 + n_2 s^3 + n_1 s^2 + n_o s}{s^6 + d_5 s^5 + d_4 s^4 + d_3 s^3 + d_2 s^2 + d_1 s + d_o} \tag{2.46}$$



$u(t)$
$\frac{K_1 s}{s^2 + a_1 s + b_1}$
$\frac{K_2 s}{s^2 + a_2 s + b_2}$
$\frac{K_3 s}{s^2 + a_3 s + b_3}$
+
+
+
$Z(t)$

Figure 2.7 Model of Ship Motion in S-domain.

Applying the Z-transformation, $B = e^{-sT}$ (2.47)

the transfer function in (2.46) becomes:

$$\frac{z}{u}(B) = \frac{\theta_o + \theta_1 B + \theta_2 B^2 + \theta_3 B^3 + \theta_4 B^4 + \theta_5 B^5 + \theta_6 B^6}{1 + \phi_1 B + \phi_2 B^2 + \phi_3 B^3 + \phi_4 B^4 + \phi_5 B^5 + \phi_6 B^6} \tag{2.48}$$

thus, (2.48) indicates that an ARMA(6,6) will describe the ship motion fully.

The ARMA model can be reduced to the state-space form [29] of

$$\underline{x}(k+1) = A(k+1,k)\ \underline{x}(k) + C \begin{pmatrix} u(k+1) \\ u(k) \end{pmatrix} \tag{2.49}$$

$$z(k+1) = \underline{d}^T\ \underline{x}(k+1) \tag{2.50}$$

where;

$$A(k+1,k) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -\phi_6 \\ 1 & 0 & 0 & 0 & 0 & -\phi_5 \\ 0 & 1 & 0 & 0 & 0 & -\phi_4 \\ 0 & 0 & 1 & 0 & 0 & -\phi_3 \\ 0 & 0 & 0 & 1 & 0 & -\phi_2 \\ 0 & 0 & 0 & 0 & 1 & -\phi_1 \end{pmatrix} \tag{2.51}$$

$$C^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \theta_o \\ \theta_6 & \theta_5 & \theta_4 & \theta_3 & \theta_2 & \theta_1 \end{pmatrix} \tag{2.52}$$

$$\underline{d}^T = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1) \tag{2.53}$$

and the states are:

$$\begin{aligned} x_1(k+1) &= -\phi_6\, x_6(k) + \theta_6\, u(k) \\ x_2(k+1) &= x_1(k) - \phi_5\, x_6(k) + \theta_5\, u(k) \\ x_3(k+1) &= x_2(k) - \phi_4\, x_6(k) + \theta_4\, u(k) \\ x_4(k+1) &= x_3(k) - \phi_3\, x_6(k) + \theta_3\, u(k) \\ x_5(k+1) &= x_4(k) - \phi_2\, x_6(k) + \theta_2\, u(k) \\ x_6(k+1) &= x_5(k) - \phi_1\, x_6(k) + \theta_1\, u(k) + \theta_o\, u(k+1) \end{aligned} \tag{2.54}$$

In this form, the parameters ϕ's and θ's in (2.51) and (2.52) can only be identified by an extended Kalman Filter algorithm [24, 32, 35, 40, 46, 47]. Not only is the estimate of the state vector $\underline{x}(k+1)$ redundant, the overhead in computation is high. The parameters must be augmented into the state vector thus pushing the effective system to an order of eighteen.

A simpler way to estimate the parameters Ø's and θ's without estimating the states is to rearrange (2.48) as

$$z(k) = - \sum_{i=1}^{6} \phi_i \, z(k-i) + \sum_{i=1}^{6} \theta_i \, u(k-i) + \theta_o \, u(k) \tag{2.55}$$

$$= \underline{d}^T(k) \, \underline{\beta} + \theta_o \, u(k) \tag{2.56}$$

where

$$\underline{d}^T(k) = (-z(k-1) \; -z(k-2) \; \ldots \; -z(k-6) \; u(k-1) \; u(k-2) \; . \; . \; u(k-6)) \tag{2.57}$$

$$\underline{\beta}^T = (\phi_1 \; \phi_2 \; \ldots \; \phi_6 \; \theta_1 \; \theta_2 \; \ldots \; \theta_6 \;) \tag{2.58}$$

By the assumption of stationarity, the ARMA process can be written mathematically as:

$$\underline{\beta}(k) = \underline{\beta}(k-1) \tag{2.59}$$

$$z(k) = \underline{d}^T(k) \; \underline{\beta}(k) + \theta_o(k) \; u(k) \tag{2.60}$$

with a priori statistics

$$E \left[u(k)\right] = 0 \; ; \; E \left[u(k) \; u(j)\right] = \delta_{jk} \tag{2.61}$$

(2.59) and (2.60) are suitable for Kalman filtering to estimate the parameter vector $\underline{\beta}$. No redundant operation exists in the algorithm because of the elimination of the state vector.

When implementing the model (2.59) and (2.60) to both canonical Kalman filter and Finite Memory Kalman filter in section 3.4, it was found that there are difficulties in estimating the moving average parameters to a satisfactory level in most cases. It is suspected that this is due to the interaction between the poles and zeroes. This deficiency is described further in Chapter 3. Due to this problem, an AR model was adopted to approximate the ship rolling instead. Assuming an ARMA model to be invertable, it can be written as an infinite order autoregressive model, such that:

$$\theta^{-1} (B) \;\; \phi(B) \;\; z(k) = u(k) \tag{2.62}$$

or: $$z(k) = -\sum_{i=1}^{\infty} \psi_i \, z(k-i) + \theta_o \, u(k) \tag{2.63}$$

$$= -\sum_{i=1}^{p} \psi_i \, z(k-i) + \theta_o \, u(k) + \varepsilon \tag{2.64}$$

Since ψ_i forms a convergent series, ε may be made arbitrarily small for a finite p [8]. The value of p to be used can be determined by using an AR(p) model to track a time series of ship rolling motion such that the mean squares error will be minimum. In fact, from the results in Chapter 6, the mean squares error decreases exponentially with p, as in Figure 2.8, when an AR(p) model is used to track an ARMA(6,6) model. Thus, there is a limit on p that no significant improvement will be achieved by going to a higher order.

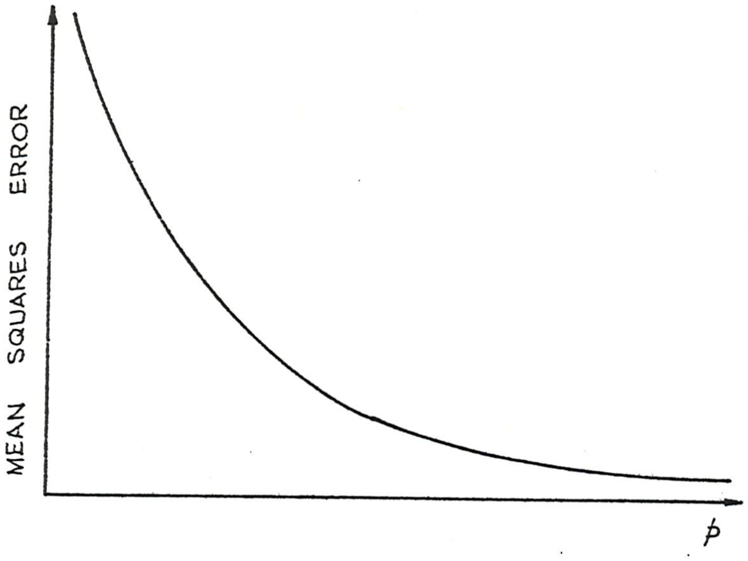


Figure 2.8 Typical Relationship between Mean Squares error and order p.

2.8 Measurement Noise in ARMA Models

So far, the ARMA and AR models mentioned in last section have not explicitly included the measurement noise. In fact, the effect of measurement noise is in general included in the parameters of the models. Considering a process with two white noise inputs $u_1(k)$ and $u_2(k)$. Each of these inputs is filtered by the following models to give outputs $z_1(k)$ and $z_2(k)$, respectively:

$$\phi_1(B) \quad z_1(k) \quad = \quad \theta_1(B) \quad u_1(k) \tag{2.65}$$

$$\phi_2(B) \quad z_2(k) \quad = \quad \theta_2(B) \quad u_2(k) \tag{2.66}$$

$z_1(k)$ may be considered to be the signal and $z_2(k)$ to be coloured noise. The sum of $z_1(k)$ and $z_2(k)$ is the observed signal plus noise, and is given by:

$$y(k) \quad = \quad z_1(k) \quad + \quad z_2(k) \tag{2.67}$$

Substituting for $z_1(k)$ and $z_2(k)$ from (2.65) and (2.66) into (2.67) yields:

$$y(k) \quad = \quad \phi_1^{-1}(B) \quad \theta_1(B) \quad u_1(k) \quad + \quad \phi_2^{-1}(B) \quad \theta_2(B) \quad u_2(k)$$

$$\phi_1(B) \quad \phi_2(B) \quad y(k) \quad = \quad \phi_2(B) \; \theta_1(B) \quad u_1(k) \quad + \quad \phi_1(B) \; \theta_2(B) \quad u_2(k) \tag{2.68}$$

The right hand side of (2.68) forms the sum of two moving average processes which may be represented by a single moving average process allowing (2.68) to be written as:

$$\phi_3(B) \; y(k) \quad = \quad \theta_3(B) \quad u_3(k) \tag{2.69}$$

where $u_3(k)$ is another white noise process. Therefore the additive noise $u_2(k)$ is included in the coefficients of $\phi_3(B)$ and $\theta_3(B)$. Consequently, there is no difference between signal noise and measurement noise. The above fact is shown by Box and Jenkins [8] in more detail.

2.9 Polynomial as Stochastic Model

Following the same procedure as in Section 2.4, the polynomial model (2.19) and (2.20) with time shift (2.22) can be written as:

$$\begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix}_{(k+1)T} = \begin{pmatrix} 1 & T & T^2 \\ 0 & 1 & 2T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix}_{kT} \tag{2.70}$$

$$y((k+1)T) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix}_{(k+1)T} + v((k+1)T) \tag{2.71}$$

Introducing plant noise to (2.70):

$$\begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix}_{(k+1)T} = \begin{pmatrix} 1 & T & T^2 \\ 0 & 1 & 2T \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix}_{kT} + \begin{pmatrix} \omega_o \\ \omega_1 \\ \omega_2 \end{pmatrix}_{kT} \tag{2.72}$$

where ω_o, ω_1 and ω_2 are mutually independent gaussian white noise processes.

Transforming (2.71) and (2.72) to Z-format:

$$\left[I - z^{-1} \begin{pmatrix} 1 & T & T^2 \\ 0 & 1 & 2T \\ 0 & 0 & 1 \end{pmatrix} \right] \begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix}_z = \begin{pmatrix} \omega_o \\ \omega_1 \\ \omega_2 \end{pmatrix}_z \tag{2.73}$$

$$y_z = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix}_z + v_z \tag{2.74}$$

From (2.73)

$$\begin{pmatrix} a_o \\ a_1 \\ a_2 \end{pmatrix} = \left[I - z^{-1} \begin{pmatrix} 1 & T & T^2 \\ 0 & 1 & 2T \\ 0 & 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} \omega_o \\ \omega_1 \\ \omega_2 \end{pmatrix}_z$$

$$= \begin{pmatrix} 1-z^{-1} & -Tz^{-1} & -T^2z^{-1} \\ 0 & 1-z^{-1} & -2Tz^{-1} \\ 0 & 0 & 1-z^{-1} \end{pmatrix}^{-1} \begin{pmatrix} \omega_o \\ \omega_1 \\ \omega_2 \end{pmatrix}_z \tag{2.75}$$

$$\det \begin{pmatrix} 1-z^{-1} & -Tz^{-1} & -T^2z^{-1} \\ 0 & 1-z^{-1} & -2Tz^{-1} \\ 0 & 0 & 1-z^{-1} \end{pmatrix} = (1 - z^{-1})^3 \qquad (2.76)$$

Therefore:

$$\begin{pmatrix} 1-z^{-1} & -Tz^{-1} & -T^2z^{-1} \\ 0 & 1-z^{-1} & -2Tz^{-1} \\ 0 & 0 & 1-z^{-1} \end{pmatrix}^{-1}$$

$$= \frac{1}{(1-z^{-1})^3} \begin{pmatrix} (1-z^{-1})^2 & Tz^{-1}(1-z^{-1}) & T^2z^{-1}(1+z^{-1}) \\ 0 & (1-z^{-1})^2 & 2Tz^{-1}(1-z^{-1}) \\ 0 & 0 & (1-z^{-1})^2 \end{pmatrix} \qquad (2.77)$$

Substituting (2.77) into (2.74) yields:

$$(1-z^{-1})^3 \, y_2 = \left((1-z^{-1})^2 \quad Tz^{-1}(1-z^{-1}) \quad T^2z^{-1}(1+z^{-1}) \right) \begin{pmatrix} \omega_o \\ \omega_1 \\ \omega_2 \end{pmatrix} + (1-z^{-1})^3 \, v_z$$

$$= (1-z^{-1})^2 \, \omega_{oz} + Tz^{-1} (1-z^{-1}) \, \omega_{1z} + T^2z^{-1} (1+z^{-1}) \omega_{2z} + (1-z^{-1})^3 \, v_z \qquad (2.78)$$

On the right hand side of (2.78) the terms ω_{oz}, ω_{1z}, ω_{2z} and v_z are all white noises which can be grouped and can be represented by one single moving average process. Thus (2.78) can be viewed as an Integrated moving average process. The coefficients of the equivalent process can be determined by equating the autocovariances of the process. The autocovariances of the right hand side of (2.78) are:

$$\begin{aligned}
\gamma_o &= 6\,\sigma_o^2 + 2\,T^2\,\sigma_1^2 + 2\,T^4\,\sigma_2^2 + 20\,\sigma_v^2 \\
\gamma_1 &= -4\,\sigma_o^2 - T^2\,\sigma_1^2 + T^4\,\sigma_2^2 - 15\,\sigma_v^2 \\
\gamma_2 &= \sigma_o^2 + 2\,\sigma_v^2 \\
\gamma_3 &= -\sigma_v^2 \\
\gamma_j &= 0 \qquad\qquad j \geqslant 4
\end{aligned} \tag{2.79}$$

where

$$\begin{aligned}
\text{Var}\,[\omega_{oz}] &= \sigma_o^2 \\
\text{Var}\,[\omega_{1z}] &= \sigma_1^2 \\
\text{Var}\,[\omega_{2z}] &= \sigma_2^2 \\
\text{Var}\,[v_z] &= \sigma_v^2
\end{aligned}$$

Since the highest order among the four moving average processes on the right hand side of (2.78) is three the equivalent single moving average process is of the order three [8]. Assuming the equivalent single moving average process is

$$y'(k) = (\theta_o + \theta_1 z^{-1} + \theta_2 z^{-2} + \theta_3 z^{-3}) \quad a(k) \tag{2.80}$$

with a(k) being a white noise process having the statistics of

$$\begin{aligned}
E\,\left[a(k)\right] &= 0 \\
\text{Var}\,\left[a(k)\right] &= 1.
\end{aligned}$$

The autocovariances of the equivalent process (2.80) are:

$$\begin{aligned} \gamma_o' &= \theta_o^2 + \theta_1^2 + \theta_2^2 + \theta_3^2 \\ \gamma_1' &= \theta_o\theta_1 + \theta_1\theta_2 + \theta_2\theta_3 \\ \gamma_2' &= \theta_o\theta_2 + \theta_1\theta_3 \\ \gamma_3' &= \theta_o\theta_3 \\ \gamma_j' &= 0 \qquad j \geqslant 4 \end{aligned} \tag{2.81}$$

The coefficients θ_o, θ_1, θ_2, and θ_3 can then be evaluated by equating (2.79) and (2.81).

The above analysis indicates that a polynomial model with its parameters as random variables is equivalent to an integrated moving average process. An m-order polynomial can be treated as an IMA(O,m,m) process. Since an integrated moving average process can be used with non-stationary time series [8] it is also true in the case of polynomial models.

2.10 Conclusions

A constant acceleration model was proposed for the target motion. Basically the constant acceleration model and the quadratic polynomial model are identical but are presented in different forms. Thus, the modelling of the target motion can be visualised as a quadratic curve fitting exercise.

The proposed model imposes an assumption of constant acceleration. This assumption can, however, be relaxed by windowing the measurements. The proposed model requires no a priori knowledge about the target as opposed to the constant velocity model advocated by Singer. The constant acceleration model can then be a truely general model.

The case of constant acceleration model with plant noise introduced was dropped by two reasons. The variance of the plant noise must be related to the characteristics of the target being tracked. Thus, an a priori knowledge of the target, which is assumed unavailable, is required to establish the required variance.

Secondly, the introduction of plant noise and the use of windowing both have similar effect on the parameter estimation process, namely preventing the covariance matrix of the estimation error from being zero.

Furthermore, it is easier to interpret the physical meaning of a window than the variance of a white noise.

A stochastic AR model was proposed for the ship rolling motion. In the early stage, it was thought that, based on the knowledge of the spectrum of the motion, an ARMA(6,6) process will fit the ship rolling motion fully. However, in the process of estimating the parameters using a recursive filtering algorithm, such as Kalman filter, a suspect poles and zeroes interaction causes difficulties in estimating the moving average parameters in the ARMA model. A sixth order AR model was demonstrated to be adequate in estimating an ARMA(6,6) model in the simulation trials. Windowing is also suggested in the modelling of ship motion because of the non-stationary characteristic of the motion. It must be stressed that the order of the AR model must be chosen based on experiments with actual ship rolling motion. Throughout the work of this thesis, the ship rolling motion is assumed to be generated by passing a white noise process through three parallel second order linear filters as in Figure 2.7.

From the derivation in Section 2.9, it was found that a polynomial model may actually be viewed as an integrated moving average model. Hence, a non-stationary process may actually be modelled using polynomial fitting. This confirms further that polynomial models can cover all types of events.

CHAPTER 3

FINITE MEMORY FILTERING

Every measurement will inevitably be corrupted with unwanted noise. This noise ought to be filtered out in some way, and furthermore, the parameters of the models need to be up-dated frequently. In modelling the two input signals: target trajectory and ship rolling motion, low order models are used. Also, the parameters may be time varying over long periods of time. Therefore, the filtering algorithm that is to be employed must be capable of rejecting the noise and give an estimate of the parameters. More important, the algorithm must be in a recursive form because this will require less computation time which best suits real-time applications.

The well established Kalman Filter named after R.E. Kalman [25, 26] who developed it in the 1960's is an optimal linear filter algorithm. In fact, many other filters algorithms, e.g. Bayes filter algorithm, the Least Squares estimator, the minimum variance estimator, etc., can be treated as equivalent to the Kalman filter. However, the original Kalman filter gives an estimate based on the whole infinite past history of sampled signals. Thus a modified version of the Kalman filter that only uses finite past data was required for this work.

3.1 Model for Random Processes

Let $\underline{x}(k)$ be the random state vector of time kT, where T is an arbitrary sample time. $\underline{x}(k)$ is assumed to obey the following recursive relationship.

$$\underline{x}(k+1) = A(k+1,k)\ \underline{x}(k) + B(k)\ \underline{u}(k) \tag{3.1}$$

where $\underline{x}(.)$ is nxl vector, $\underline{u}(.)$ is pxl vector. A(k+1,k) is a nxn transition matrix, B(.) is a nxp control matrix.

The measurement at time (k+1)T is denoted by:

$$\underline{z}(k+1) = D(k+1)\ \underline{x}(k+1) + \underline{v}(k+1) \tag{3.2}$$

Here $\underline{z}(.)$ and $\underline{v}(.)$ are mxl vectors and D(.) is a mxn observation matrix. $\underline{v}(.)$ is a gaussian random variable with mean value zero. i.e.

$$E\left[\underline{v}(k)\right] = 0 \tag{3.3}$$

and $E[\underline{v}(k)\ \underline{v}^T(j)] = R(k)\ \delta_{jk}$ (3.4)

where; $\delta_{jk} = \begin{cases} 0 & ; \quad j \neq k \\ 1 & ; \quad j = k \end{cases}$ (3.5)

The measurement noise, $\underline{v}(.)$ is assumed to be uncorrelated with the state vector, $\underline{x}(.)$. The initial state vector is assumed to be random with a guassian distribution and is independent to the measurement noise.

3.2 Statement of Problem

The problem is to estimate the value of the state vector $\underline{x}(k)$ in the model (3.1) given the data from time k-ℓ up to the current time. i.e. z(k), $\underline{z}(k-1)$, $\underline{z}(k-2)$, . . ., $\underline{z}(k-\ell)$ and $\underline{u}(k-1)$, u(k-2), . . u(k-ℓ-1), where ℓ+1 is the number of data for estimation (Figure 3.1). The notation $\hat{\underline{x}}(k/j,\ell)$ is used to mean the estimate of $\underline{x}(k)$ based on ℓ+1 data points from time instant j-ℓ upto and including the time j. The time k will always be greater than or equal to j. i.e. the state vector is estimated at current time and is predicted into the future.

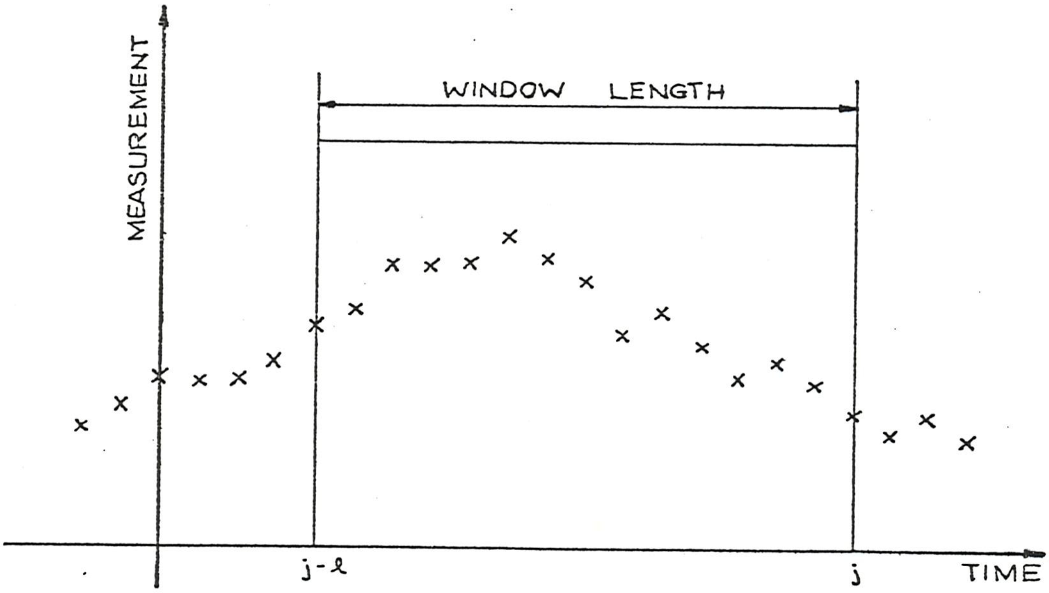


Figure 3.1 Windowing of Time Series

The developed estimator should be a linear combination of the measurements and would give minimum variance of error. In other words, it is going to be a linear minimum variance estimator.

3.3 Finite Memory Kalman Filter

Based on the idea of adding one new measurement and dropping one old measurement from the estimate simultaneously a finite memory version of the Kalman filter can then be developed either from a minimum variance approach or from a Least Squares approach, etc. The algorithm of the Finite Memory Kalman filter is stated without proof as follows:

$$\underline{\hat{x}}(k+1/k,\ell) = A(k+1,k)\ \underline{\hat{x}}(k/k,\ell) + B(k)\ \underline{u}(k) \tag{3.6}$$

$$P(k+1/k,\ell) = A(k+1,k)\ P(k/k,\ell)\ A^T(k+1,k) \tag{3.7}$$

$$G(k+1) = P(k+1/k,\ell)\ M^T(k+1)\ \left[M(k+1)\ P(k+1/k,\ell)\ M^T(k+1) + S(k+1)\right]^{-1} \tag{3.8}$$

$$\underline{\hat{x}}(k+1/k+1,\ell) = \left[I - G(k+1)\ M(k+1)\right]\ \underline{\hat{x}}(k+1/k,\ell) + G(k+1)\ U(k+1) \tag{3.9}$$

$$P(k+1/k+1,\ell) = \left[I - G(k+1)\ M(k+1)\right]\ P(k+1/k,\ell) \tag{3.10}$$

with
$$M(k+1) = \begin{pmatrix} D(k+1) \\ D(k-\ell)\ A(k-\ell,\ k+1) \end{pmatrix} \tag{3.11}$$

$$S(k+1) = \begin{pmatrix} R(k+1) & 0 \\ 0 & -R(k-\ell) \end{pmatrix} \tag{3.12}$$

$$U(k+1) = \begin{pmatrix} \underline{z}(k+1) \\ \underline{z}'(k-\ell) \end{pmatrix} \tag{3.13}$$

$$\underline{z}'(k-\ell) = \underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k+1) \tag{3.14}$$

and
$$\begin{aligned}\beta(k-\ell,\ k+1) &= A(k-\ell,\ k-\ell-1)\ \beta(k-\ell-1,k) \\ &+ A(k-\ell,\ k+1)\ B(k)\ \underline{u}(k) \\ &- B(k-\ell-1)\ \underline{u}(k-\ell-1)\end{aligned} \tag{3.15}$$

The term $\beta(k-\ell,\ k+1)$ arises from the projection of the new estimate back to the time of the old measurement being dropped out of the window. The whole algorithm preserves the appearance of the canonical Kalman filter, but with the sizes of the variance matrix S(.) and the vectors U(.) and M(.) being doubled. The whole derivation can be found in Appendix B. The block diagram of the filter is shown in Figure 3.2a.

For completeness, the canonical form of the Kalman filter for the same set of system and observation equations (3.1) and (3.2) is given below:

$$\underline{\hat{x}}(k+1/k) = A(k+1,k)\ \underline{\hat{x}}(k/k) + B(k)\ \underline{u}(k) \tag{3.16}$$

$$P(k+1/k) = A(k+1,k)\ P(k/k)\ A^T(k+1,k) \tag{3.17}$$

$$K(k+1) = P(k+1/k)\ D^T(k+1) \left[D(k+1)\ P(k+1/k)\ D^T(k+1) + R(k+1) \right]^{-1} \tag{3.18}$$

$$\underline{\hat{x}}(k+1/k+1) = \left[I - K(k+1)\ D(k+1) \right] \underline{\hat{x}}(k+1,k) + K(k+1)\ \underline{z}(k+1) \tag{3.19}$$

$$P(k+1/k+1) = \left[I - K(k+1)\ D(k+1) \right]\ P(k+1/k) \tag{3.20}$$

The block diagram of the canonical Kalman filter is shown in Figure 3.2b. By comparison, if the vector $D(k-\ell)$ in (3.11) is set to be a null vector, both algorithms give the same result.

3.4 Experimental Results

3.4.1 Implementation on Deterministic Model.

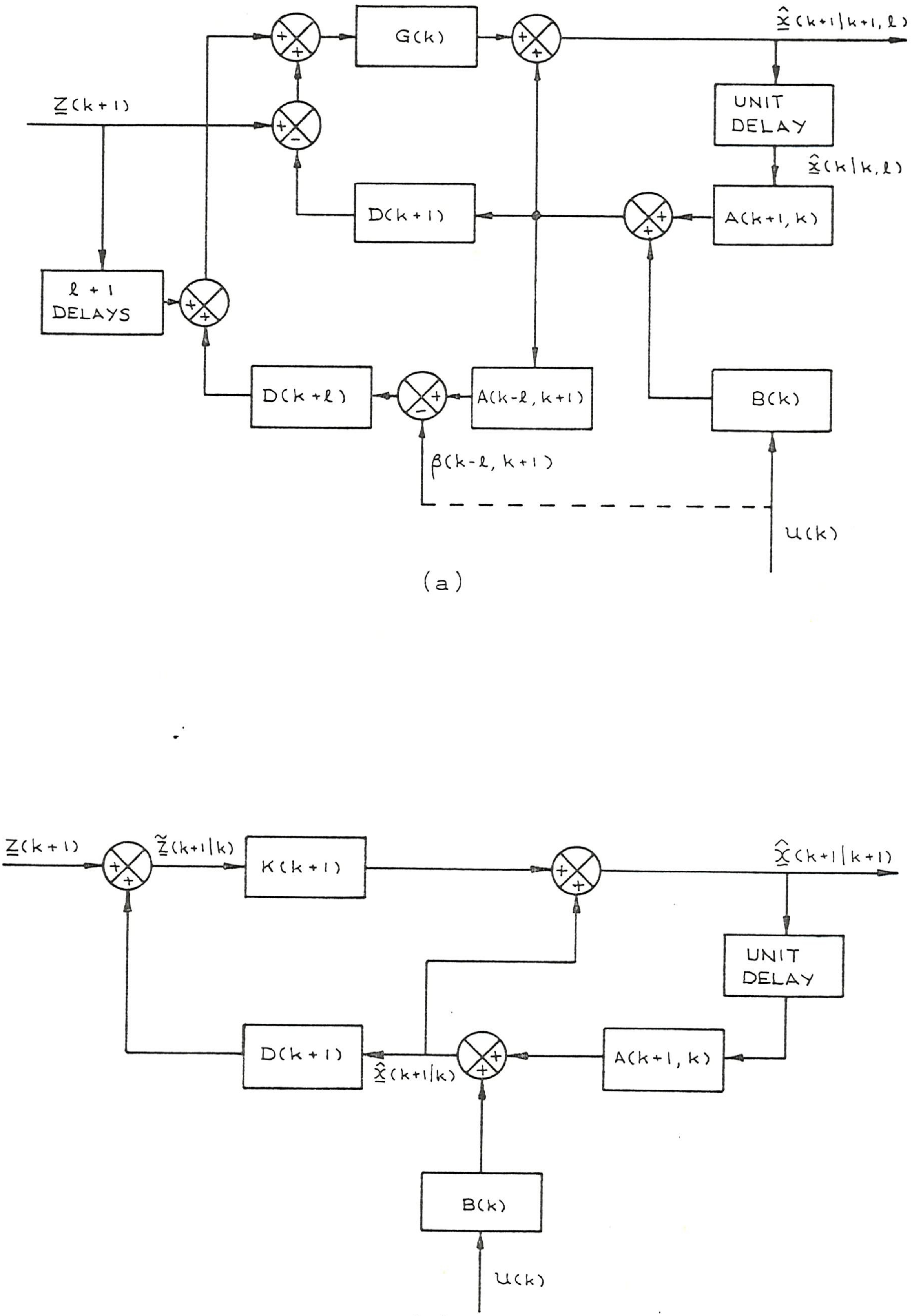
The developed Finite Memory Kalman Filter algorithm was tried out as a parameter estimator. The parameters, a_o, a_1 and a_2 of a quadratic polynomial

$$y(t) = a_o + a_1 t + a_2 t^2 \tag{3.21}$$

were estimated from the measurements:

$$z(t) = y(t) + v(t) \tag{3.22}$$

where the measurement noise, v(t), has a variance, R, of 10.0.



(a)

(b)

Figure 3.2 Block diagram of Kalman Filter
(a) Finite Memory (b) Canonical

In the experiment, (3.21) and (3.22) were re-written in the vector form:

$$\underline{x}(k+1) = \underline{x}(k) \tag{3.23}$$

$$z(k+1) = d^T(k+1) \; \underline{x}(k+1) + v(k+1) \tag{3.24}$$

where $$\underline{x}^T(k+1) = (\; a_o \quad a_1 \quad a_2 \;)_{k+1} \tag{3.25}$$

$$\underline{d}^T(k+1) = (\; 1 \quad (k+1) \quad (k+1)^2 \;) \tag{3.26}$$

For the first ℓ+1 measurements, a canonical form of Kalman filter was used. From the (ℓ+2)th measurements onwards, the estimation was switched to the Finite Memory Kalman Filter. In this case the complexity of the algorithm has been reduced due to the simplicity of (3.23) - the transition matrix is an identity matrix. The algorithm becomes:

$$G(k+1) = P(k/k,\ell) \; M^T(k+1) \; \left[M(k+1) \; P(k/k,\ell) \; M^T(k+1) + S(k+1)\right]^{-1} \tag{3.27}$$

$$\underline{\hat{x}}(k+1/k+1,\ell) = \left[I - G(k+1) \; M(k+1)\right] \; \underline{\hat{x}}(k/k,\ell) + G(k+1) \; U(k+1) \tag{3.28}$$

$$P(k+1/k+1,\ell) = \left[I - G(k+1) \; M(k+1)\right] \; P(k/k,\ell) \tag{3.29}$$

with $$M(k+1) = \begin{pmatrix} \underline{d}^T(k+1) \\ \underline{d}^T(k-\ell) \end{pmatrix} \tag{3.30}$$

$$S(k+1) = \begin{pmatrix} R(k+1) & 0 \\ 0 & -R(k+1) \end{pmatrix} \tag{3.31}$$

$$U(k+1) = \begin{pmatrix} z(k+1) \\ z(k-\ell) \end{pmatrix} \tag{3.32}$$

However (3.26) is undesirable for computation since the term $(k+1)^2$ will soon become exceptionally large. The parameter vector was then time shifted after each cycle of the whole algorithm

$$\underline{x}'(k+1) = C \; \underline{x}(k+1) \tag{3.33}$$

where $\underline{x}'(k+1)$ = time-shifted state vector,

and

$$C = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.34)$$

The matrix C in (3.33) is the same as the matrix S in (2.28). In each time-shift operation, the covariance matrix has to be up-dated, too.

$$P'(k+1/k+1,\ell) = C \; P(k+1/k+1,\ell) \; C^T \quad (3.35)$$

With the help of the time-shift operation, (k+1) and $(k+1)^2$ remain constant throughout the whole simulation trial.

The estimate, a_o, a_1 and a_2 are plotted against time in Figures 3.4, 3.5 and 3.6. In Figure 3.4, it clearly shows that the estimation with a memory length of a hundred data points adapts to sudden changes in parameters easily whereas the one with memory length of four hundred data points shows difficulty in accommodating the step changes in a_o. The results reveal that a Finite Memory Kalman Filter with suitable window length gives good estimation even with a model that does not anticipate any external disturbances. However, the uncertainty associated with the estimates is highly dependent upon the window length, which can be observed from Figures 3.10, 3.11 and 3.12. If the window length is so short that it is in the same order as the degrees of freedom in the estimation process, the uncertainty will be unacceptable.

The results from the above experiment are also applicable to the proposed constant acceleration model since the polynomial model is exactly the same as the proposed model. By treating the state vector as the coefficients of the polynomial, the true state vector will then be invariant with time. It then allows a clearer view in the performance of the estimator during the whole experiment. If the constant acceleration model was used, the first two elements in the state vector will be time dependent and will become very large eventually.

The performance of the parameter estimator may be improved by making the window length adaptive to sudden target manoeuvres and to unmatching model order (Figure 3.3). If target menoeuvres exist , shorter window length will accelerate the adaption of parameter changes. If the model order is too low, once again a shorter window length will remedy the deficiency. The criterion used in the adaptor may be to minimise the mean square error of the estimator by means of the windowing length. If a longer window length is desired, a canonical Kalman Filter will be implemented instead of the finite memory version. If a shorter window length is required, an algorithm for dropping old measurements only (such as one in Appendix B), will be used together with the finite memory Kalman Filter.

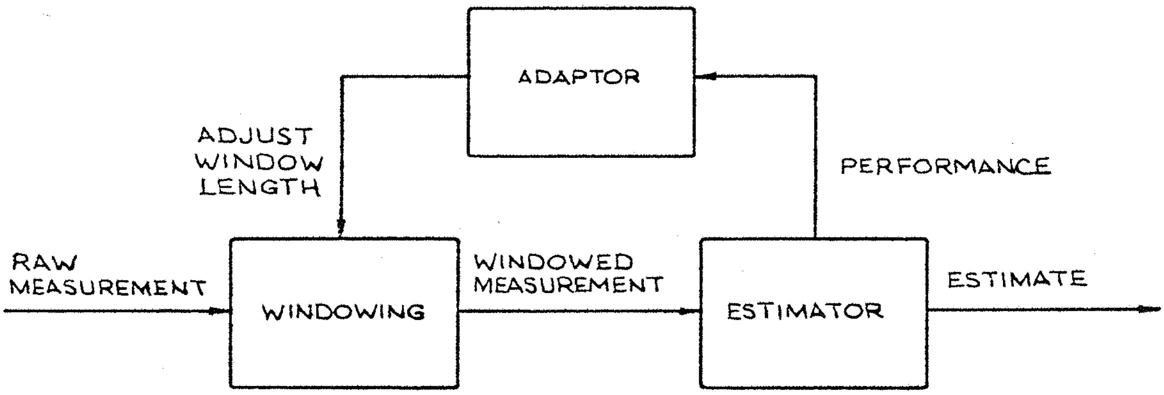


Figure 3.3 Adaptive Window Length.

3.4.2 Implementation of ARMA Models

Restating the state-space form of an ARMA(m,n) model (2.59) and (2.60) here:

$$\underline{\beta}(k) = \underline{\beta}(k-1) \qquad (3.36)$$

$$z(k) = \underline{d}^T(k)\ \underline{\beta}(k) + \theta_o(k)\ u(k) \qquad (3.37)$$

$\underline{\beta}$ and $\underline{d}$ are (m+n)x1 vectors. z, θ_o and u are scalar.

(3.36) and (3.37) are similar to (3.23) and (3.24). Hence, the Finite Memory Kalman Filter algorithm may be applied to estimate the parameter vector . It is shown by Box and Jenkins [8] that the input u(k) can be obtained from the one-step ahead prediction error, i.e.

$$\hat{\theta}_o(k)\ u(k) = z(k) + \sum_{i=1}^{m} \hat{\phi}_i(k)\ z(k-i) - \sum_{i=1}^{n} \hat{\theta}_i(k)\ u(k-i) \tag{3.38}$$

Therefore:

$$u(k) = \frac{1}{\hat{\theta}_o(k)}\ \varepsilon(k) \tag{3.39}$$

where;

$$\varepsilon(k) = z(k) + \sum_{i=1}^{m} \hat{\phi}_i(k)\ z(k-i) - \sum_{i=1}^{n} \hat{\theta}_i(k)\ u(k-i) \tag{3.40}$$

The parameter $\theta_o(k)$ is estimated from (3.39) using the assumption that u(k) is white and with unit spectral height.

$$E\left[\left|\ \hat{\theta}_o(k)\ u(k)\right|^2\right] = \hat{\theta}_o^{\,2}(k) = E\left[\varepsilon^2(k)\right] \tag{3.41}$$

where $E\left[\varepsilon^2(k)\right]$ can be computed as $E\left[\varepsilon^2(k)\right] = \frac{1}{\ell} \sum_{i=o}^{\ell} \varepsilon^2(k-i)$

(3.42)

$(\ell+1)$ is the number of data points that the estimate $\hat{\underline{\beta}}$ and $\hat{\theta}_o$ are based on. The estimation algorithm as (3.27), (3.28) and (3.29).

$$\hat{\underline{\beta}}(k+1/k+1,\ell) = \left[I - G(k+1)\ M(k+1)\right]\ \hat{\underline{\beta}}(k/k,\ell) + G(k+1)\ U(k+1) \tag{3.43}$$

$$G(k+1) = P(k/k,\ell)\ M^T(k+1)\ \left[M(k+1)\ P(k/k,\ell)\ M^T(k+1) + S(k+1)\right]^{-1} \tag{3.44}$$

$$P(k+1/k+1,\ell) = \left[I - G(k+1)\ M(k+1)\right]\ P(k/k,\ell) \tag{3.45}$$

with $$S(k+1) = \begin{pmatrix} \hat{\theta}_o^{2}(k+1) & 0 \\ 0 & -\hat{\theta}_o^{2}(k-\ell) \end{pmatrix} \tag{3.46}$$

These, in conjunction with (3.39) and (3.42), form a complete algorithm for identifying the parameters of the ARMA model.

This formation of ARMA model, (3.36) and (3.37) was first appeared in Srinath and Rejacekaran [46]. They originally gave one example employing the canonical Kalman filter to estimate the parameters. It has been extended here to the Finite Memory Kalman Filter. The combination of (3.36), (3.37) and Kalman Filter is the most efficient one. Box and Jenkins [8] suggested an algorithm which is not suitable for on-line implementation and requires human interference and pattern recognition. Lee [29] and Gersch [19] attacked the problem of estimating the autoregressive parameters of a mixed ARMA model. However no attempt was made by either to estimate the moving-average parameters. Mehra [33], Krause and Graupe [28] estimate the parameters by forming the ARMA model to the state-space form as:

$$\underline{x}(k+1) = A(k+1,k)\ \underline{x}(k) + B \begin{vmatrix} u\ (k+1) \\ \underline{u}\ (k) \end{vmatrix} \qquad (3.47)$$

and

$$z(k+1) = \underline{d}^T\ \underline{x}(k+1) \qquad (3.48)$$

where the parameters of the ARMA model are in the matrices A and B. The estimation becomes a non-linear problem.

The algorithm, (3.43), (3.44) and (3.45) was used on several types of ARMA processes both in the expanding memory form and finite memory form. The estimates are all plotted out in Figures 3.13 to 3.34. The mean absolute percentage error (MAPE) are listed in Tables 3.1 and 3.2. Srinath and Rejacekaran [46] gave only an example of a simple ARMA(2.1) as the one in Figure 3.15. They showed that the estimates are consistent. The estimates converge to the true value with probability one. From the results, it is clear that convergence is guaranteed. The rate of convergence is, however, linked to the individual model. The most disturbing discovery was from the four ARMA(2.1) processes. It was found that if the poles and zeroes are located on the same side of the imaginary axis on the z-plane, the rate of convergence is terribly slow, (Figures 3.19, 3.22). This has the implication that the adaptation to changes in parameters will be poor. When the same processes used in finite memory algorithm the estimates fluctuated widely around the true values (Figures 3.28, 3.32). One possible explanation to this phenomenon is there exists some sort of pole and zero interaction. Therefore, from a numerical estimation point of view, it concluded that the ARMA process is not a suitable choice for modelling the stochastic ship rolling even though it is parsimonious. It is then envisaged that a higher order AR model may be more suitable, to approximate the parsimonious ARMA model.

3.5 Conclusions

A finite memory version of the Kalman filter is presented in this chapter. The finite memory version resembles the canonical Kalman filter. It possesses the merits of the Kalman filter and removes some problems inherited with the Kalman filter.

It is noted that both the canonical Kalman filter and the finite memory version shown in Figure 3.2 closely resemble the original structure of the system model, (3.1) and (3.2). In the case of the canonical Kalman filter, it is driven by the innovation $\underline{\tilde{z}}(k+1/k)$. In the finite memory version case, it is driven by two innovations, $\underline{\tilde{z}}(k+1/k,\ell)$ and $\underline{\tilde{z}}(k-\ell/k,\ell)$. Both $\underline{\tilde{z}}(k+1/k)$ and $\underline{\tilde{z}}(k+1/k,\ell)$ are identical. In both algorithms, the estimates are up-dated with the prediction together with the weighted innovations.

The Finite Memory Kalman Filter has its own defects. Firstly, if a window of ℓ data points is planned, the first ℓ measurements must be filtered with a canonical Kalman filter or some other filtering algorithms. It is because in the first ℓ measurements, no old measurement is to be discarded. It must be stressed that initialising the old measurements with zeroes cannot be used during the first ℓ measurements due to the properties of the gain matrix and the covariance matrix described in later paragraphs. During the first ℓ measurements, the Finite Memory Kalman filter may still be used only by setting the terms in $D(k-\ell)$ in (3.11) to zero. This arrangement turns the filter to a canonical Kalman filter. However, a high proportion of the computation time is wasted in multiplying numbers with zero. From the $(\ell+1)$th measurement onwards the normal Finite Memory Kalman filter can then be used. The second imperfection of the Finite Memory Kalman filter is the inability of removing old measurement alone. It is because the algorithm is derived based on the one step prediction. The closely related algorithm for removing old measurements only is in a slight different form and is presented in Appendix B. This measurement removal algorithm is very useful if adaptive windowing is employed.

The gains, K(k+1) and G(k+1), of both algorithm depend on the signal to noise ratio. In general, the gain decreases as the measurement noise variance increases. This result is intuitively appealing, since as R(.) increases, the gain will decrease in order to avoid introducing excessive measurement noise into the estimate. In the case of the canonical Kalman Filter, it can be shown that [40] the gain, K(j), asymptotically approaches zero for large j. As K(.) approaches zero, the error covariance matrix, P(.), appraoches zero, too. Eventually the estimation procedure becomes decoupled from, or independent of, the measurement. It enters a condition known as data saturation. This condition can lead to serious divergence problems. One of the major causes of divergence is inaccuracies in the modelling process used to determine the system and measurement model, due to failure of linearisation, lack of complete knowledge of the physical problem or the simplifying assumptions necessary for mathematical tractability. Errors in the statistical modelling of noise variances and means or unknown input may also lead to divergence. A plant noise, $\underline{\omega}(.)$, in the system model, introduced either from the numerical or modelling points of view, prevents the occurance of such divergence. The presence of the plant noise inhibits the gain matrix being zero and sets a lower bound on the covariance matrix. The gain matrix, with plant noise added, decreases as the variance of the plant noise, Q(.), decreases. However, it still has the problem of the need to assess the variance of the plant noise. On the other hand, the Finite Memory Kalman filter has a built-in property of stopping the gain and covariance matrices of going to zero due to the term $-R(k-\ell)$ in (3.12). If the measurement noise is stationary, i.e. R is constant over time, the gain matrix and the covariance matrix also will be constant once the Finite Memory Kalman Filter starts, as in the case of the parameter estimation example given in Section 3.4.1 (Figures 3.7 - 3.12). The constant covaraince matrix can be explained from the view of a Least Square estimation method since Kalman Filter is also a least square estimator. The covariance depends only on the number of available data points. The more data points, the more accurate the estimate.

Another source of divergence is round-off errors, inherent in any digital implementation of the filter algorithm, which may cause the error variance matrix to lose its positive definiteness or symmetry. This sort of divergence is still associated with the Finite Memory Kalman Filter. One possible way to get around of this sort of divergence is to form an expression calculating the square root of the covariance matrix instead of the actual covarinace matrix [37, 40]. Where the covariance matrix is required in the algorithm, the square of the square root covariance matrix is used which guarantees symmetry.

The Finite Memory Kalman filter is superior to the canonical Kalman filter with plant noise incorporated in the system model on the basis that it adapts to any changes in the state which is not anticipated in the system model more easily. In Figures 3.4 and 3.5 it can be found that the Finite Memory Kalman filter adapts to the new parameter quickly after one window length. This shows that the algorithm actually gives estimates based on the measurements inside the window. One might argue that the canonical Kalman filter with appropriate plant noise in the system model can have the same, or even better, response to the changes. However, if the plant noise has a large variance the response to sudden changes is fast, but with penalties of high uncertainty and large proportions of measurement noise on the estimate. On the other hand, if the plant noise variance is low, the opposite is true. There is no guidelines on setting the variance.

Due to the incapability of the Finite Memory Kalman filter to start the estimation process, the algorithm is therefore powerless to remedy the error introduced in the initial guess of the state. There are ways of minimizing, or eliminating, the initial error. If the canonical Kalman filter is used to start the process, the initial error effect can be minimized by using very large initial covariance matrix implies there is no confidence on the initial guess of the state. However, if the window length is short compared with the time constant of the canonical Kalman Filter, the covariance matrix may still be artifically very high in entering the Finite Memory Kalman Filter. This may then affect the performance of the algorithm. The best way to eliminate the initial error is to apply either a Bayes Filter [35] or a non-recursive Least Squares Estimator on the first few measurements to provide a better estimate before starting the Kalman filter. Although the Bayes Filter is essentially the same as the Kalman Filter [35], the formulation of the Bayes Filter allows it to be started with no a priori data.

The developed Finite Memory filtering is not the only way of tackling the problems of mismatching models and sudden changes in model parameters. A Fading Memory Kalman filter described by Morrison [35] has a similar effect as the Finite Memory version. The Fading Memory Kalman Filter is derived from minimising the cost.

$$\underset{\underline{x}(k/k)}{\text{Min}}\ J = \underset{\underline{x}(k/k)}{\text{Min}} \sum_{i=o}^{k-1} \underline{U}^{T}(k-i)\ R^{-1}(k-i)\ \underline{U}(k-i)\ \beta^{t_k - t_{k-i}} \quad (3.49)$$

where

$$\underline{U}(k-i) = \underline{z}(k-i) - D(k-i)\ A(k-i,k)\ \underline{x}(k/k) \quad (3.50)$$

The characteristic of this cost function is the term $\beta^{t_k - t_{k-i}}$ where $0<\beta\leqslant 1$. The observation $\underline{z}(k-i)$ is being made to enter into the estimate of $\underline{x}(k/k)$ with an importance depending on its staleness (i.e. the elapsed time $(t_k - t_{k-i})$ since $z(k-i)$ was obtained). The introduction of the stress function $\beta^{t_k - t_{k-i}}$ with $\beta<1$ means that thememory length does not expand steadily as in the case of the canonical Kalman filter, but that the observations are forgotten or discounted at a rate exponentially proportional to their age. At the extreme when $\beta=1$, the Fading Filter becomes a canonical Kalman filter. At the other extreme, when $\beta=0$, the estimate is based purely on the single most recent observation. This then suggests that by making β sufficiently small the effective memory length can be kept sufficiently short. However, the exact window length is not available.

Both the Fading Memory Filter and the Finite Memory Filter have their usefulness and weakness. The Finite Memory Kalman Filter cannot work with system models having a plane noise term. This is due to the inability to establish the correlations between the estimation error and the plant noise in the past. However, it can be certain that any error arises from a sudden change in the state is completely eliminated from the estimates after a time equivalent to one window length. The Fading Memory Filter, on the other hand, cannot provide the same guarantee on the estimates but can work with models having plant noise in it.

The most intriguing result from the experiments was the behaviour of the algorithms in estimating the parameters of ARMA models. It was found that the moving average part, $\hat{\theta}$'s, were in general more difficult to converge to the true value than their counterparts $\hat{\phi}$'s. Moreover, the positions of the poles and zeroes of the ARMA models do play a part in the performance of the algorithms. It suggests that there are pole and zero interactions in the algorithm. Nevertheless, the results indicated that the Finite Memory Kalman Filter is a useful tool with a lot of potential.

FILTER ALGORITHM	FIGURE	MODEL	MEAN ABSOLUTE PERCENTAGE ERRORS AT SAMPLE INTERVAL				
			260	1,000	1,500	15,000	30,000
CANONICAL KALMAN FILTER	3.13	$\frac{y}{u}(Z) = \frac{1.0}{1.0 + 0.5Z^{-1}}$	3.73	3.04	1.44	-	-
	3.14	$\frac{y}{u}(Z) = \frac{1.0}{1.0 + 1.5Z^{-1} + 0.625Z^{-2}}$	3.48	0.59	1.98	-	-
	3.15	$\frac{y}{u}(Z) = \frac{1.0}{1.0 - 1.5Z^{-1} + 0.625Z^{-2}}$	5.14	2.22	0.93	-	-
	3.16	$\frac{y}{u}(Z) = 1.0 + 0.5Z^{-1}$	3.69	6.90	3.35	-	-
	3.17	$\frac{y}{u}(Z) = 1.0 + 0.5Z^{-1} + 0.625Z^{-2}$	3.92	7.99	4.72	-	-
	3.18	$\frac{y}{u}(Z) = 1.0 - 0.5Z^{-1} + 0.625Z^{-2}$	4.97	4.15	1.26	-	-
	3.19	$\frac{y}{u}(Z) = \frac{1.0 + 0.5Z^{-1}}{1.0 + 1.5Z^{-1} + 0.625Z^{-2}}$	10.01	27.54	21.94	3.20	2.81
	3.20	$\frac{y}{u}(Z) = \frac{1.0 + 0.5Z^{-1}}{1.0 - 1.5Z^{-1} + 0.625Z^{-2}}$	14.44	5.71	3.47	1.43	0.76
	3.21	$\frac{y}{u}(Z) = \frac{1.0 - 0.5Z^{-1}}{1.0 + 1.5Z^{-1} + 0.625Z^{-2}}$	5.73	6.10	3.89	1.28	1.01
	3.22	$\frac{y}{u}(Z) = \frac{1.0 - 0.5Z^{-1}}{1.0 - 1.5Z^{-1} + 0.625Z^{-2}}$	41.65	33.37	23.76	4.18	1.38
	3.23	$\frac{y}{u}(Z) = \frac{1.0}{1.0 - 1.618Z^{-1} + 2.618Z^{-2} - 2.617Z^{-3} + 2.617Z^{-4} - 1.618Z^{-5} + 0.9997\,Z^{-6}}$	2.64	0.74	0.50	-	-

TABLE 3.1: MEAN ABSOLUTE PERCENTAGE ERRORS FROM CANNONICAL KANMAN FILTER

FILTER ALGORITHM	FIGURE	MODEL	MEAN ABSOLUTE PERCENTAGE ERRORS AT SAMPLE INTERVAL				
			260	1,500	7,000	15,000	30,000
FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH OF 1,000	3.24	$\frac{y}{u}(z) = \frac{1.0}{1.0 + 0.5z^{-1}}$	1.90	1.63	-	-	-
	3.25	$\frac{y}{u}(z) = \frac{1.0}{1.0 + 1.5z^{-1} + 0.625z^{-2}}$	1.50	3.20	-	-	-
	3.26	$\frac{y}{u}(z) = \frac{1.0}{1.0 - 1.5z^{-1} + 0.625z^{-2}}$	2.49	1.27	-	-	-
	3.27	$\frac{y}{u}(z) = 1.0 + 0.5z^{-1}$	5.06	0.78	-	-	-
	3.28	$\frac{y}{u}(z) = 1.0 + 0.5z^{-1} + 0.625z^{-2}$	7.08	4.32	-	-	-
	3.29	$\frac{y}{u}(z) = 1.0 - 0.5z^{-1} + 0.625z^{-2}$	3.21	1.86	-	-	-
	3.30	$\frac{y}{u}(z) = \frac{1.0 + 0.5z^{-1}}{1.0 + 1.5z^{-1} + 0.625z^{-2}}$	33.12	23.42	22.51	13.82	26.74
	3.31	$\frac{y}{u}(z) = \frac{1.0 + 0.5z^{-1}}{1.0 - 1.5z^{-1} + 0.625z^{-2}}$	2.65	2.55	2.53	2.46	4.73
	3.32	$\frac{y}{u}(z) = \frac{1.0 - 0.5z^{-1}}{1.0 + 1.5z^{-1} + 0.625^{-2}}$	7.57	4.93	3.60	5.78	5.69
	3.33	$\frac{y}{u}(z) = \frac{1.0 - 0.5z^{-1}}{1.0 - 1.5z^{-1} + 0.625z^{-2}}$	3.80	10.55	4.44	3.23	13.53
	3.34	$\frac{y}{u}(z) = \frac{1.0}{1.0 - 1.618z^{-1} + 2.618z^{-1} - 2.613z^{-3} + 2.617z^{-4} - 1.618z^{-5} + 0.99971z^{-6}}$	0.94	0.82	-	-	-

TABLE 3.2: MEAN ABSOLUTE PERCENTAGE ERRORS FROM FINITE MEMORY KALMAN FILTER

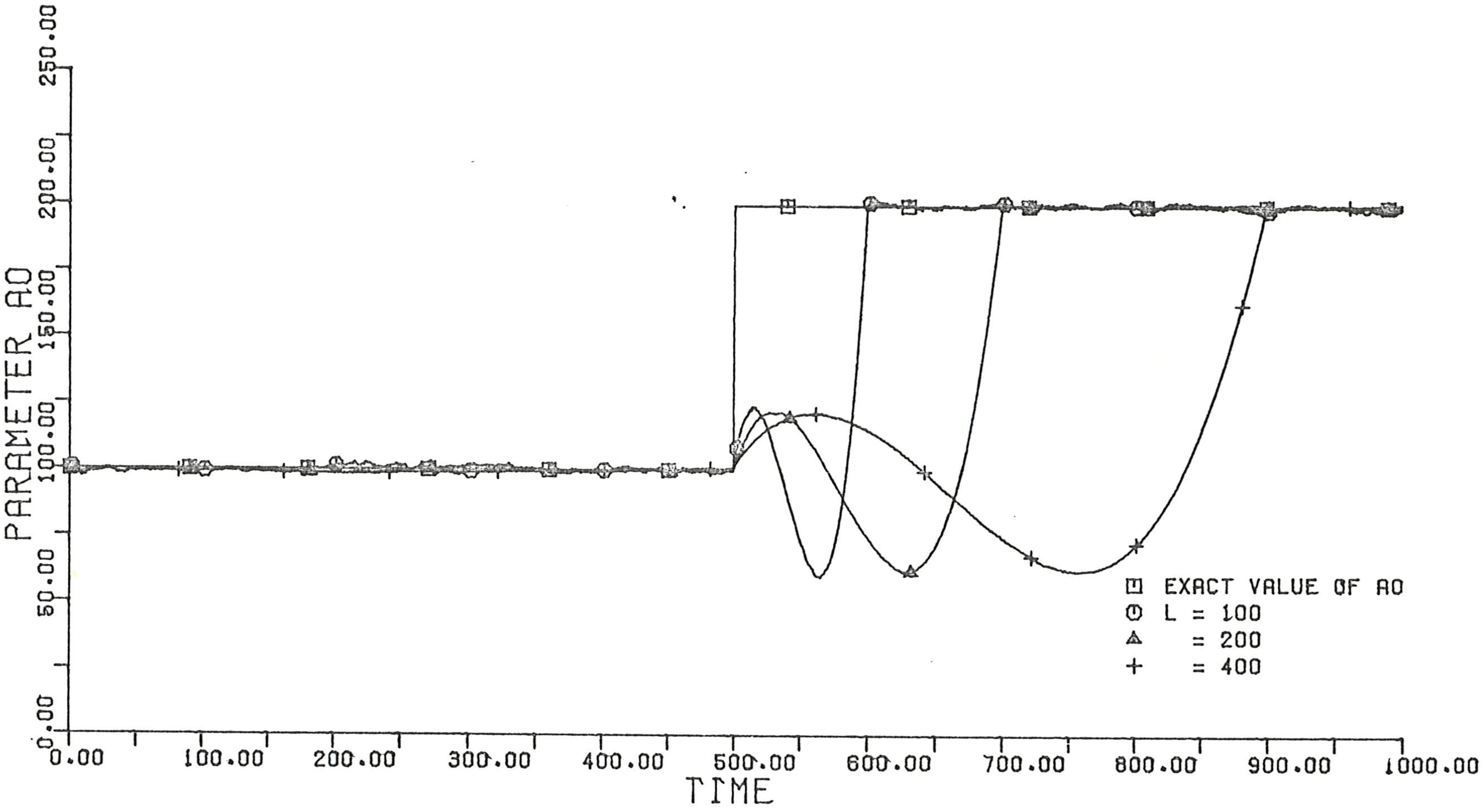


Figure 3.4 ESTIMATION OF PARAMETER AO
VARIANCE OF MEASUREMENT NOISE IS 10.0)

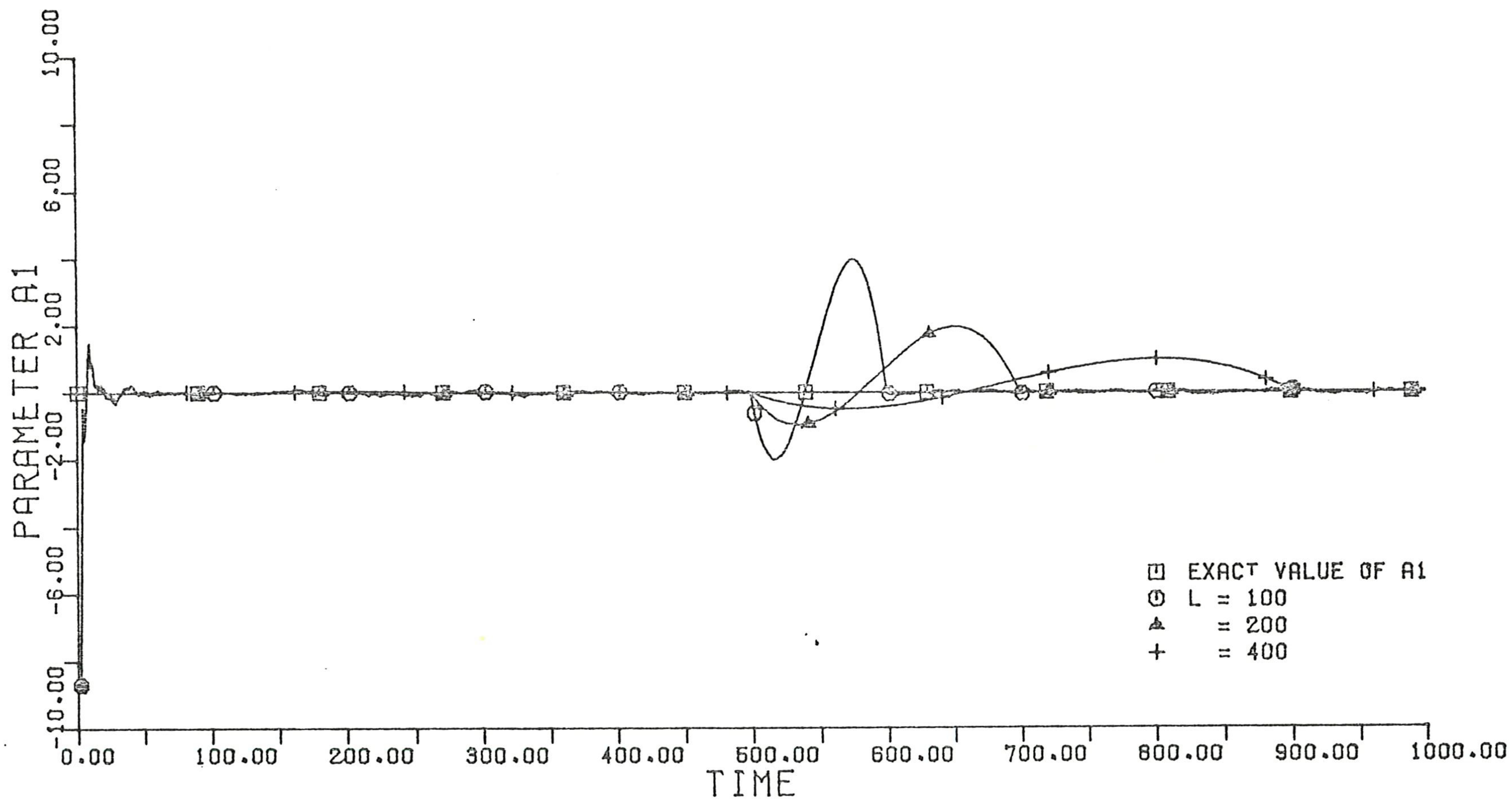


Figure 3.5 ESTIMATION OF PARAMETER A1
VARIANCE OF MEASUREMENT NOISE IS 10.0)

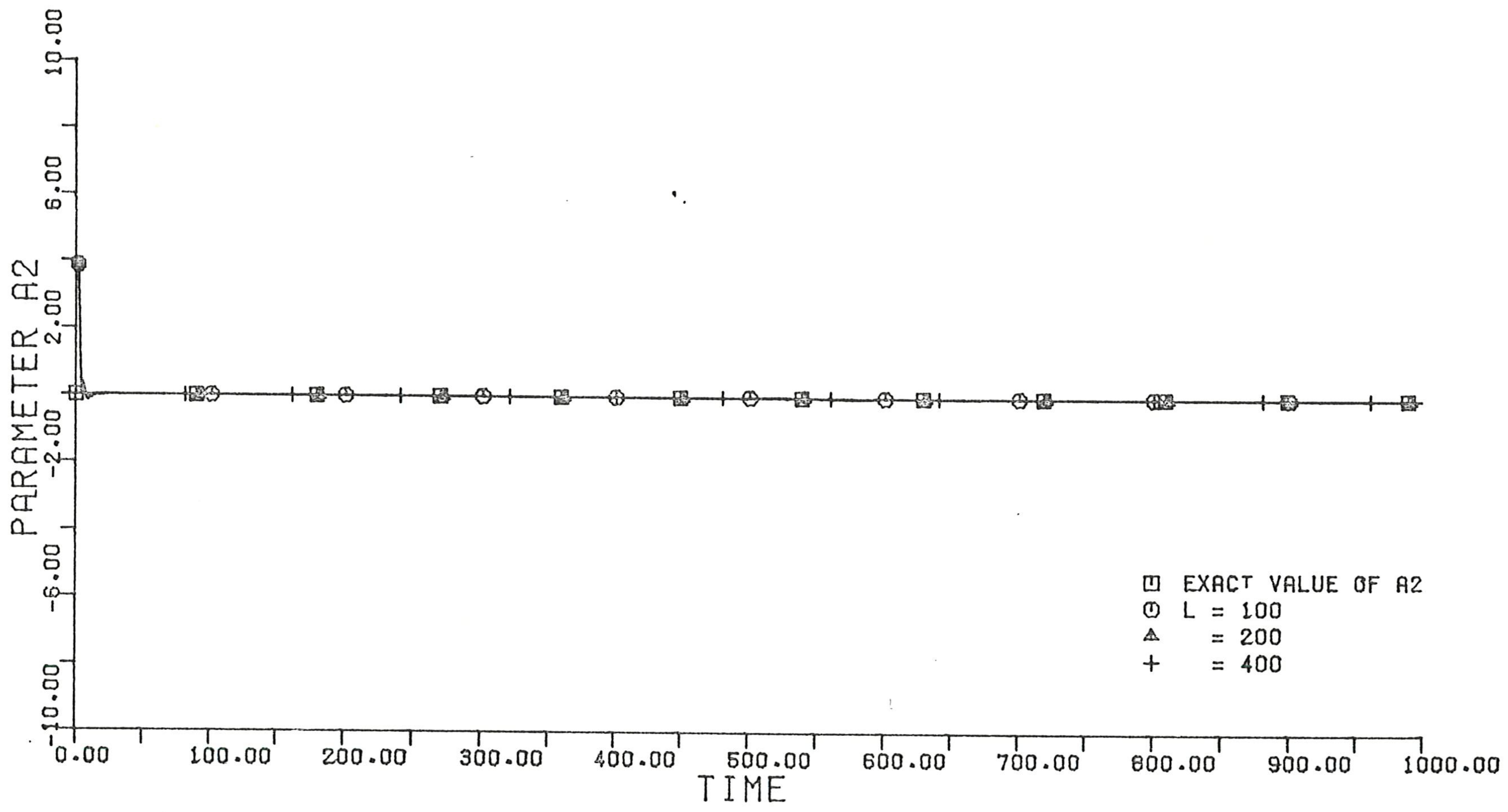


Figure 3.6 ESTIMATION OF PARAMETER A2
VARIANCE OF MEASUREMENT NOISE IS 10.0)

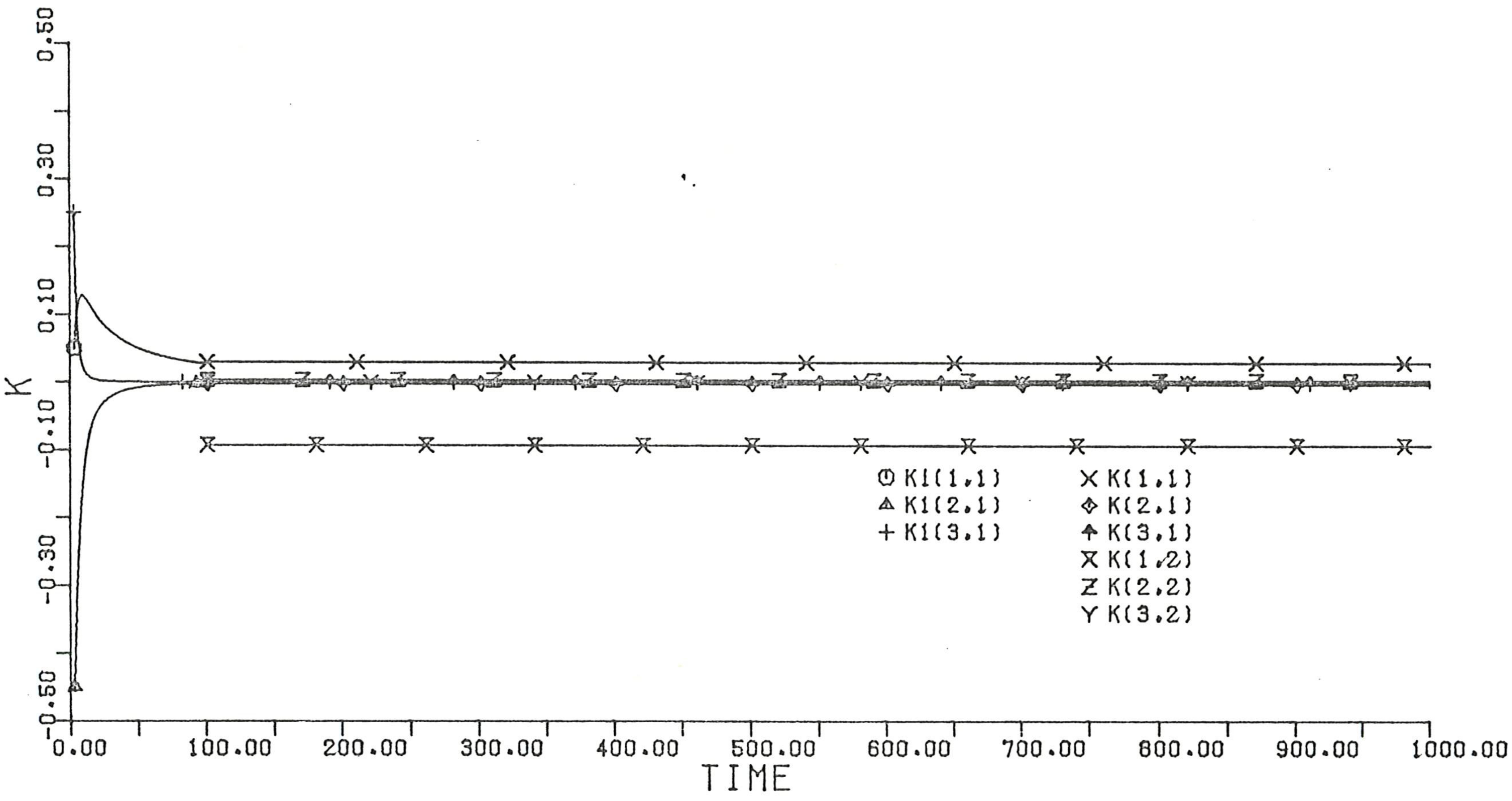


Figure 3.7 KALMAN GAIN K
(DATA LENGTH IS 100)

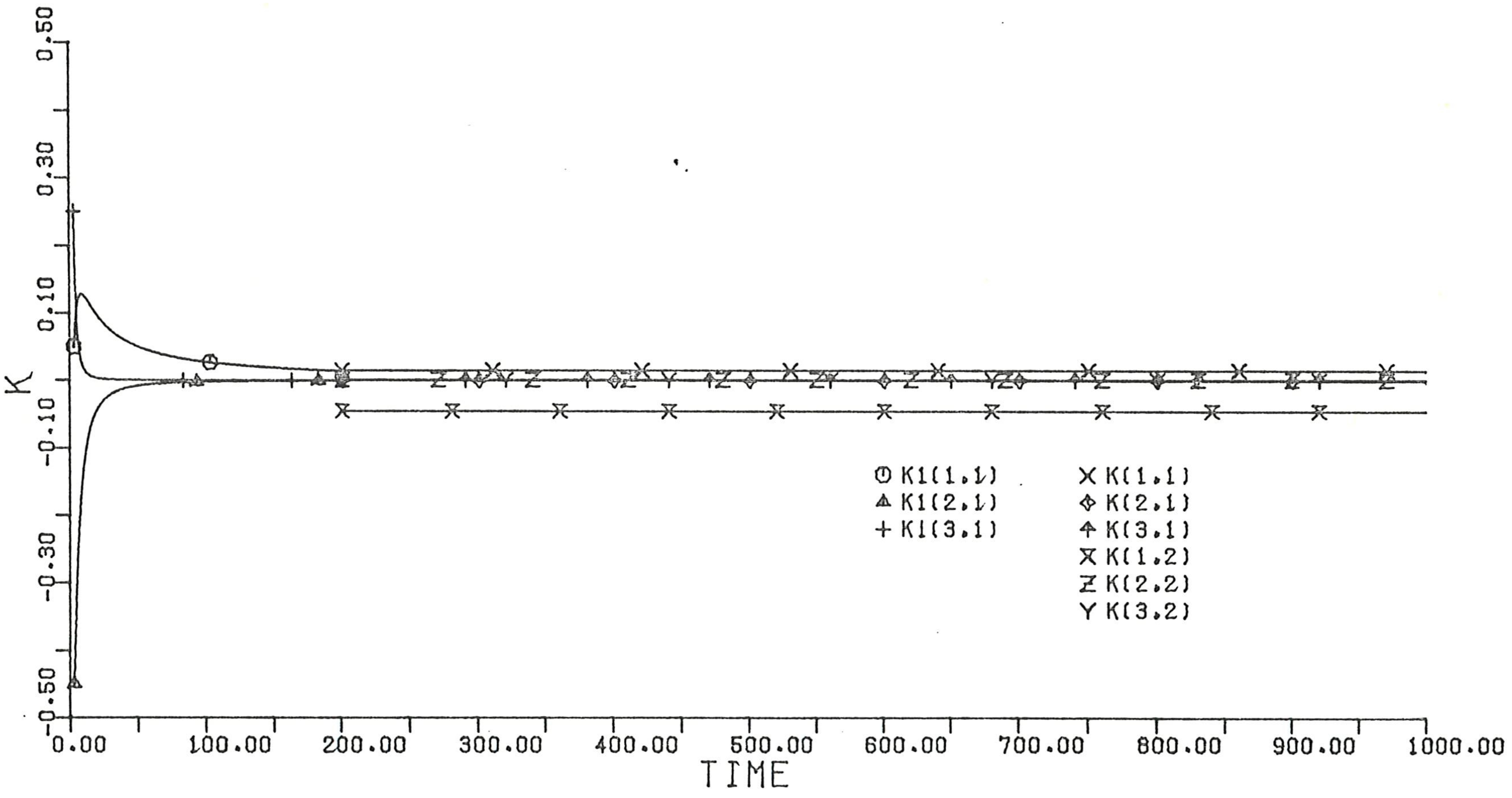


Figure 3.8 KALMAN GAIN K
(DATA LENGTH IS 200)

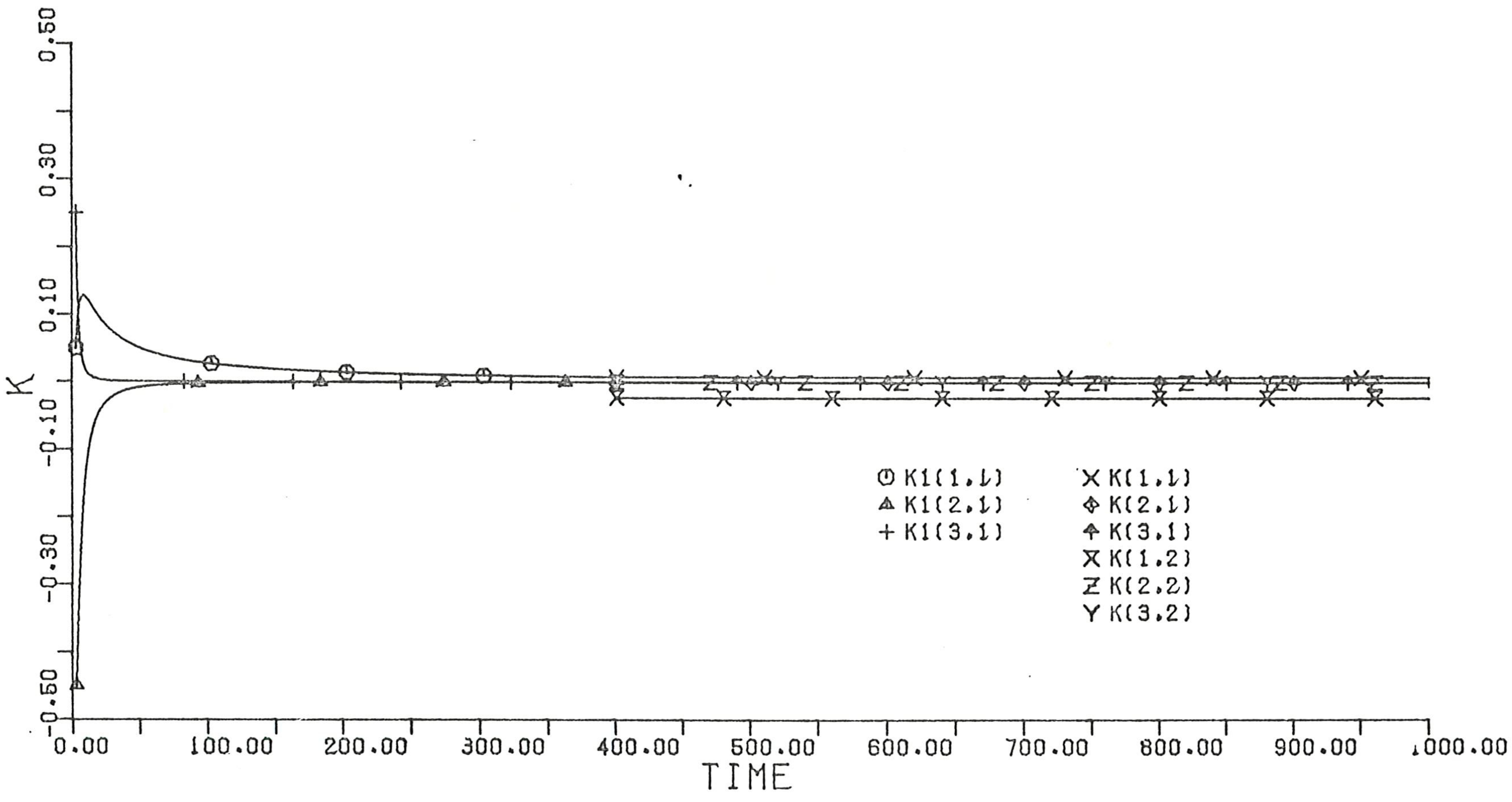


Figure 3.9 KALMAN GAIN K
(DATA LENGTH IS 400)

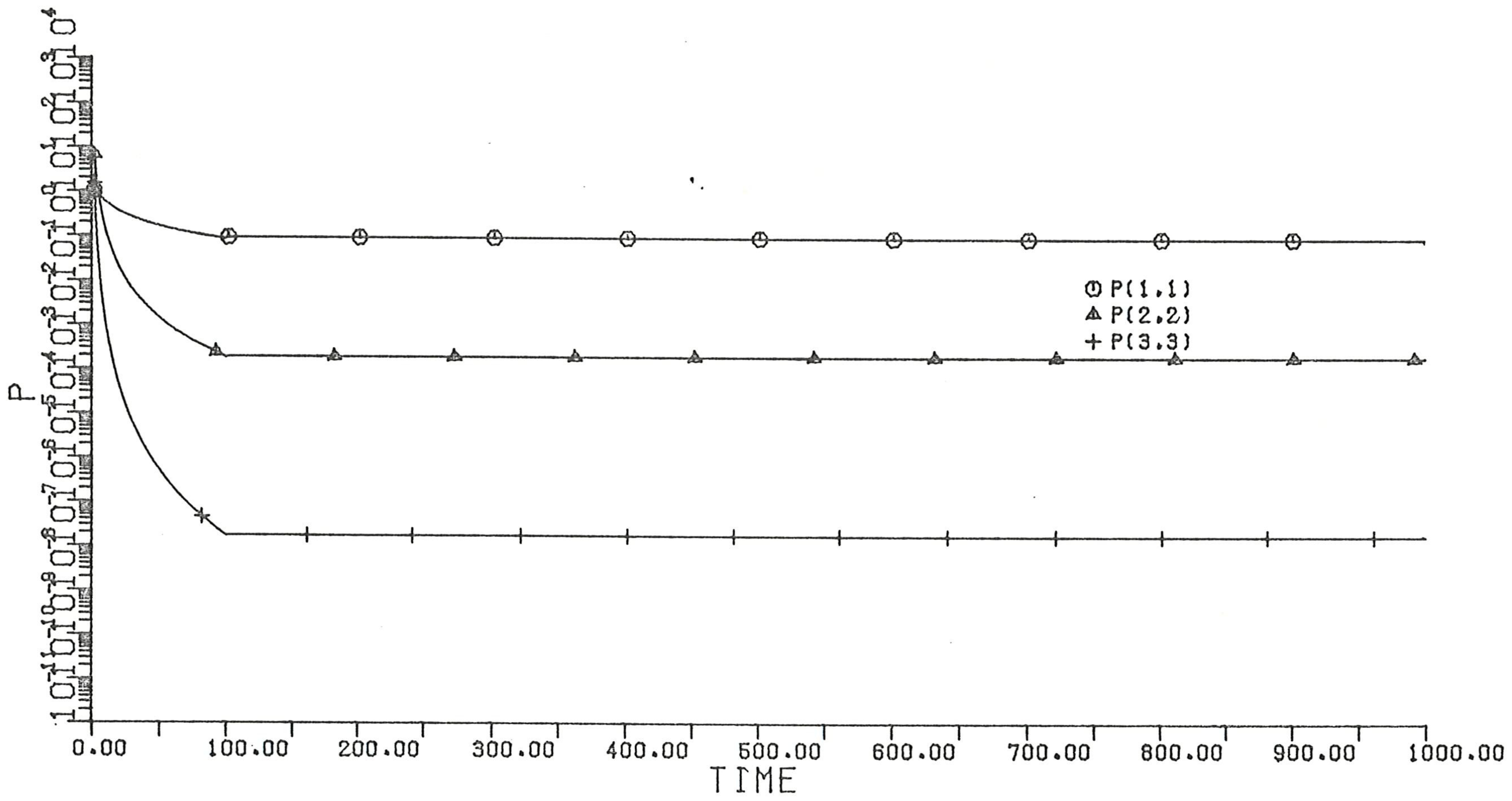


Figure 3.10 DIAGONAL ELEMENTS OF VARIANCE MATRIX P
(DATA LENGTH IS 100)

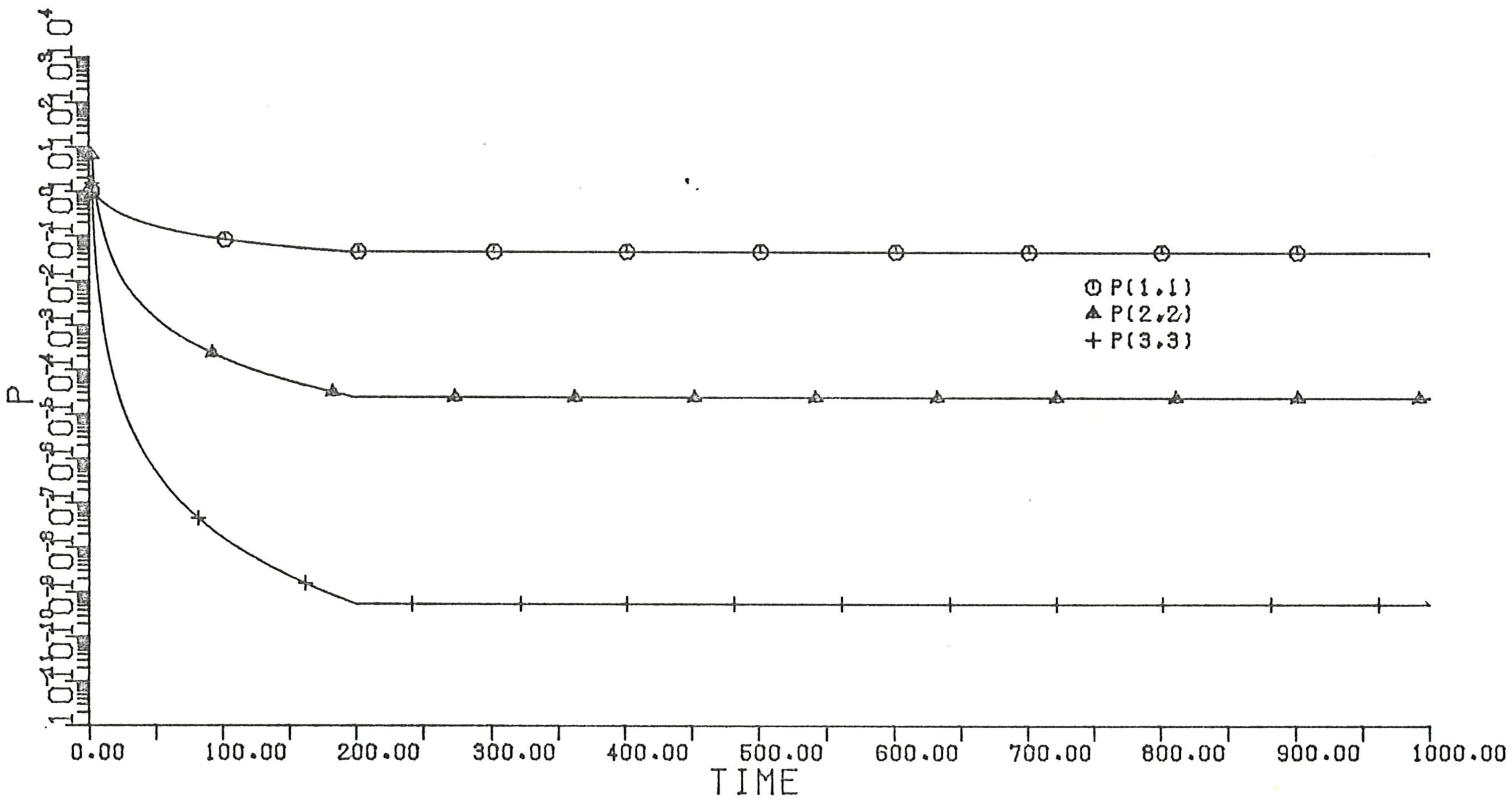


Figure 3.11 DIAGONAL ELEMENTS OF VARIANCE MATRIX P
(DATA LENGTH IS 200)

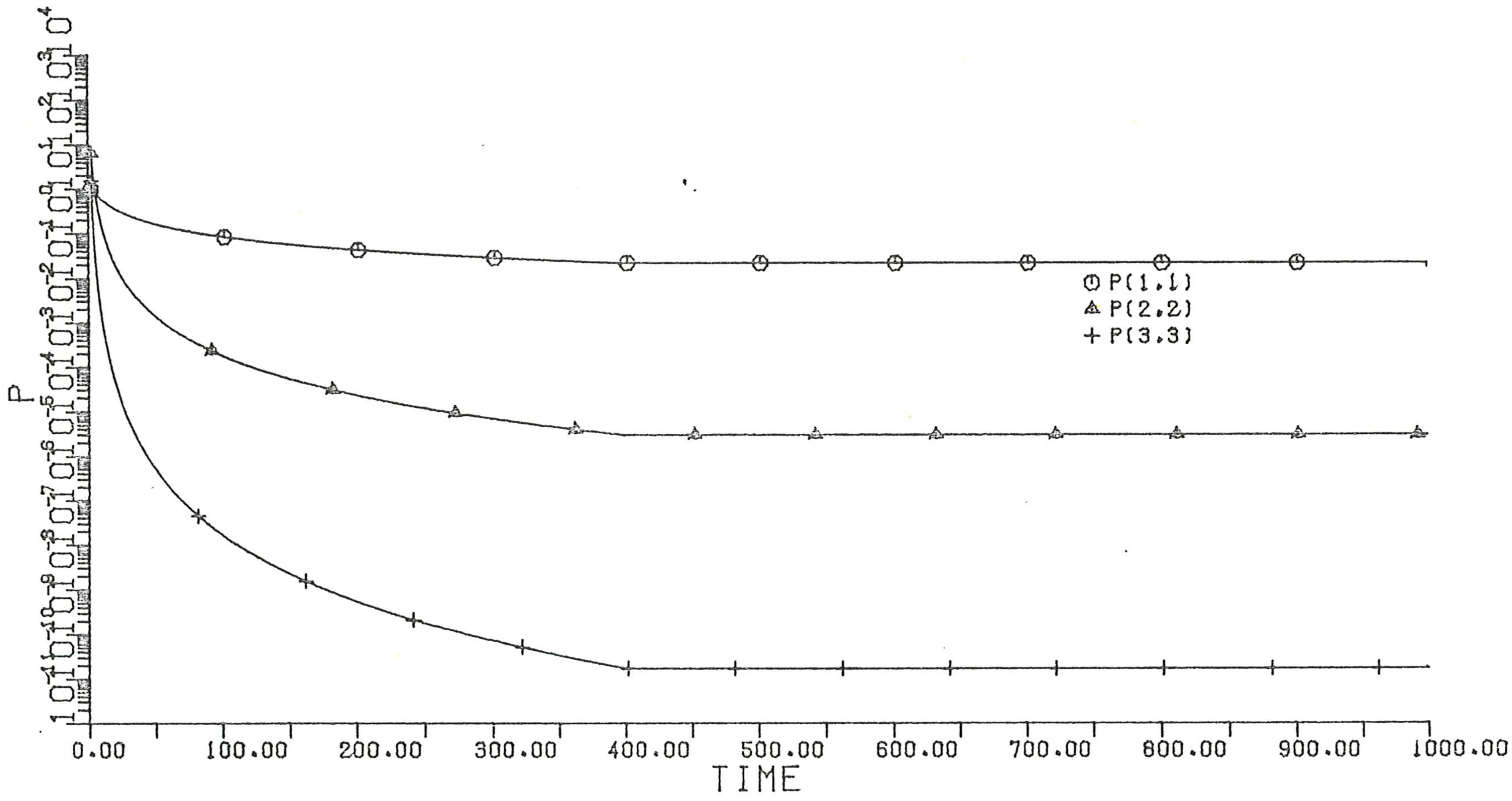


Figure 3.12 DIAGONAL ELEMENTS OF VARIANCE MATRIX P
(DATA LENGTH IS 400)

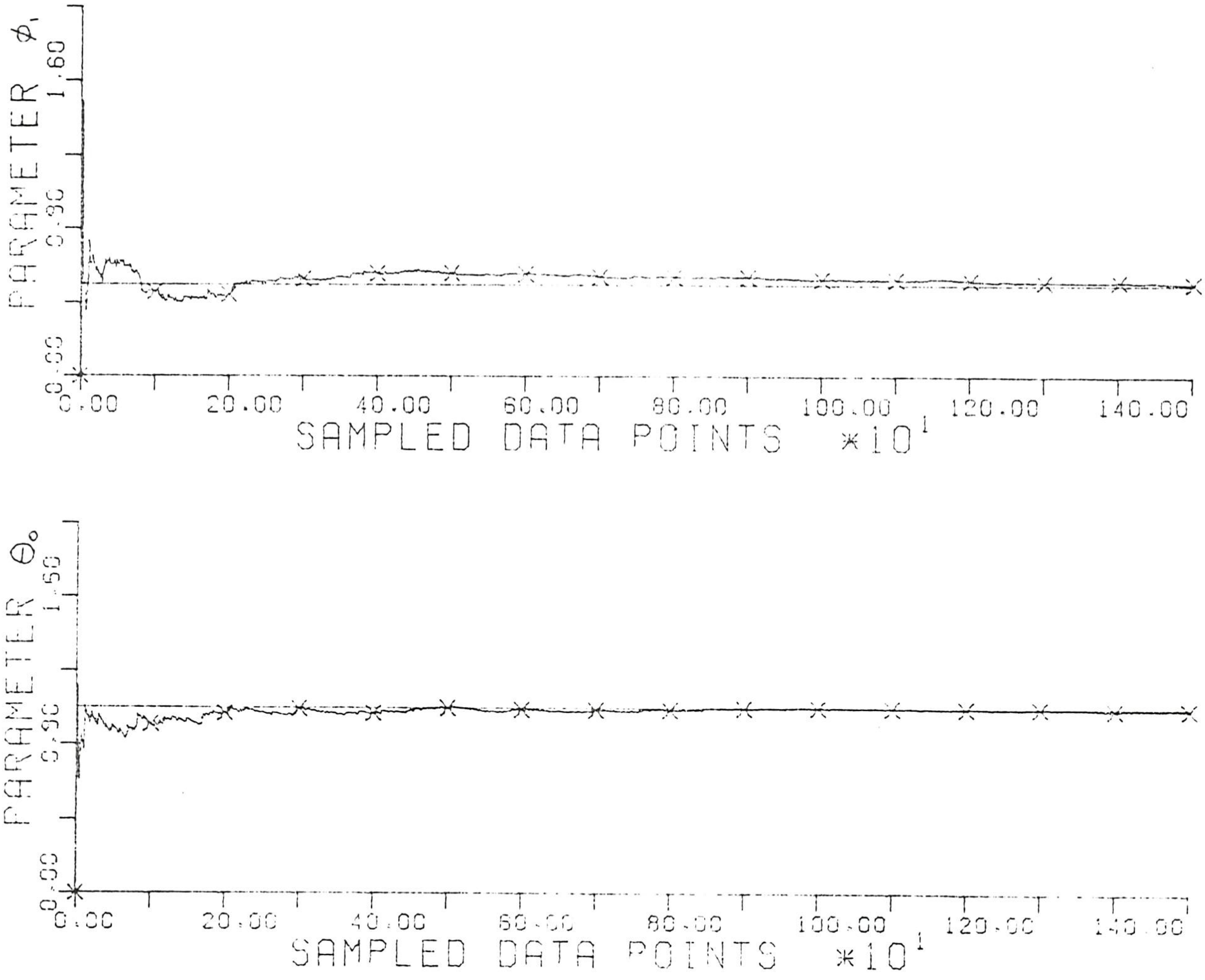


Figure 3.13 PARAMETER ESTIMATION OF AR(1) USING CANONICAL KALMAN FILTER

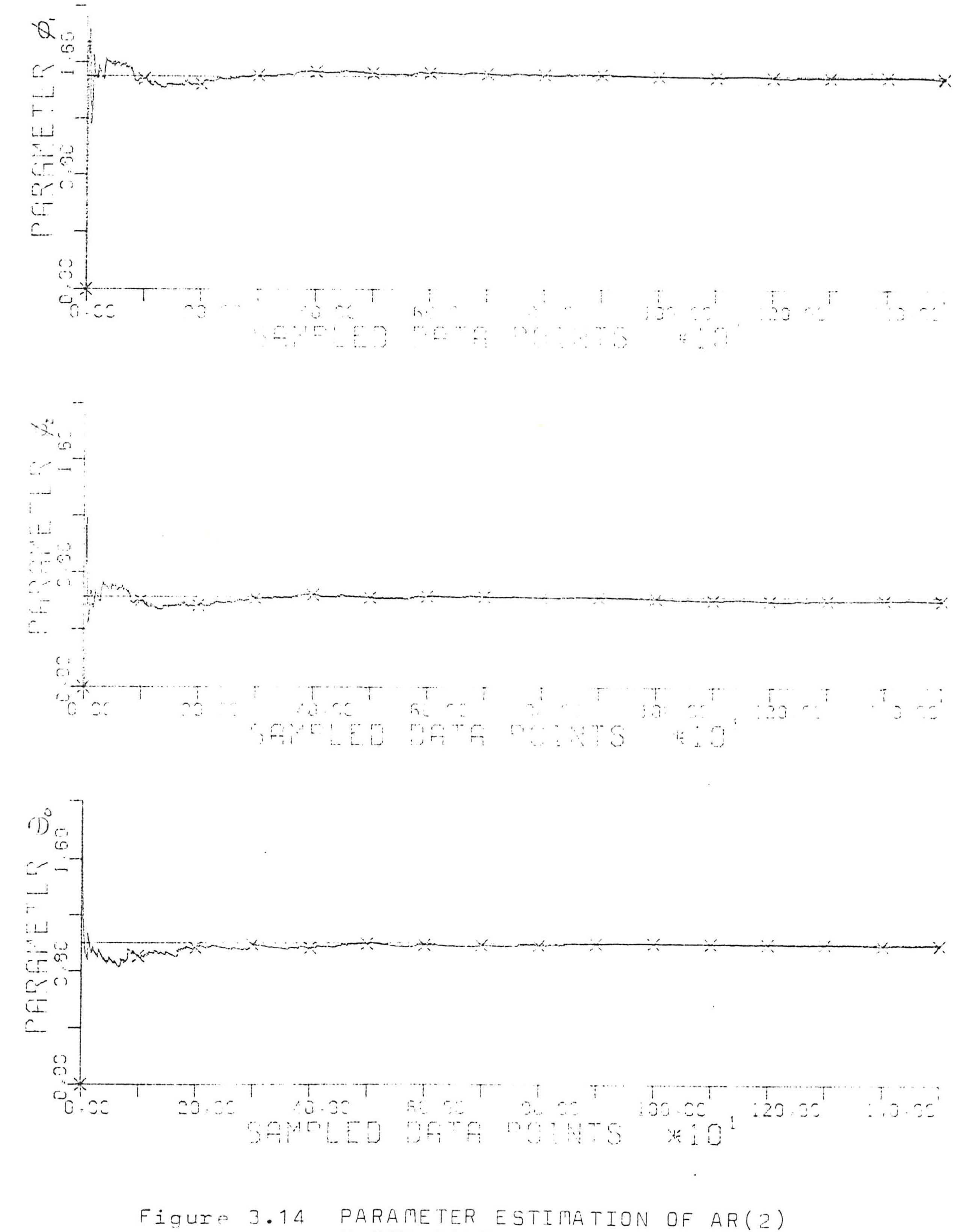


Figure 3.14 PARAMETER ESTIMATION OF AR(2)
USING CANONICAL KALMAN FILTER

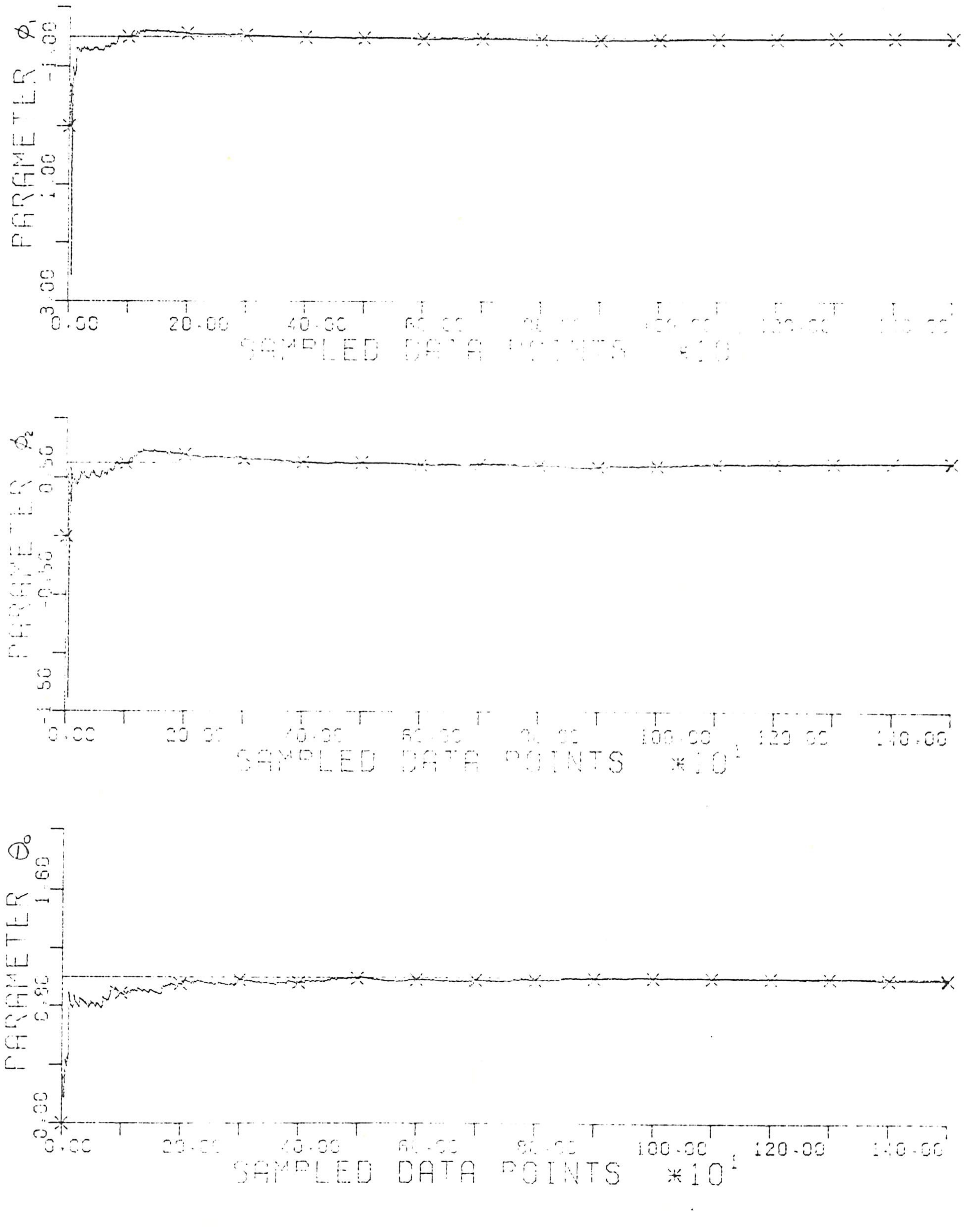


Figure 3.15 PARAMETER ESTIMATION OF AR(2) USING CANONICAL KALMAN FILTER

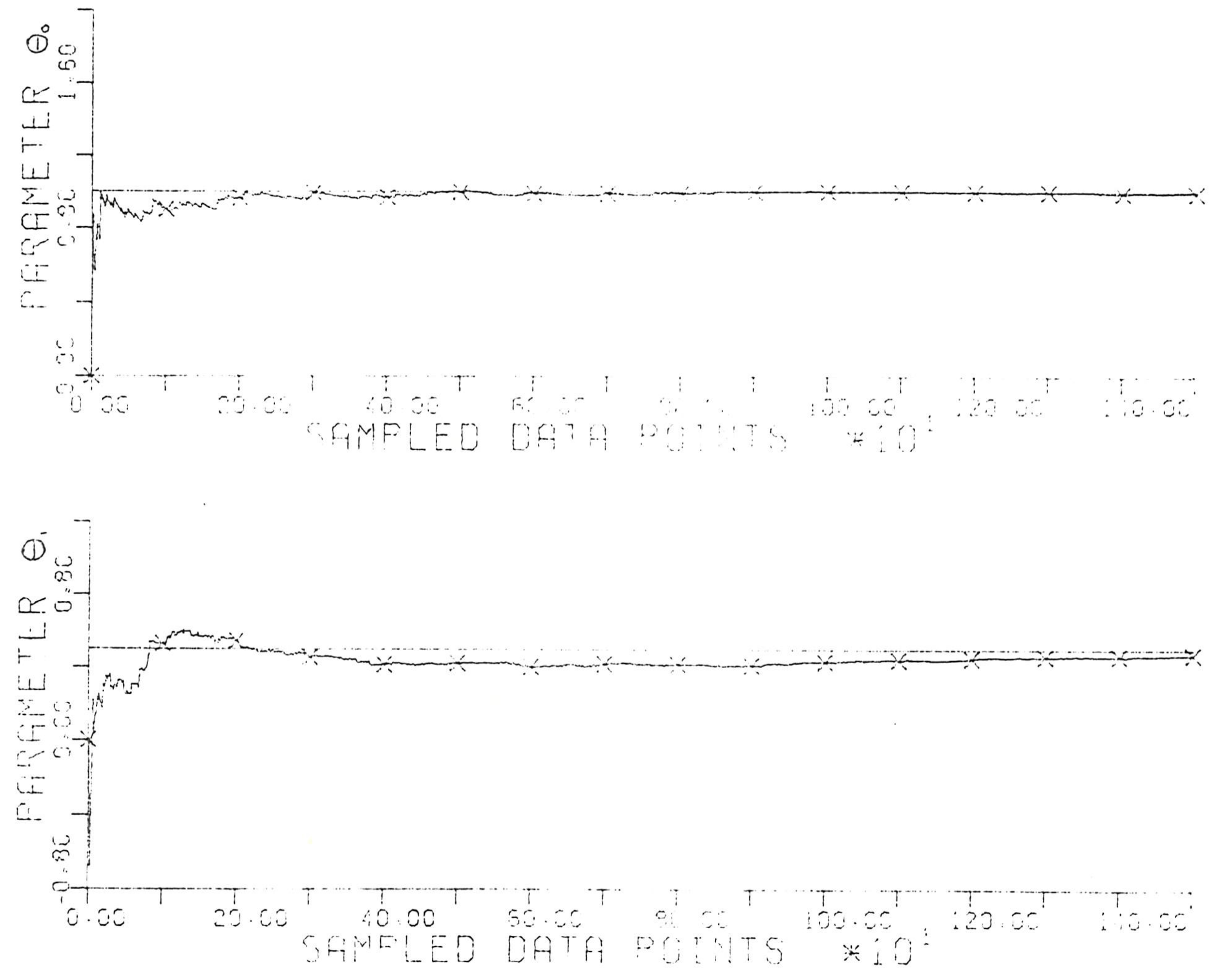


Figure 3.16 PARAMETER ESTIMATION OF MA(1) USING CANONICAL KALMAN FILTER

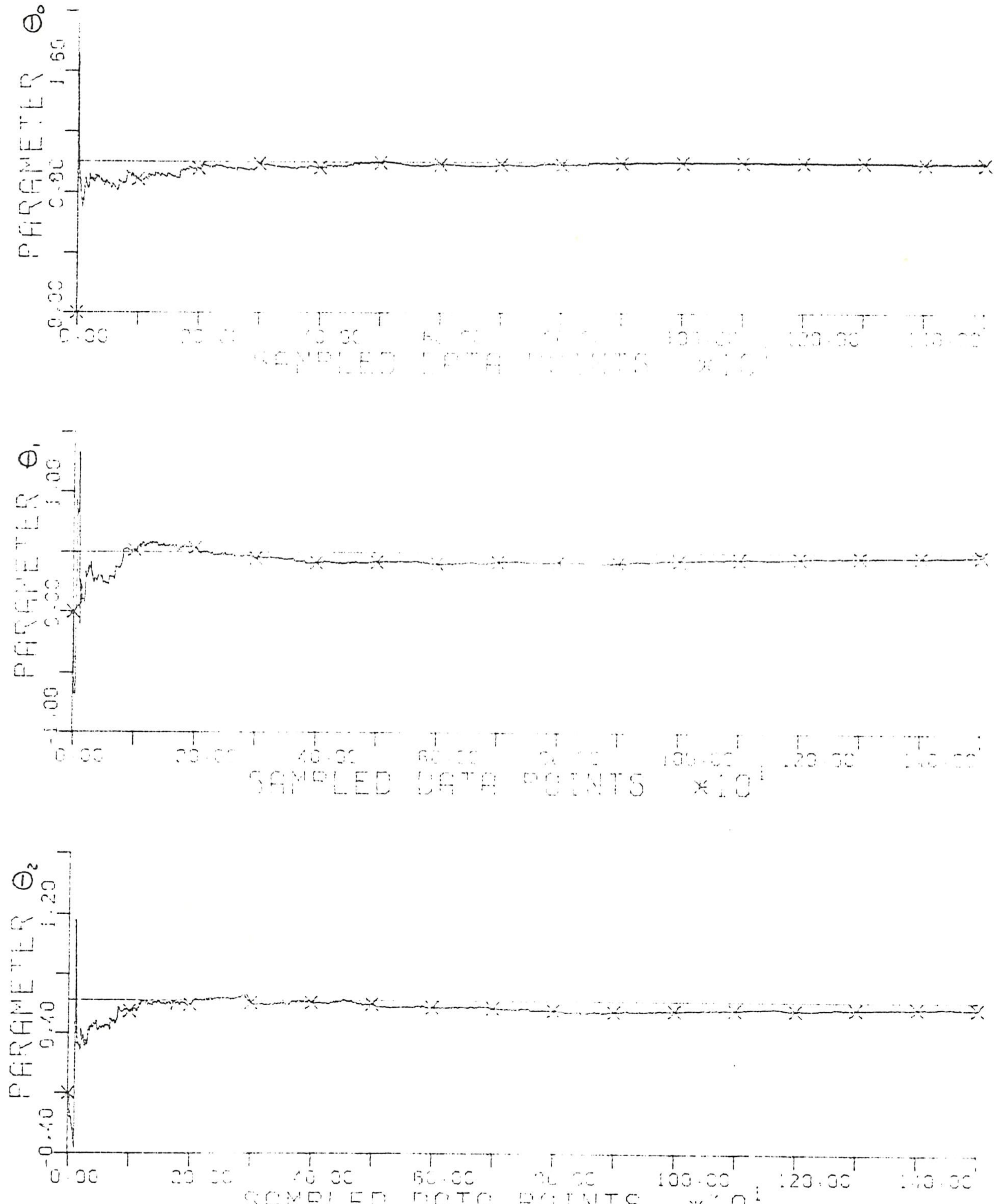


Figure 3.17 PARAMETER ESTIMATION OF MA(2) USING CANONICAL KALMAN FILTER

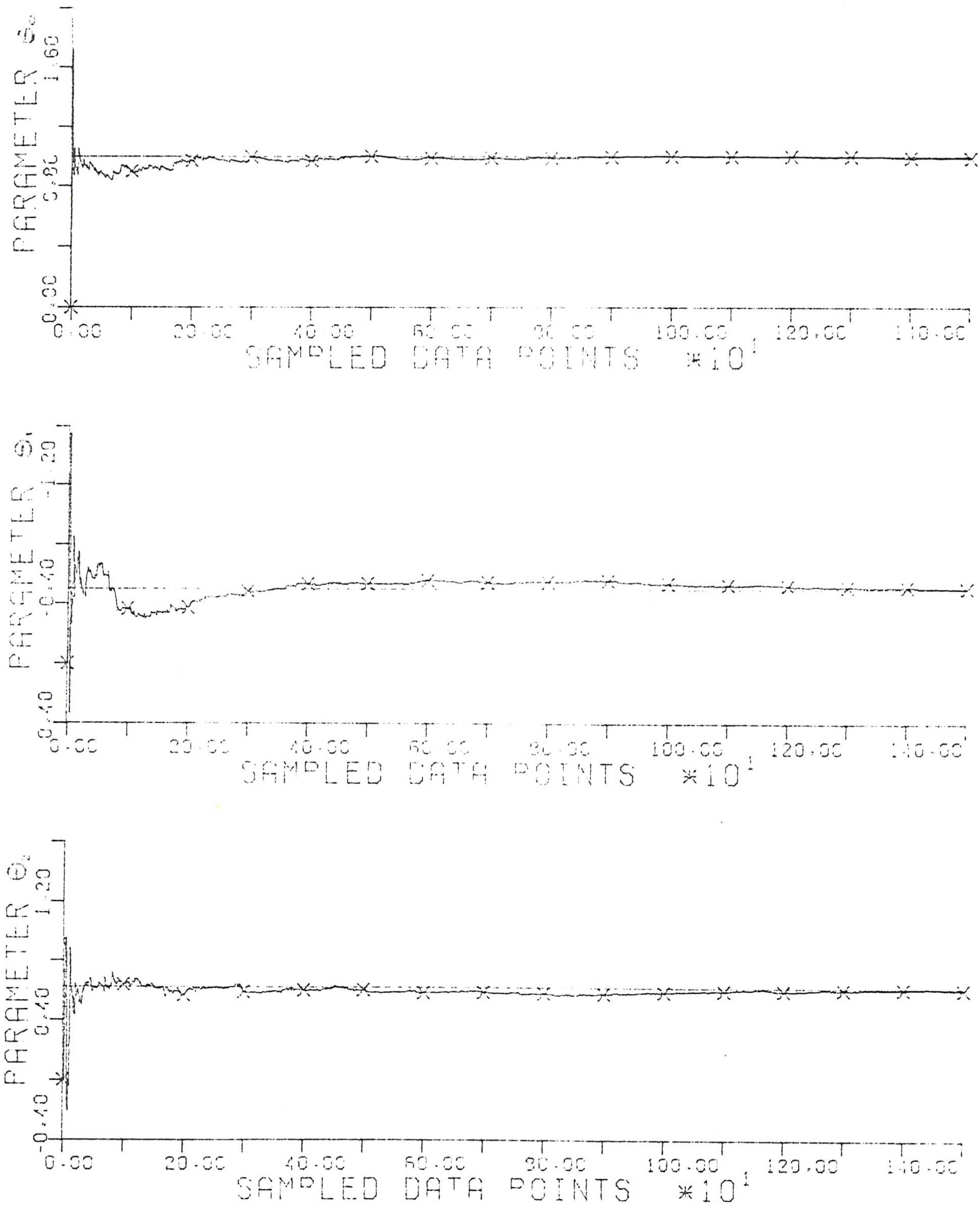


Figure 3.18 PARAMETER ESTIMATION OF MA(2)
USING CANONICAL KALMAN FILTER

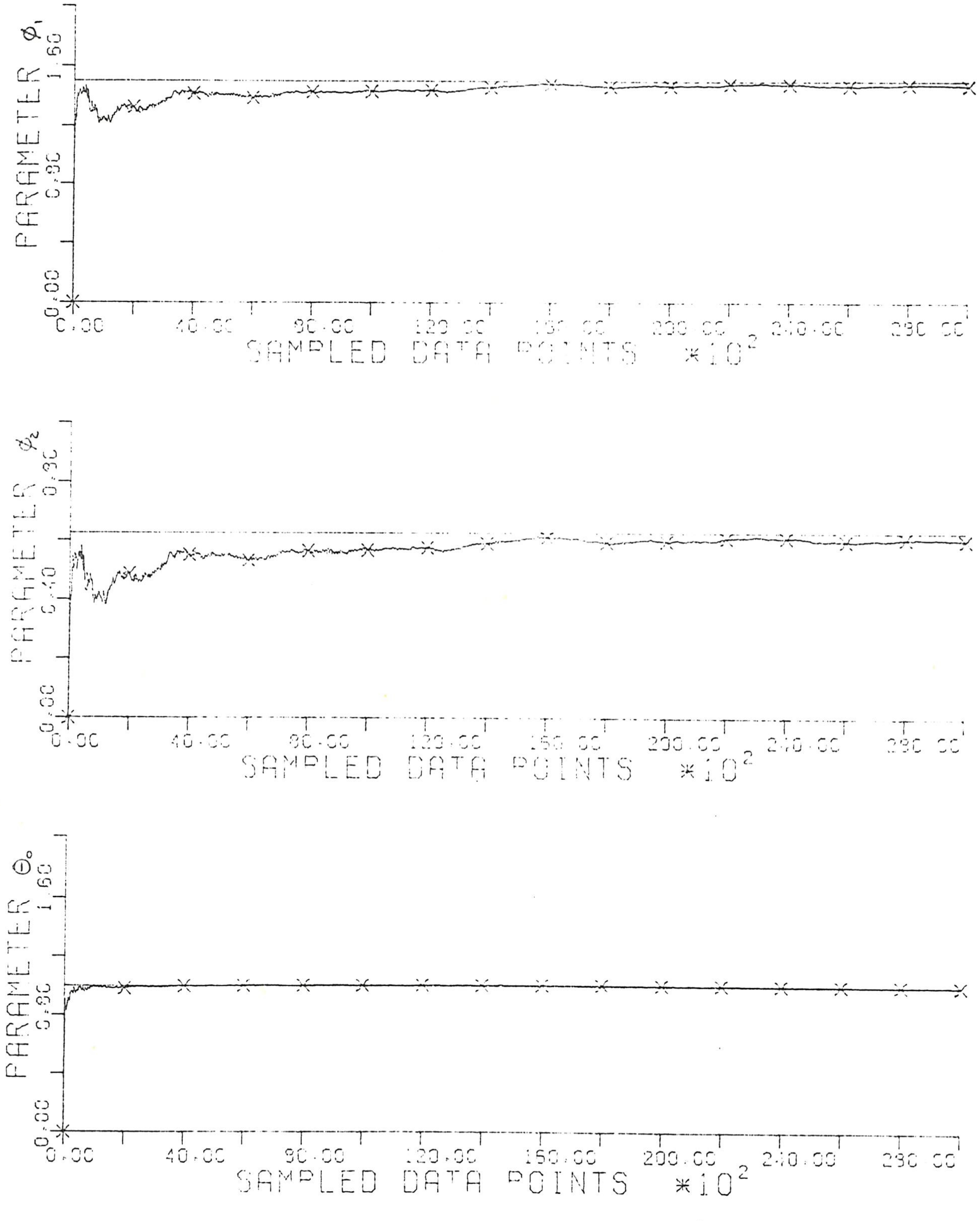


Figure 3.19a PARAMETER ESTIMATION OF ARMA(2,1) USING CANONICAL KALMAN FILTER

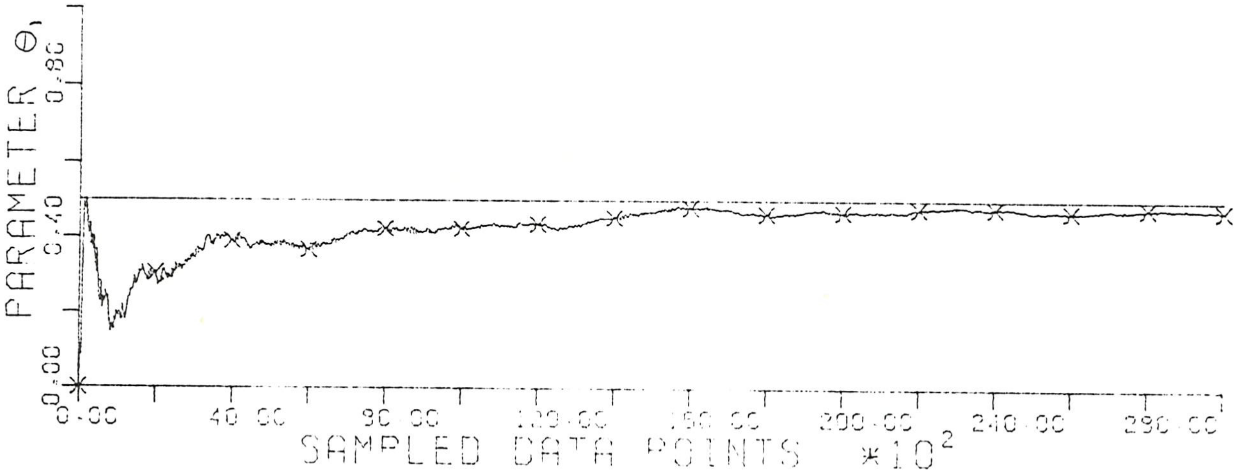

PARAMETER Θ_1
0.00
0.40
0.80
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS $*10^2$

Figure 3.19b

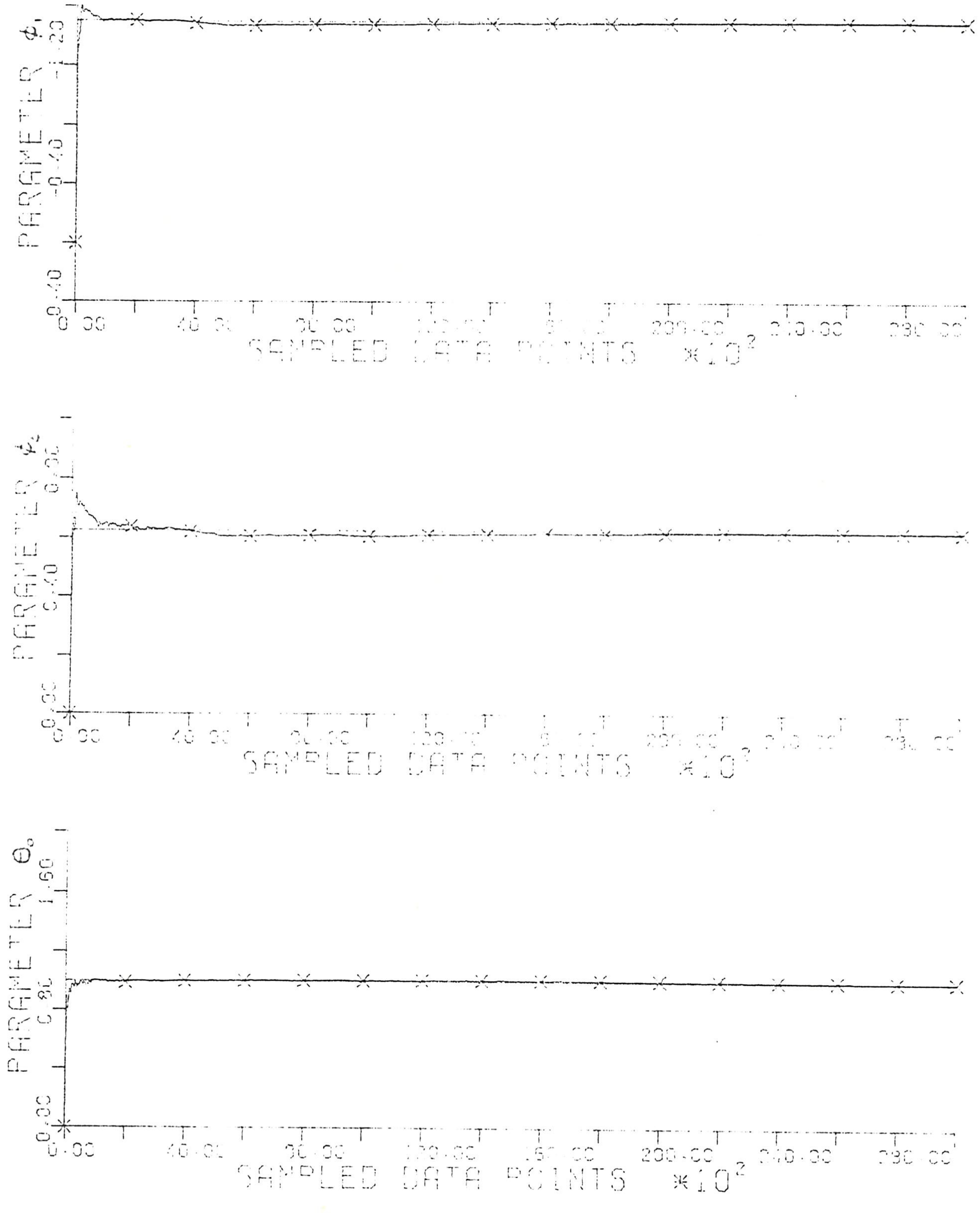


Figure 3.20a PARAMETER ESTIMATION OF ARMA(2,1) USING CANONICAL KALMAN FILTER

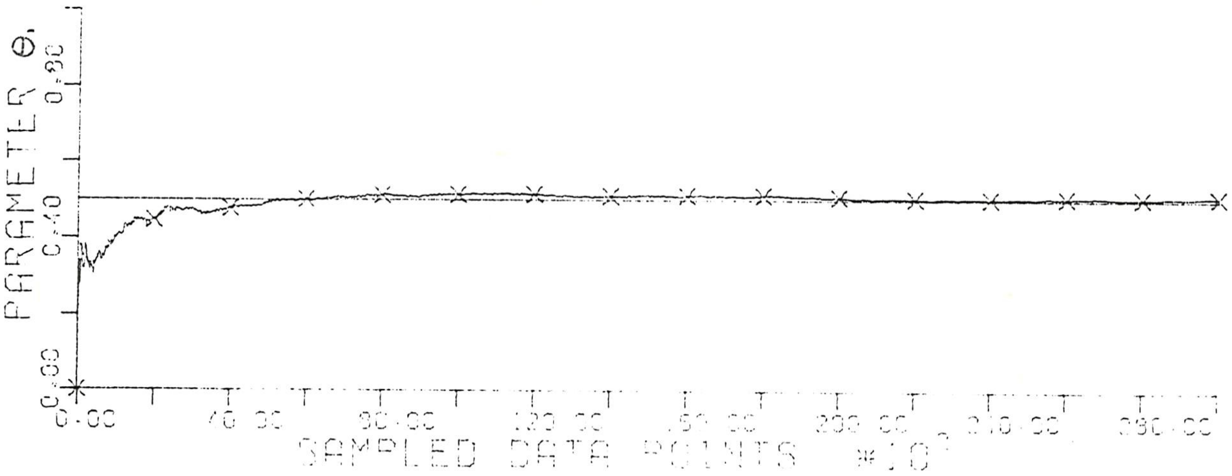

PARAMETER θ_1
0.00
0.40
0.80
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS $*10^2$

Figure 3.20b

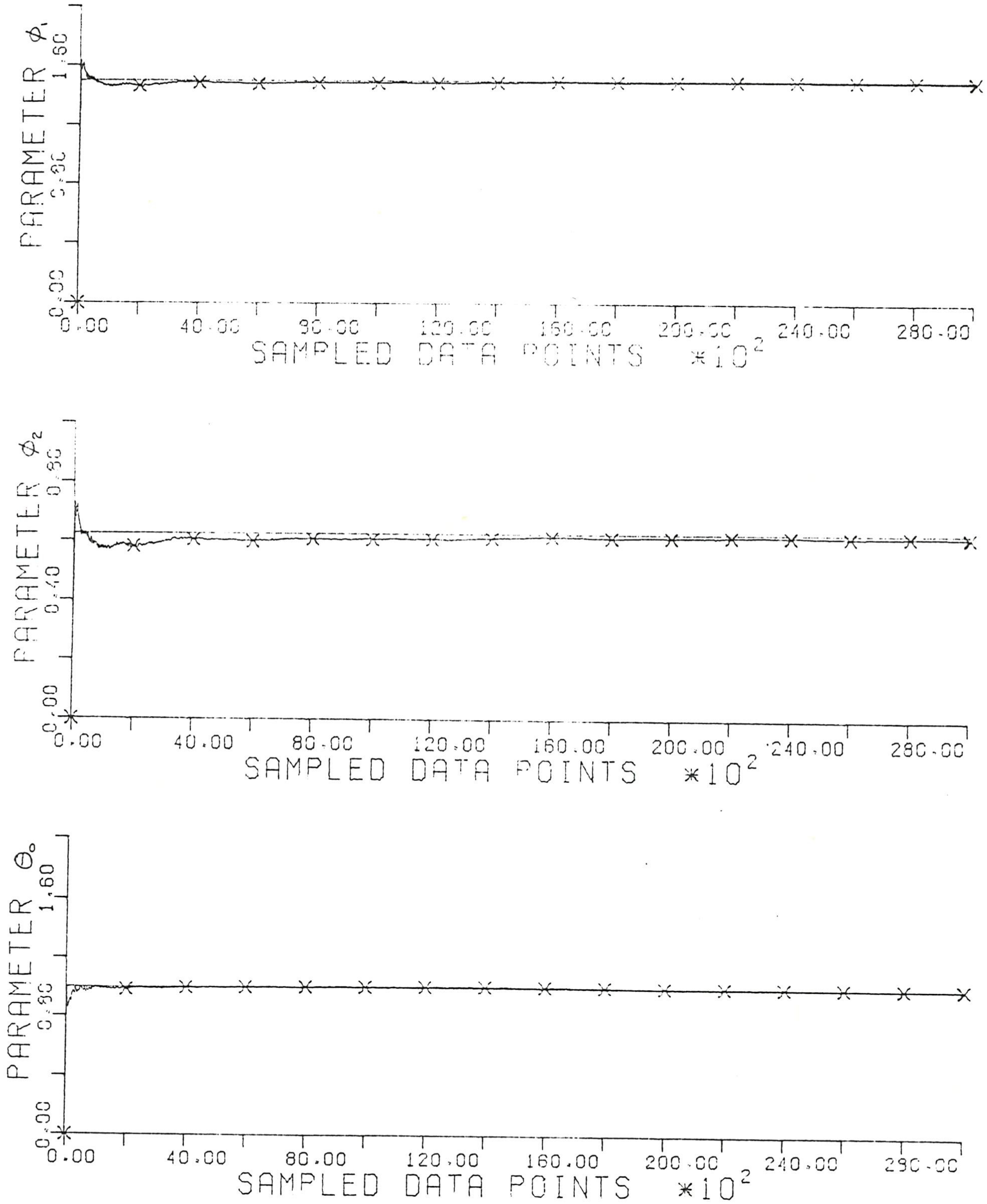


Figure 3.21a PARAMETER ESTIMATION OF ARMA(2,1) USING CANONICAL KALMAN FILTER

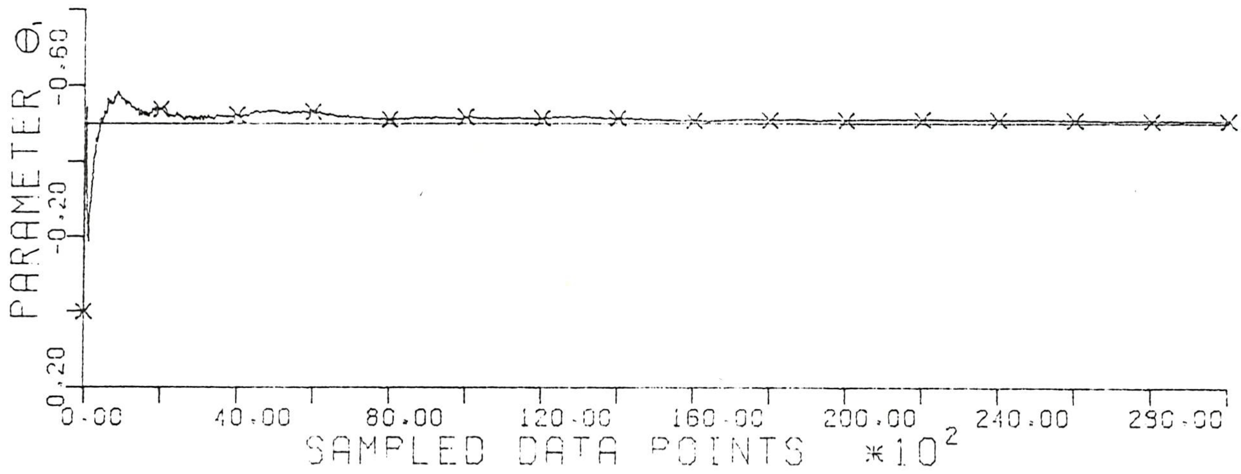

PARAMETER θ1
-0.60
-0.20
0.20
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10^2

Figure 3.21b

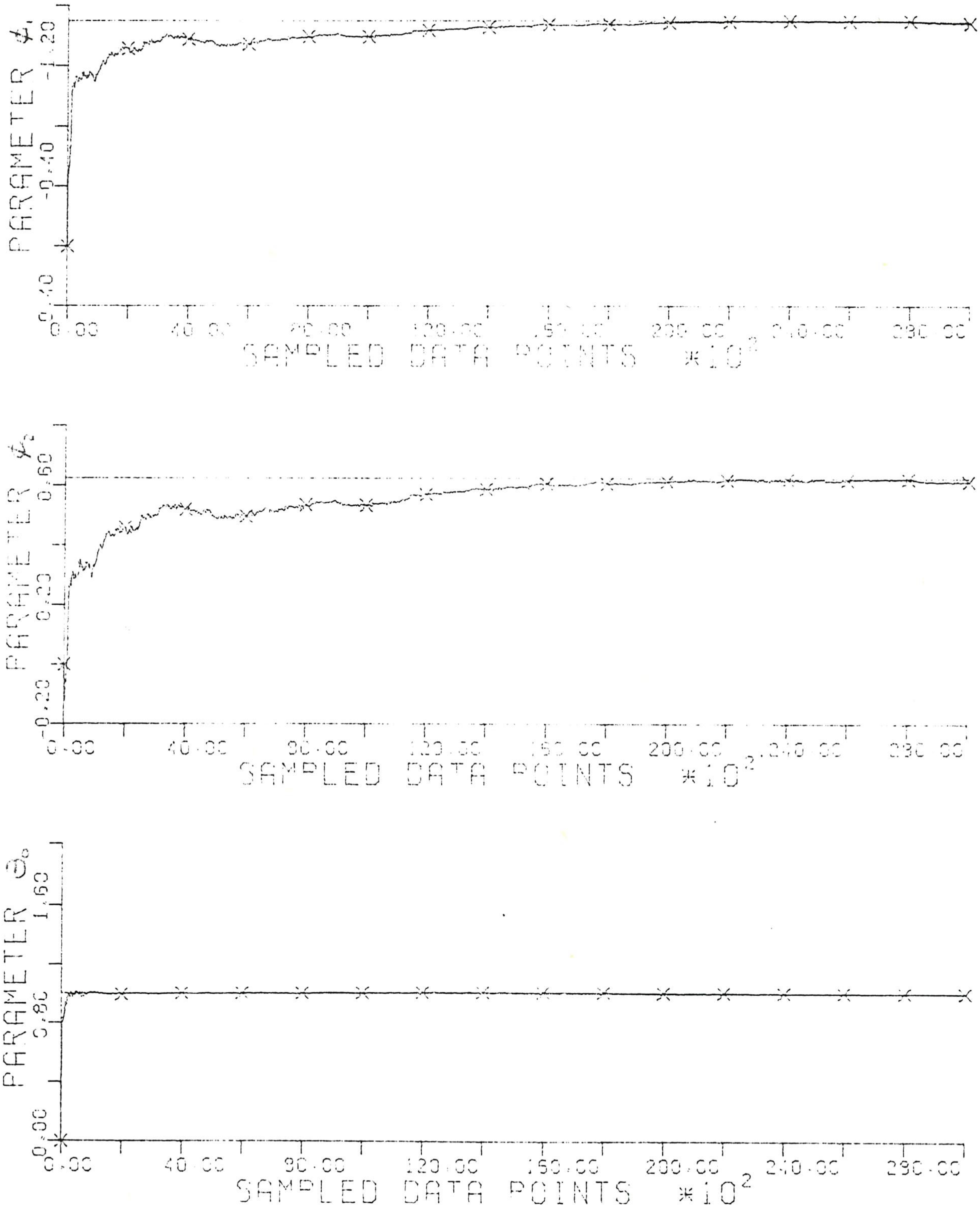


Figure 3.22a PARAMETER ESTIMATION OF ARMA(2,1)
USING CANONICAL KALMAN FILTER

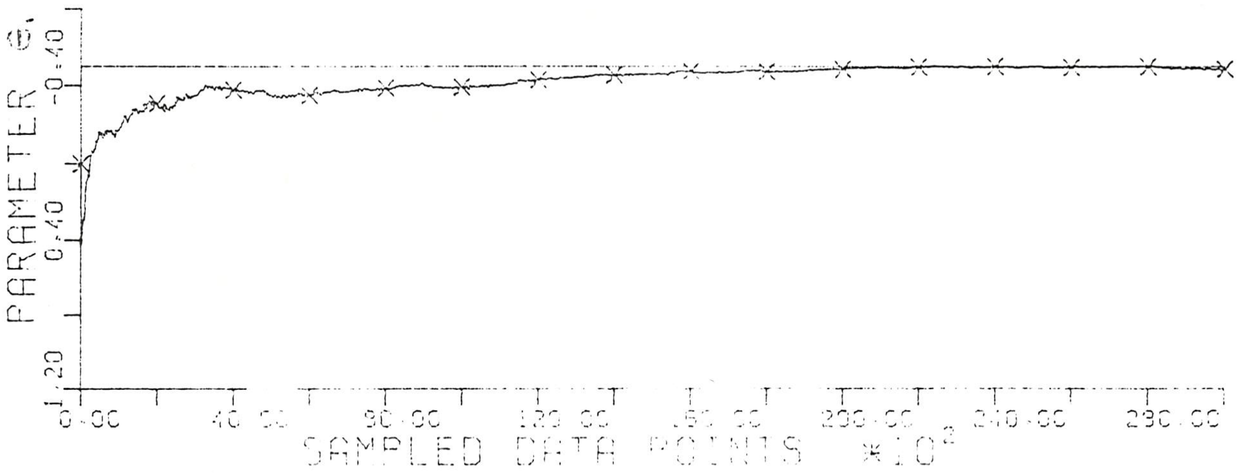

PARAMETER θ_1
-0.40
0.40
1.20
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS $*10^2$

Figure 3.22b

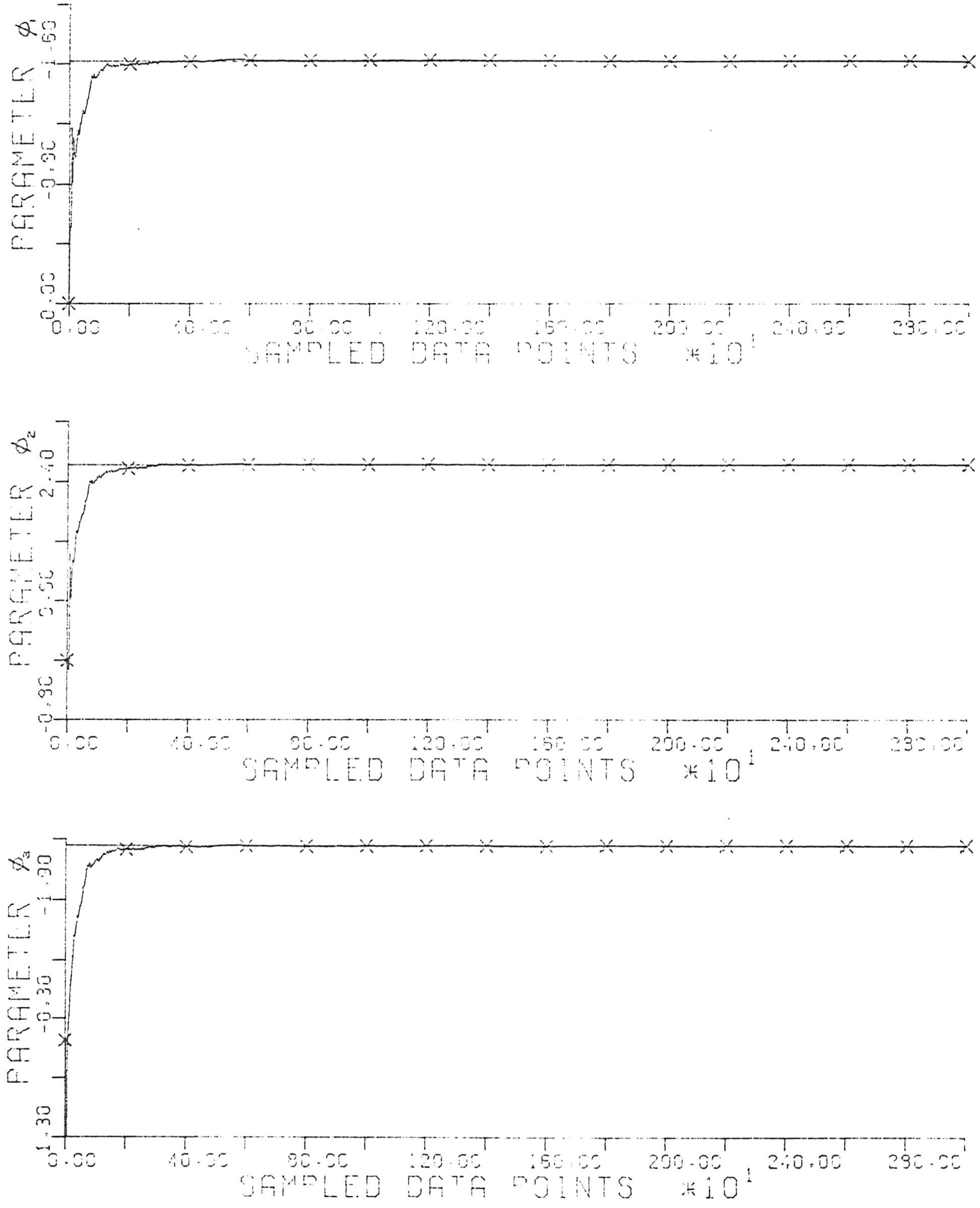


Figure 3.23a PARAMETER ESTIMATION OF AR(6)
USING CANONICAL KALMAN FILTER

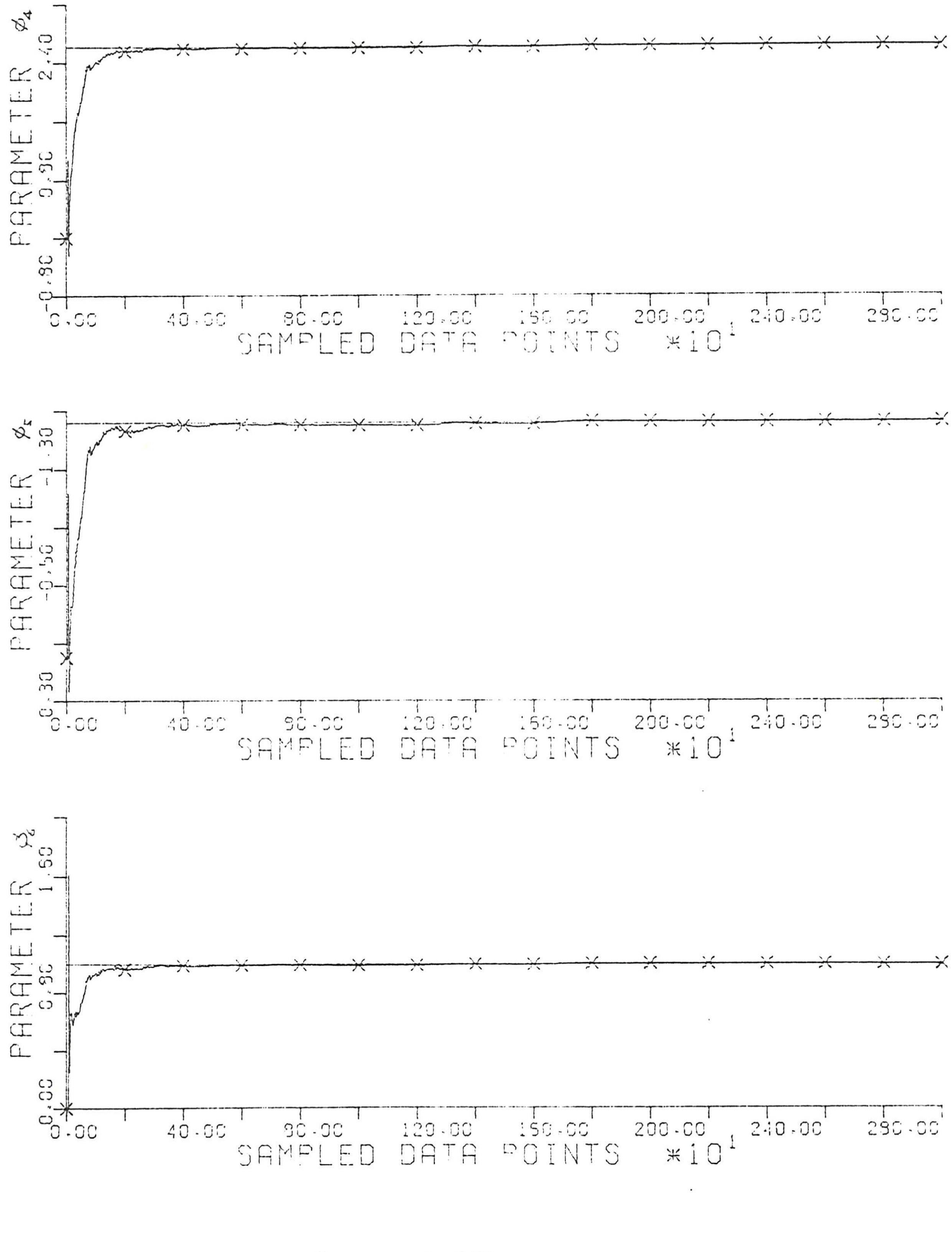

PARAMETER ϕ_4
-0.80
0.80
2.40
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10^1
PARAMETER ϕ_5
0.30
-0.50
-1.30
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10^1
PARAMETER γ_2
0.00
0.80
1.60
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10^1

Figure 3.23b

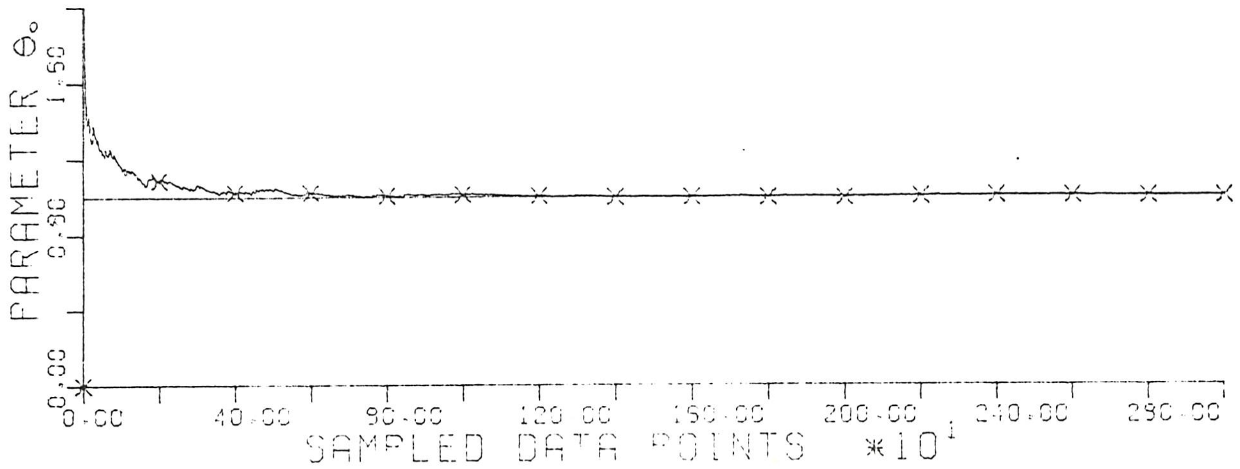

PARAMETER θ_0
0.00
0.60
1.20
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS $*10^1$

Figure 3.23c

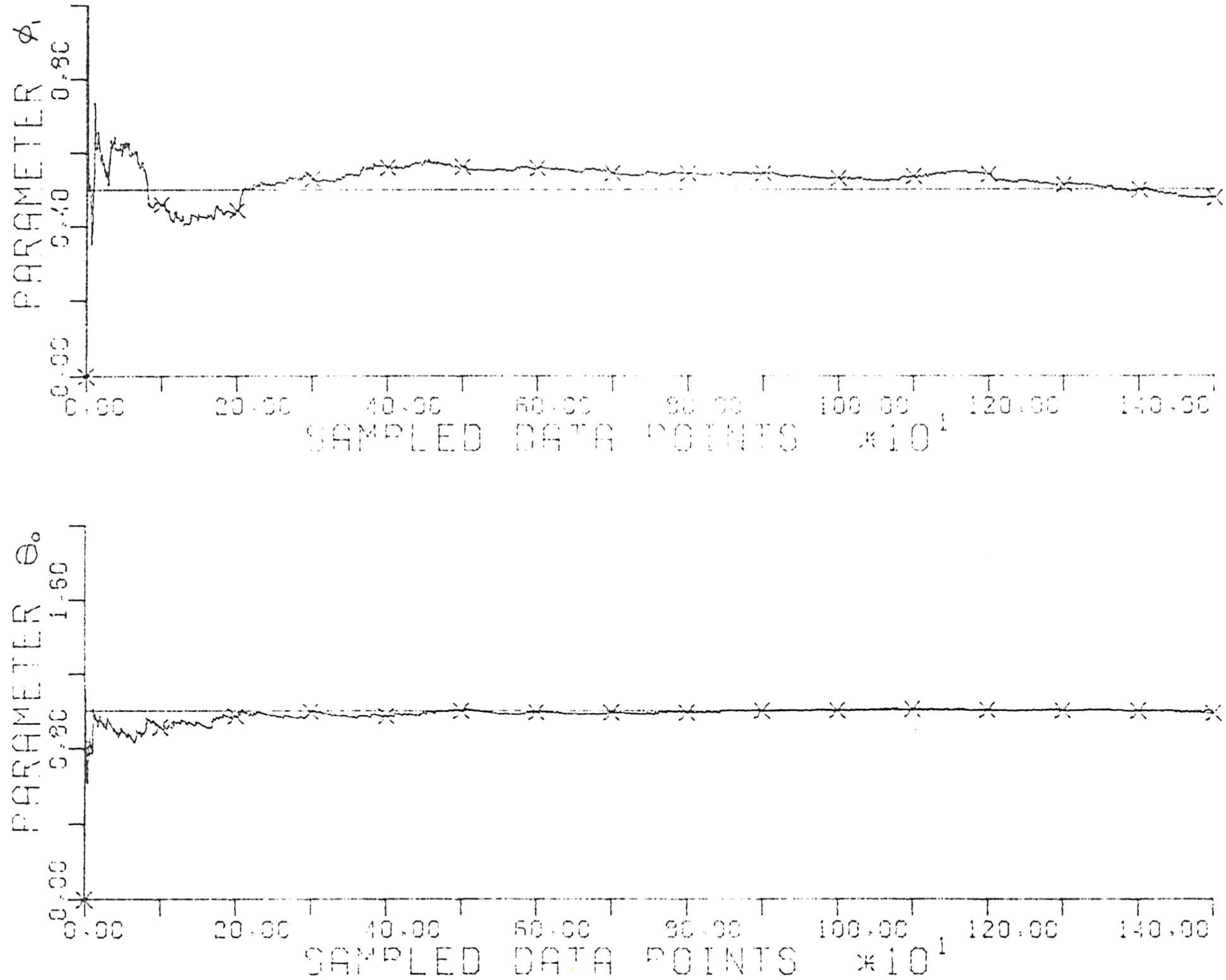


Figure 3.24 PARAMETER ESTIMATION OF AR(1) USING FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH 1000

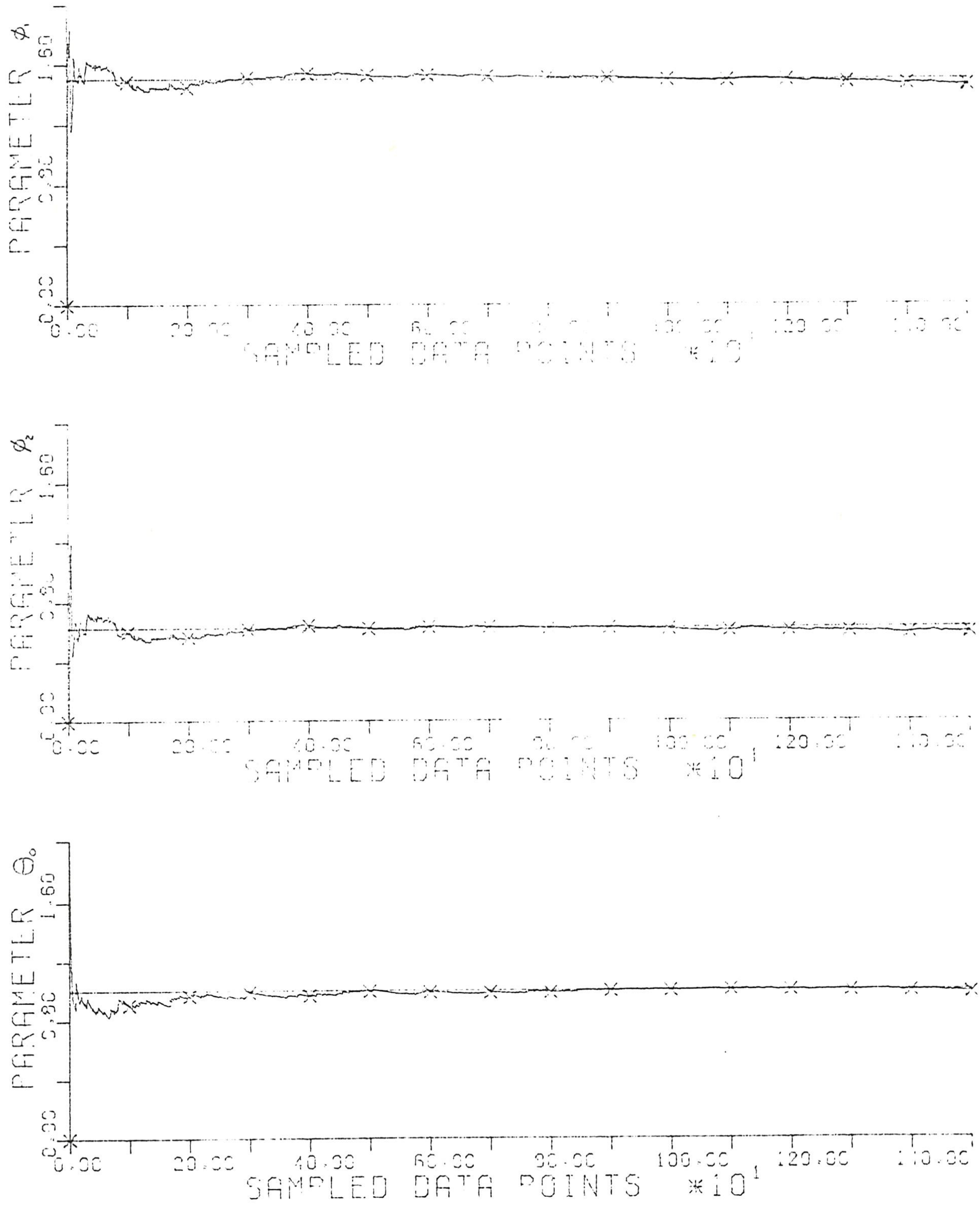


Figure 3.25 PARAMETER ESTIMATION OF AR(2)
USING FINITE MEMORY KALMAN FILTER
WITH WINDOW LENGTH 1000

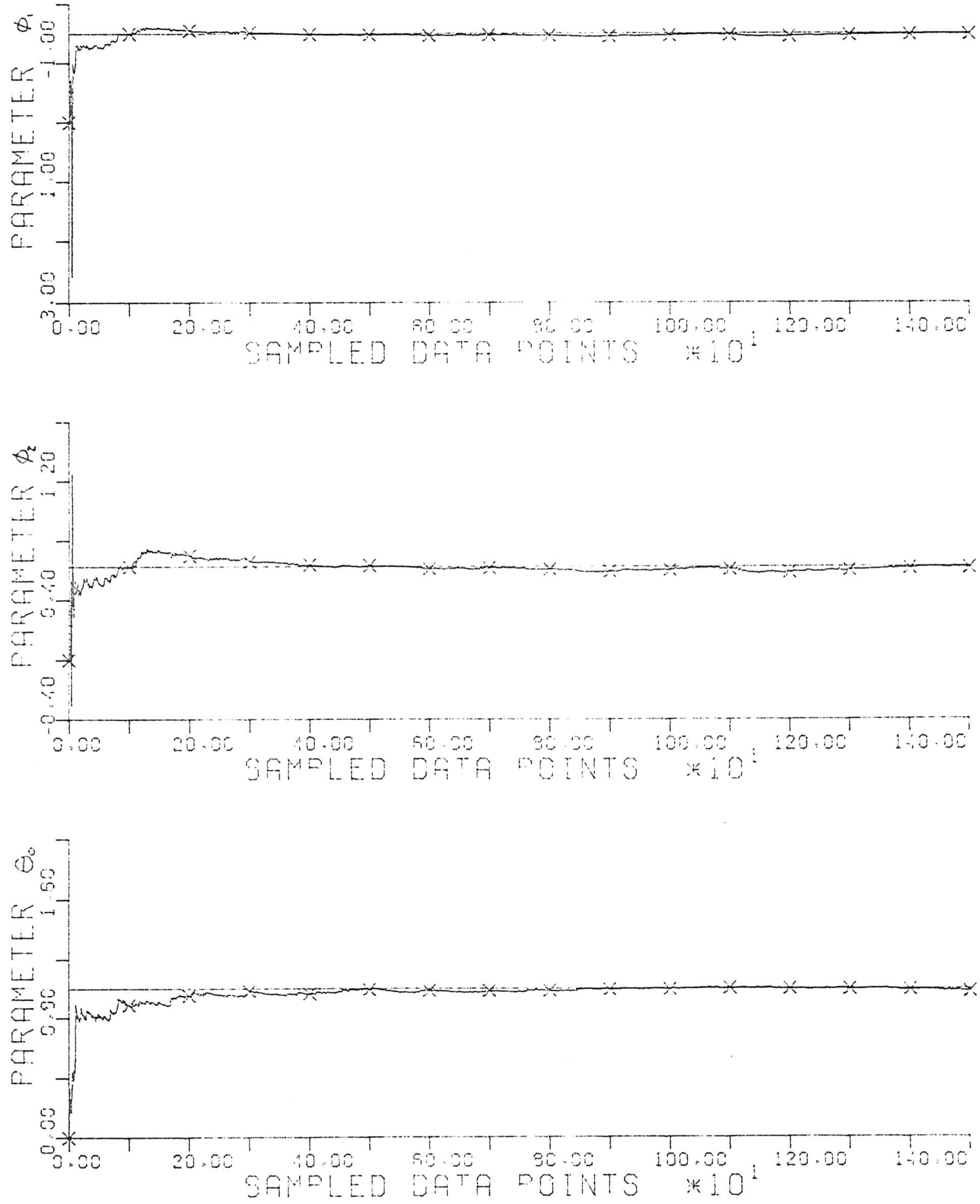


Figure 3.26 PARAMETER ESTIMATION OF AR(2)
USING FINITE MEMORY KALMAN FILTER
WITH WINDOW LENGTH 1000

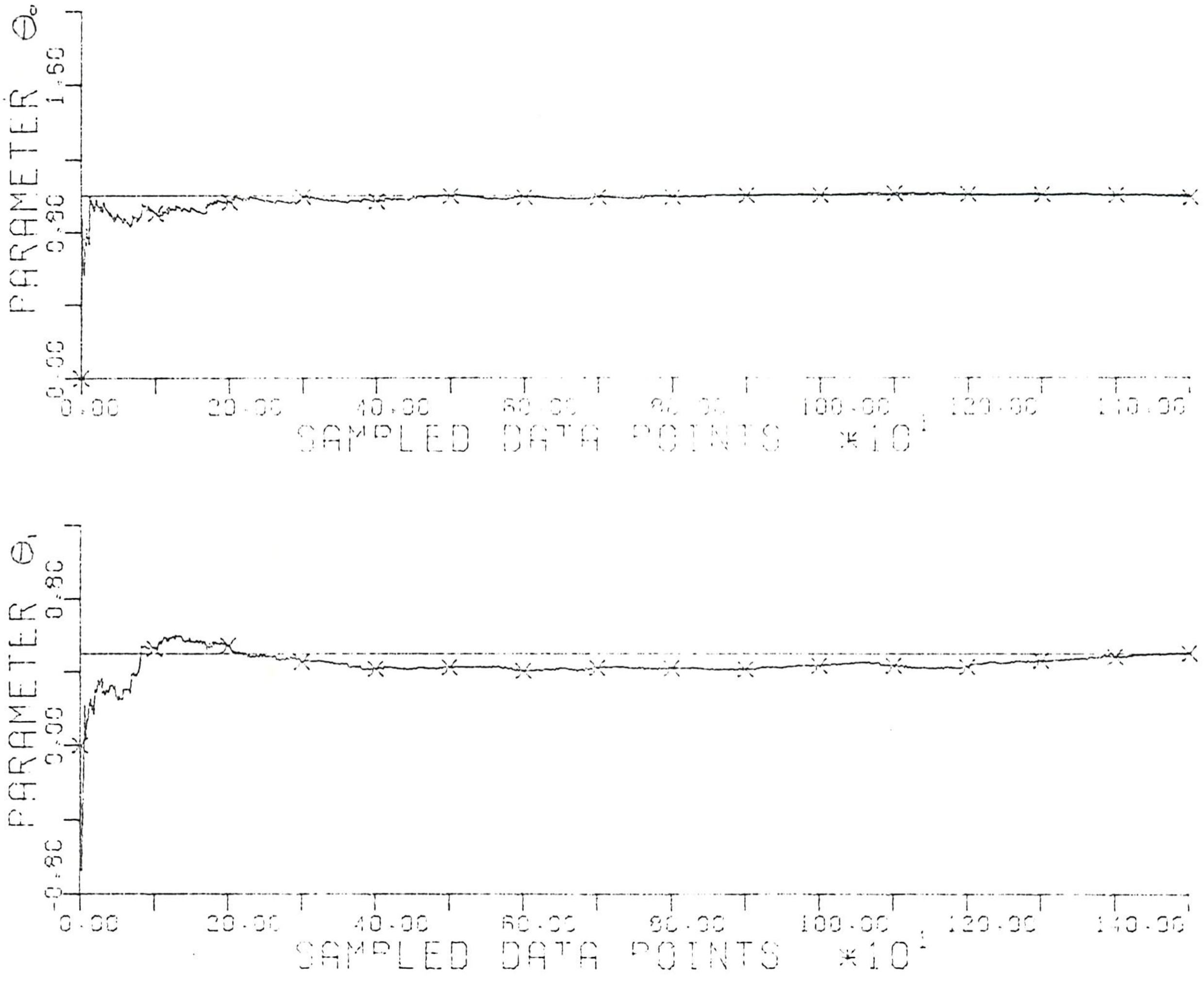


Figure 3.27 PARAMETER ESTIMATION OF MA(1) USING FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH 1000

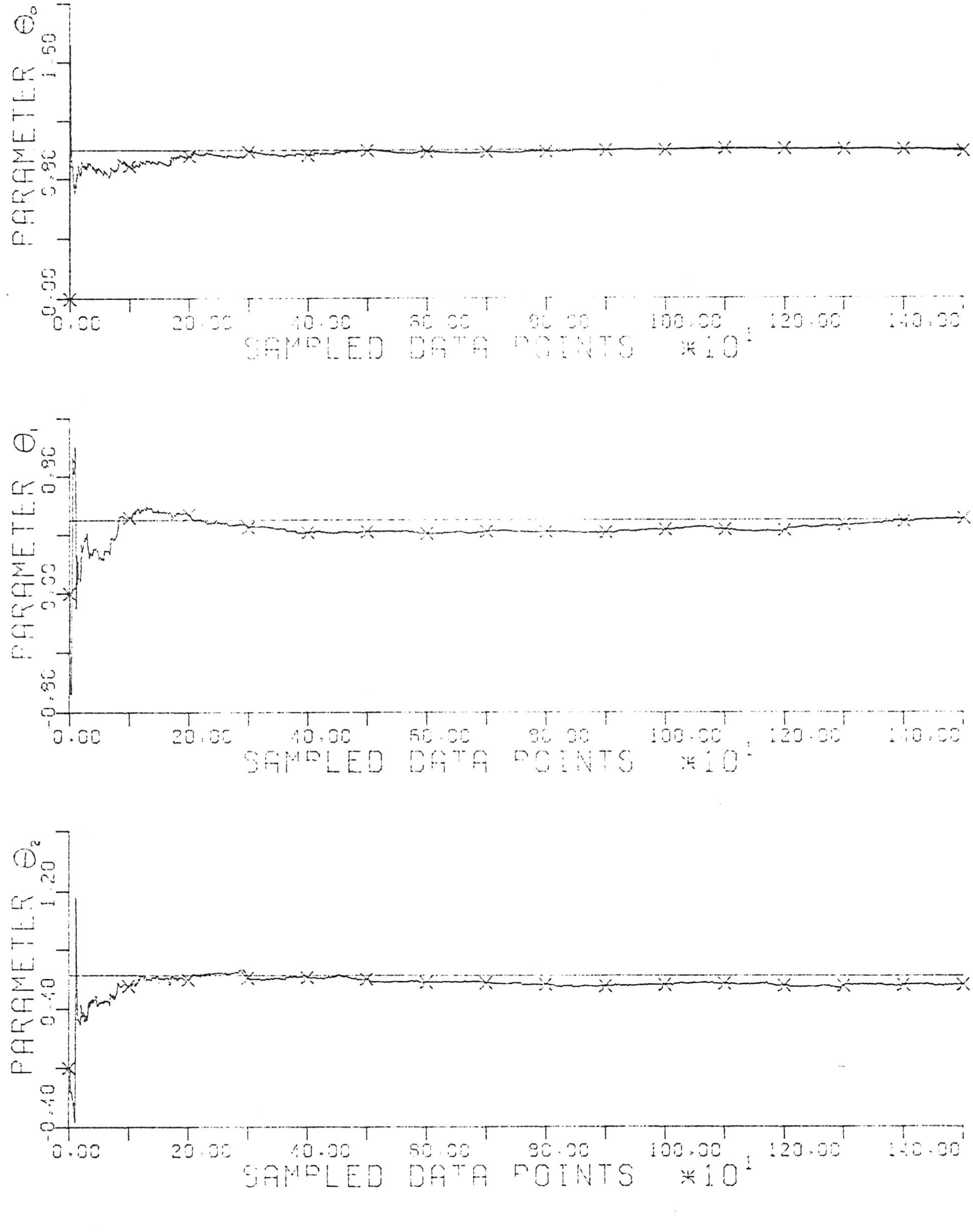


Figure 3.28 PARAMETER ESTIMATION OF MA(2) USING FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH 1000

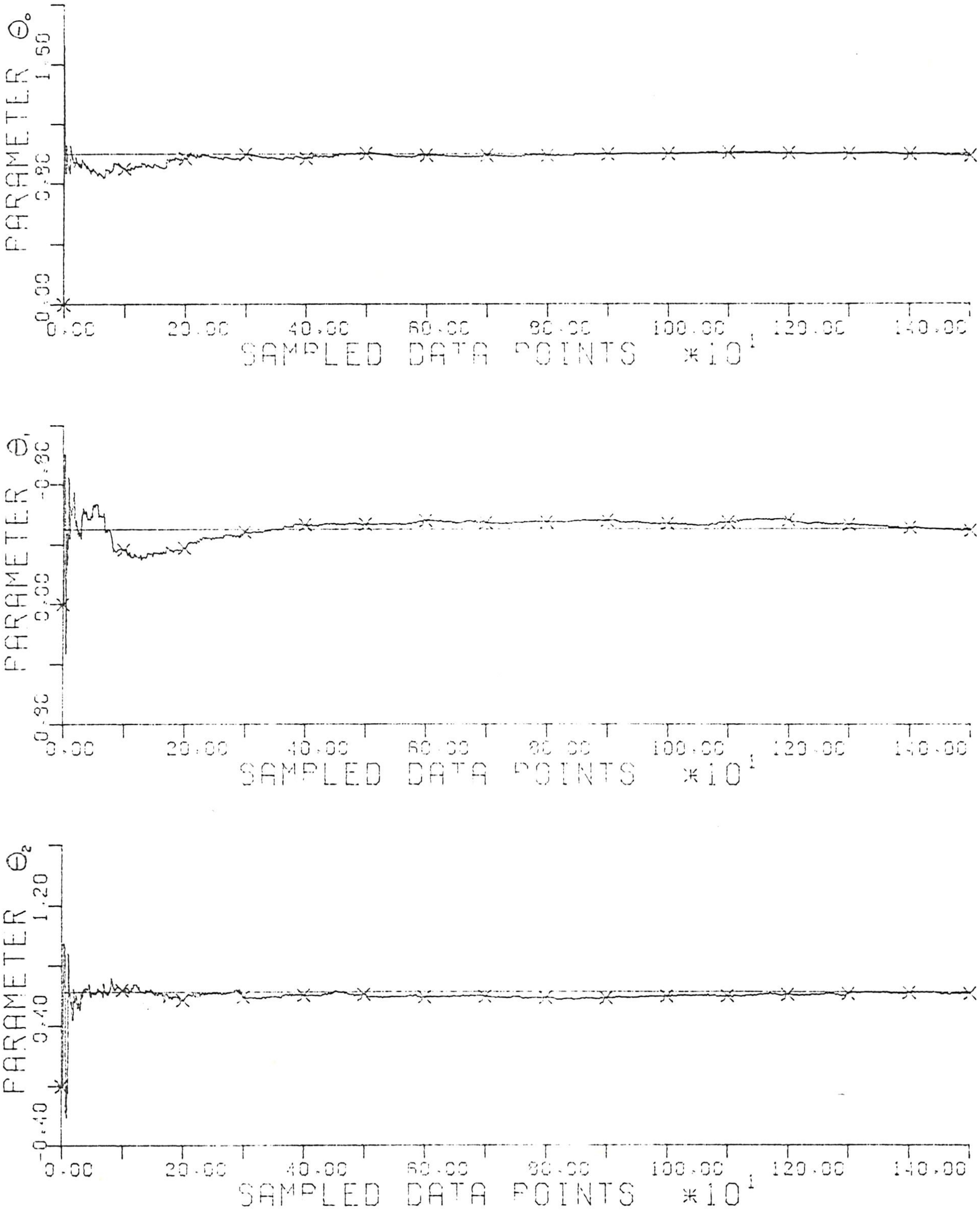


Figure 3.29 PARAMETER ESTIMATION OF MA(2) USING FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH 1000

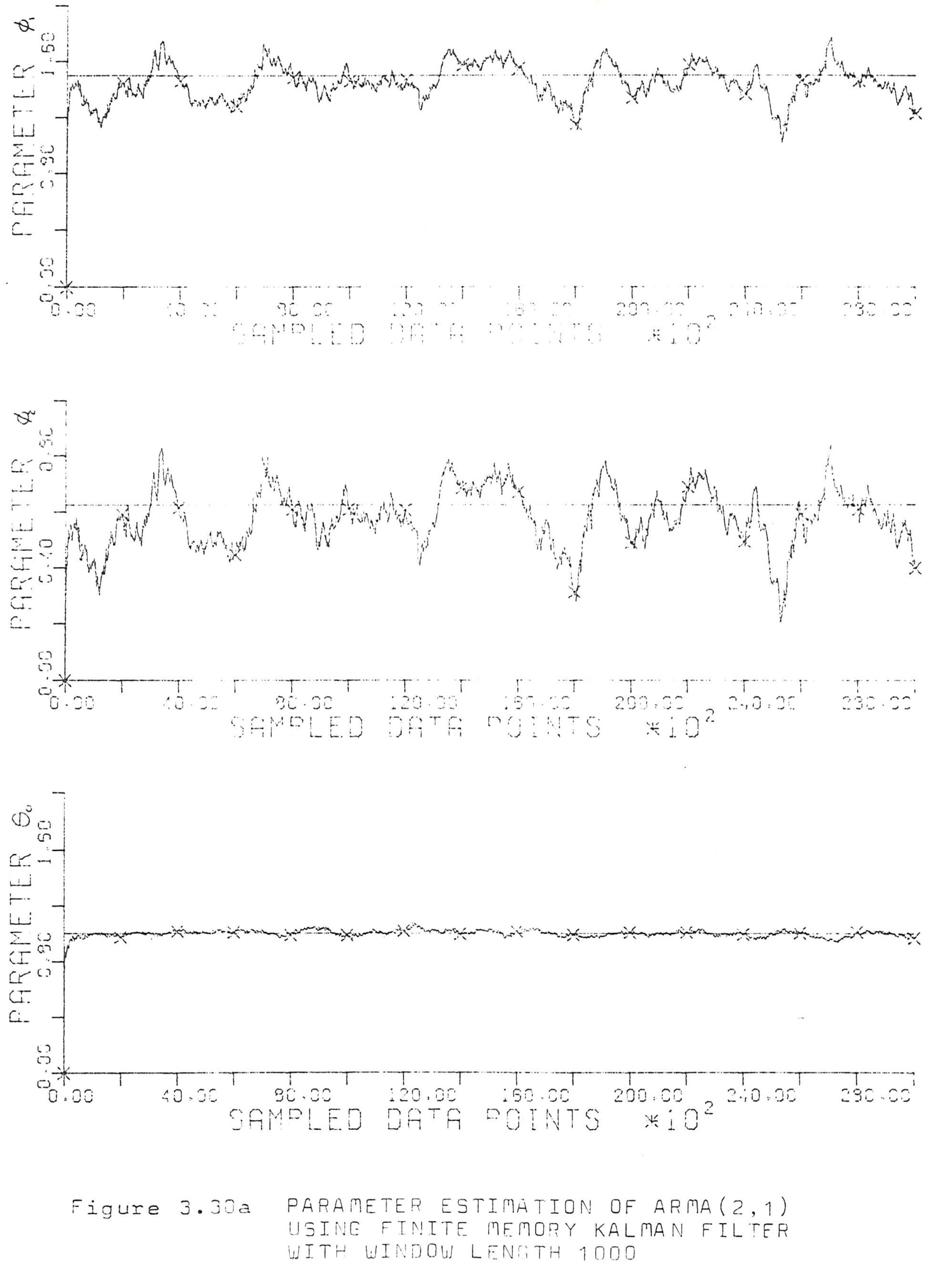


Figure 3.30a PARAMETER ESTIMATION OF ARMA(2,1) USING FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH 1000

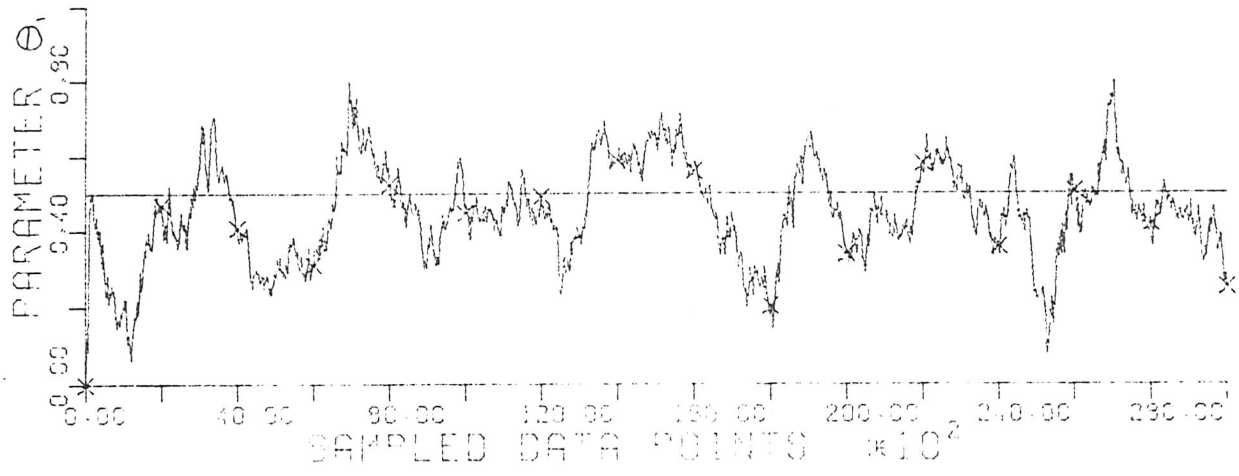

PARAMETER Θ1
0.00
0.40
0.80
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10²

Figure 3.30b

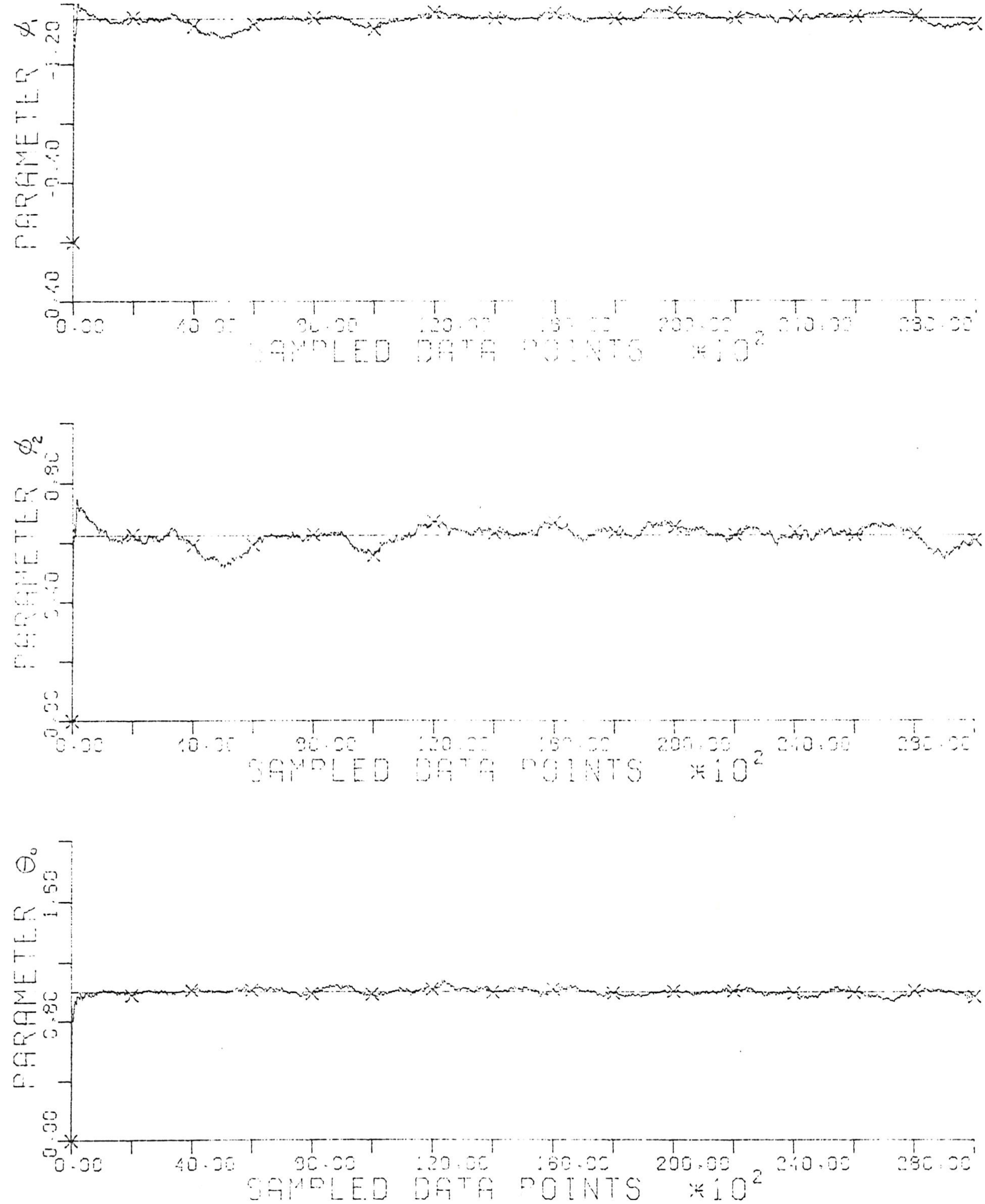


Figure 3.31a PARAMETER ESTIMATION OF ARMA(2,1) USING FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH 1000

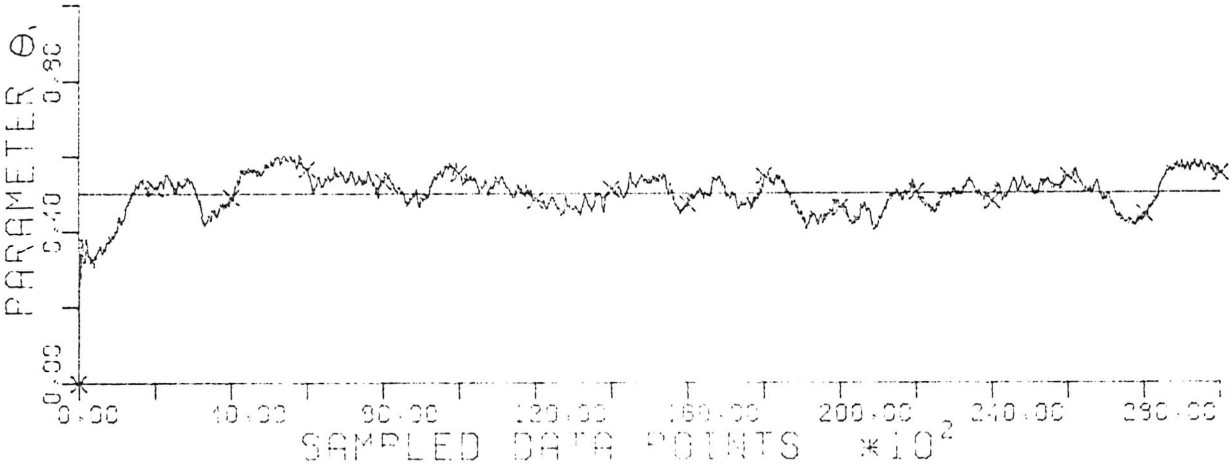

PARAMETER θ1
0.00
0.40
0.80
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10^2

Figure 3.31b

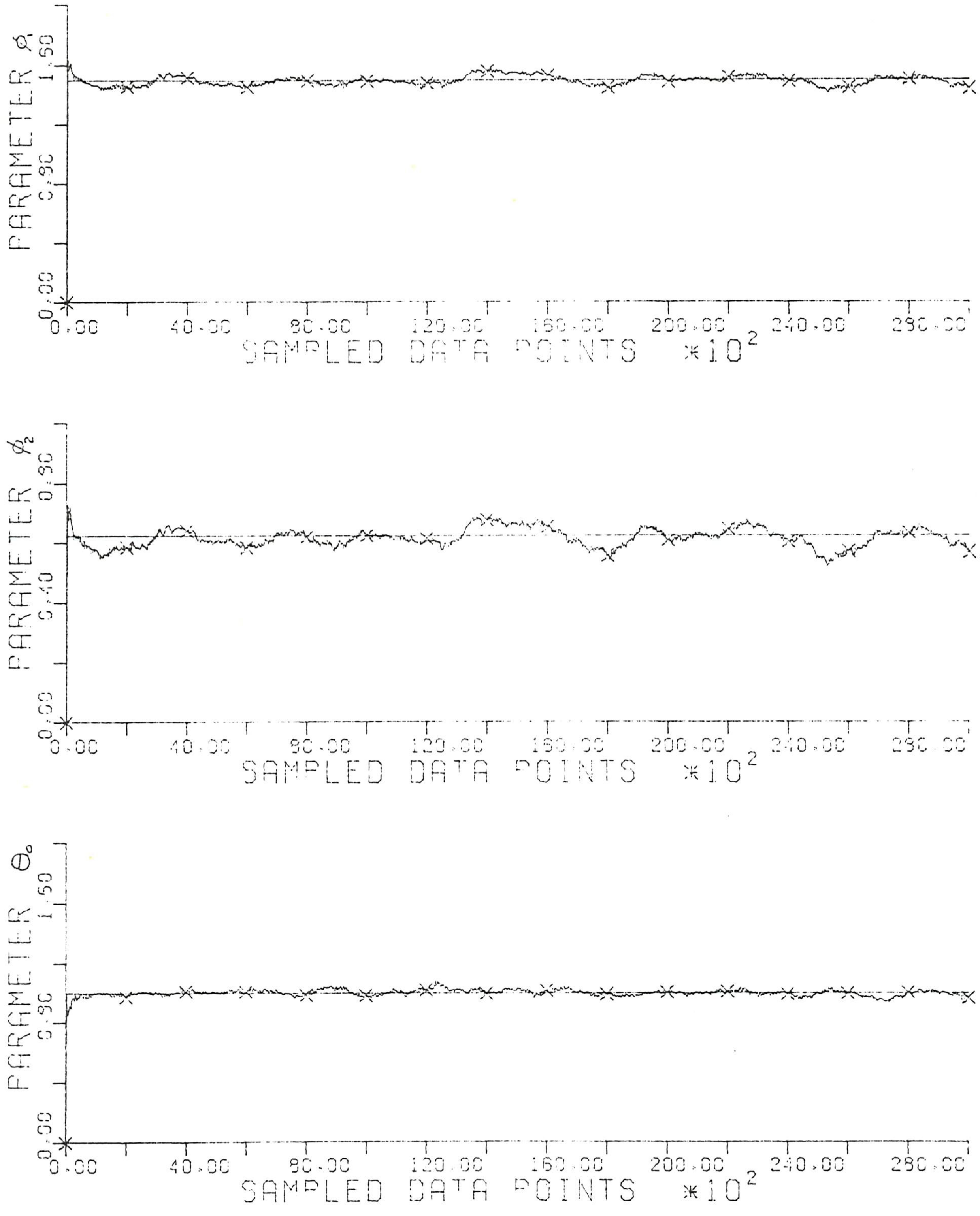


Figure 3.32a PARAMETER ESTIMATION OF ARMA(2,1) USING FINITE MEMORY KALMAN FILTER WITH WINDOW LENGTH 1000

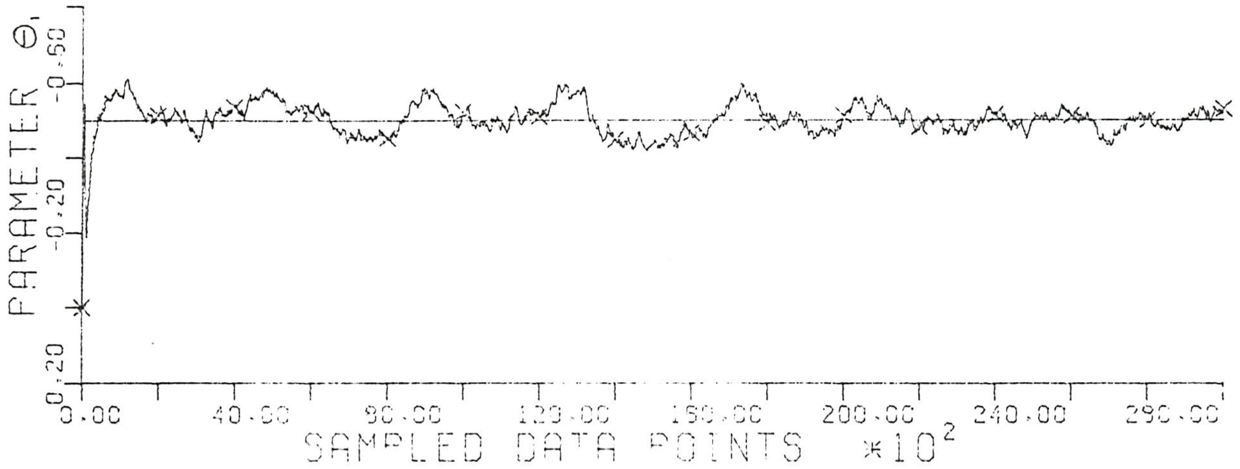

PARAMETER Θ1
-0.60
-0.20
0.20
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10²

Figure 3.32b

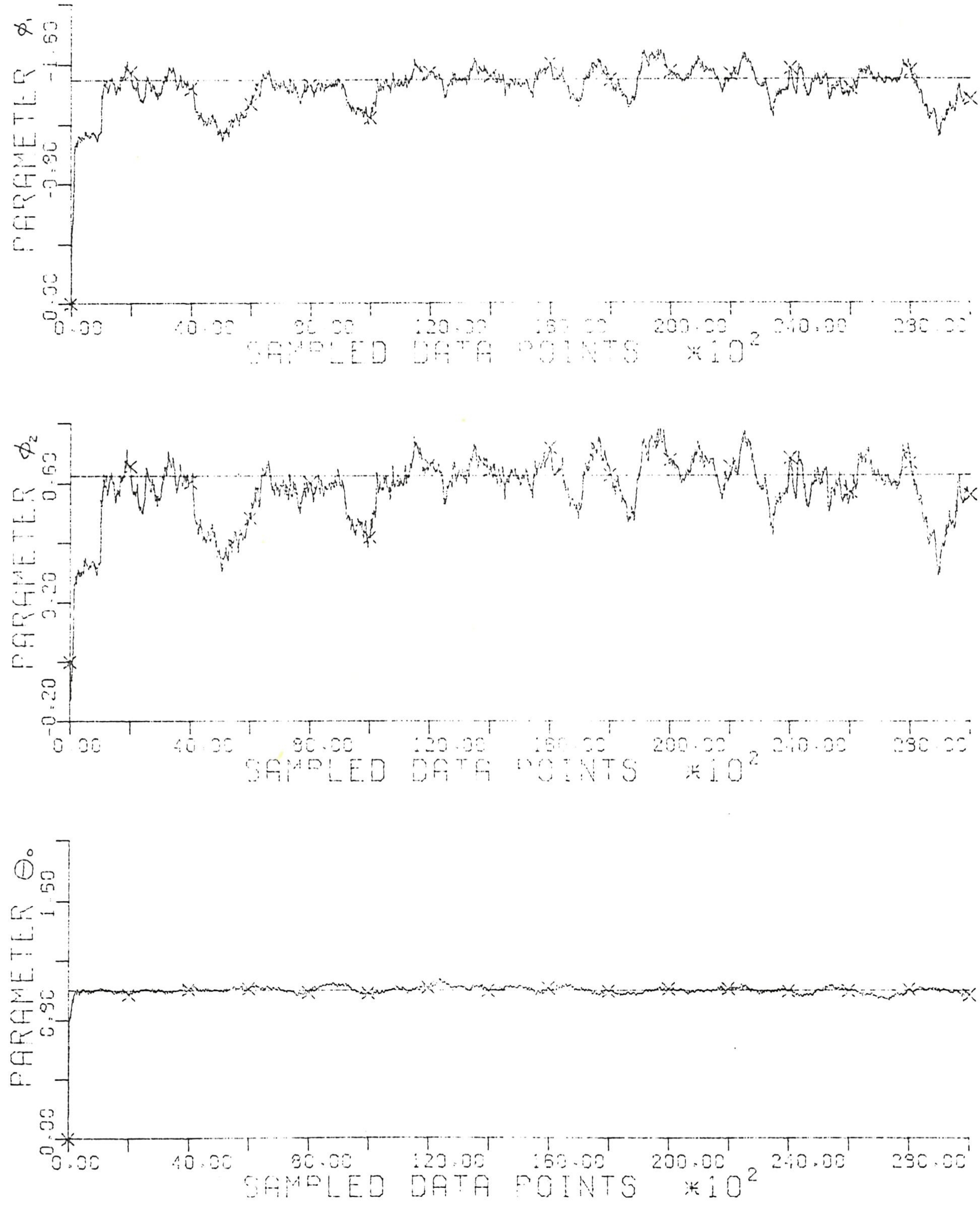


Figure 3.33a PARAMETER ESTIMATION OF ARMA(2,1)
USING FINITE MEMORY KALMAN FILTER
WITH WINDOW LENGTH 1000

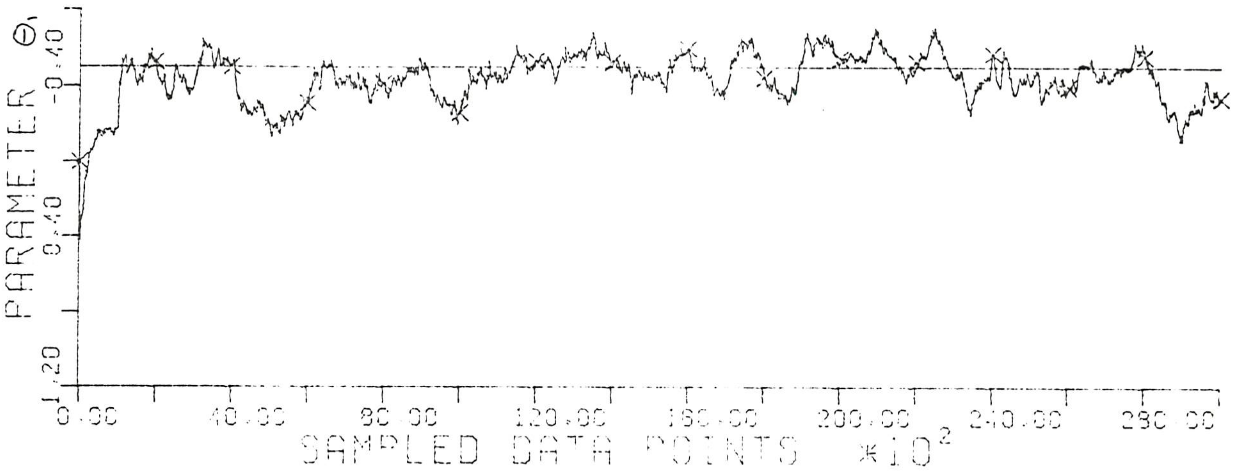

PARAMETER Θ_1
-0.40
0.40
1.20
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10^2

Figure 3.33b

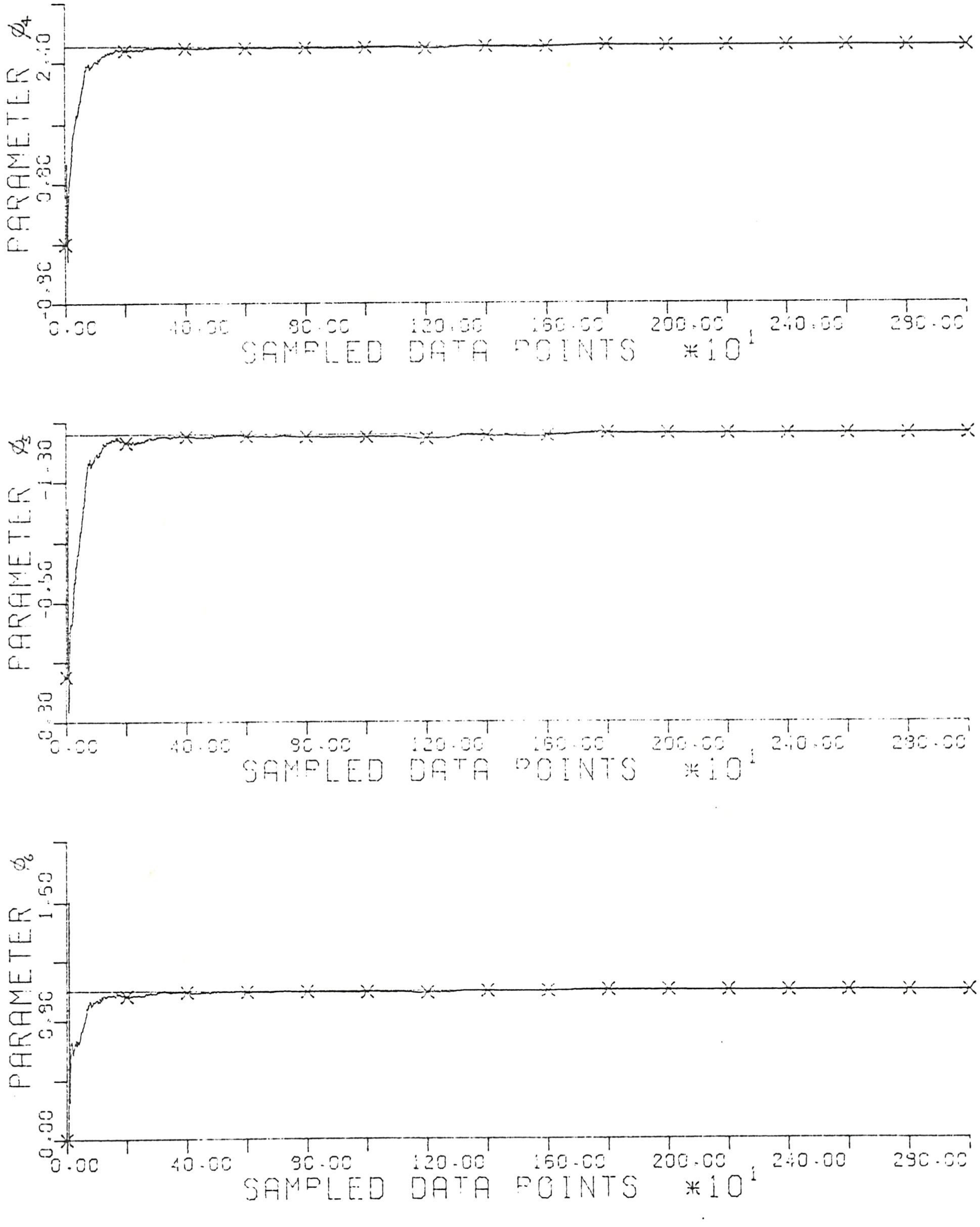


Figure 3.34a PARAMETER ESTIMATION OF AR(6)
USING FINITE MEMORY KALMAN FILTER
WITH WINDOW LENGTH 1000

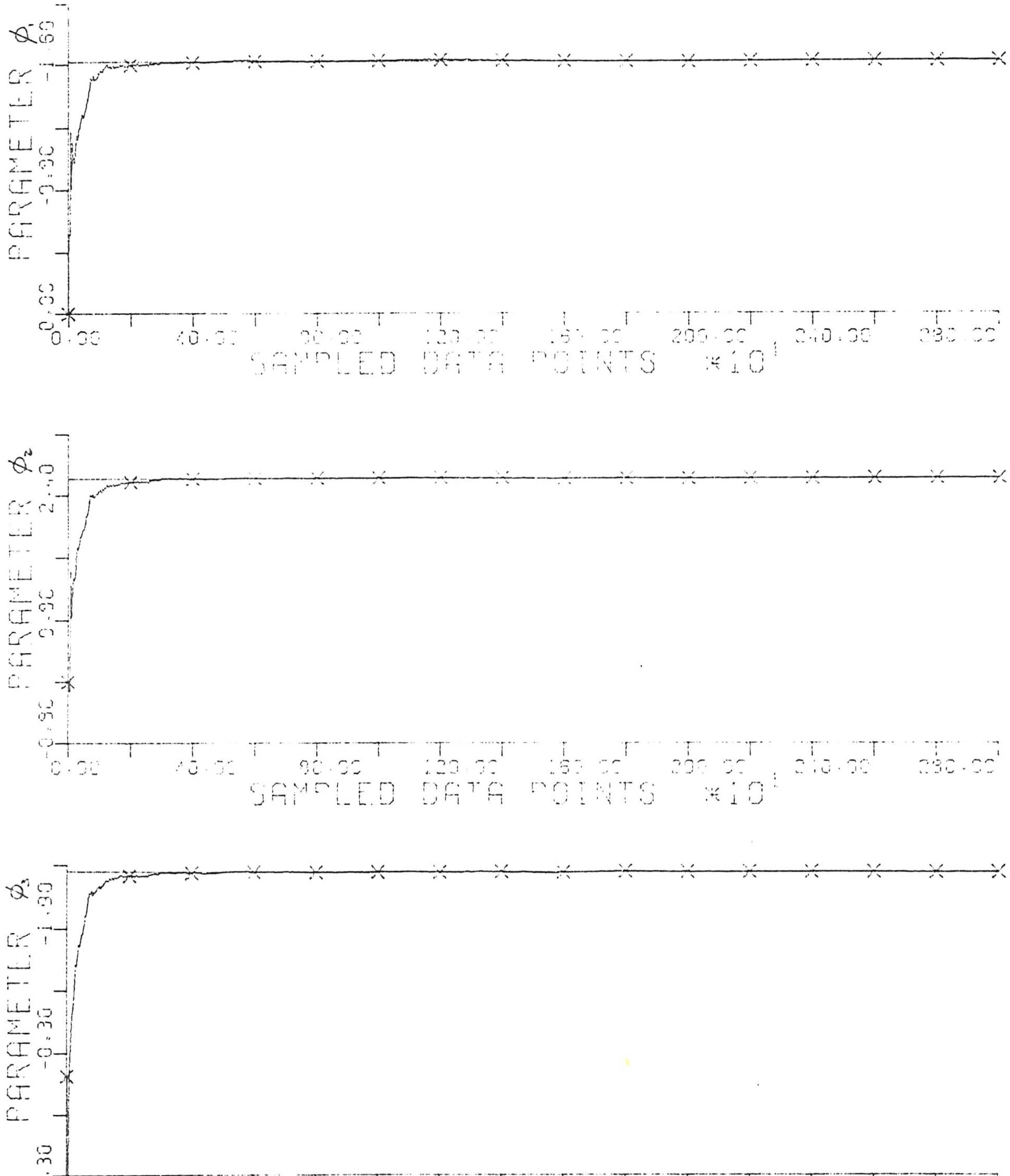

PARAMETER ϕ_1
0.30
-0.30
-1.60
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10¹
PARAMETER ϕ_2
-0.90
0.90
2.40
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10¹
PARAMETER ϕ_3
1.30
-0.30
-1.30
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10¹

Figure 3.34b

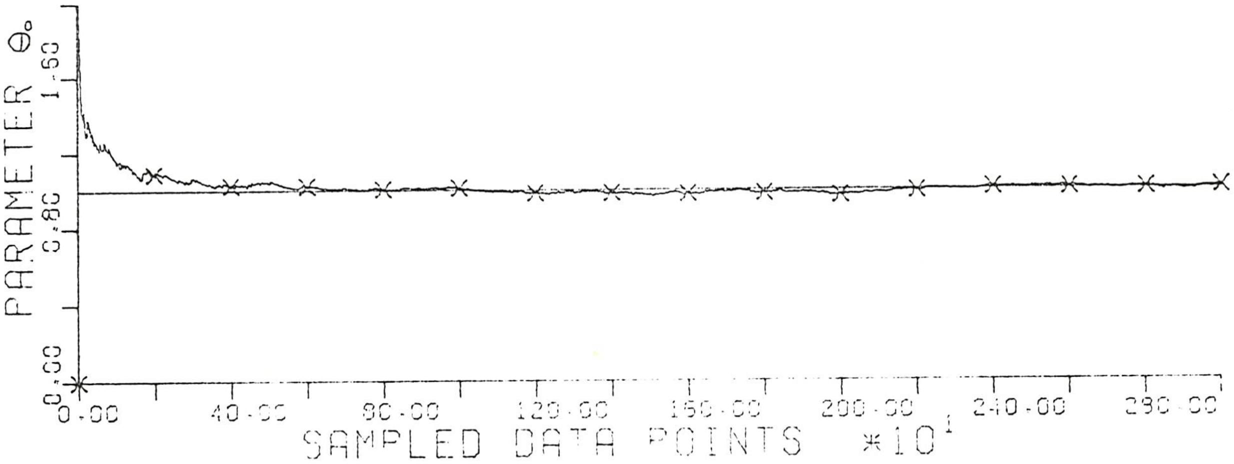

PARAMETER Θ_0
1.60
0.80
0.00
0.00
40.00
80.00
120.00
160.00
200.00
240.00
280.00
SAMPLED DATA POINTS *10¹

Figure 3.34c

CHAPTER 4

FORECASTING

Prediction of the input signal, target motion and ship rolling, plays an important part in deciding the optimum strategy opted for the slew motor control loop. The minimum variance forecast of both motions is developed in this chapter.

4.1 Deterministic Model

In Chapter 2, a constant acceleration model was proposed for the target trajectory. The model is:

$$\underline{x}(k+1) = A(k+1,k) \ \underline{x}(k) \qquad (4.1)$$

$$z(k+1) = \underline{d}^T \ \underline{x}(k+1) + v(k+1) \qquad (4.2)$$

where

$$A(k+1,k) = \begin{pmatrix} 1 & T & \tfrac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \qquad (4.3)$$

$$\underline{d}^T = (1 \quad 0 \quad 0) \qquad (4.4)$$

The best estimate $\underline{\hat{x}}(k/k,\ell)$ is given by the Kalman Filter algorithm described in the last chapter. This is the best estimate of the state at time k based upon ℓ measurements from the k-ℓ to time k inclusive. The Orthogonal Projection Lemma [32]; states that the linear minimum variance estimate of the state based on the measurements is the orthogonal projection of the state onto the linear measurement space. Denoting the measurement space by Y(k), the estimate $\underline{\hat{x}}(k/k,\ell)$ is in the space of Y(k), i.e. $\underline{\hat{x}}(k/k,\ell) \in Y(k)$. The estimation error, $\underline{\tilde{x}}(k/k,\ell)$ is orthogonal to the vector space Y(k). Thus, if $\underline{y}$ is any vector in Y(k);

$$E\left[\underline{\tilde{x}}^T(k/k,\ell) \ \underline{y}\right] = 0 \qquad (4.5)$$

The predicted target position at time k+p will be:

$$\underline{\hat{x}}(k+p/k,\ell) = A(k+p,k) \ \underline{\hat{x}}(k/k,\ell) \qquad (4.6)$$

Since $\hat{\underline{x}}(k/k,\ell)$ belongs to the measurement space Y(k) any linear transformation of $\hat{\underline{x}}(k/k,\ell)$ will also be belonging to the vector space Y(k). $\hat{\underline{x}}(k+p/k,\ell)$ in (4.6) is truely a linear transformation of $\hat{\underline{x}}(k/k,\ell)$ and hence belongs to the vector space Y(k), too.

Considering any vector $\underline{y}$ which belongs to the vector space Y(k) and defining T as:

$$T \;=\; E\left[\{\underline{x}(k+p) - A(k+p,k)\;\; \hat{\underline{x}}(k/k,\ell)\}^T\; \underline{y}\,\right] \tag{4.7}$$

from (4.1)

$$\underline{x}(k+p) \;=\; A(k+p,k)\;\; \underline{x}(k) \tag{4.8}$$

and the decomposition of $\underline{x}(k)$

$$\underline{x}(k) \;=\; \hat{\underline{x}}(k/k,\ell) \;+\; \tilde{\underline{x}}(k/k,\ell) \tag{4.9}$$

(4.7) becomes:

$$T \;=\; E\left[\tilde{\underline{x}}^T(k/k,\ell)\;\; A^T(k+p,k)\;\; \underline{y}\,\right] \tag{4.10}$$

The term $A^T(k+p,k)\underline{y}$ is a linear transformation of $\underline{y}$. $A^T(k+p,k)\underline{y}$ is in the vector space Y(k), but $\tilde{\underline{x}}(k/k,\ell)$ is orthogonal to the vector space Y(k). Thus, T in (4.10) is zero showing that $\hat{\underline{x}}(k+p/k,\ell)$ in (4.6) is an orthogonal projection of $\underline{x}(k+p)$ onto the measurement space Y(k). The prediction $\hat{\underline{x}}(k+p/k,\ell)$ is also a minimum variance forecast because of the orthogonal projection lemma. The covariance matrix of the forecast is:

$$\begin{aligned} \mathrm{Var}\;(\hat{\underline{x}}(k+p/k,\ell)) \;&=\; \mathrm{Var}\;(A(k+p,k)\;\hat{\underline{x}}(k/k,\ell)) \\ &=\; A(k+p,k)\;\; P(k/k,\ell)\;\; A^T(k+p,k) \end{aligned} \tag{4.11}$$

Therefore:

$$\hat{z}(k+p) \;=\; \underline{d}^T\;\; \hat{\underline{x}}(k+p/k,\ell) \tag{4.12}$$

and

$$\mathrm{Var}(\hat{z}(k+p)) \;=\; \underline{d}^T\, A(k+p,k)\; P(k/k,\ell)\;\; A^T(k+p,k)\; \underline{d} \tag{4.13}$$

The variance of the predicted target position increases to the fourth power of the lead time p.

Assuming the measurement noise v(k) in (4.2) is gaussian, the measurement z(k) will also be gaussian. Any linear combination of a set of gaussian distributed random variables is also gaussian. Hence, the estimate $\underline{\hat{x}}(k/k,\ell)$ and the forecast $\underline{\hat{x}}(k+p/k,\ell)$, which are the linear combination of the measurements, are gaussian, too. Consequently the forecast $\hat{z}(k+p)$ might be treated as gaussian.

One must be cautious that the above statements are made on the assumption that the model fits the target trajectory exactly. In fact, this is not the case. Converting the constant acceleration model back to polynomial form,

$$z(k+p) = a_o + a_1(k+p) + a_2(k+p)^2 \qquad (4.14)$$

But, as presented in Chapter 2, the target trajectory will only be represented exactly by an infinite power series, i.e.

$$z^*(k+p) = a_o + a_1(k+p) + a_2(k+p)^2 + \ldots \qquad (4.15)$$

Thus, the true prediction error is:

$$\begin{aligned} \tilde{z}^*(k+p) &= z^*(k+p) - z(k+p) + z(k+p) - \hat{z}(k+p) \\ &= 0\left[(k+p)^3\right] + \tilde{z}(k+p) \\ &= \tilde{a}_o + \tilde{a}_1(k+p) + \tilde{a}_2(k+p)^2 + 0\left[(k+p)^3\right] \qquad (4.16) \end{aligned}$$

The estimates from the Kalman Filter algorithm are then not truely unbiased. Therefore, the forecast is biased with $\tilde{z}^*(k+p)$. (4.13) can only be viewed as the lower bound variance of the forecast. The bias can however, be reduced by using short window length.

It is then necessary to emphasise that the lead time, p, should not be too large. Otherwise, the bias of the forecast will be too large. Furthermore, the confidence on the forecasts reduces quickly with time as can be noted from the expression of the lower bound variance.

4.2 Stochastic Model

The prediction of ARMA type models has been thoroughly discussed by Box and Jenkins. It is briefly described in this section.

An ARMA process can always be expressed as an infinite order moving average (MA) process.

$$z(k) = \theta_o u(k) + \theta_1 u(k-1) + \ldots + \theta_n u(k-n) + \ldots \tag{4.17}$$

or as an infinite order autoregressive (AR) process:

$$z(k) + \phi_1 z(k-1) + \ldots + \phi_m z(k-m) + \ldots = \theta_o u(k) \tag{4.18}$$

Let the present time be t, a forecast at time t+p which is to be a linear function of current and previous measurements z(t), z(t-1), z(t-2), . . ., will also be a linear function of current and previous random inputs u(t), u(t-1), u(t-2)....

Suppose the best forecast at t+p is:

$$\hat{z}(t+p) = \hat{\theta}_p u(t) + \hat{\theta}_{p+1} u(t-1) + \hat{\theta}_{p+2} u(t-2) + \ldots \tag{4.19}$$

Using (4.17), the prediction error is:

$$\tilde{z}(t+p) = z(t+p) - \hat{z}(t+p)$$

$$= \theta_o u(t+p) + \theta_1 u(t+p-1) + \ldots + \theta_{p-1} u(t+1)$$

$$+ \sum_{j=o}^{\infty} \{ \theta_{p+j} - \hat{\theta}_{p+j} \} u(t-j) \tag{4.20}$$

Using the assumption that u(t) is a zero mean white noise and

$$E\left[u^2(t)\right] = \sigma_u^2 \tag{4.21}$$

The variance of the prediction error is:

$$\mathrm{Var}\left[\tilde{z}(t+p)\right] = (\theta_o^2 + \theta_1^2 + \ldots + \theta_{p-1}^2) \sigma_u^2 \tag{4.22}$$

$$+ \sum_{j=o}^{\infty} \{\theta_{p+j} - \hat{\theta}_{p+j}\}^2 \sigma_u^2$$

It will be minimised by setting $\hat{\theta}_{p+j} = \theta_{p+j}$. Then the optimum prediction error is:

$$\tilde{z}(t+p) = \theta_o u(t+p) + \theta_1 u(t+p-1) + \ldots + \theta_{p-1} u(t+1) \qquad (4.23)$$

Then:

$$\hat{z}(t+p) = \theta_p u(t) + \theta_{p+1} u(t-1) + \ldots$$

$$= E\left[z(t+p/t)\right] \qquad (4.24)$$

where $E\left[z(t+p/t)\right]$ denotes the conditional expectation of z(t+p) given knowledge of all the z's up to time, t. Thus, the minimum variance forecast for time t+p is the conditional expectation of z(t+p) at time t.

$$\text{Since } E\left[\tilde{z}(t+p)\right] = 0 \qquad (4.25)$$

the forecast is unbiased. The variance of the prediction error is then:

$$\text{Var}\,(\tilde{z}(t+p)) = (\theta_o^2 + \theta_1^2 + \ldots + \theta_{p-1}^2)\,\sigma_u^2 \qquad (4.26)$$

From (4.23), it shows that:

$$u(t+1) = \tilde{z}(t+1) / \theta_o. \qquad (4.27)$$

Once again, the forecast is unbiased and has minimum variance only if the order of the model matches the time series. A low order AR model may still give a small bias if the ship rolling motion is closely an invertable ARMA model.

The simplest and most efficient way to obtain a forecast for time t+p at current time, t, is through the difference equation of the model. In the case of an AR(m) model (4.18) is used. The measurements z(t-j), j=0, 1, 2, . . . , m, which have already happened are left unchanged. The measurements, z(t+j), j=1, 2, . . ., p-1, which have not yet happened are replaced by their forecast z(t+j). The random input u(t+p) which have not happened is replaced by its expected value, zero. The process can then be predicted recursively into the future. The lower bound variance is available from (4.26). The coefficients θ_o, θ_1, . . ., can easily be evaluated from (4.28) recursively.

$$\theta_o = 1/\phi_o$$

$$\theta_1 = (-\phi_1 \theta_o)/\phi_o$$

$$\theta_2 = (-\phi_2 \theta_o - \phi_1 \theta_1)/\phi_o$$

$$\theta_j = -(\sum_{i=1}^{j} \phi_i \theta_{j-i})/\phi_o \quad ; \quad 1 \leqslant j \leqslant m$$

and $$\theta_j = -(\sum_{i=1}^{m} \phi_i \theta_{m-i})/\phi_o \quad ; \quad j>m \qquad (4.28)$$

4.3 Conclusions

In this chapter, the minimum variance predictions for the target trajectory and the ship rolling motion were shown based on the chosen models, constant acceleration model and autoregressive model. These forecasts are normally biased due to the unmatching models. Thus, the target trajectory and the ship rolling motion are predicted in a sub-optimal way. The significance of the biases can be reduced by fitting the model to the motions over a short period of time, and also by not predicting the motions too far ahead. The latter restriction is not unrealistic. If the motions are predicted too far into the future, the variance on the forecasts will become so high that there is no point of considering them in the determination of the control strategy.

CHAPTER 5

DECISION MAKING

The strategy proposed for the novel dual drive tracking servomechanism was described in Section 1.2. It reduces the complexity of the novel system to a single drive system most of the time. The slew motor will only be put into action if and only if there is a high possibility of losing the target. The main theme is to realign the slew motor as infrequently as possible while keeping the tracking error down to a minimum. The decision making process is to determine the optimal time and optimal position for the slew motor if realignment is required. The word "optimal" has no absolute meaning. It implies that there must be some kind of measure based on which the decision is superior to all the other alternatives. A decision which is optimum in one measure may not necessarily be optimum in another measure. Thus, the choice of the measure must be closely related to the particular application. The measure must be able to reflect the relative importance of the various objectives. In the case of the dual drive tracking servomechanism, the criteria in determining the optimal strategy for the slew motor are to track the target at high accuracy and minimum interference from the slew motor. The interference arises from the combination of the proposed piggy-back configuration and the fact that the slew motor has a lower torque characteristic, and hence a lower bandwidth, than the tracking motor. If the output shaft of the slew motor is not clamped by external means, the tracking motor will not have a firm base to react any generated torque to its inertial load. The proposed method of eliminating the interference is to apply a brake on the slew motor output shaft when it is de-activated.

The non-linearity of the proposed cost funcation makes a complete analytical solution to the optimisation very difficult. In Section 5.2, an approximation is available for the optimal position of the slew motor between two realignments. Alternatively, if the number of realignments is constrained to one (Section 5.3) the optimal amount of realignment, but not the optimal time, can be expressed analytically. The proof of this expression is by mathematical induction from an approximate solution.

5.1 Proposed Cost Function

One possible cost function based on the previously mentioned criteria was proposed:

$$J = \int_{t_o}^{t_p} w_1(t) \; \{ \; 1 - \Pr\,(\theta_s(t) - \theta_\ell \leqslant \theta_T(t) \leqslant \theta_s(t) + \theta_\ell) \; \} \; dt$$

$$+ \int_{t_o}^{t_p} w_2(t) \{ \mathrm{sign}(\theta_s^+(t) - \theta_s^-(t))\}^2 \, dt \qquad (5.1)$$

where

$$\Pr (a \leqslant x \leqslant b) = \int_a^b p(x) \, dx$$

$$p(x) = \text{probability density function of } x$$

and

$$\mathrm{sign}(x) = \begin{cases} -1 & x<0 \\ 0 & x=0 \\ 1 & x>0 \end{cases}$$

The proposed cost function only considers the predicted trajectory of the target relative to the ship, $\theta_T(.)$, over a prediction period ahead from the current time, t_o, to the end of the period, t_p. The first definite integral is the weighted sum of the probability of target loss over the period t_o to t_p. The second integral, which involves the sign function, relates to the number of realignments within the same period. The term $\theta_s^+(t)$ is the slew motor position at a time interval infinitesimal small after the time t and $\theta_s^-(t)$ is the slew motor position at a time interval infinitesimal small before the time t. The appearance of the sign function is due to the fact that only the number of realignments is important. The amount of each realignment is of no importance from the point of interference once the time of realignment is fixed. The terms $w_1(.)$ and $w_2(.)$ are time variant weighting factors on the probability of target loss and number of realignments, respectively.

The weighting factors $w_1(.)$ and $w_2(.)$ put different emphasis on the two objectives: probability of target loss and frequency of realignments, according to the requirements of the system. A target will first appear on the radar when it is still very far away. During this stage, the accuracy on tracking the target may not be so crucial. Should the target be lost during this time there will be sufficient time to re-establish the presence of the target on the radar. In other words, frequent re-alignments of the slew motor is acceptable. As time progresses, with the target moving closer and closer the risk of losing the target is becoming progressively greater. A lot of damage may be done if the target were lost during the second stage, when the target is extremely close to the radar, the driving system of the antenna is no longer capable of the high rate of movement to keep up with the target. Even the novel drive will not have the physical characteristics required to cope with the situation in the final stage. Hence, the performance of the tracking servomechanism is not considered during the closing stage of the tracking. The profile of the acceptable risk in the first and second stages of an engagement with a target may be treated as an exponential function as shown in Figure 5.1. The weighting factors $w_1(.)$ and $w_2(.)$ will then increase exponentially in the forms of:

$$w_1(\tau) = Q \left[1 - C_1 \exp(-a_1 \tau)\right] \tag{5.2}$$

$$w_2(\tau) = R \left[1 - C_2 \exp(-a_2 \tau)\right] \tag{5.3}$$

where the origin of τ is the time of the target's initial appearance on the radar.

The control variable of the cost function is the predicted trajectory of the slew motor during the prediction period. The resulting trajectory for the slew motor will not be a continuous function. An idealised slew motor trajectory will be of the form as shown in Figure 5.2. The slew motor is assumed to be able to perform an ideal step movement. The anticipated slew motor trajectory is a summation of a series of step functions occuring at different, unevenly spaced, time intervals. The optimisation problem is to determine the number of realignments within the prediction period, and the timing and amount of each realignment. To the knowledge of the author, this cost function is unique to this work. No optimisation of this form has yet been treated. Due to the peculiar non-linearity and the uncertainty of the degrees of freedom on the control variable, it is envisaged that an analytical solution is not feasible. It will therefore be necessary to resort to numerical methods. Before an attempt on the numerical solution to the problem, the characteristics of the proposed cost function with certain assumptions were studied. It gave some insight to the behaviour of the cost function.

5.2 Optimal Position for Slew Motor between Two Realignments

In Figure 5.2, if the times of two consecutive realignments, t_1, and t_2, were known, the slew motor position between these two realignments must be optimal. The cost function for the time intervals concerned is:

$$J = \int_{t_1}^{t_2} w(t) \left\{ 1 - \Pr\left(\theta_s(t) - \theta_\ell \leqslant \theta_T(t) \leqslant \theta_s(t) + \theta_\ell\right)\right\} dt \tag{5.4}$$

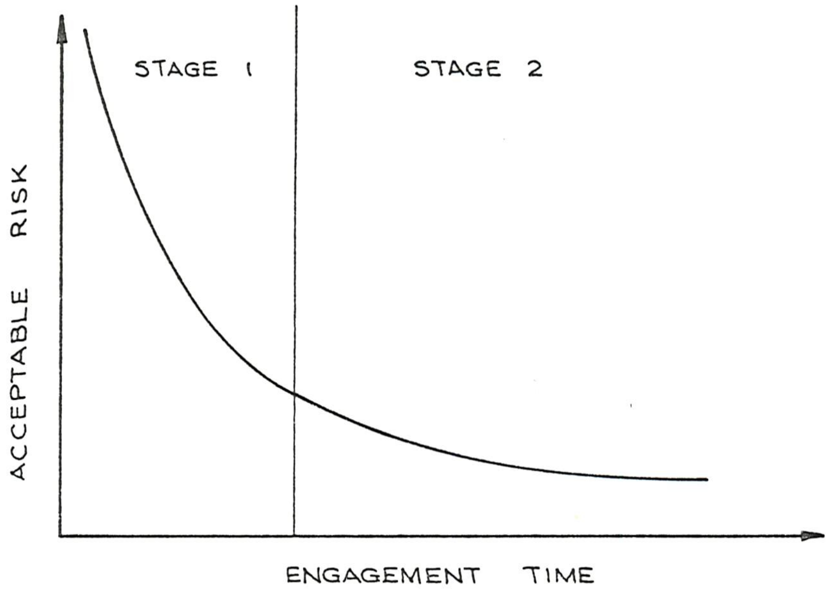


Figure 5.1 Profile of acceptable risk

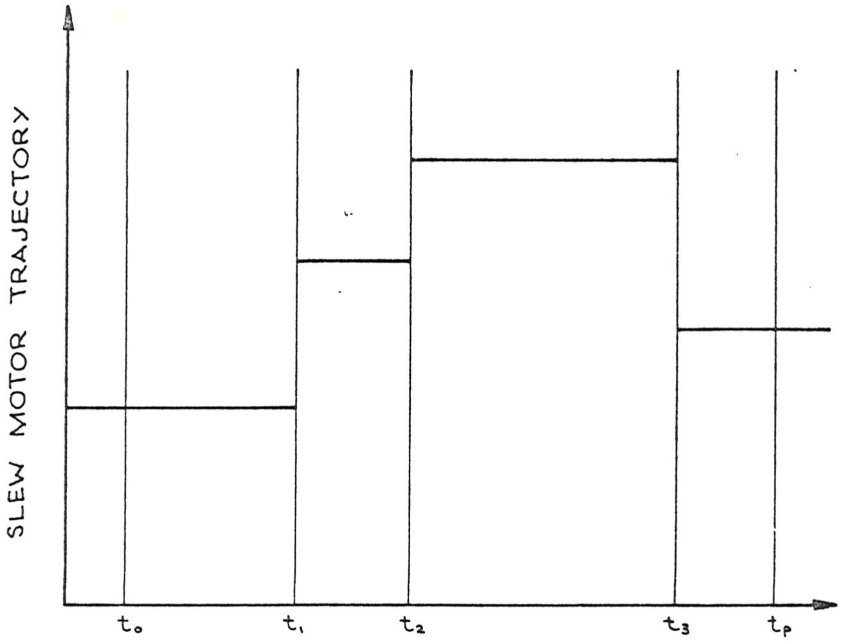


Figure 5.2 Idealised slew motor trajectory

As it is reasonable to assume that the measurement noise in the signal is gaussian. Together with the fact that the estimates obtained from the finite memory Kalman filter are linear combinations of the measured data, the predicted signal will also be gaussian. Therefore, the probability density function of being on target at time t is:

$$p(\theta_T(t)) = \frac{1}{\sqrt{2\pi}\,\sigma(t)} \exp\left\{ \frac{-(\theta_T(t) - \hat{\theta}_T(t))^2}{2\,\sigma^2(t)} \right\} \tag{5.5}$$

where $\hat{\theta}_T(t)$ and $\sigma^2(t)$ are the mean and variance of the target position at time t. Thus (5.4) becomes:

$$J = \int_{t_1}^{t_2} w(t) \left\{ 1.0 - \int_{\theta_s-\theta_\ell}^{\theta_s+\theta_\ell} p(\theta_T(t))\, d\theta_T(t) \right\} dt \tag{5.6}$$

The optimal slew motor position, θ_s*, is obtained by minimising (5.6), i.e. setting the derivative w.r.t. θ_s to zero.

$$\frac{dJ}{d\theta_s} = 0 \tag{5.7}$$

Differentiating (5.6) w.r.t. θ_s gives:

$$\frac{dJ}{d\theta_s} = - \int_{t_1}^{t_2} w(t)\left\{ p(\theta_T(t))\big|_{\theta_s+\theta_\ell} - p(\theta_T(t))\big|_{\theta_s-\theta_\ell} \right\} dt \tag{5.8}$$

Hence if θ_s is optimal:

$$\int_{t_1}^{t_2} w(t)\; p(\theta_T(t))\big|_{\theta_s*+\theta_\ell}\, dt = \int_{t_1}^{t_2} w(t)\; p(\theta_T(t))\big|_{\theta_s*-\theta_\ell}\, dt \tag{5.9}$$

Substituting (5.5) into (5.9)

$$\int_{t_1}^{t_2} \frac{w(t)}{\sqrt{2\pi}\ \sigma(t)} \quad \exp\ \{ -\tfrac{1}{2} \left(\frac{\theta_s^* + \theta_\ell - \theta_T(t)}{\sigma(t)}\right)^2 \}\ dt$$

$$= \int_{t_1}^{t_2} \frac{w(t)}{\sqrt{2\pi}\ \sigma(t)} \quad \exp\ \{ -\tfrac{1}{2} \left(\frac{\theta_s^* - \theta_\ell - \theta_T(t)}{\sigma(t)}\right)^2 \}\ dt \qquad (5.10)$$

In order to obtain an approximate explicit expression for θ_s^*, the exponential function is approximated by the first two terms of its Binomial series:

$$\exp(-x) \quad \simeq \quad 1 - x \qquad (5.11)$$

and (5.10) reduces to:

$$\int_{t_1}^{t_2} \frac{w(t)}{\sigma(t)} \quad \{ 1 - \tfrac{1}{2} \left(\frac{\theta_s^* + \theta_\ell - \theta_T(t)}{\sigma(t)}\right)^2 \} \quad dt$$

$$= \int_{t_1}^{t_2} \frac{w(t)}{\sigma(t)} \quad \{ 1 - \tfrac{1}{2} \left(\frac{\theta_s^* - \theta_\ell - \theta_T(t)}{\sigma(t)}\right)^2 \} \quad dt$$

Or

$$\int_{t_I}^{t_2} \frac{w(t)}{\sigma^3(t)} \quad (\theta_s^* + \theta_\ell - \theta_T(t))^2\ dt$$

$$= \int_{t_1}^{t_2} \frac{w(t)}{\sigma^3(t)} \quad (\theta_s^* - \theta_\ell - \theta_T(t))^2\ dt$$

$$\int_{t_1}^{t_2} \frac{w(t)}{\sigma^3(t)} \quad \theta_s^*\ dt \quad = \quad \int_{t_1}^{t_2} \frac{w(t)}{\sigma^3(t)} \quad \hat{\theta}_T(t)\ dt$$

Therefore the optimal slew motor position is given by:

$$\theta_s{}^* = \frac{\int_{t_1}^{t_2} \frac{w(t)}{\sigma^3(t)} \hat{\theta}_T(t)\, dt}{\int_{t_1}^{t_2} \frac{w(t)}{\sigma^3(t)}\, dt} \tag{5.12}$$

If $\hat{\theta}_T(t)$ is given in discrete form between intervals a and b

$$\theta_s{}^* = \frac{\sum_{i=a}^{b} \frac{w(i)}{\sigma^3(i)} \hat{\theta}_T(i)}{\sum_{i=a}^{b} \frac{w(i)}{\sigma^3(i)}} \tag{5.13}$$

(5.13) and (5.14) show that the variances of the forecast play an important role in determining the optimal slew motor position. The forecasts with higher variances (i.e. less confidence) will contribute less to the optimal value. If the terms w(t) and $\sigma(t)$ are constant with time, the optimal position will simply be the mean value of the forecasts within the time period. This is actually the case due to the symmetry of the gaussian probability density function.

5.3 One Realignment During the Prediction Period

In (5.1), the weighting of the realignments are explicitly expressed in the cost function. However, the cost function may be simplified by constraining one realignment within the prediction period. (5.1) can be rewritten as:

$$J = \int_{t_o}^{t_p} w(t) \quad \{ 1.0 - Pr(\theta_s(t) - \theta_\ell \leq \theta_T(t) \leq \theta_s(t) + \theta_\ell) \quad dt \tag{5.14}$$

The weighting on the number of realignments can now be implicitly introduced by varying the duration of the prediction period. One realignment within a long period will mean less interference from the slew motor. Thus, the probability of keeping track with the target is higher. On the other hand, one realignment within a short duration will result with frequent positioning of the slew motor overall. Since minimising

(5.14) is the same as maximising:

$$J^1 = \int_{t_o}^{t_p} w(t)\ Pr(\theta_s(t) - \theta_\ell \leqslant \theta_T(t) \leqslant \theta_s(t) + \theta_\ell)\ dt \tag{5.15}$$

The optimisation is concentrated on (5.15) only. Again, assuming the forecasts, $\theta_T(t)$, to be normally distributed as in Section 5.2:

$$J^1 = \int_{t_o}^{t_p} \frac{w(t)}{\sqrt{2\pi}\ \sigma(t)} \int_{\theta_s(t)-\theta_\ell}^{\theta_s(t)+\theta_\ell} \exp\{-\tfrac{1}{2}\ (\frac{\theta_T(t)-\hat{\theta}_T(t)}{\sigma(t)})^2\}\ d\ \theta_T(t)\ dt \tag{5.16}$$

$$\text{Let}\quad z = \frac{\theta_T(t) - \hat{\theta}_T(t)}{\sqrt{2}\ \sigma(t)} \tag{5.17}$$

$$dz = \frac{d\ \theta_T(t)}{\sqrt{2}\ \sigma(t)} \tag{5.18}$$

Substituting (5.17) and (5.18) into (5.16) yields:

$$J^1 = \int_{t_o}^{t_p} \frac{w(t)}{\sqrt{\pi}} \int_{z_1}^{z_2} \exp(-z^2)\ dz\ dt \tag{5.19}$$

where

$$z_1 = \frac{\theta_s(t) - \theta_\ell - \hat{\theta}_T(t)}{\sqrt{2}\ \sigma(t)} \tag{5.20}$$

$$z_2 = \frac{\theta_s(t) + \theta_\ell - \hat{\theta}_T(t)}{\sqrt{2}\ \sigma(t)} \tag{5.21}$$

By definition, the error function is

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_o^x \exp(-w^2)\ dw \tag{5.22}$$

The cost function (5.19) becomes:

$$J^1 = \int_{t_o}^{t_p} \frac{w(t)}{2} \{ \mathrm{erf}(z_2) - \mathrm{erf}(z_1) \} \, dt \tag{5.23}$$

Since the slew motor position, $\theta_s(t)$, is assumed to be a step function as

$$\theta_s(t) = \theta_o + \Delta\theta \, u(t-\tau) \tag{5.24}$$

where θ_o is the initial slew motor position at time t_o, $\Delta\theta$ is the amount of the realignment and

$$u(t-\tau) = \begin{cases} 1 & t \geqslant \tau \\ o & t < \tau \end{cases} \tag{5.25}$$

The control variables of the cost function are $\Delta\theta$ and τ. The optimal values of $\Delta\theta$ and τ are obtained by differentiating the cost function w.r.t. $\Delta\theta$ and τ and by equating them to zero respectively. i.e.

$$\frac{\partial J^1}{\partial \tau} = 0 \quad ; \quad \frac{\partial J^1}{\partial \Delta\theta} = 0$$

Differentiating (5.23) w.r.t. τ gives

$$\frac{\partial J^1}{\partial \tau} = \int_{t_o}^{t_p} \frac{w(t)}{2} \{ \frac{\partial}{\partial \tau} \mathrm{erf}(z_2) - \frac{\partial}{\partial \tau} \mathrm{erf}(z_1) \} \, dt \tag{5.26}$$

and differentiating (5.23) w.r.t. $\Delta\theta$ gives

$$\frac{\partial J^1}{\partial \Delta\theta} = \int_{t_o}^{t_p} \frac{w(t)}{2} \{ \frac{\partial}{\partial \Delta\theta} \mathrm{erf}(z_2) - \frac{\partial}{\partial \Delta\theta} \mathrm{erf}(z_1) \} \, dt \tag{5.27}$$

(5.26) and (5.27) can be simplified by approximating the error function by a MacLaurin series

$$\mathrm{erf}(x) \simeq \frac{2}{\sqrt{\pi}} (x - \frac{x^3}{3} + \frac{x^5}{2!5} - \frac{x^7}{3!7} + \ldots) \tag{5.28}$$

Consider the first two terms only in (5.28), and substitute into (5.26) yields:

$$\frac{\partial J^1}{\partial \tau} \simeq \int_{t_o}^{t_p} \frac{w(t)}{\sqrt{\pi}} \frac{\partial}{\partial \tau} (z_2 - \frac{z_2^3}{3} - z_1 + \frac{z_1^3}{3}) \, dt$$

$$= \int_{t_o}^{t_p} \frac{w(t)}{\sqrt{\pi}} \frac{\partial}{\partial \tau} \left\{ \frac{2\theta_\ell}{\sqrt{2}\,\sigma(t)} - \frac{\theta_\ell\,(\theta_s(t) - \hat{\theta}_T(t))^2}{\sqrt{2}\quad \sigma^3(t)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(t)} \right\} dt \tag{5.29}$$

Using the expression (5.24)

$$\frac{\partial J^1}{\partial \tau} = \frac{\partial}{\partial \tau} \int_{t_o}^{\tau} \frac{w(t)}{\sqrt{\pi}} \left[\frac{2\theta_\ell}{\sqrt{2}\,\sigma(t)} - \frac{\theta_\ell(\theta_o - \hat{\theta}_T(t))^2}{\sqrt{2}\quad \sigma^3(t)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(t)} \right] dt$$

$$+ \frac{\partial}{\partial \tau} \int_{\tau_o}^{t_p} \frac{w(t)}{\sqrt{\pi}} \left[\frac{2\theta_\ell}{\sqrt{2}\,\sigma(t)} - \frac{\theta_\ell\,(\theta_o + \Delta\theta - \hat{\theta}_T(t))^2}{\sqrt{2}\quad \sigma^3(t)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\quad \sigma^3(t)} \right] dt$$

$$\frac{\partial J^1}{\partial \tau} = \frac{w(\tau)}{\sqrt{\pi}} \left[\frac{2\theta_\ell}{\sqrt{2}\,\sigma(\tau)} - \frac{\theta_\ell\,(\theta_o - \hat{\theta}_T(\tau))^2}{\sqrt{2}\quad \sigma^3(\tau)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(\tau)} \right]$$

$$- \frac{w(\tau)}{\sqrt{\pi}} \left[\frac{2\theta_\ell}{\sqrt{2}\,\sigma(\tau)} - \frac{\theta_\ell\,(\theta_o + \Delta\theta - \hat{\theta}_T(\tau))^2}{\sqrt{2}\quad \sigma^3(\tau)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(\tau)} \right] \tag{5.30}$$

Equating (5.30) to zero yields:

$$(\theta_o - \hat{\theta}_T(\tau))^2 = (\theta_o + \Delta\theta - \hat{\theta}_T(\tau))^2 \qquad \text{OR}$$

$$\Delta\theta = 2(\hat{\theta}_T(\tau) - \theta_o) \tag{5.31}$$

Similarly, (5.27) becomes:

$$\frac{\partial J^1}{\partial \Delta\theta} \simeq \frac{\partial}{\partial \Delta\theta} \int_{t_o}^{\tau} \frac{w(t)}{\sqrt{\pi}} \left\{ \frac{2\theta_\ell}{\sqrt{2}\,\sigma(\tau)} - \frac{\theta_\ell\,(\theta_o - \hat{\theta}_T(\tau))^2}{\sqrt{2}\quad \sigma^3(\tau)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(\tau)} \right\} dt$$

$$+ \frac{\partial}{\partial \Delta\theta} \int_{\tau}^{t_p} \frac{w(t)}{\sqrt{\pi}} \left\{ \frac{2\theta_\ell}{\sqrt{2}\,\sigma(\tau)} - \frac{\theta_\ell\,(\theta_o + \Delta\theta - \hat{\theta}_T(\tau))^2}{\sqrt{2}\quad \sigma^3(\tau)} \right.$$

$$\left. - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(\tau)} \right\} dt$$

$$\frac{\partial J^1}{\partial \Delta\theta} = - \int_{\tau}^{t_p} \frac{\sqrt{2}\,w(t)\,\theta_\ell}{\sigma^3(t)} (\theta_o + \Delta\theta - \hat{\theta}_T(t))\ dt \qquad (5.32)$$

Equating (5.32) to zero yields:

$$\int_{\tau}^{t_p} \frac{w(t)}{\sigma^3(t)} (\ \theta_o + \Delta\theta - \hat{\theta}_T(t))\ dt = 0 \qquad (5.33)$$

The optimal values of τ and $\Delta\theta$ are obtained by solving (5.31) and (5.33).

Now consider the error function to be approximated by the first three terms in (5.28), the cost function (5.23) becomes:

$$J^1 = \int_{t_o}^{\tau} \frac{w(t)}{\sqrt{\pi}} \left\{ \frac{2\theta_\ell}{\sqrt{2}\,\sigma(t)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(t)} + \frac{\theta_\ell^{\,5}}{20\sqrt{2}\,\sigma^5(t)} \right.$$

$$+ \left(\frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^5(t)} - \frac{\theta_\ell}{\sqrt{2}\,\sigma^3(t)}\right) (\theta_o - \hat{\theta}_T(t))^2$$

$$\left. + \frac{\theta_\ell}{4\sqrt{2}\,\sigma^5(t)} (\theta_o - \hat{\theta}_T(t))^4 \right\} \quad dt$$

$$+ \int_{\tau}^{t_p} \frac{w(t)}{\sqrt{\pi}} \left\{ \frac{2\theta_\ell}{\sqrt{2}\,\sigma(t)} - \frac{\theta_\ell^{\,3}}{3\sqrt{2}\,\sigma^3(t)} + \frac{\theta_\ell^{\,5}}{20\sqrt{2}\,\sigma^5(t)} \right.$$

$$+ \left(\frac{\theta_\ell^{\,3}}{2\sqrt{2}\,\sigma^5(t)} - \frac{\theta_\ell}{\sqrt{2}\,\sigma^3(t)}\right) (\theta_o + \Delta\theta - \hat{\theta}_T(t))^2$$

$$\left. + \frac{\theta_\ell}{4\sqrt{2}\,\sigma^5(t)} (\theta_o + \Delta\theta - \hat{\theta}_T(t))^4 \right\} \ dt \qquad (5.34)$$

Differentiating (5.34) w.r.t. τ and equating it to zero yields:

$$\left(\frac{\theta_\ell^{3}}{2\sqrt{2}\,\sigma^{5}(\tau)} - \frac{\theta_\ell}{\sqrt{2}\,\sigma^{3}(\tau)}\right)(\theta_o - \hat{\theta}_T(\tau))^{2}$$

$$+ \frac{\theta_\ell}{4\sqrt{2}\,\sigma^{5}(\tau)}(\theta_o - \hat{\theta}_T(\tau))^{4}$$

$$= \left(\frac{\theta_\ell^{3}}{2\sqrt{2}\,\sigma^{5}(\tau)} - \frac{\theta_\ell}{\sqrt{2}\,\sigma^{3}(\tau)}\right)(\theta_o + \Delta\theta - \hat{\theta}_T(\tau))^{2}$$

$$+ \frac{\theta_\ell}{4\sqrt{2}\,\sigma^{5}(\tau)}(\theta_o + \Delta\theta - \hat{\theta}_T(\tau))^{4} \qquad (5.35)$$

It is clear that (5.31) is also a solution to (5.35) regardless of the number of terms taken in approximating the error function. Hence, by the method of mathematical induction, the optimal amount of realignment, Δθ is exactly twice the difference between the forecast at the time of realignment and the current slew motor position. This statement holds even the exact integral for the error function is used.

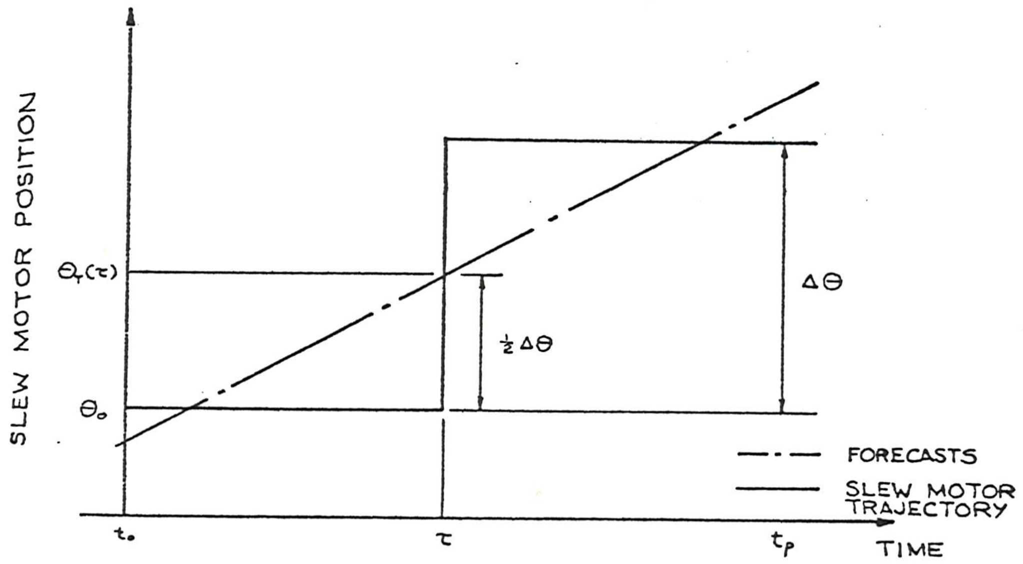


Figure 5.3 Realignment of Slew Motor.

The analysis shows that the amount of realignment can easily be obtained from (5.31) once the time of realignment is known. However, the difficulty is to determine the time of realignment which depends on the predicted trajectory and its associated variance. In the view of this, analytical solution is not feasible. However, the optimisation of (5.14) or (5.15) reduces to a one-dimensional problem with a constraint of (5.31). This can be done by numerical methods more easier than the original cost function (5.1).

The characteristics of the cost function (5.15) were studied by assuming the target trajectory to be a quadratic polynomial:

$$z(t) = a_o + a_1 t + a_2 t^2 \tag{5.36}$$

and the variance of the forecast was:

$$\text{Var}\ (z(t)) = b_o + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 \tag{5.37}$$

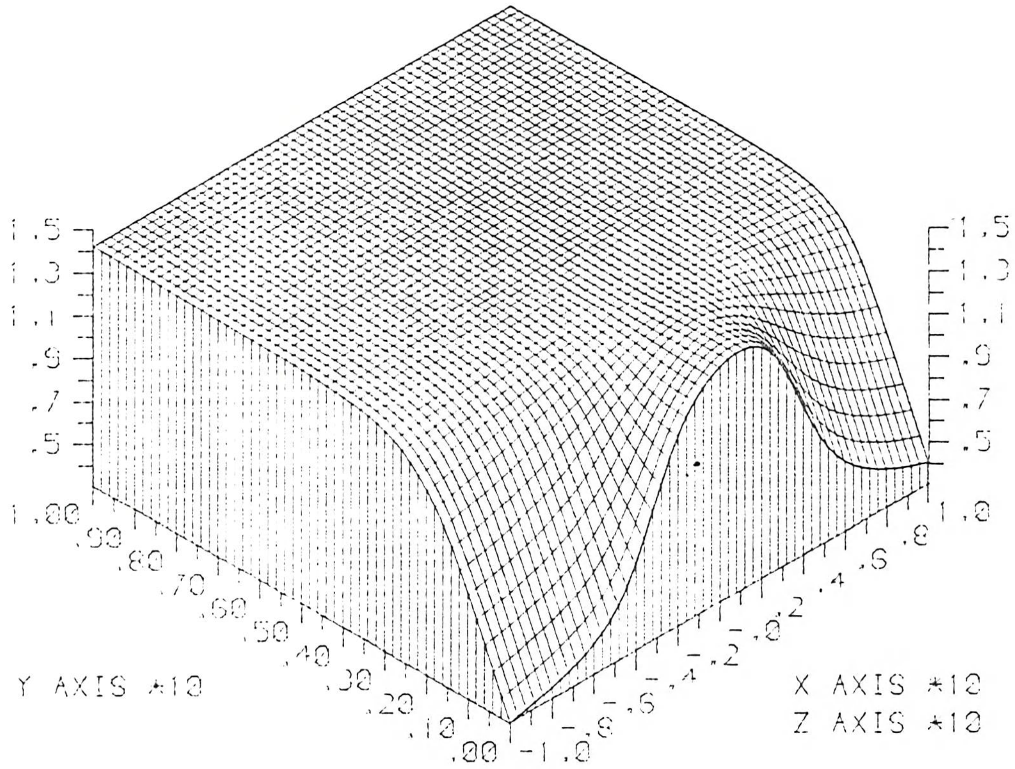
This trajectory and variance correspond to the case where there is no ship rolling motion. The shapes of the cost function under three different target trajectories were plotted in Figures 5.4 - 5.6. The plots showed the variation of the cost function at various times of realignment and with various amounts of realignment. The initial position of slew motor at time t_o were zero in all three cases. The coefficients of the polynomials (5.36) and (5.37) were shown under the plots. TAR means target trajectory and VAR stands for variance. From the three graphs, it showed that the cost function is a nice surface having only one single maxima.

5.4 Dynamic Programming Method

Restating the proposed cost function here:

$$J = \int_{t_o}^{t_p} w_1(t) \{ 1 - \Pr (\theta_s(t) - \theta_\ell \leqslant \theta_T(t) \leqslant \theta_s(t) + \theta_\ell) \} dt$$

$$+ \int_{t_o}^{t_p} w_2(t) \{ \text{sign} (\theta_s^+(t) - \theta_s^-(t)) \}^2 dt \tag{5.38}$$

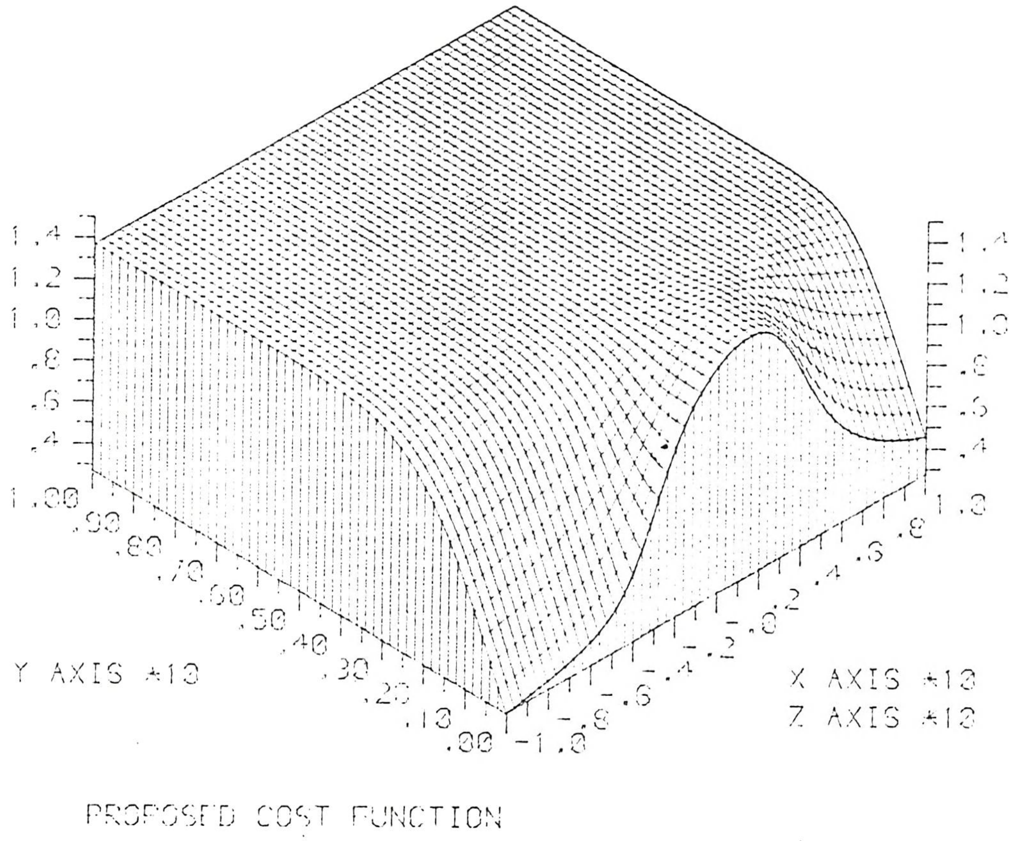


PROPOSED COST FUNCTION

Maximum point at
Tau (Y-axis) = 1.0
Delta theta (X-axis) = 1.2

	t^0	t^1	t^2	t^3	t^4	t^5
Tar	.00	.70	.00	.00	.00	.00
Var	1.00	.00	1.00	.00	1.00	.00

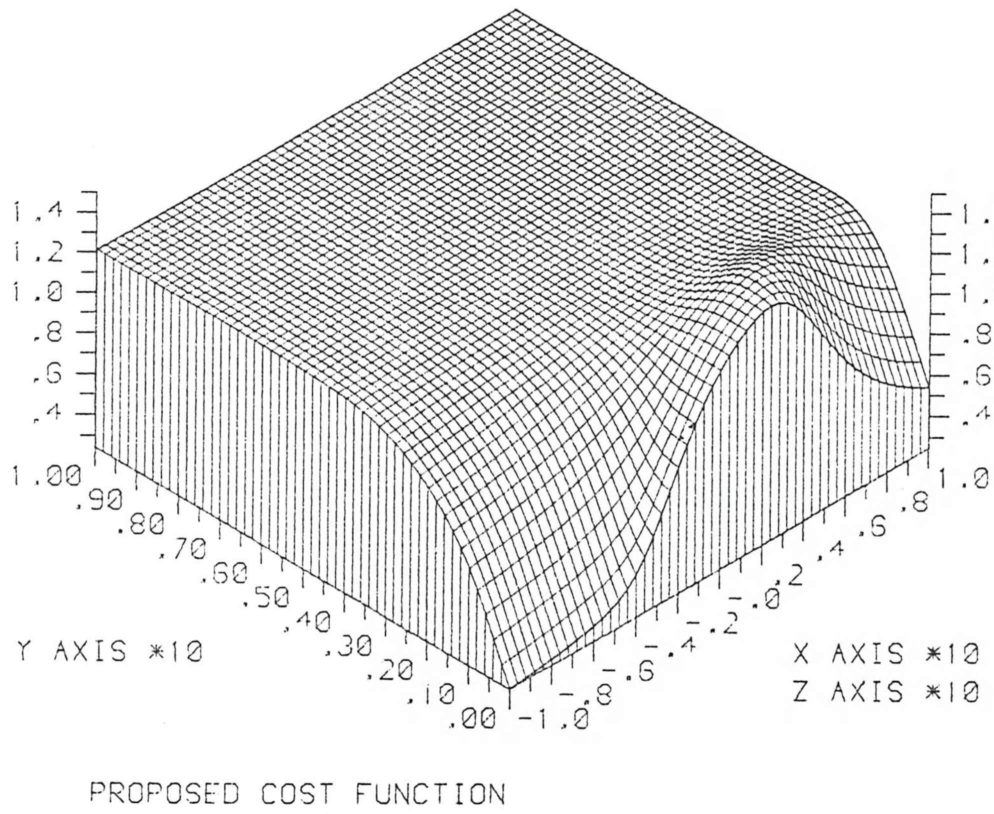
Figure 5.4 Surface of cost function



Maximum point at
Tau (Y-axis) = .8
Delta theta (X-axis) = 2.0

	t^0	t^1	t^2	t^3	t^4	t^5
Tar	.00	1.40	-.07	.00	.00	.00
Var	1.00	.00	1.00	.00	1.00	.00

Figure 5.5 Surface of cost function



Maximum point at
Tau (Y-axis) = .8
Delta theta (X-axis) = 3.2

	t^0	t^1	t^2	t^3	t^4	t^5
Tar	.00	2.80	-.28	.00	.00	.00
Var	1.00	.00	1.00	.00	1.00	.00

Figure 5.6 Surface of cost function

The anticipated slew motor trajectory will be:

$$\theta_s(t) = \theta_o + \sum_{i=1}^{n} \Delta\theta_i \, u(t + t_i) \tag{5.39}$$

where n is the number of realignments. Both (5.38) and (5.39) are discontinuous functions. Furthermore, the forecasts of the motions will be discrete because of the models used. It is then appropriate to solve the optimal trajectory by means of numerical methods. In discrete form the cost function (5.38) is:

$$J = \sum_{m=0}^{N_p} w_1(m) \; \{ \; 1 - \Pr\,(\theta_s(m) - \theta_\ell < \theta_T(m) < \theta_s(m) + \theta_\ell) \; \}$$

$$+ \sum_{m=1}^{N_p} w_2(m) \; \{ \; \mathrm{sign}\,(\theta_s(m) - \theta_s(m-1) \; \}^2 \tag{5.40}$$

The problem becomes a Np-variable one determining the optimal values for $\theta_s(m)$, m=0, 1, 2, . . ., Np. This Np-variable may be solved by the application of a few classical unconstrained optimisation techniques. For instance, the direct search methods [38] may be applied to the problem. However, these methods are not practical in association with problems having large numbers of variables. They will demand huge computation time before the result converged. Other more powerful and faster methods, such as the conjugate gradients method, etc. [15, 16, 17, 38] , are handicapped by the highly non-linear cost function. These methods generally require a cost function to be continuous and continuously differentiable.

The Np-variable problem may be viewed as a serial multistage decision problem. It can be represented in block diagram form as shown in Figure 5.7. Considering a single stage, i, it is characterised by an input paramters, $\theta_s(i-1)$, a decision variable, $\Delta\theta_i$, and an output parameters, $\theta_s(i)$, representing the outcome obtained as a result of making the decision. There is a return function, J_i, which measures the effectiveness of the decision. The objective of the multistage decision problem is, therefore, to find $\Delta\theta_o$, $\Delta\theta_1$, . . . , so as to optimise some function of the individual stage return functions, $f(J_o, J_1, \ldots, J_{Np})$. The function for this particular Np-variable problem is the summation of all the return functions.

$$J = \sum_{i=0}^{N_p} J_i(\theta_s(i-1), \Delta\theta_i) \tag{5.41}$$

which is the cost function (5.40). Each return function is:

$$J_i(\theta_s(i-1), \Delta\theta_i) = w_1(i)\{1 - \Pr(\theta_s(i) - \theta_\ell \leq \theta_T(i) \leq \theta_s(i) + \theta_\ell)\}$$

$$+ w_2(i)\{\text{sign}(\theta_s(i-1) - \theta_s(i))\}^2 \qquad (5.42)$$

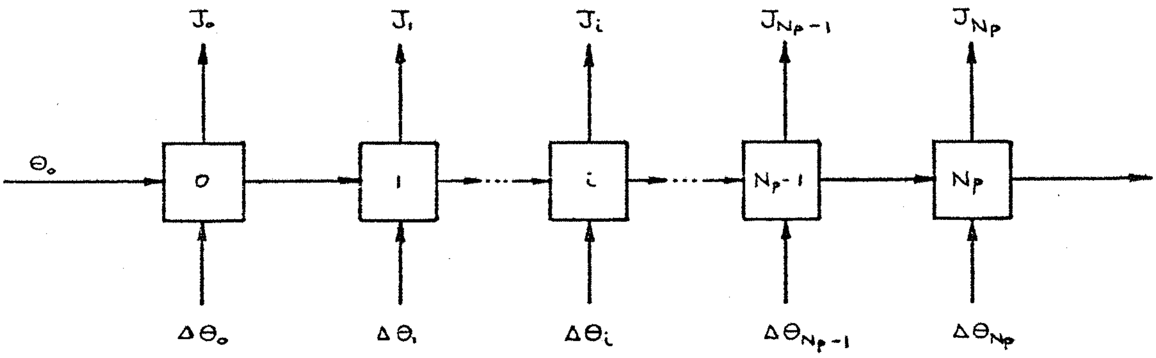


Figure 5.7 Multistage Decision Problem

The relationship between the input and the output parameters is: $\theta_s(i) = \theta_s(i-1) + \Delta\theta_i$ (5.43)

Thus, a realignment at time interval, i, is represented by a non zero $\Delta\theta_i$.

The method of Dynamic Programming is a mathematical technique well suited for the optimisation of multistage decision problems. This technique was developed by Bellman in the early 1950's [3]. The method makes use of the concept of sub-optimisation and the principle of optimality in solving the problem. The statement of the principle of optimality is [38]:

> "An optimal policy (or a set of decisions), has the property that whatever the initial state and initial decision are, the resulting decisions must constitute an optimal policy with regard to the state resulting from the first decision".

The process of optimisation is carried out in the reverse order as illustrated in Figure 5.8. The first sub-problem starts at the final stage, i = Np. The cost function J_{Np} is a function of the input parameter $\theta_s(Np-1)$ and the decision $\Delta\theta_{Np}$. Once the parameter $\theta_s(Np-1)$ is specified, the decision must be made irrespective of what happens in the other stages. Therefore, the optimal cost function depends only on the input parameter, $\theta_s(Np - 1)$. i.e.

$$J^*_{Np} (\theta_s(Np-1)) = \min_{\Delta\theta_{Np}} \{J_{Np}\} \tag{5.44}$$

The next sub-problem is to group the last two stages together. The optimal cost function is then a function of the input parameter, $\theta_s(Np-2)$. i.e.

$$J^*_{Np-1} (\theta_s (Np - 2)) = \min_{\Delta\theta_{Np-1},\ \Delta\theta_{Np}} \{ J_{Np-1} + J_{Np}\} \tag{5.45}$$

The principle of optimality requires that $\Delta\theta_{Np}$ be selected so as to optimise J_{Np} for a given $\theta_s(Np-1)$. Since $\theta_s(Np-1)$ can be obtained once $\theta_s(Np-2)$ is specified and the decision $\Delta\theta_{Np-1}$ is made. Thus, the optimisation of this second stage becomes:

$$J^*_{Np-1} (\theta_s (Np-2)) = \min_{\Delta\theta_{Np-1}} \{ J_{Np-1} + J^*_{Np} (\theta_s (Np-1))\} \tag{5.46}$$

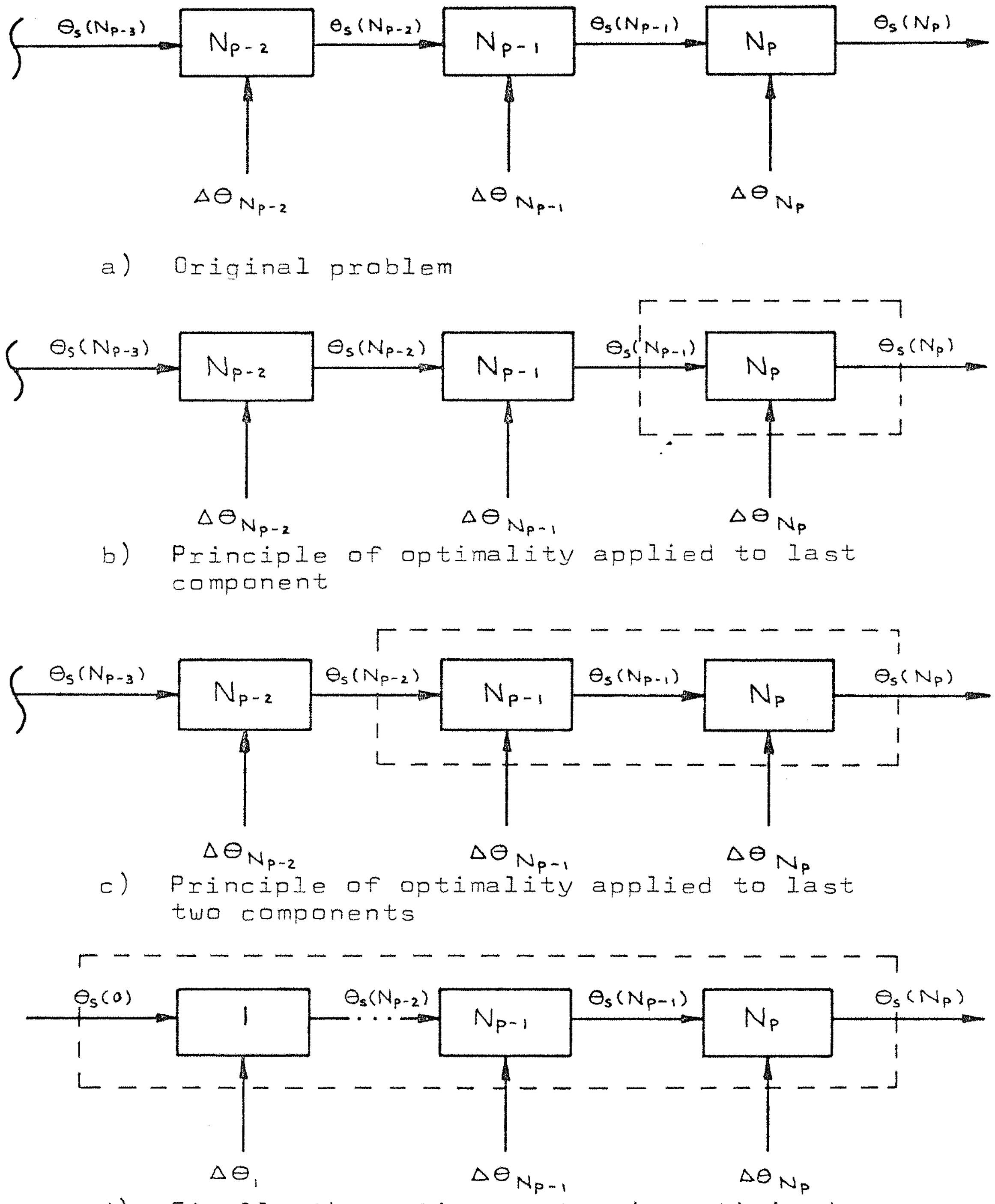
Similarly, the idea can be extended to the ith sub-problem which is defined as:

$$J^*_i (\theta_s(i-1)) = \min_{\Delta\theta_i,\ \Delta\theta_{i-1},\ \ldots,\ \Delta\theta_{Np}} \{ J_i + J_{i+1} + \ldots + J_{Np}\} \tag{5.47}$$

The principle of optimality allows the problem to be simplified as:

$$J^*_i(\theta_s(i-1)) = \min_{\Delta\theta_i} \{J_i + J^*_{i+1} (\theta_s(i)) + \ldots + J^*_{Np} (\theta_s (Np-1)) \} \tag{5.48}$$

Therefore, by applying the principle of optimality, the original multi-variables optimisation problem of (5.47) is decomposed into a series of separate problems, each involving only one decision variable.



a) Original problem

b) Principle of optimality applied to last component

c) Principle of optimality applied to last two components

d) Finally the entire system is optimised

Figure 5.8 Illustration of principle of optimality

The implementation of dynamic programming method to the cost function (5.40) must resort to tabulation form. This is totally due to the odd characteristic of the cost function. The optimal cost function in each sub-problem is not a continuous function of the input variable $\theta_s(i-1)$. At each stage, the relationship between the return function J_i and the decision $\Delta\theta_i$ for a given input $\theta_s(i-1)$ is shown in Figure 5.9. There is a singular point at $\Delta\theta = 0$. The depth of the 'ditch' at the singular point is the weighting on a realignment at this stage. A change of the input $\theta_s(i-1)$ will only shift the curve to the right or left without changing the overall shape, except that the singular point remains at $\Delta\theta = 0$. A discrete range of possible values of the input variable $\theta_s(i-1)$ is given in each sub-problem. A table of optimal decisions and the corresponding cost functions is then constructued for the values of $\theta_s(i-1)$ as shown in Figure 5.10. The sub-optimisation for a given value of $\theta_s(i-1)$ is done by searching through a range of values of the decision $\Delta\theta_i$ and by picking out the optimum. An input $\theta_s(i-1)$ and a decision $\Delta\theta_i$ will fix the output $\theta_s(i)$. The optimal decisions from stage i+1 to the final stage will then be found from the tables set up in previous sub-problems. The cost function for the given pair, $\theta_s(i-1)$ and $\Delta\theta_i$, can be evaluated using (5.48) The optimal decision $\Delta\theta_i^*$ and its corresponding cost function $J_i^*(\theta_s(i-1))$ are then registered in the row of the table associated with the input $\theta_s(i-1)$. Since the initial value, θ_o, is known the optimal value $\Delta\theta^*_1$ is then looked up from the table for stage 1. At this stage, $\theta_s(1)$ is then fixed and $\Delta\theta^*_2$ can be found from the table of stage 2, and so on. The optimal decisions, $\Delta\theta^*_1$, $\Delta\theta^*_2$, . . ., $\Delta\theta^*_{Np}$ are then determined. The step sizes of $\theta_s(i-1)$ and $\Delta\theta_i$ are compromised with the time to compute the tables. The number of tables required to determine the optimal slew motor trajectory is the number of time intervals in one prediction period.

The dynamic programming method was simulated on the proposed cost function. The forecasts of the motion was deliberately chosen to be a sine wave with unit amplitude. It was so chosen because the total motion of the predicted target trajectory and the predicted ship rolling motion may be decomposed into two or three sine wave components. The actual amplitude is immaterial, only the number of realignments is interesting. The variances of the forecasts followed the expression:

$$\sigma^2(n) = 1.0 + 0.05\,n \qquad (5.49)$$

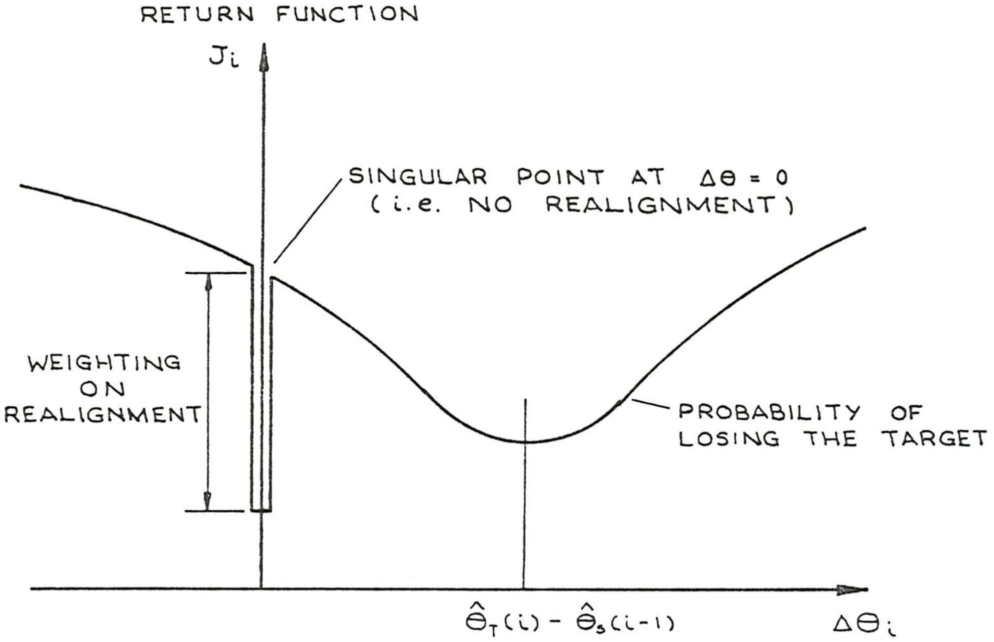


Figure 5.9 Plot of return function against realignment at time i

$\theta_s(i-1)$	$\Delta\theta^*(i)$	$J_i^*(\theta_s(i-1))$	$\theta_s(i)$
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

Figure 5.10 Table constructed in each suboptimization

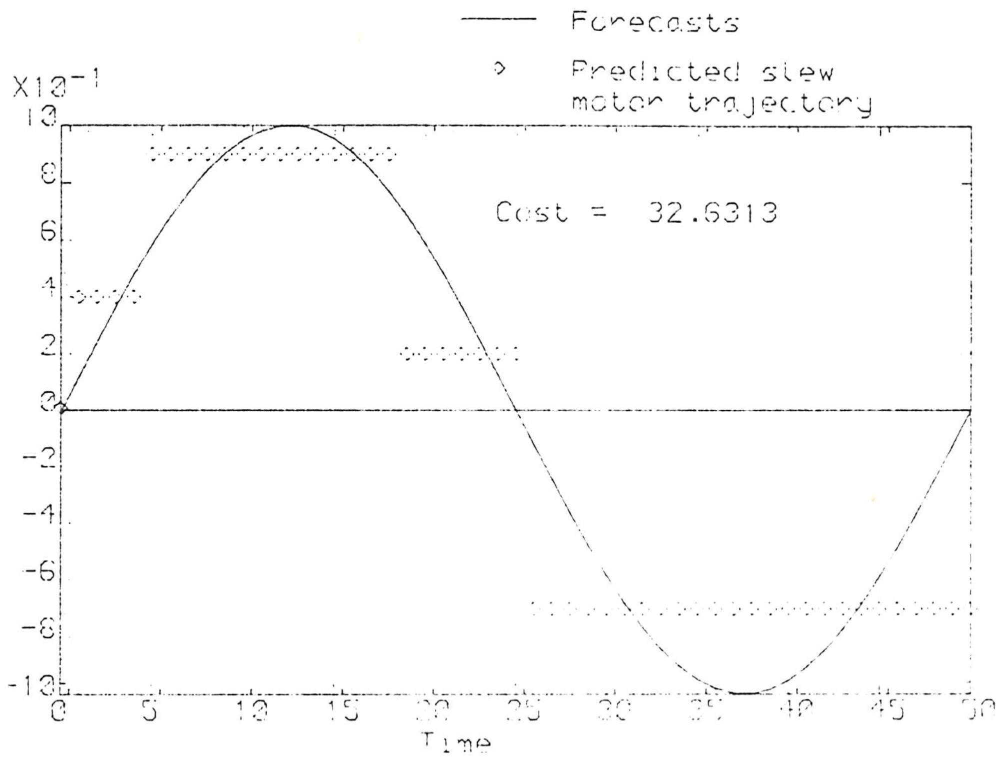


Figure 5.11 Resulting optimal slew motor trajectory from dynamic programming method

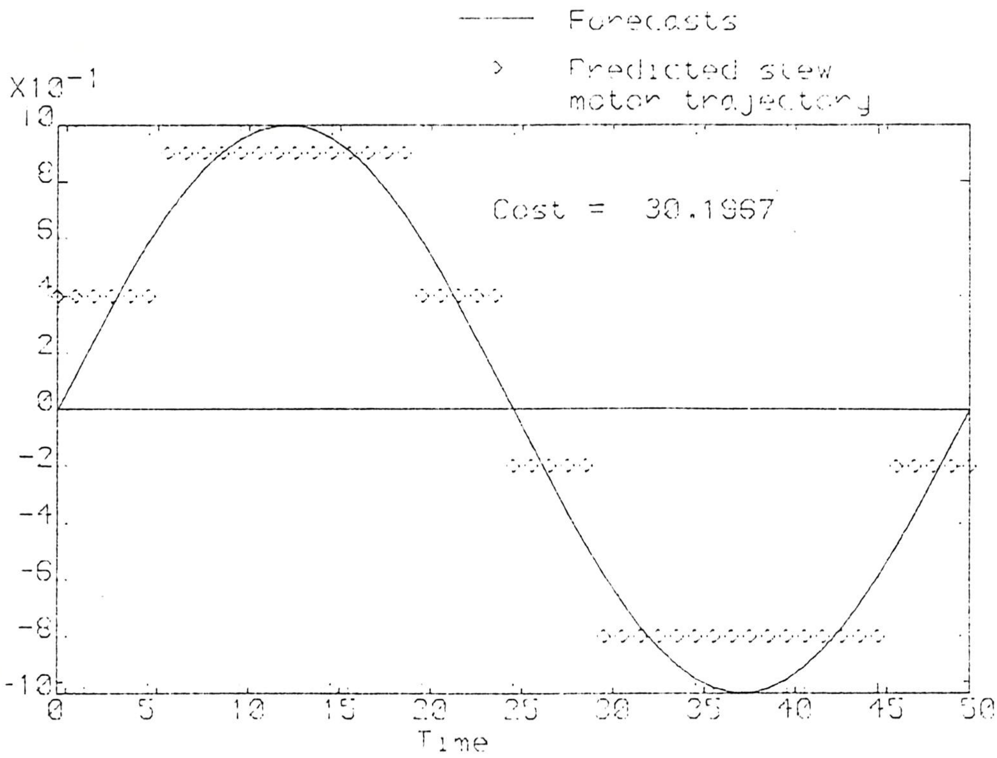


Figure 5.12 Resulting optimal slew motor trajectory from numerical search method

The expressions for the two weightings in the cost function were set as:

$$w_1(n) = 1.0 \ \{1 - \exp(-0.01\ n)\} \tag{5.50}$$

$$w_2(n) = 0.01 \ \{1 - \exp(-0.01\ n)\} \tag{5.51}$$

In effect, the weightings had a ratio of a hundred irrespective of the lead time. The limit on the tracking motor was set to 0.5, and the initial slew motor position was $\theta_o = 0$. The step sizes of $\theta_s(i-1)$ and $\Delta\theta_i$ in each sub-problem were both fixed at 0.1 and within the range of -2.0 to 2.0. The resulting slew motor trajectory was then determined as in Figure 5.11.

5.5 Numerical Search Method

In the previous section, the cost function was optimised by the use of dynamic programming method in tabular form. It constrained the possible positions of the slew motor and the amount of realignment in each sub-problem to a set of discrete values. However, the movement of the slew motor is not constrained to these values. It is possible that the optimal trajectory of the slew motor obtained from the previous section is not the true optimal value due to the discretisation process. Furthermore, the computation time required in constructing the tables will be unbearably long for a real-time application. The computation time is proportional to the number of time intervals within the prediction period and the product of the numbers of discrete points in the possible slew motor position and the amount of realignments. By increasing the step sizes of the slew motor positions and the amount of realignments, the computation time can be reduced, but the correctness of the obtained optimal solution is doubtful. Ideally, from the computation point of view, it is desired to have a recursive algorithm performing the optimisation. Due to the variation in the forecasts after each filtering and the non-linearity of the cost function, only a non-recursive numerical search method was established for optimising the proposed cost function.

The development of the search method is based on the results obtained in Sections 5.2 and 5.3. If the timings of the realignments are given, the optimal slew motor positions between two consecutive realignments are governed by the expression (5.13). The amount of each realignment is also governed by the expression (5.31). The effect of any variation in the slew motor position at one stage may not be propagated to other stages far away from the origin of variation. This is partially due to the step changes in the optimal slew motor trajectory. The realignments act as buffers partially isolating the steps of the slew motor trajectory from each other. The whole prediction period can be divided into regions with the realignments as their boundaries.

The developed numerical search method was a result of logical analysis of the problem. Mathematical proof of the validity of the algorithm has not yet been established. However, the results from the algorithm was compared with the one obtained from the dynamic programming method mentioned in the previous section. It showed that the results obtained from the numerical search method gave a better cost function value when compared with the dynamic programming results (Figures 5.11 and 5.12). The algorithm is explained in the following paragraphs.

The philosophy of the algorithm is to first divide the whole prediction period into several regions. At each boundary a realignment is assumed. Therefore, the corresponding slew motor position in each region can then be determined by obtaining the weighted mean within that region by means of (5.13). The assumed number of regions is so chosen that it is actually more than the anticipated optimal number of realignments. The next step is to merge two realignments into one if it gives a better cost function value by doing so. It then adjusts the timings of the remaining realignments. Finally, the decision of moving the slew motor at current time is made by considering the slew motor position in the first region only. It assumes that by this time, the timings of the realignments are optimal, and hence the regions separated by the realignments are independent to each other. Based on this approach, the algorithm is divided into four stages (5.15).

In the first stage, the potential realignments are established by making use of the expression (5.31) which states that the optimal amount of realignment is twice the difference between the target position and the slew motor position at the time of realignment. Several assumptions are made during this stage. It first assumes that any decision on realignments is correct and the realignment defines the boundary of a region. Once a realignment is established, the regions before the realignment are ignored in the determination of next realignment. The task then concentrates on the forecasts after the realignment until another realignment is established. The decision on a realignment is based on the compar

-ison of two costs J_a and J_b.

$$J_a = \sum_{k=t_r}^{i-1} w_1(k) \; \{ 1 - \Pr(\theta_1 - \theta_\ell \leq \hat{\theta}_T(k) \leq \theta_1 + \theta_\ell) \} \tag{5.52}$$

and

$$J_b = \sum_{k-t_r}^{i-1} w_1(k) \; \{ 1 - \Pr(\theta^* - \theta_\ell \leq \hat{\theta}_T(k) \leq \theta^* + \theta_\ell) \}$$

$$+ \; W_2(i) \tag{5.53}$$

where t_r is the time of last realignment and i is the prediction time interval under consideration. The cost J_a assumes no realignment at time i, whereas J_b assumes there is one. The term θ^* in J_b is the optimal slew motor position between last realignment at time t_r and a realignment at time i. Its value is determined by the expression (5.13) The term θ_1 in J_a is the slew motor position from time t_r if there is no realignment at time i. In both costs, J_a and J_b, the probability of losing the target at time i is not included. It is assumed that if a realignment occurs at time i the slew motor will be moved to position θ_1, The crucial part is to give a value to θ_1, in order to allow a comparison between the J_a and J_b. The term θ_1 is evaluated recursively following the expression:

$$\theta_1 = \theta_1 + 2 \, (\hat{\theta}_T(i) - \theta_1) \tag{5.54}$$

The term θ_1 is re-initialised every time after a realignment is established. The recursion starts with the value of the forecast at the time of last realignment. The flow chart of this stage is as shown in Figure 5.16. The predicted slew motor trajectory during the first stage is not the optimal one. The number of realignments determined in this stage is highly probable to be more than the optimum. This is because of the choice of the values for θ_1 in the process. As shown in Figure 5.13, the position of θ_1 will amost certainly have a higher probability of losing the target. Unless the weight on a realignment is very large or the forecasts are nearly flat, a realignment is likely to be set.

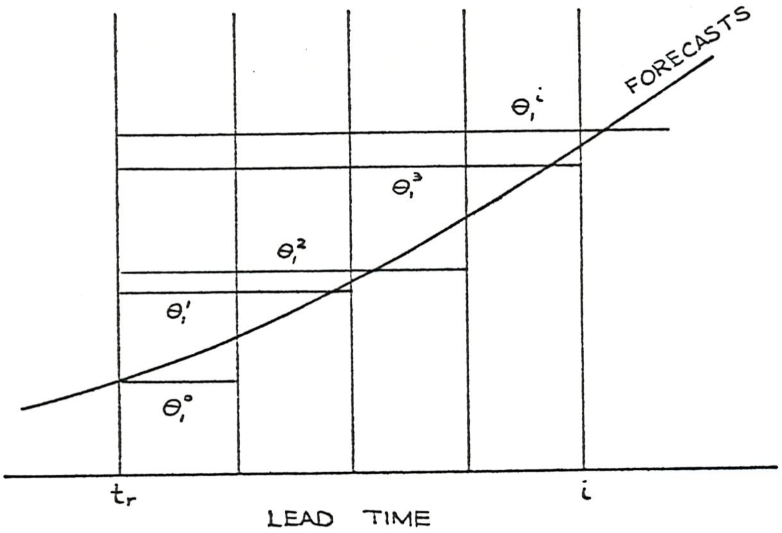


Figure 5.13 Positions of θ_1

The second stage in the search method is aimed at removing the redundant realignments. It checks two realignments in succession at times t_1 and t_2 whether they can be replaced by one realignment at t_s between the two (Figure 5.14). The time for the new realignment is approximated at the time that the forecast is equal to or larger than the average value of the predicted slew motor positions before the realignment at time t_1 and after the realignment at time t_2. The flow chart of the second stage is shown in Figure 5.17.

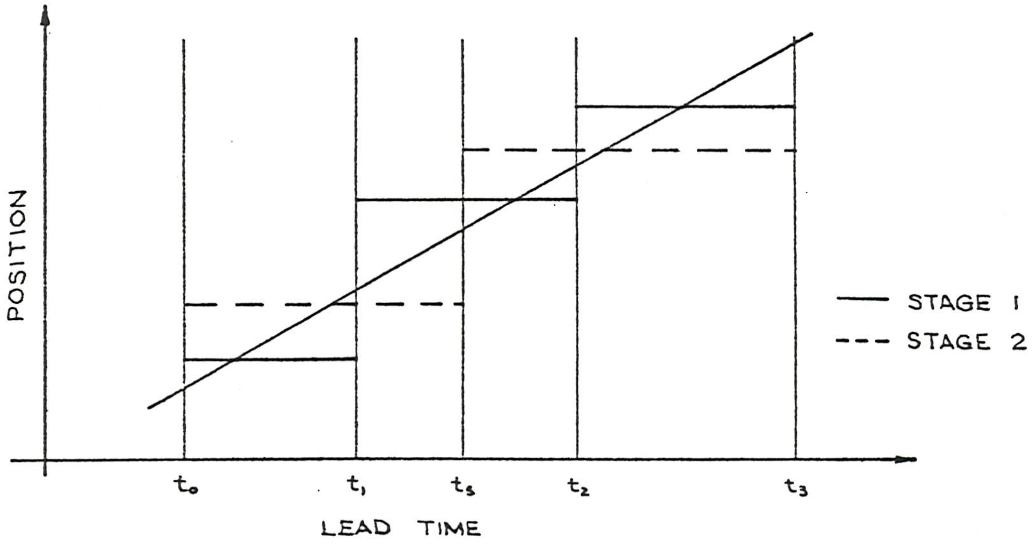


Figure 5.14 Removing Redundant Realignments.

The third stage adjusts one by one the timing of the realingments determined in the second stage. It evaluates the change in cost function if a realignment is brought forward, or is postponed, one interval. If a change in the time of the realignment gives a smaller cost, the realignment is changed accordingly until the cost is minimum. The process is described in Figure 5.18. Referring to Figure 5.14, the third stage only concentrates at the time between t_1 and t_3 in the testing of the realignment at t_2.

The final stage makes the decision whether there should be an actual realignment at current time. The previous three stages are intended to divide the whole prediction period into regions separated by the realignments. However, all three previous stages assume a realignment at current time. The final stage is to determine the validity of this assumption. After the third stage of the search method, the slew motor position of the first region will be the weighted mean of the forecasts within that region according to expression (5.13). This value θ^* will be different from the initial value of the slew motor position, θ_o. Therefore, a decision is made between the two costs:

$$J_a = \sum_{k=0}^{t_1-1} w_1(k) \{ 1 - Pr (\theta^* - \theta_\ell \leqslant \hat{\theta}_T(k) \leqslant \theta^* + \theta_\ell)\} + w_2(o) \tag{5.55}$$

and

$$J_b = \sum_{k=0}^{t_1-1} w_1(k) \{ 1 - Pr (\theta_o - \theta_\ell \leqslant \hat{\theta}_T(k) \leqslant \theta_o + \theta_\ell)\} \tag{5.56}$$

where t_1 is the time of the first realignment. J_a assumes a realignment at current time and J_b assumes the slew motor remains at its initial position. The summation term in J_a must be smaller than J_b since θ^* is the optimum position in the first region. Therefore, if $w_2(o)$ is very large, the decision is to keep the slew motor at its initial position. Otherwise, the slew motor will be commanded to the position θ *. In fact, if the decision is to keep the slew motor position at its initial position, the timing of the realignments obtained from the previous stages will not be optimum, particularly the first few realignments due to the new slew motor position in the first region. The realignments will be brought earlier. However, this alteration will be small. Furthermore, the changes on the future realignment is immaterial since the main objective is to determine the optimum position of the slew motor at current time.

The numerical search method was tested using the same set of parameters in the case of dynamic programming method. The result was plotted out in Figure 5.12. The comparison of the results illustrated that the search method gave a smaller cost. The search method performs better over the dynamic programming method is mainly due to the absence of the discretisation on the possible positions for the slew motor. The slew motor is no longer restricted to a set of values as in the case of the dynamic programming method. In the computation point of view, the search method also performs better than the dynamic programming method. The search method requires less memory size as compared with the dynamic programming method which requires a series of tables. The dissappearance of the tables reduces the huge computation time in generating these tables. In the search method, there is no need to compute the whole cost function at any one time since the prediction period is separated into regions. Moreover, the partial cost function required in the search method can be evaluated recursively which reduces the computation time further.

5.6 Conclusions

A cost function was finally proposed. It incorporates the probabilities of losing the target and the frequency of realignments. The weightings on both terms vary exponentially with time. This is because of the physical requirement in tracking moving targets. However, the cost function is a discontinuous function. Analytical solution is not available, except in a few circumstances, in the optimisation process. Among the well established optimisation techniques, only did the dynamic programming method seem feasible at a time. Later, it was found that the results from the dynamic programming method were not truely optimal. A numerical search method was finally established which was shown experimentally to be superior than the dynamic programming method. Experimentally and logically, the numerical search method was proved to be the best possible way. However, a mathematical prove is not completed.

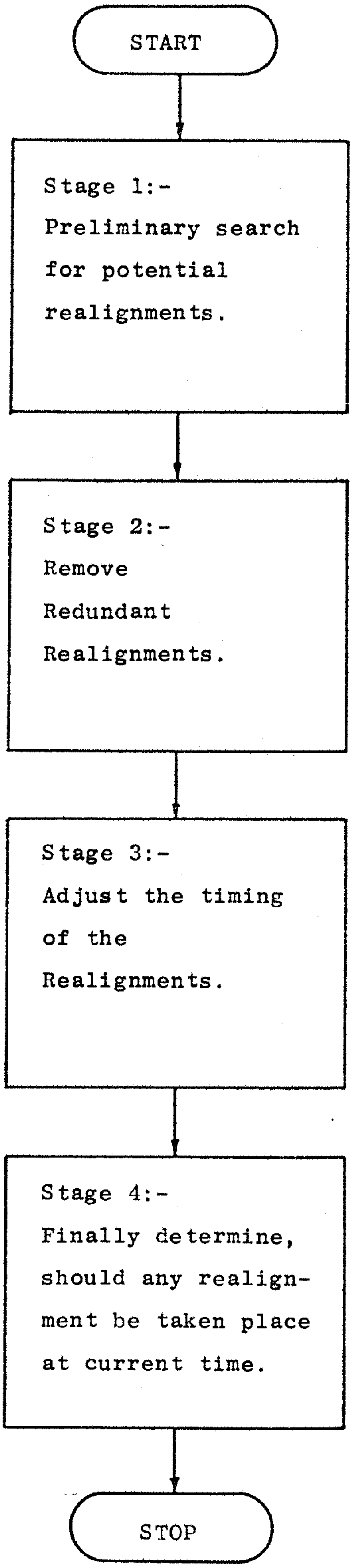

START
Stage 1:-
Preliminary search for potential realignments.
Stage 2:-
Remove Redundant Realignments.
Stage 3:-
Adjust the timing of the Realignments.
Stage 4:-
Finally determine, should any realignment be taken place at current time.
STOP

Figure 5.15 Numerical Search Method

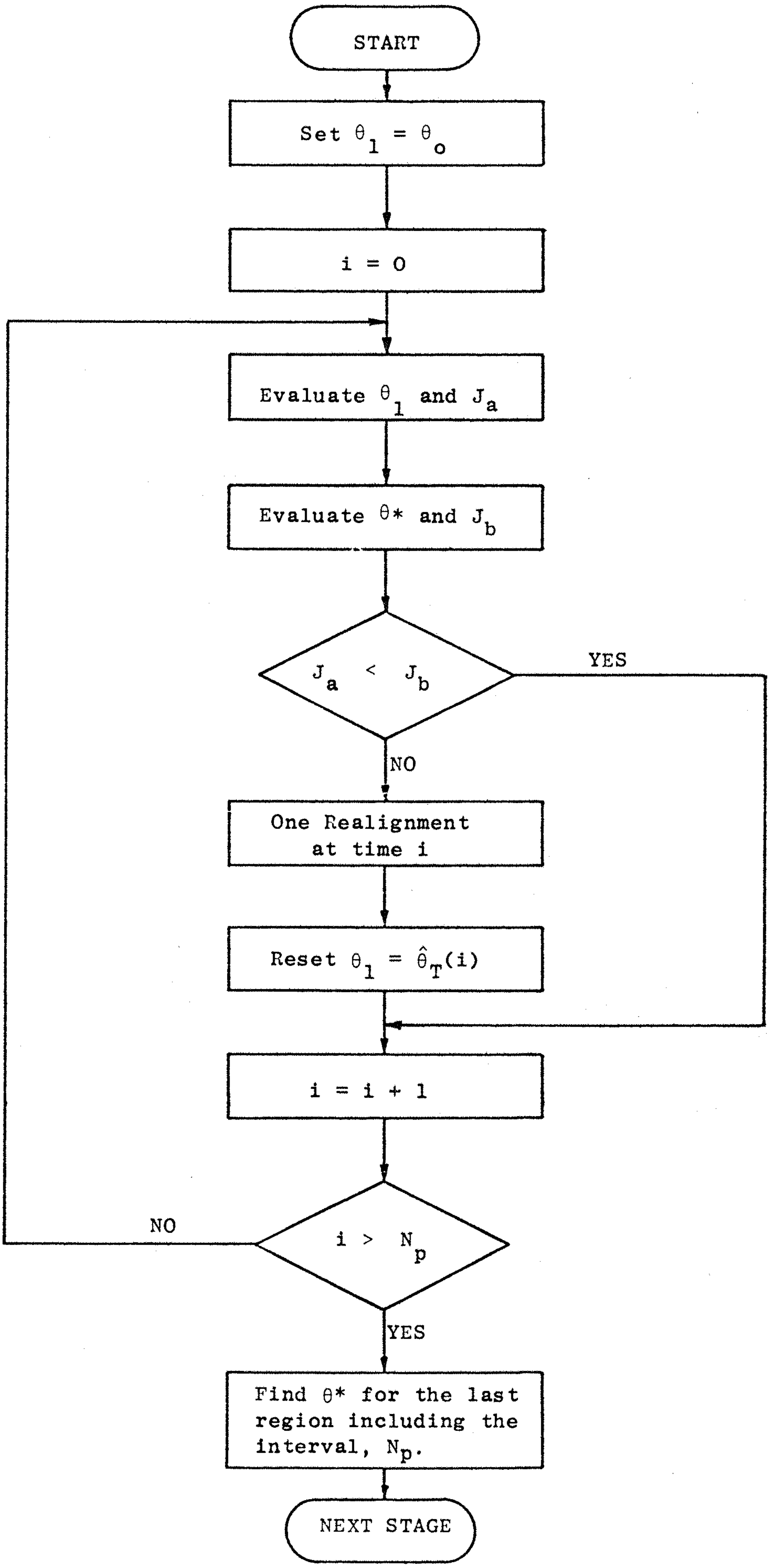

START
Set $\theta_1 = \theta_o$
$i = 0$
Evaluate θ_1 and J_a
Evaluate θ^* and J_b
$J_a < J_b$
YES
NO
One Realignment at time i
Reset $\theta_1 = \hat{\theta}_T(i)$
$i = i + 1$
$i > N_p$
NO
YES
Find θ^* for the last region including the interval, N_p.
NEXT STAGE

Figure 5.16 Flow Chart of First Stage.

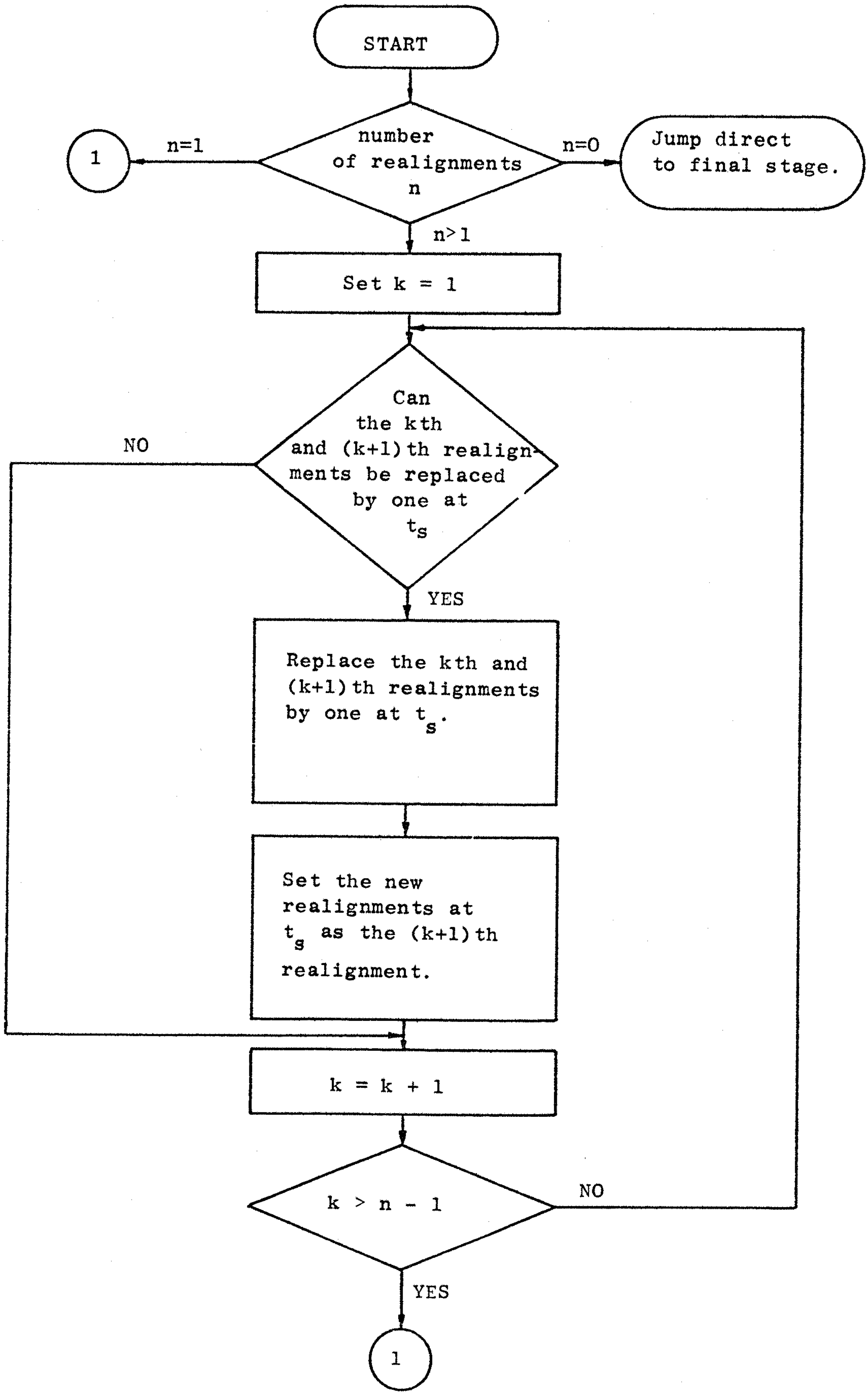

START
number of realignments n
n=1
1
n=0
Jump direct to final stage.
n>1
Set k = 1
Can the kth and (k+1)th realignments be replaced by one at t_s
NO
YES
Replace the kth and (k+1)th realignments by one at t_s.
Set the new realignments at t_s as the (k+1)th realignment.
k = k + 1
k > n - 1
NO
YES
1

Figure 5.17(a) Flow Chart of Second Stage.

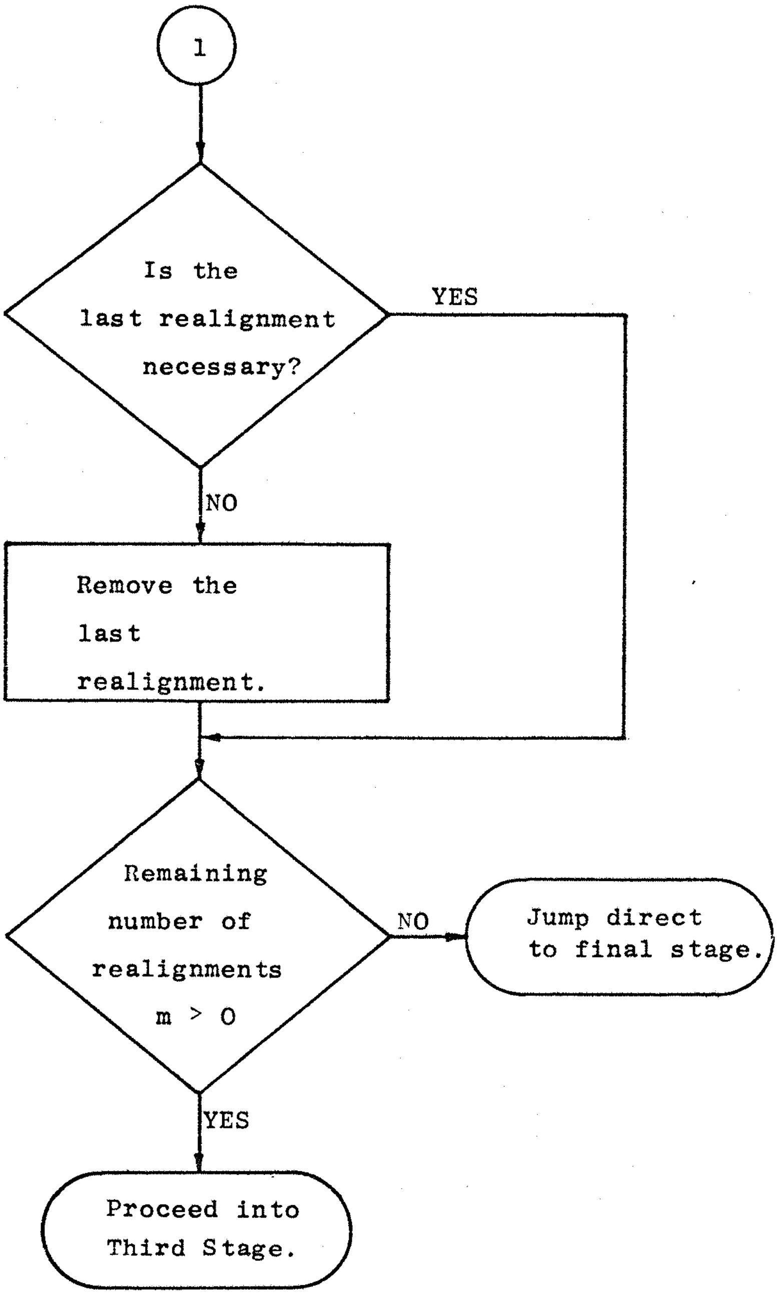

1
Is the last realignment necessary?
YES
NO
Remove the last realignment.
Remaining number of realignments m > 0
NO
Jump direct to final stage.
YES
Proceed into Third Stage.

Figure 5.17(b)

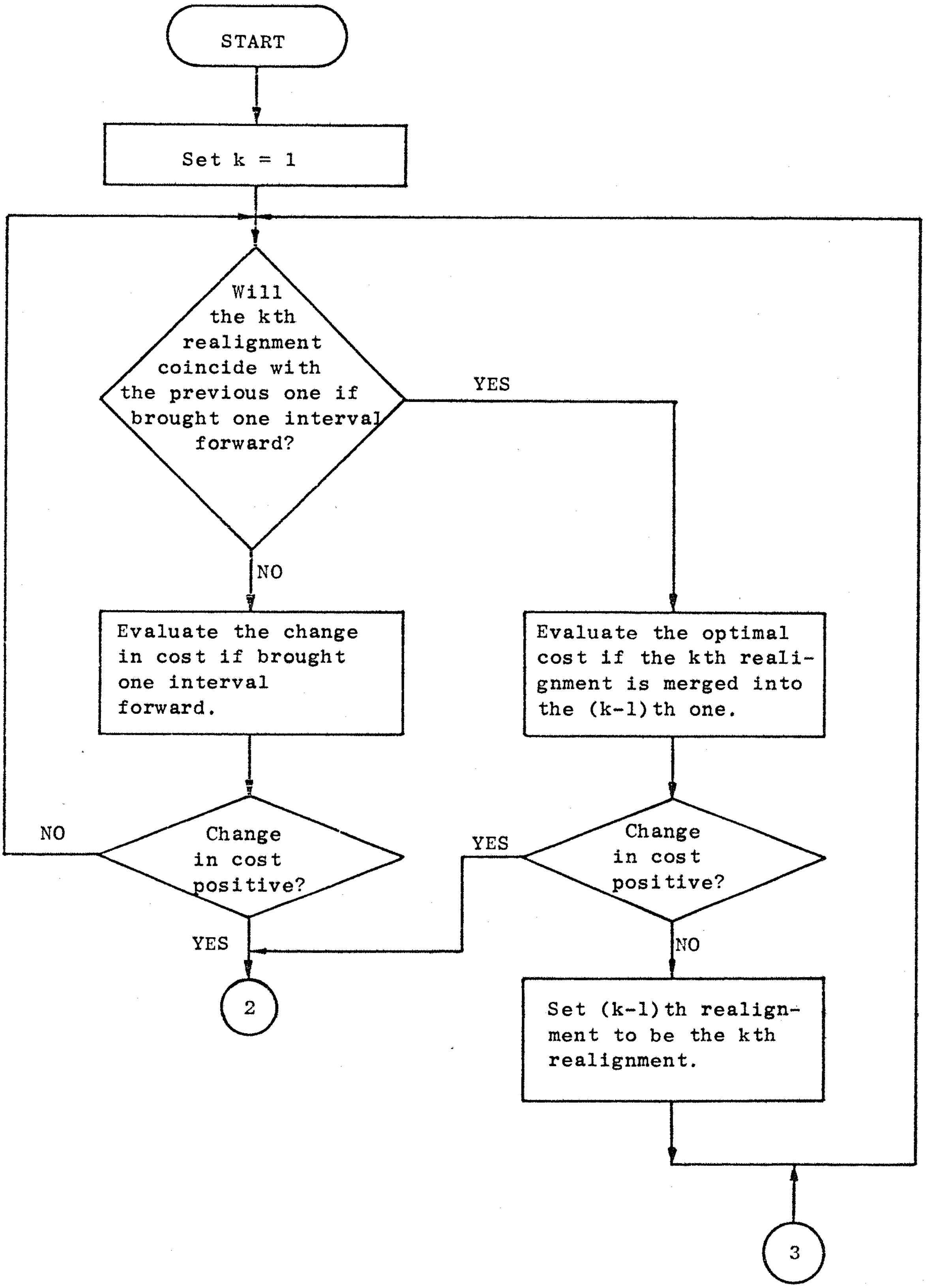

START
Set k = 1
Will the kth realignment coincide with the previous one if brought one interval forward?
YES
NO
Evaluate the change in cost if brought one interval forward.
Evaluate the optimal cost if the kth realignment is merged into the (k-1)th one.
NO
Change in cost positive?
YES
Change in cost positive?
YES
NO
2
Set (k-1)th realignment to be the kth realignment.
3

Figure 5.18(a) Flow Chart of Third Stage (Bringing forward the realignment)

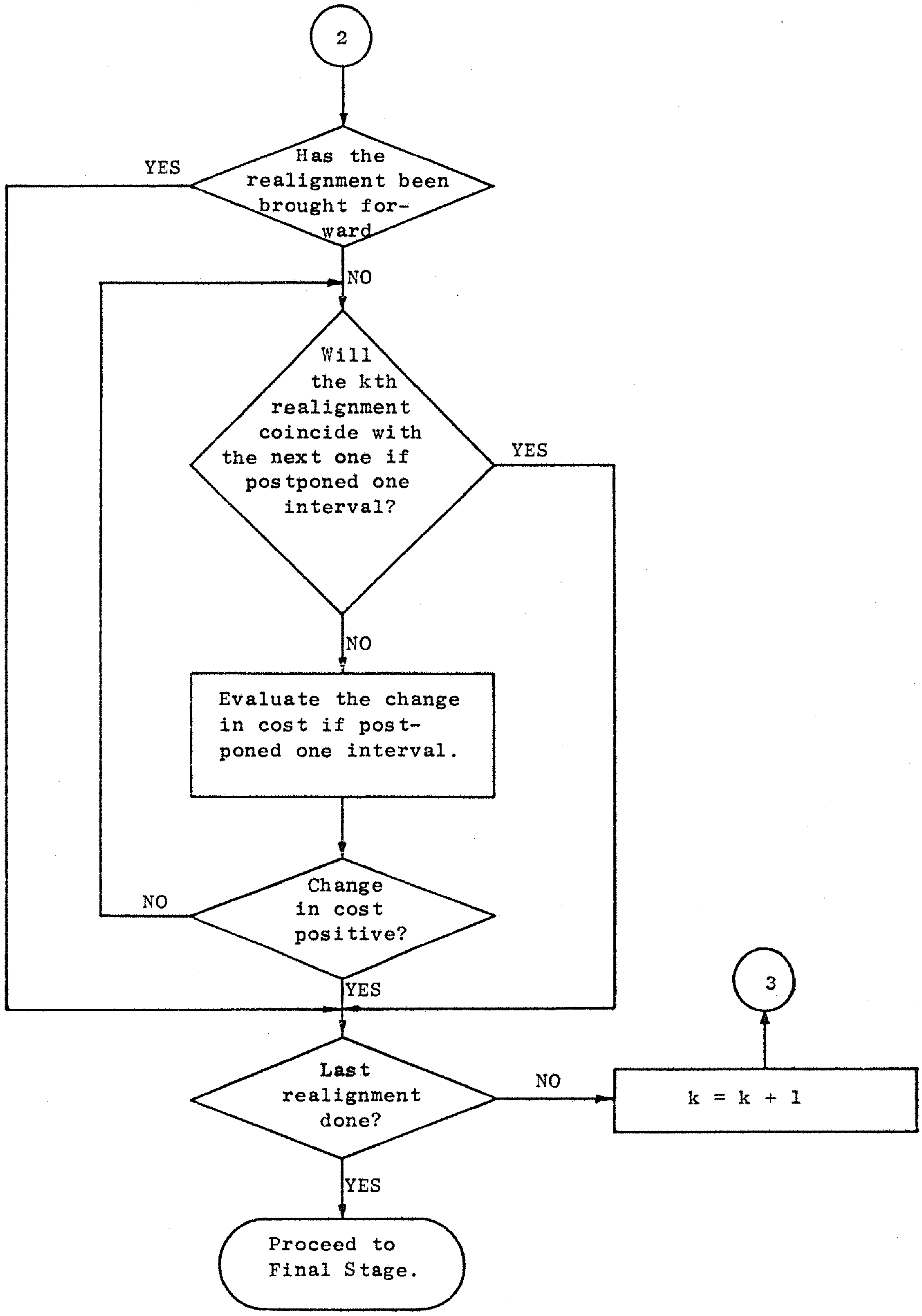

2
Has the realignment been brought forward
YES
NO
Will the kth realignment coincide with the next one if postponed one interval?
YES
NO
Evaluate the change in cost if postponed one interval.
Change in cost positive?
NO
YES
3
Last realignment done?
NO
k = k + 1
YES
Proceed to Final Stage.

Figure 5.18(b) Flow Chart of Third Stage (postponing the realignment)

CHAPTER 6

DIGITAL SIMULATIONS

Digital simulation of target tracking by the proposed dual-drive tracking servomechanism is essential to establish the performance and the validity of the proposed control policy. Furthermore, the proposed cost function is given in its general form only in the last chapter. The parameters of the weighting profiles on the probability of losing the target and on the frequency of realignments have yet to be determined, particularly the ratio between them. Putting too much emphasis on either one of the objectives of the cost function will result in poor tracking performance. It is, therefore, necessary to choose an optimal set of the parameters by means of digital simulation. The digital simulation was planned as a preliminary evaluation of the proposed control policy before the whole work is tested on a scaled demonstration rig. The whole specifications of the rig is listed in the next chapter. It is sufficient, at this stage, to mention that the demonstration rig is designed to run four times slower than the full size proposed system.

6.1 Ship Motion

There was no authentic ship rolling motion available for experiments. The ship motion could only be generated by passing a random white noise through three filters in parallel as described in Figure 2.7 of Chapter 2. Each filter is a second order system possessing different natural frequencies and damping ratios. Since all simulated motions are slowed down by a factor of four the scaled ship motion will then have three dominant frequencies four times slower than the true motion. By trial-and-error method, it was eventually agreed that the three filters should have the characteristics listed in Tabel 6-1. A typical time history of the scaled motion and its spectrum are shown in Figure 6.1 and Figure 6.2.

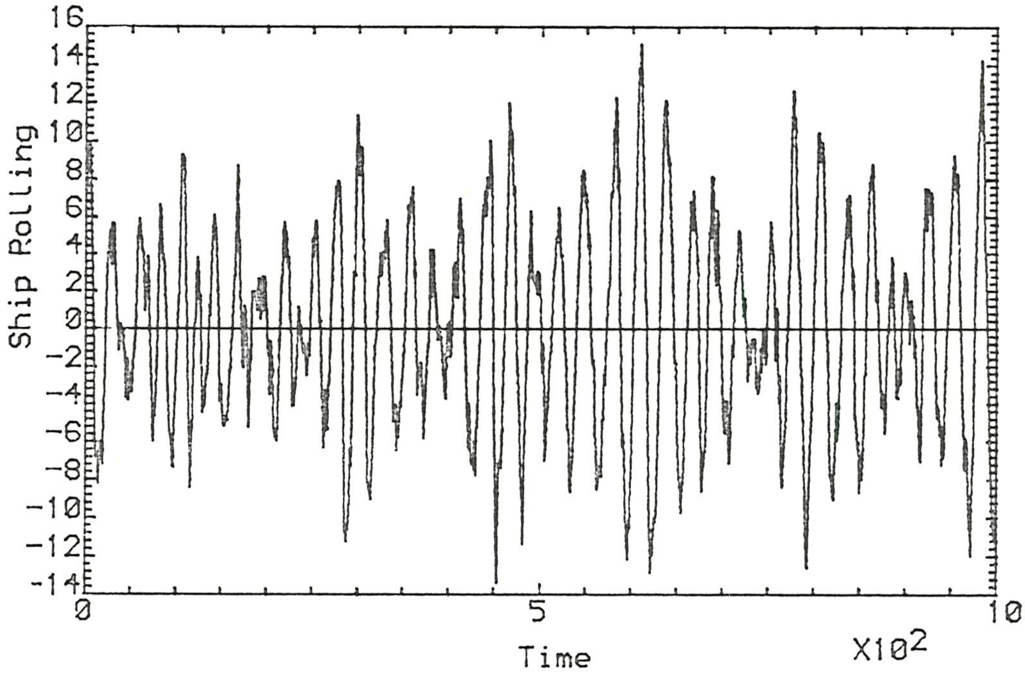


Figure 6.1 Time series of scaled ship motion

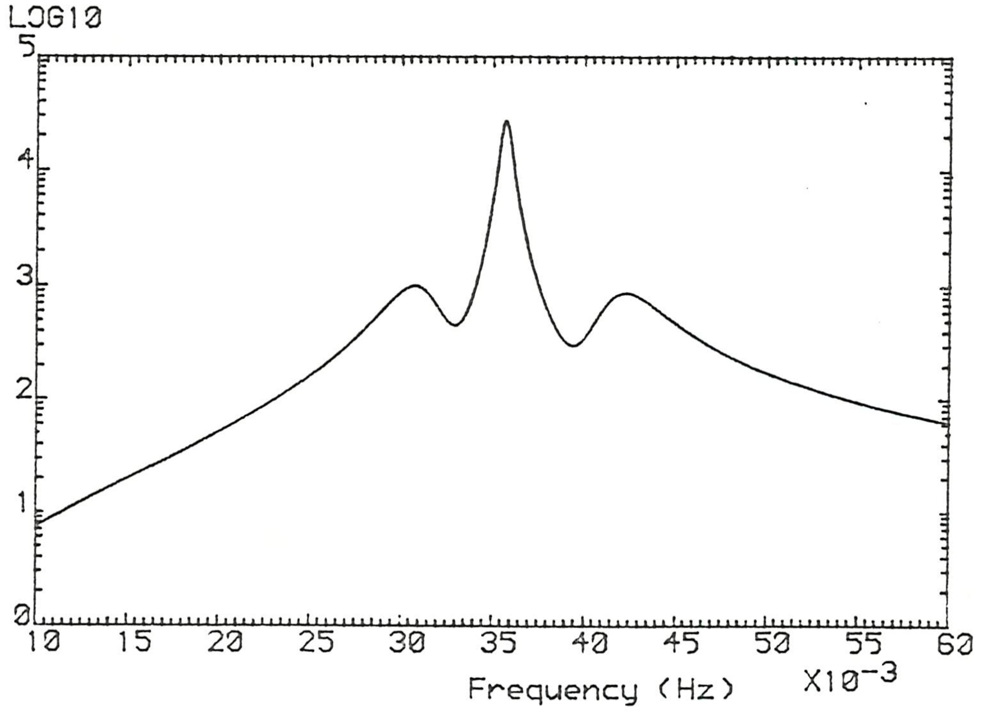


Figure 6.2 Spectrum of scaled ship motion

	Filter 1	Filter 2	Filter 3
Natural Frequency (Hz)	0.04167	0.03571	0.03125
Damping ratio	0.05	0.01	0.06
Gain	0.4	0.5	0.4

Table 6-1 Characteristics of Ship Motion Generator Filters.

It was, as discussed in Chapter 2, the ship motion is possibly best approximated by an autoregressive-moving-average model, ARMA(6,6). However, the parameter estimation process forces an autoregressive model, AR, to be used instead since the filtering algorithm has problems in tracking the parameters in the moving-average part. Theoretically, an ARMA(6,6) model can be represented by an infinite order of AR model. In practice, one has to compromise the order between accuracy and computation time. The computation time increases rapidly with the order of the model. In order to choose a reasonable order for the AR model to model the ship motion, the mean squares errors in estimating the simulated ship motion using various orders were studied. The results were plotted in Figure 6.3 and Figure 6.4. Figure 6.3 shows the effect of the sampling rates on the accuracy with various orders. On the other hand, the effect of the data length in estimating the parameters under various model orders is shown in Figure 6.4. Both graphs illustrate that the mean square errors decrease exponentially with model order. This follows from the fact that an invertable ARMA model can be written as a converging infinite order autoregressive model as mentioned in Section 2.7. Reiterating, an ARMA model is represented by

$$\phi(B)\ z(k) \ = \ \theta(B)\ u(k) \qquad (6.1)$$

(6.1) can be expressed as :

$$\theta^{-1}(B)\ \phi(B)\ z(k) \ = \ u(k) \qquad (6.2)$$

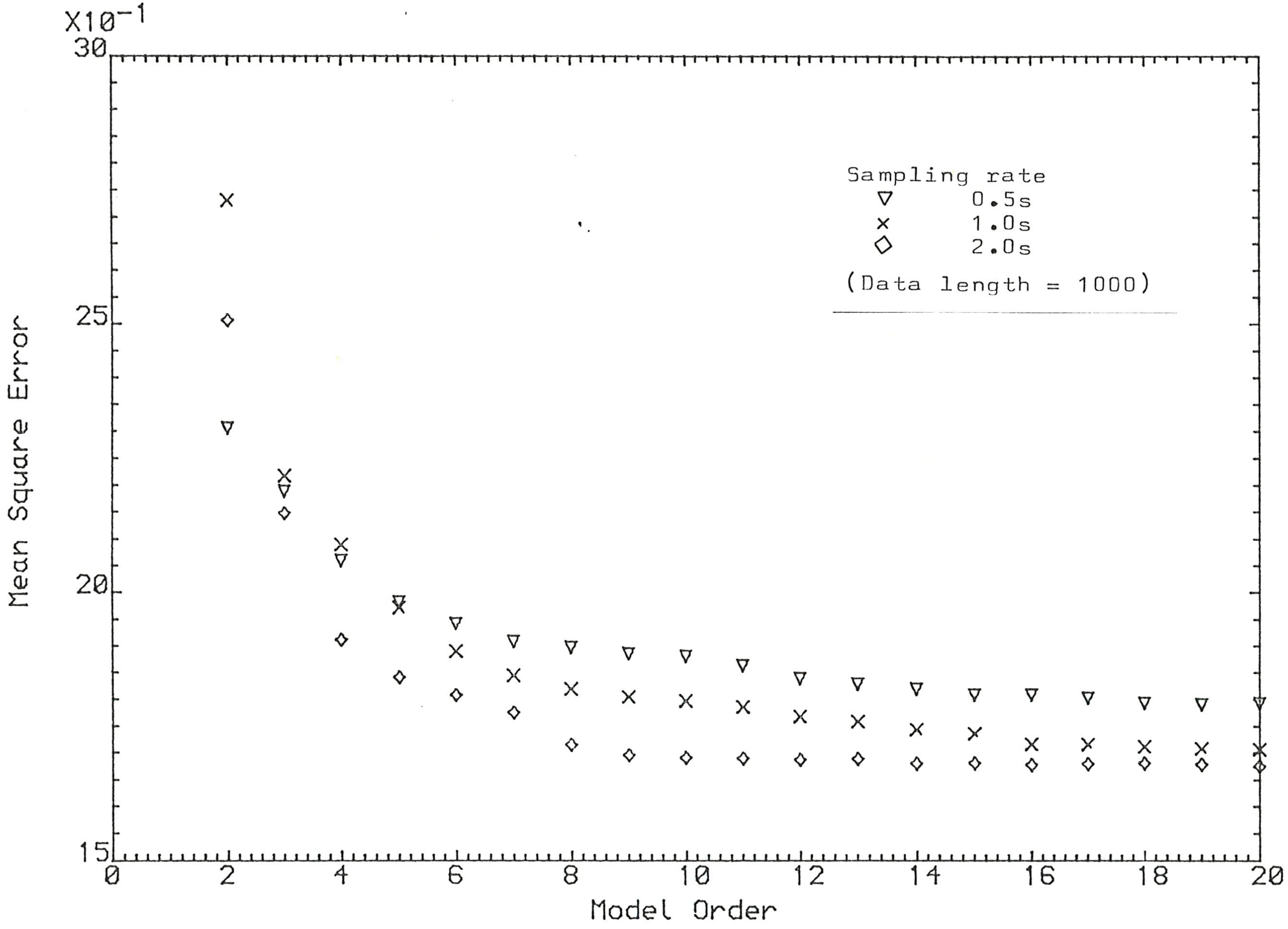


Figure 6.3 Mean squares error vs Model order at various sampling rate

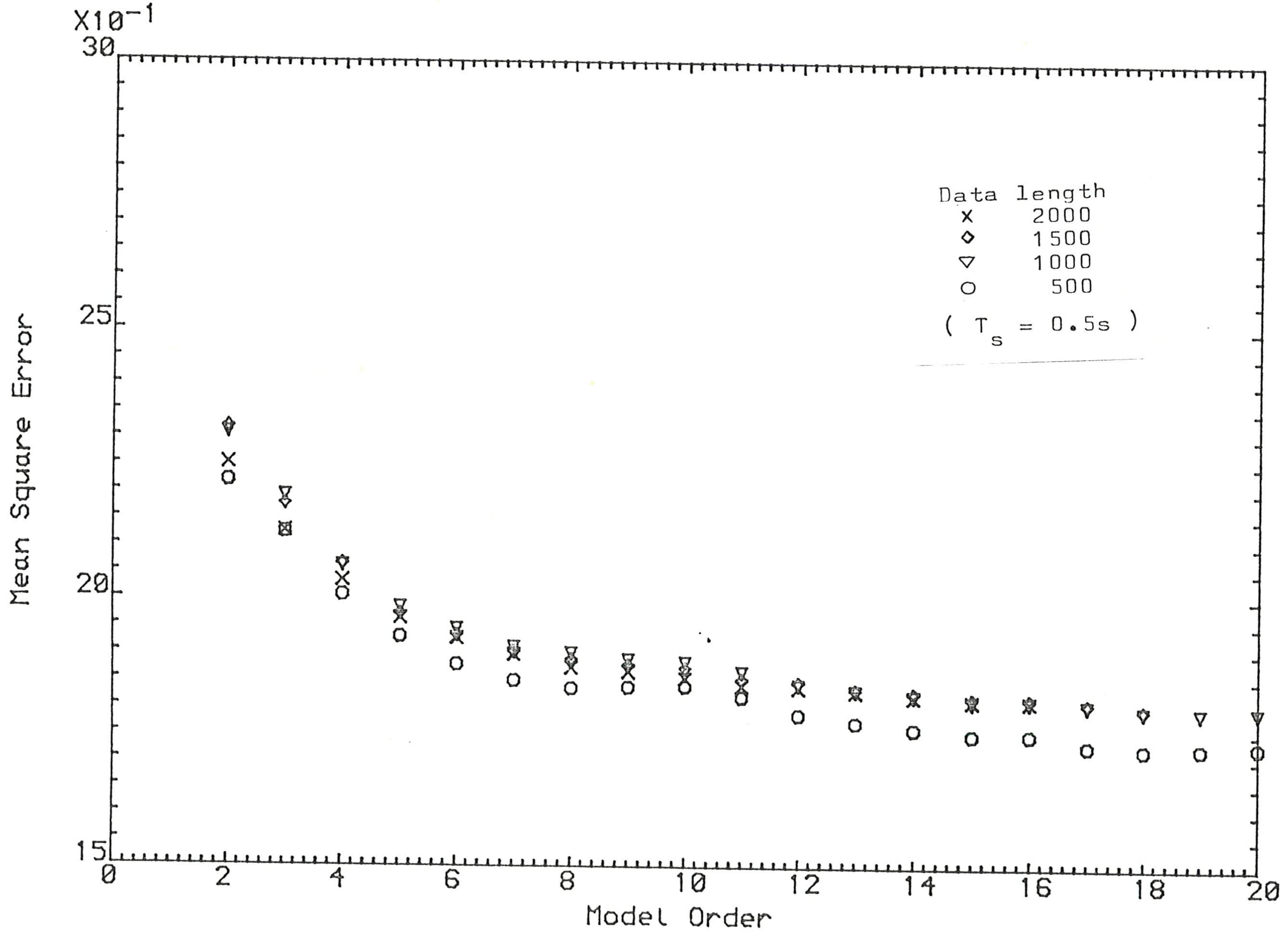


Figure 6.4 Mean squares error vs Model order at various data length

$$\text{OR} \quad z(k) = -\sum_{i=1}^{\infty} \psi_i \, z(k-i) + \theta_o \, u(k)$$

$$= -\sum_{i=1}^{p} \psi_i \, z(k-i) + \theta_o \, u(k) + \varepsilon \qquad (6.3)$$

If the roots of $\theta(B)$ lie outside the unit circle in the B-plane (6.3) is a converging series. The truncation error ε decreases with the order p. In Figure 6.3, it revealed that the mean squares errors are insensitive to the sampling rates. This is because the sampling rates, tried in the experiments, were much higher than twice the highest dominant frequency of the ship motion. This means that the ship motion was sampled without aliasing effect. The characteristics of any ARMA process are related to the positions of the poles in the B-plane regardless of the sampling rate. As long as the roots of $\theta(B)$ are outside of the unit circle, an AR process is sufficient to approximate the ARMA process. Figure 6.4 showed that the windowing length does not play a significant part in the estimation. This is because there were enough data points to perform the estimation. It is therefore suggested that the orders of four to eight may be chosen to model the ship motion. No significant improvement on accuracy can be benefitted from using higher order. It must, however, emphasize that the true ship rolling motion is not exactly an ARMA(6,6) process. Therefore, the suggested order is only true in the case of the simulated motion generated from three parallel filters. A sixth order AR model was used throughout the digital simulation.

6.2 Implementation and Simulation Results

The main objective of the simulations was to evaluate the performance of the proposed control policy. Besides, the simulations were used to study the proposed cost function. The cost function proposed in the previous chapter is only in its general form. The profiles of the weights and some parameters have not yet been fixed for optimum tracking performance. There are at least nine variables in all in the simulation. Six of them are in the cost function and the seventh one is the target dynamics. The lengths of prediction period and window may also affect the performance. Thousands of simulation runs are required to study the effect of each variable on the performance. It was not possible to perform so many simulation runs in the time available. Therefore, a preliminary study of the variables was performed to pick out the essential ones.

Considering the weights in the cost function:

$$w_1(\tau) = Q\left[1 - C_1 \exp(-a_1\tau)\right] \qquad (6.4)$$

$$w_2(\tau) = R\left[1 - C_2 \exp(-a_2\tau)\right] \qquad (6.5)$$

there are six variables, Q, R, C_1, C_2, a_1 and a_2, that have to be determined. The reason of having the weights in the cost function is to put different emphasis on different objectives. In this case, there are only two objectives - minimum probability of losing the target, and minimum number of realignments. It is the ratio between the weights which is important. The ratio can be constant throughout an engagement because the relative importance between the two objectives remains constant. Therefore, the two weights can then be reduced to

$$w_1(\tau) = 1 - \exp(-a\tau) \qquad (6.6)$$

$$w_2(\tau) = r\ w_1(\tau) \qquad (6.7)$$

where a is time constant and r is the ratio between the two weights. The profiles of the weights remain to be an exponential decay to reflect the acceptable risk at various time during an engagement. Thus, only two variables need consideration in the weights.

Among the other parameters in the simulation, the target trajectory and the ship motion cannot be ignored. These two are the essential signals in the evaluation of the proposed control strategy performance.

The length of the prediction period was not chosen to be studied at this stage because it was felt that it is meaningless to have a very long prediction period as the confidence on the forecast decreases as the fourth power of the lead time. Moreover, the optimal predicted slew motor trajectory is a series of step functions. The decision on a realignment at current time will not be affected by the length of the prediction period. The decision is mainly due to the forecasts between the current time and the first predicted realignment.

From the result in the last section, the window length will have little effect on the estimation process. Hence, it can also be ignored at this stage of simulation.

In summary, there are only four essential variables in the simulation. They are:

i) target trajectory

ii) ship rolling motion

iii) ratio between the two weights

iv) decaying time constant of the weight profile

The simulation was performed as shown in Figure 6.5. Each cycle involved the generation of the simulated target trajectory and the ship motion, the parameters estimation using either the canonical Kalman filter or the finite memory version, the prediction of the motions, optimisation of the cost function, and finally the realignment of the slew motor if needed. The simulation program was written in modular form and FORTRAN was the chosen language. Each module performed one particular task. The main body of the program then comprised of a series of subroutine calls. Writing in modular form simplifies the development of programs. It permits programs to be debugged more easily and they are less prone to errors. Each subroutine can be tested individually for error. However, there is a price to pay. It generally requires more computation time than those programs written in pipe-line form.

The significance of the selected variables were studied individually. One of the four variables was varied while the other three were kept fixed. In all, five sets of simulations were carried out. Among them, three different target trajectories, three different ship rolling motions, five different decaying time constants and five different weight ratios were used.

The simulated target trajectory was shown in Figure 6.6. The target was assumed to be flying at constant altitude with constant velocity initially. After a certain time, the target performed a manoeuvre by initiating a constant acceleration and a change in the heading. This kind of trajectory is closely related to a typical engagement, except that the true trajectory should be smoother at the transition between the pre-manoeuvre and the post-manoeuvre period. The parameters of the three target trajectories used in the simulation were listed in Table 6-2. One of the three trajectories represented a target flying at constant altitude with constant velocity during the whole engagement. The accelerations and angles of manoeuvres in the second and the third trajectories were chosen randomly. The acceleration

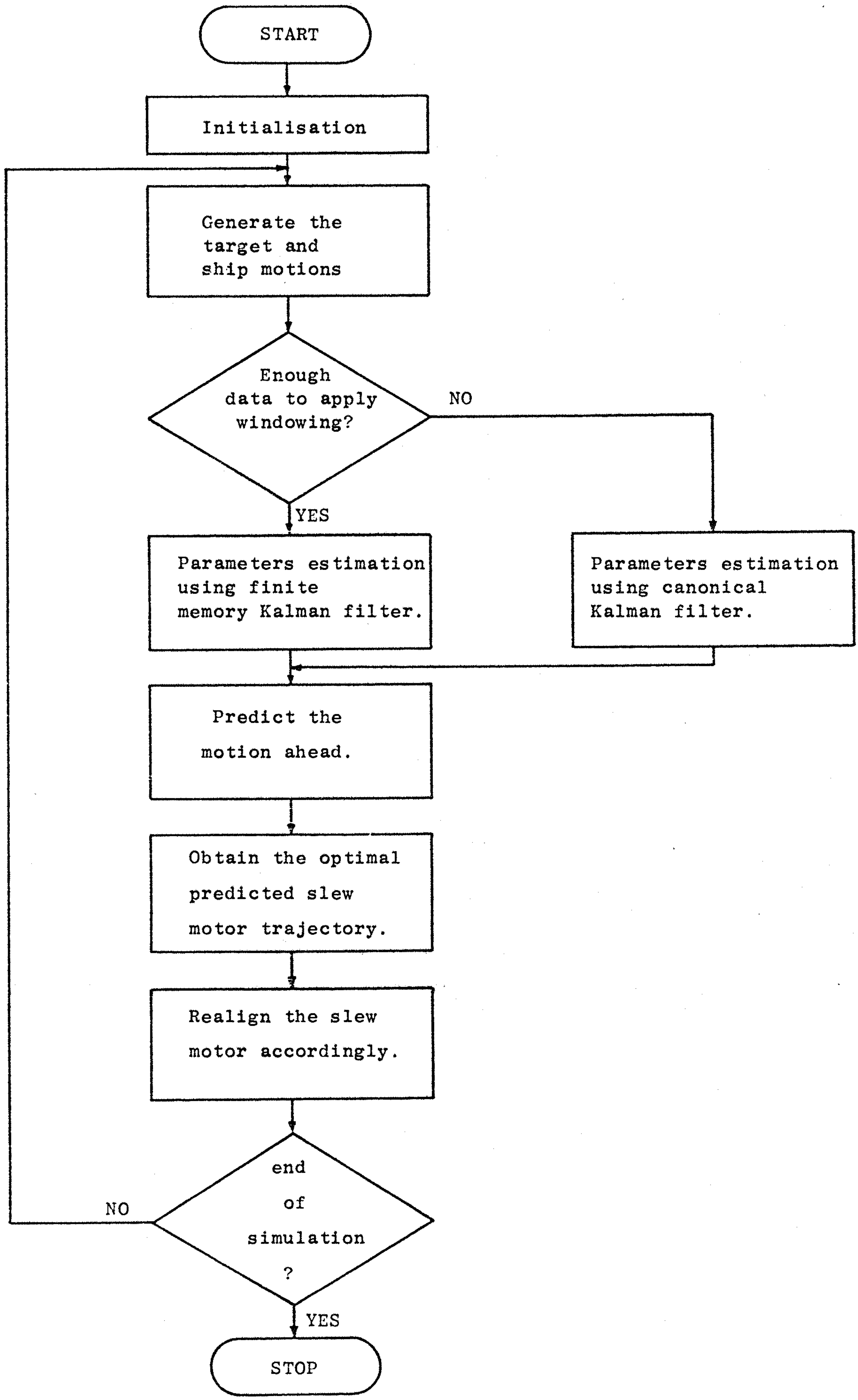

START
Initialisation
Generate the target and ship motions
Enough data to apply windowing?
NO
YES
Parameters estimation using finite memory Kalman filter.
Parameters estimation using canonical Kalman filter.
Predict the motion ahead.
Obtain the optimal predicted slew motor trajectory.
Realign the slew motor accordingly.
end of simulation ?
NO
YES
STOP

Figure 6.5 Flow Chart of Digital Simulation

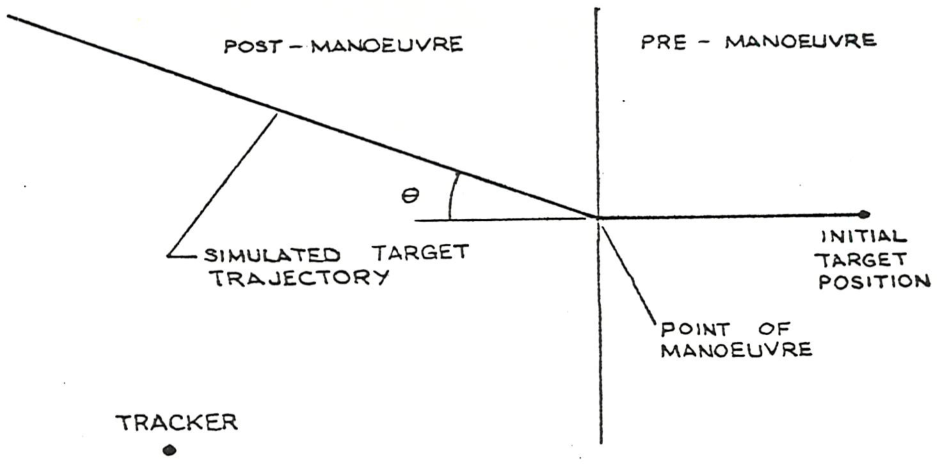


Figure 6.6 Simulated Target Trajectory

were taken from the range of 0 to 9g, whereas the changes in the angle of heading were within the range of 0° to 90°. The times of the manoeuvres took place were also randomly chosen. The trajectories assumed the target to be able to change its acceleration and angle of heading instantly at the times of manoeuvres.

The ship rolling motion was created by passing a pseudo random number sequence to three parallel filters with characteristics as shown in Table 6-1. Three different time history of the ship rolling motion were achieved by using three different pseudo random number generators.

Besides the four selected variables, there were other parameters needed to be fixed a priori. First of all, a sampling period of 0.4 seconds was chosen for the simulation. Because of the scaling factor in time, the full size system would be sampling at a rate of 0.1 second, i.e. the highest possible bandwidth of the motion that the slewing system could cover was five Hertz . The track motor would, however, have a much higher bandwidth.

The window length used was fifty data points long, and the prediction period was chosen as forty data points. The chosen window length and prediction period were mainly governed by the limitation of the computer storage and the time to perform the optimisation. Nevertheless, a prediction period of sixteen seconds in simulation, sixty-four seconds in real-time, should be long enough to anticipate the slew motor trajectory. In real-time, a target moving at a speed of Mach 2 will cover over forty-three thousand meters in sixty-four seconds.

Parameter	Unit	Target Trajectory		
		1st	2nd	3rd
Distance from tracker at initial engagement	m	50,000	50,000	50,000
Altitude at initial engagement	m	1,000	1,000	1,000
Velocity at initial engagement	m/s	150.0	150.0	150.0
Heading at initial engagement	°	0.0	0.0	0.0
Starting time of manoeuvre	s	0.0	46.0	66.0
Duration of manoeuvre	s	0.0	144.0	80.0
Acceleration of manoeuvre	m/s^2	0.0	5.46	2.54
Change of heading of manoeuvre	°	0.0	36.9	79.2

TABLE 6-2 Target Trajectories Tested

Since any measurement is always corrupted by noise the observations of the target were assumed to have a variance of five degrees. Each simulation run lasted two hundred seconds in scaled time. The working range of the tracking motor was limited to five degrees.

As a convenient means of representing the success/failure of the technique, the percentage of time that the tracker lost its target and the percentage of realignment time were evaluated. The interaction between the slew motor and the tracking motor was ignored in the simulation. It was assumed that the realignment was taken instantly. The percentages from the simulation runs were listed in Tables 6-4 to 6-8. The five weight profiles used in the simulations were plotted in Figure 6.7. A selection of the simulation runs were graphically shown in the Appendix. A typical simulation result is shown in Figures 6.13, 6.14, and 6.15.

6.3 Conclusions

From the simulation results (Figures 6.13, 6.14, 6.15 and Appendix C) they illustrate that the estimations of the target trajectory and the ship motion were very effective. Throughout the simulation, the estimated combined motion was very close to the noise free motion. This indicates that the sixth order autoregressive model was adequate in modelling the autoregressive-moving-average process. The prediction of the combined motion was also capable of indicating the trend of the motion (Figure 6.14). The simulated ship rolling motion seemed to be unrealistically random. The true ship rolling motion might not fluctuate as violently as simulated.

The simulation results (e.g. Figure 6.13) also revealed that the simulated ship rolling motion dominated the performances due to its wild fluctuations. In comparison, the target trajectory was smoother than the ship motion. Thus, the ship rolling motion was the main cause that the proposed tracking system would lose its target. Work has to be concentrated on the problems associated with the ship rolling motion. Particularly, it would be beneficial to investigate the possibility of modelling the ship motion with a deterministic model, e.g. a polynomial function.

The results in Tables 6-4 to 6-8 showed that the performance was insensitive to the weight profile. This may be the result of the application of a short prediction period. However long predictions are not recommended. It is partly because the models used in the estimation process do not match the motions exactly. Long prediction periods will then inevitably incur large biases on the forecasts near the end of the prediction period. Furthermore, the confidence on the forecasts drops at a rate to the fourth power of the lead time. Therefore it is feasible to use an uniform weighting instead of an exponential one. This simplifies the cost function to:

$$J = \sum_{m=0}^{Np} \{ 1 - \Pr(\theta_s(t+m) - \theta_\ell \leqslant \theta_T(t+m) \leqslant \theta_s(t+m) + \theta_\ell)\}$$

$$+ \; W \sum_{m=1}^{Np} \left| \text{sign}\,(\theta_s(t+m) - \theta_s(t+m-1) \right| \qquad (6.8)$$

There is a spin-off of employing an uniform weight profile. The time constants of the five profiles used in the simulation were chosen based on an assumption that the engagement lasted for two hundreds seconds in scaled time. The absolute values of the time constants are meaningless to the cost function. It is the ratio between the rise time and the duration of an engagement which is important. A long rise time with a long engagement will have the same effect as a short rise time with a short engagement. The two profiles may be exactly the same if they are plotted with the time axis normalised. In real life, the duration of an engagement is an uncertainty. There is no way to determine the duration of an engagement at the time a target is first established. Hence, it is impossible to establish the time constant for the weight profile. The problem of the uncertainty on the duration of an engagement is completely eliminated by employing an uniform weight profile over the whole engagement.

The results in Figures 6.8 to 6.12 demonstrated nicely that the percentage of miss[†] is inversely related to the percentage of realignment. Frequent realignment gives low percentage of miss and vice versa. The results also revealed that as the weight ratio increased, i.e. the emphasis on the realignment increased, the percentage of miss changed from low to high with a sharp transition within the range of weight ratio of one to three. On the other hand the percentage of realignment moved from high to low more gently. By plotting the axis of weight ratio in logarithmic as it was in Figures 6.8 to 6.12, the percentage of realignment demonstrated an exponential relationship with the weight ratio. If the weight ratio was very high, i.e. the realignment was more heavily weighted, no realignment would take place. The misses would be saturated at its maximum value. This maximum value of percentage in miss depended on the motion. On the other hand, a very small weight ratio would give a zero

[†] A miss in the simulation is when the target not covered by the tracking motor.

percentage of miss, but a high percentage of realignment dependent on the motion, too. The objective of the control strategy is to have minimum realignments and minimum misses. Therefore, the best weight ratio could be the cross-over point of the two curves of realignments and misses. Figures 6.8 to 6.12 suggested that such a cross-over point occurred at a weight ratio of between two and three regardless of the target trajectory and the ship motion. This means that the realignments was best weighted twice, or triple, the probability of losing target.

The simulation results also suggested that the restriction of one realignment within the prediction period was justified. The three dimensional plots of the evolution of the predicted slew motor trajectory (e.g. Figure 6.15) showed that, at most, only one realignment was predicted at each stage. This phenomenon might be due to the characteristic of the forecasts. In Figure 6.14, it showed that the forecasts at one time behaved like an exponential function. The forecasts tended to settle down to a steady state value. The steady state phenomenon is the typical characteristic of the stochastic autoregressive model [8]. It was also due to the fact that the changes in the target trajectory was insignificant compared with the ship motion. The heavily damped transient of the forecasts was the characteristic of the estimated autoregressive model. This steady state phenomenon in the forecasts also substantiates the previous point that long prediction period is not necessary. Once the forecast settles down to its steady state, the predicted slew motor trajectory will then remain constant. Therefore, any realignment is likely to take place during the transient. It is unreasonable to expect more than one realignment during the transient if the working limit of the tracking motor is comparable with the changes in the forecasts. In Figure 6.15, it also revealed that the predicted slew motor trajectory at one stage was not closely related to the one predicted in the previous stage. This made it hard to devise a recursive method to up-date the predicted slew motor trajectory. The uncorrelation might be because of the unrealistically fluctuated ship motion.

The reason of using three different ship rolling motion was to study the consistance of the simulation results. It also aimed to eliminate any possibility that one ship motion was favourable to the proposed control strategy. However, the results showed consistent performance over the three simulated ship rolling motion.

In summary, the simulation results suggested that the cost function could have an uniform weight with a weight ratio between two and three. The results also pointed to the application of a short prediction period. The exact length of the period depends on the characteristic of the estimated autoregressive model. It happened that a prediction period of forty data points was sufficient. The simulation results also revealed that the restriction of one alignment is realistic. Therefore the task of optimisation can then be simplified. It will then only involve the search of the point where the amount of realignment is twice the difference between the present slew motor position and the forecast. This will then reduce the computation overhead further.

	Target Trajectory		
	1st	2nd	3rd
1st ship motion	Table 6-4 Fig. 6.8	Table 6-5 Fig. 6.9	Table 6-6 Fig. 6.10
2nd ship motion	Table 6-7 Fig. 6.11	/ /	/ /
3rd ship motion	Table 6-8 Fig. 6.12	/ /	/ /

Table 6-3 Index to Tables 6-4 to 6-8 and Figures 6.8 to 6.12.

Weight	Reciprocal of Weight Profile Time Constant.				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1	61.6	61.2	61.2	61.2	61.2
0.5	41.2	41.6	41.4	41.4	41.6
1.0	32.3	32.2	32.4	32.4	32.4
2.0	20.4	20.8	21.0	21.6	21.6
3.5	12.2	12.6	12.6	12.6	12.6
5.0	7.0	7.2	7.4	7.4	8.6
10.0	0.6	0.6	0.6	0.6	0.6
20.0	0.0	0.0	0.0	0.0	0.0

PERCENTAGE OF REALIGNMENTS

Weight	Reciprocal of Weight Profile Time Constant.				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1	1.0	1.0	1.0	1.0	1.0
0.5	3.8	4.2	4.4	4.4	4.0
1.0	7.8	7.4	7.4	7.4	7.4
2.0	14.4	14.0	13.8	13.6	13.8
3.5	40.6	38.6	38.8	38.8	38.8
5.0	60.4	59.8	59.6	59.6	52.4
10.0	74.8	74.8	74.8	74.8	74.8
20.0	75.2	75.2	75.2	75.2	75.2

PERCENTAGE OF MISS

Note:- $x = \dfrac{\text{Weight on Realignments}}{\text{Weight on Probability of Target Loss}}$

Weight Profile $= 1.0 - 0.99 \times \text{Exp}(-Axt)$

Table 6-4 Simulation Results

Weight	Reciprocal of Weight Profile Time Constant.				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1		58.4			58.4
0.5	41.8	42.2	42.0	42.0	42.0
1.0	32.4	32.4	32.6	32.6	32.6
2.0	21.8	22.2	21.6	22.4	22.4
3.5	12.4	12.8	12.8	12.8	12.8
5.0	6.2	6.4	6.4	6.2	7.4
10.0		1.2			
20.0		0.2			

PERCENTAGE OF REALIGNMENTS.

Weight	Reciprocal of Weight Profile Time Constant.				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1		1.0			1.0
0.5	4.0	4.4	4.8	4.8	4.8
1.0	8.4	8.0	8.0	8.0	8.0
2.0	15.0	14.6	14.8	14.8	15.0
3.5	41.0	39.0	39.2	39.2	39.2
5.0	64.6	64.0	64.0	64.8	60.8
10.0		76.0			
20.0		86.2			

PERCENTAGE OF MISS

Note:- $x = \dfrac{\text{Weight on Realignments}}{\text{Weight on Probability of Target Loss}}$

Weight Profile $= 1.0 - 0.99 \times \text{Exp}(-Axt)$

Table 6-5 Simulation Results

Weight	Reciprocal of Weight Profile Time Constant				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1			59.0		
0.5		40.2	40.2	40.4	
1.0		31.4	31.2	31.4	
2.0		22.8	22.8	23.4	
3.5		13.6	13.6	14.4	
5.0		8.0	7.6	8.0	
10.0			1.2		
20.0					

PERCENTAGE OF REALIGNMENTS

Weight	Reciprocal of Weight Profile Time Constant				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1			1.0		
0.5		4.2	4.0	4.2	
1.0		7.4	7.4	7.4	
2.0		14.6	14.8	14.4	
3.5		37.8	37.4	37.0	
5.0		61.0	61.4	60.8	
10.0			74.8		
20.0					

PERCENTAGE OF MISS

Note:- $x = \frac{\text{Weight on Realignments}}{\text{Weight on Probability of Target Loss}}$

Weight Profile = 1.0 - 0.99 x Exp(-Axt)

Table 6-6 Simulation Results

Weight	Reciprocal of Weight Profile Time Constant				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1					
0.5	43.6	44.2	44.0	44.0	43.8
1.0	33.2	33.6	33.8	33.8	33.6
2.0	22.6	22.8	22.0	22.4	22.6
3.5	15.0	14.8	14.8	14.8	15.0
5.0	7.4	7.6	7.4	7.4	7.4
10.0					
20.0					

PERCENTAGE OF REALIGNMENTS

Weight	Reciprocal of Weight Profile Time Constant				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1					
0.5	3.8	3.6	3.6	3.6	3.6
1.0	5.8	6.0	5.8	5.8	5.8
2.0	14.8	14.8	14.6	14.6	14.6
3.5	28.0	27.0	26.8	27.0	27.2
5.0	46.6	46.6	46.8	46.8	46.6
10.0					
20.0					

PERCENTAGE OF MISS

Note:- $x = \dfrac{\text{Weight on Realignments}}{\text{Weight on Probability of Target Loss}}$

Weight Profile = 1.0 - 0.99 x Exp(-Axt)

Table 6-7 Simulation Results

Weight	Reciprocal of Weight Profile Time Constant				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1					
0.5	44.6	44.6	44.6	44.0	44.0
1.0	31.6	31.6	31.6	32.6	32.8
2.0	2o.0	21.0	21.2	21.2	21.2
3.5	13.2	13.2	12.6	12.6	12.2
5.0	7.4	7.4	8.2	7.8	7.8
10.0					
20.0					

PERCENTAGE OF REALIGNMENTS

Weight	Reciprocal of Weight Profile Time Constant				
Ratio	0.1	0.025	0.01	0.005	0.0025
0.1					
0.5	2.8	2.6	2.6	2.6	2.6
1.0	5.6	5.4	6.0	6.0	6.0
2.0	13.0	13.0	12.6	12.4	12.4
3.5	31.4	31.6	31.4	31.4	32.4
5.0	40.8	40.8	39.4	41.0	41.0
10.0					
20.0					

PERCENTAGE OF MISS

Note:- $x = \dfrac{\text{Weight on Realignments}}{\text{Weight on Probability of Target Loss}}$

Weight Profile = 1.0 - 0.99 x Exp(-Axt)

Table 6-8 Simulation Results

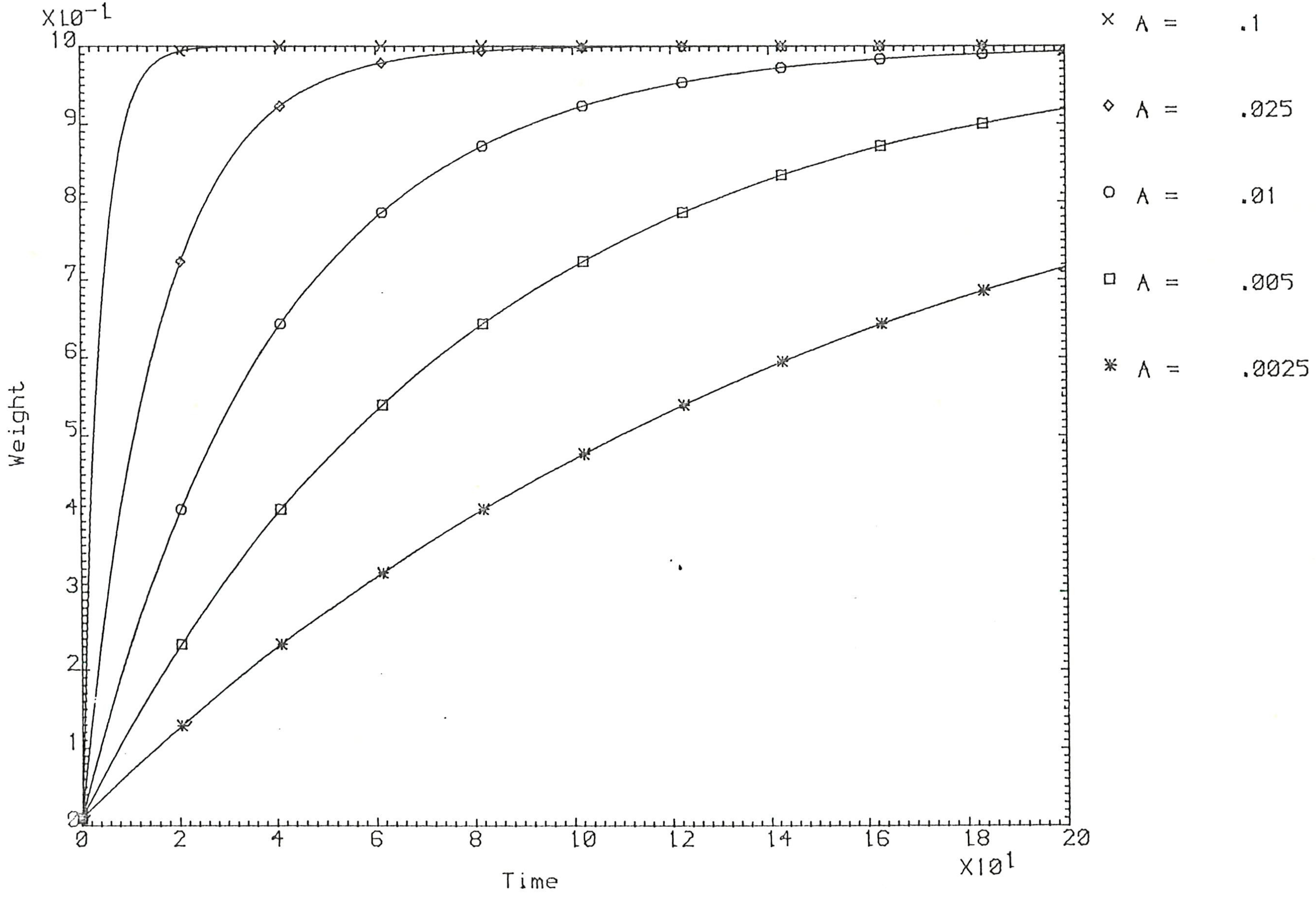


Figure 6.7 Weight profiles used in simulation

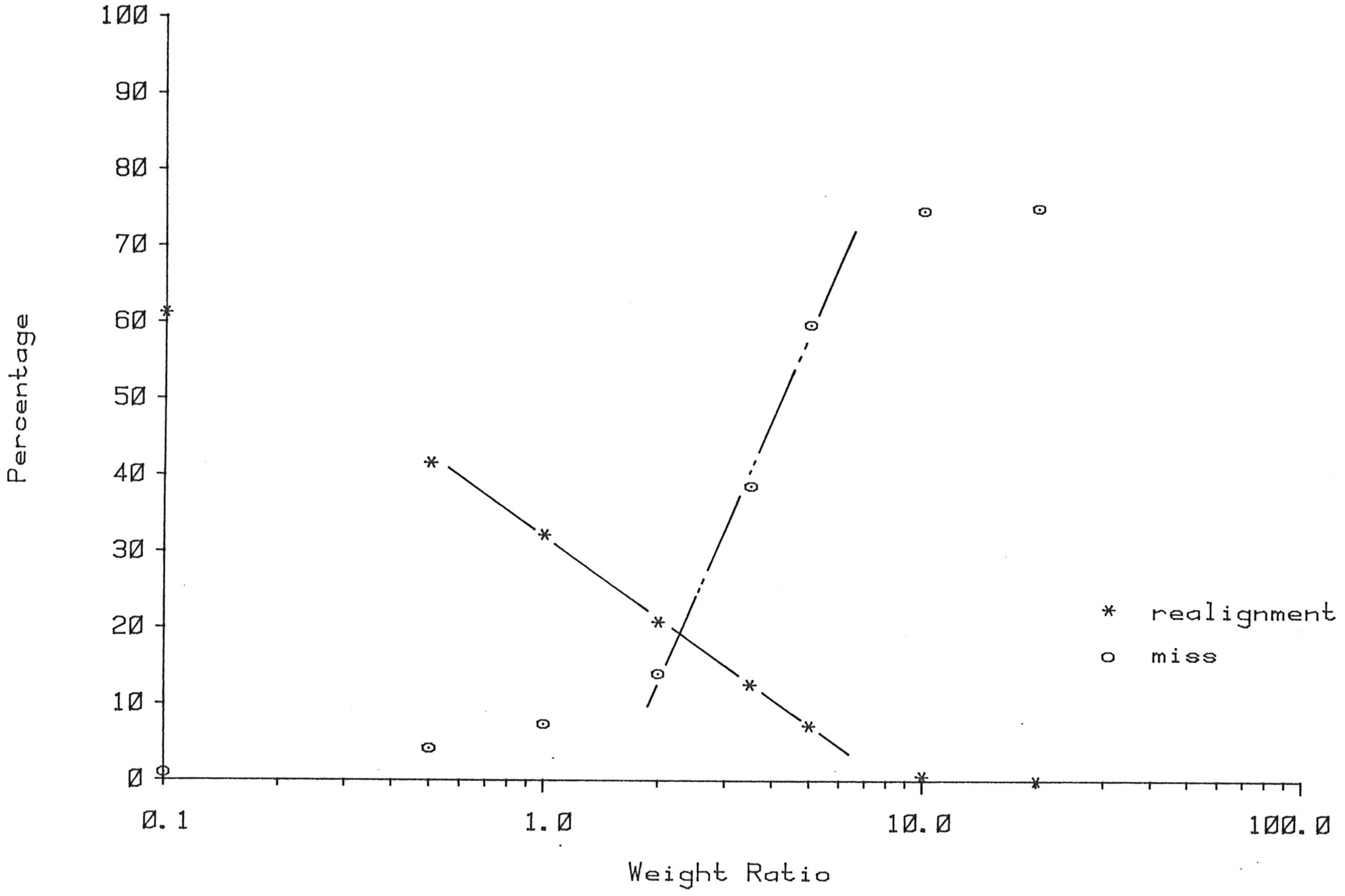


Figure 6.8 Performance of proposed strategy vs weight ratio

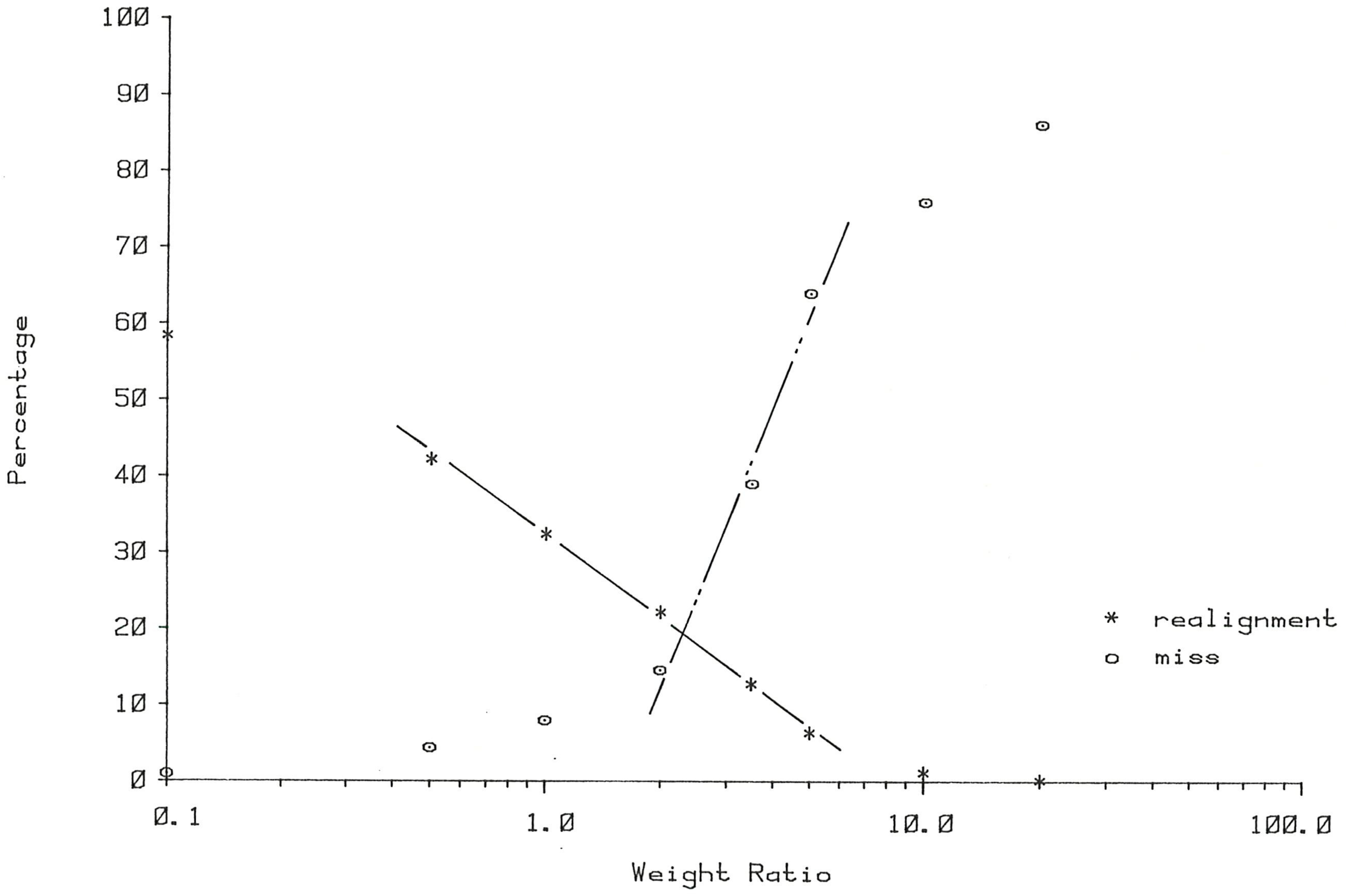


Figure 6.9 Performance of proposed strategy vs weight ratio

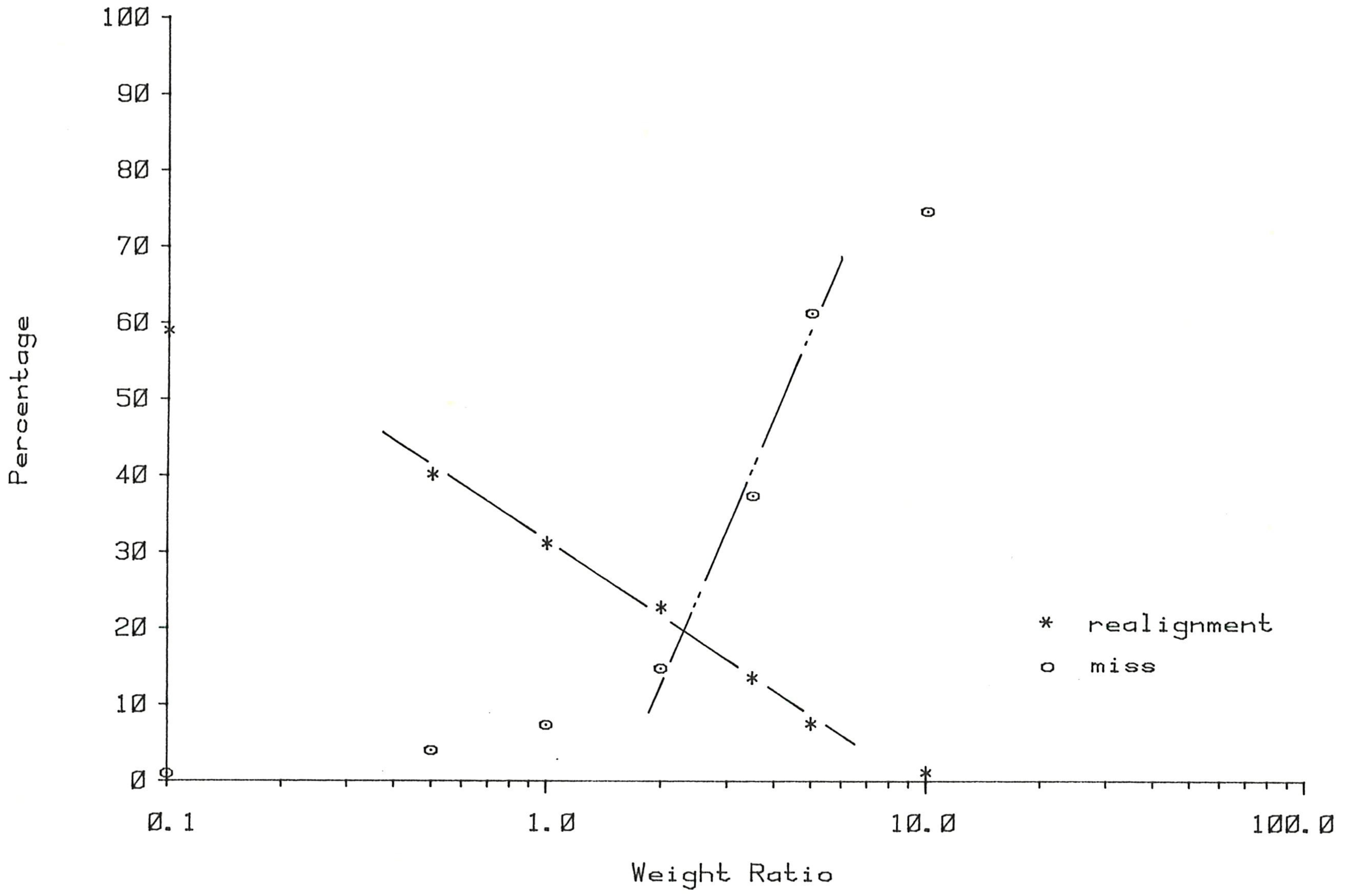


Figure 6.10 Performance of proposed strategy vs weight ratio

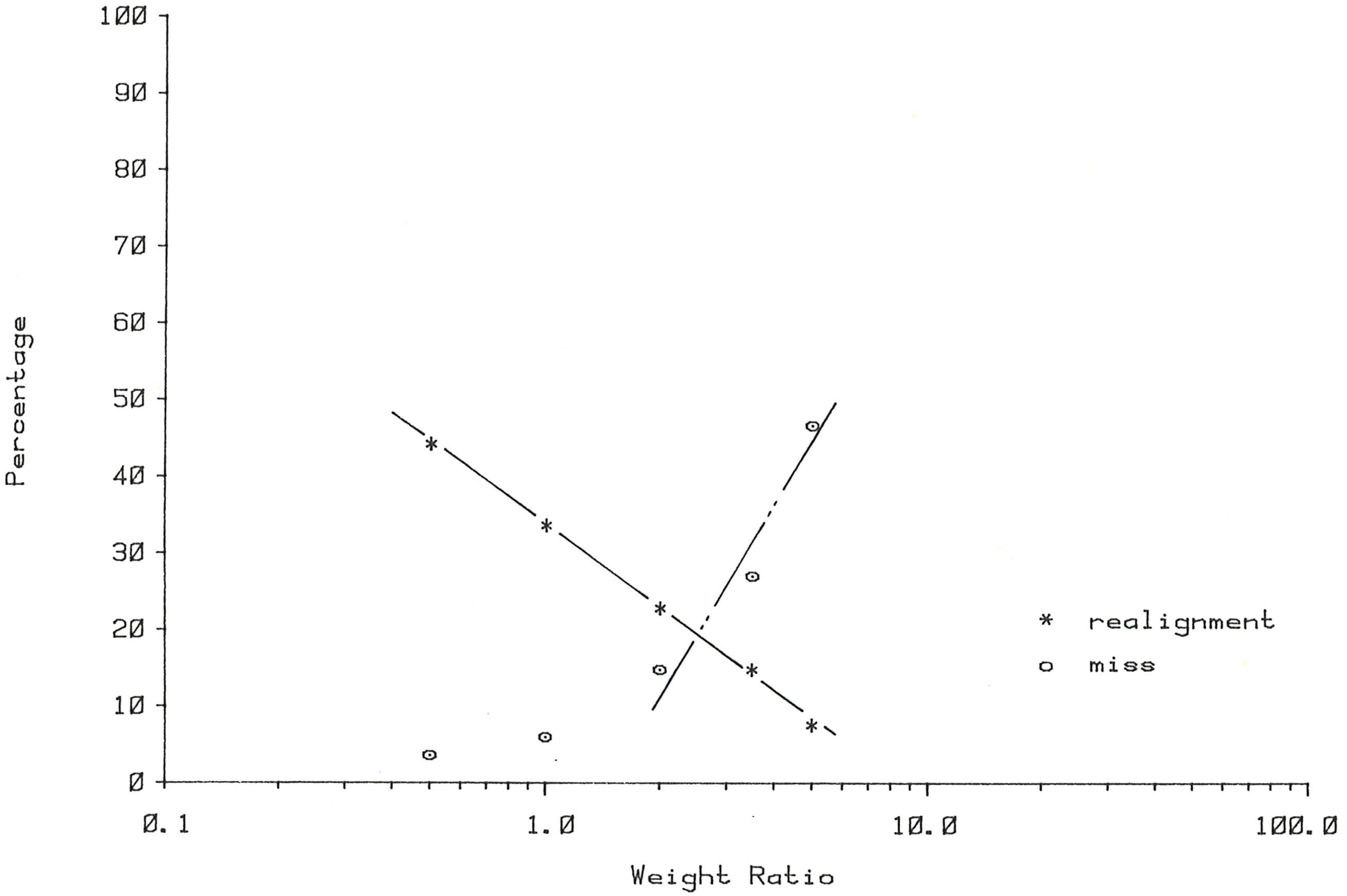


Figure 6.11 Performance of proposed strategy vs weight ratio

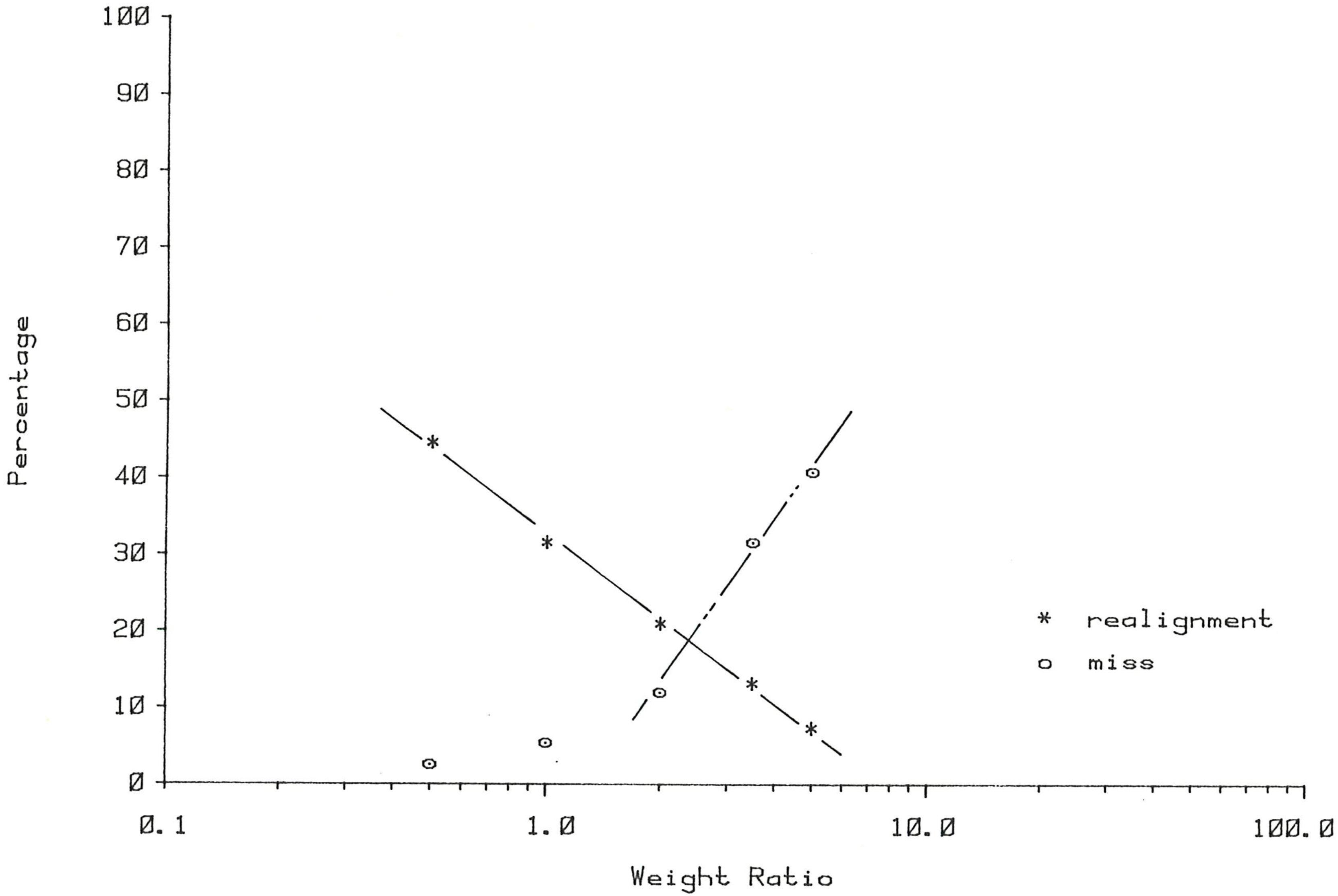


Figure 6.12 Performance of proposed strategy vs weight ratio

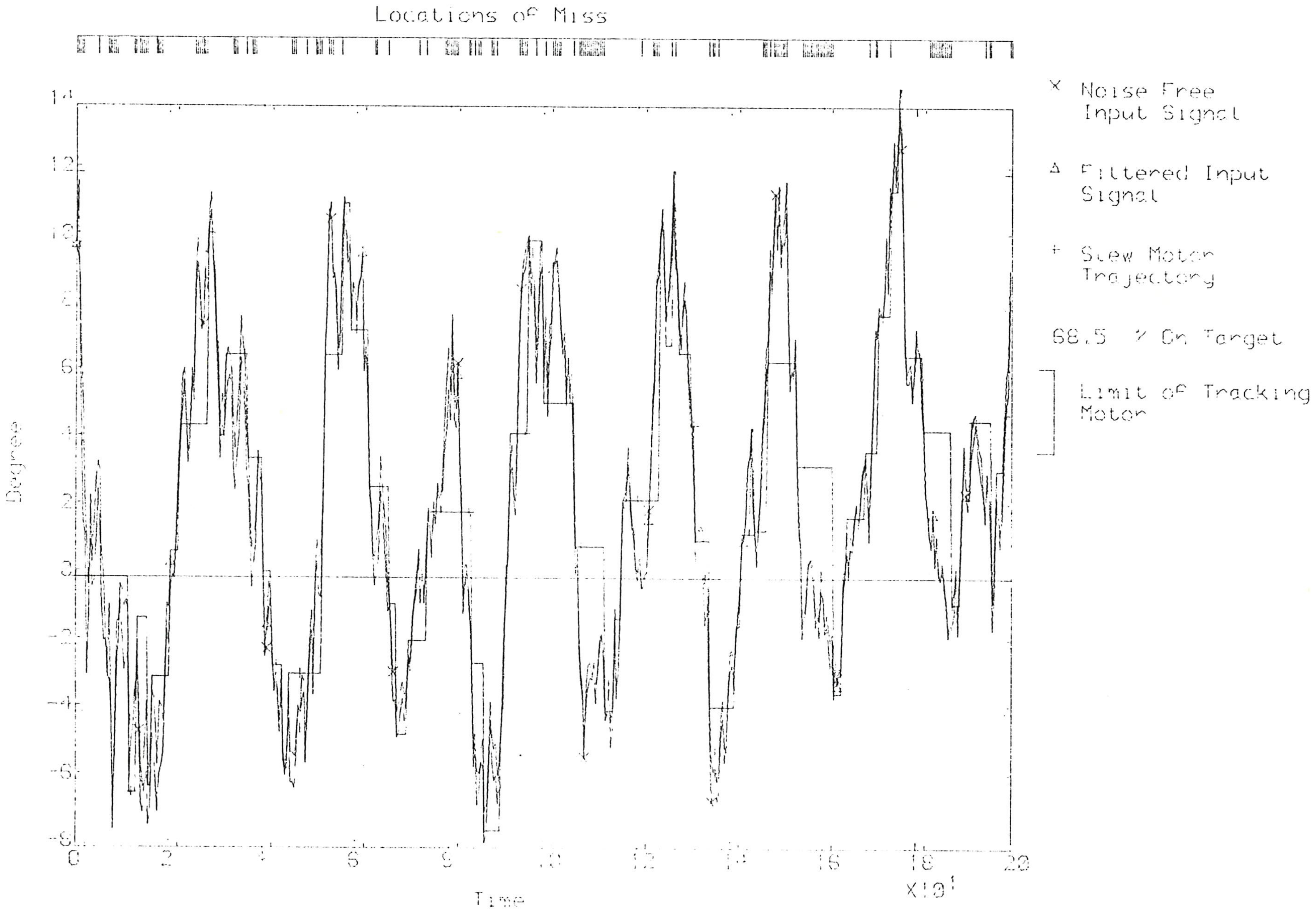


Figure 6.13 History of simulated engagement with a weights ratio of 3.5 and a weight profile of time constant 1/0.025

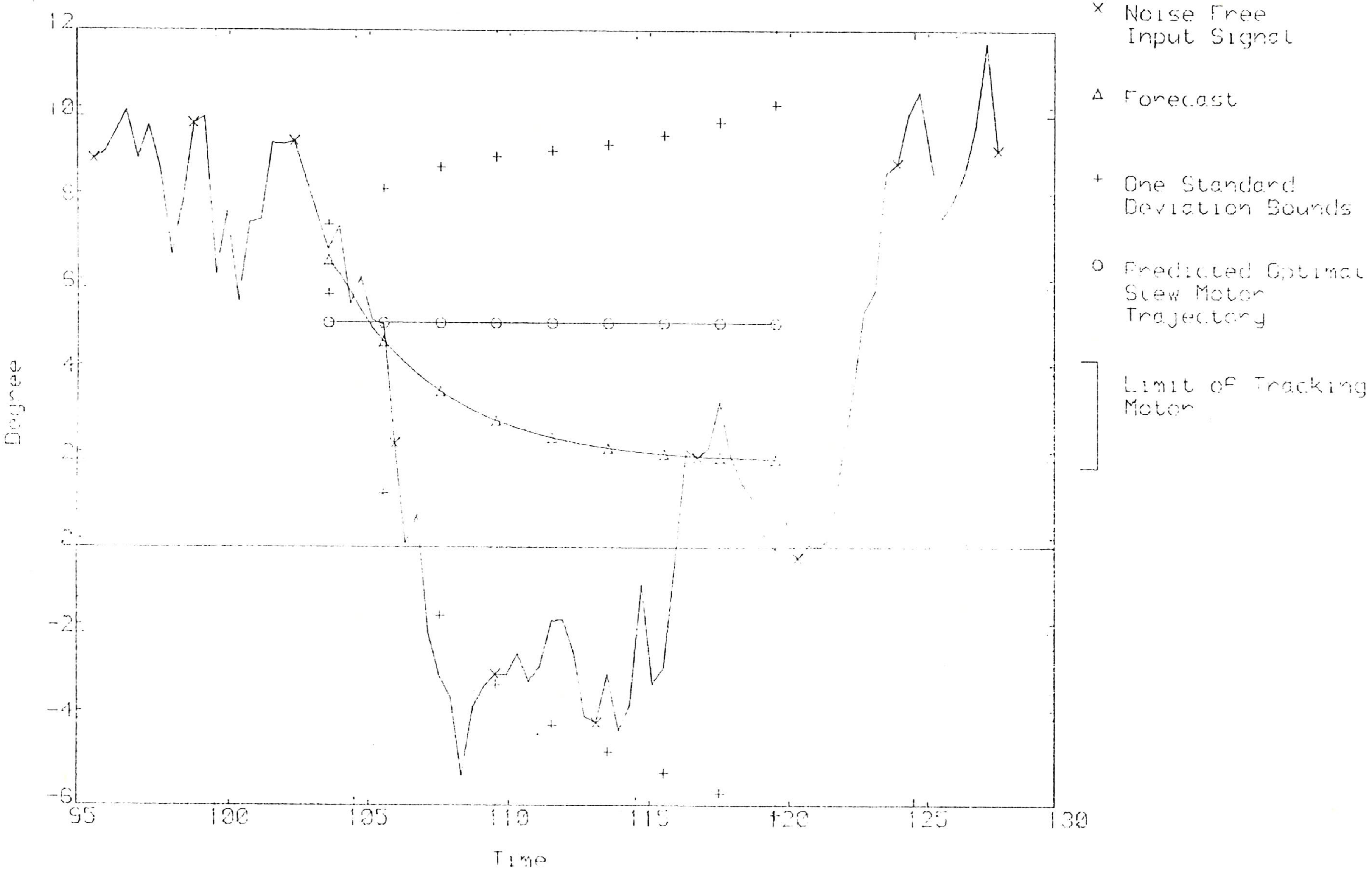


Figure 6.14 Forecasts and predicted optimal slew motor trajectory at time interval 100

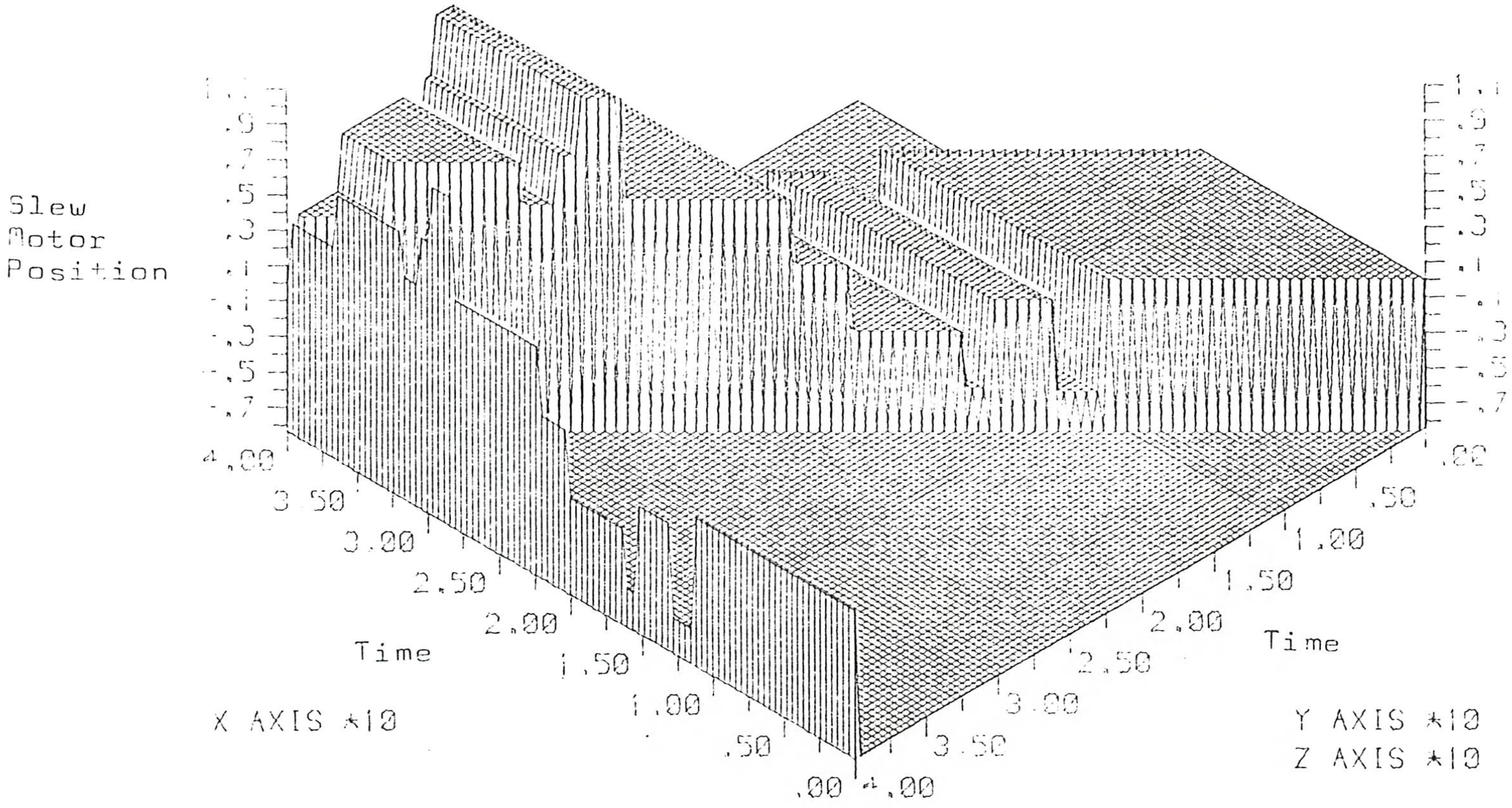


Figure 6.15 Predicted optimal slew motor trajectories at the first 40 intervals

CHAPTER 7

DEMONSTRATION RIG

A demonstration rig was planned from the beginning to demonstrate and to validate the control law and the principles involved. The rig should then possess the same characteristics as the actual system. It is not good practice to apply any new idea on a full scale prototype without first trying it out on a scaled down model. The eventual full size operational system is proposed to be a microprocessor controlled system. A microprocessor based monitor is the heart of the system. It is required to perform the task of modelling, forecasting, and decision making. The monitor also sends command signals to realign the slew motor. Therefore, the demonstration rig should comprise of two units: The microprocessor based monitor, and the driving unit.

7.1 Specification of Demonstration Rig

It was mentioned that the demonstration rig is only a scaled model of the actual proposed system. The scaling is applied to the physical dimensions and time. The scaling on the physical dimensions permits any physical simulation to be performed without the requirement of large power sources. It also allows the simulation to be carried out in a laboratory. The microprocessor for the demonstration rig is only a general one. The computing power is not as powerful as the most sophisticated one available at the present time. Hence, a time scaling is necessary on the demonstration rig. The scaling factors adopted were one-fifth and one-fourth for linear physical dimension and time respectively.

The full specifications for both actual proposed system and the demonstration rig are:

	Actual System	Dem. Rig
i) Accuracy (radian)	10×10^{-6}	as high as poss.
ii) Bandwidth (Hertz)		
slew motor servo	as high as poss.	2
tracking motor servo	50 - 100	20

	Actual	Demonstration
iii) Inertia load (kg-m^2)		
slew motor servo	12	19.2×10^{-3}
tracking motor servo	0.4	0.64×10^{-3}
iv) Working range (degree)		
slew motor servo	±180	±180
tracking motor servo	±2.5	±2.5

7.2 Monitor

The monitoring unit of the demonstration rig is basically a microcomputer system. Its computer power is a single card computer supplied by Cromenco. The single card computer was built around the versatile and commonly used Z-80A eight-bit microprocessor. The processor runs at a clock rate of four megahertz. The single card computer is compatible to the industrial S-100 communication bus. Hence, a lot of peripheral supporting hardwares are easily obtained from the existing market. The computer board has only one kilobyte of dynamic memory. It was then extended with a sixty-four kilobytes dynamic memory board. In order to allow the monitor to control the motors, an analogue input/output board was also attached to the system allowing the microcomputer system to communicate with the real world. An eight inch dual floppy diskette system was adopted as the mass storage media for developing required software. An arithmetic processing unit was added to the system to increase the speed of the system in performing the floating point arithmetic operations. A visual display unit was used in the communication between operators and the microcomputer. Hardcopies could be obtained from the dot matrix printer attached to the system. The block diagram of the microprocessor system is shown in Figure 7.1.

The computing power of this system is, however, not the fastest available in today's microcomputer market. It has been chosen because back-up services are easily obtained within the department of Mechanical Engineering. It was also the most cost-effective system at the time the system was ordered. Comparing its computing power with the more advanced microprocessor system, it was initially felt that it is four times slower. This is the ratio used as the scaling factor on time for the demonstration rig. Actually, there are a lot of factors affecting the computing speed. For instance, the efficiency of the high level language compilers.

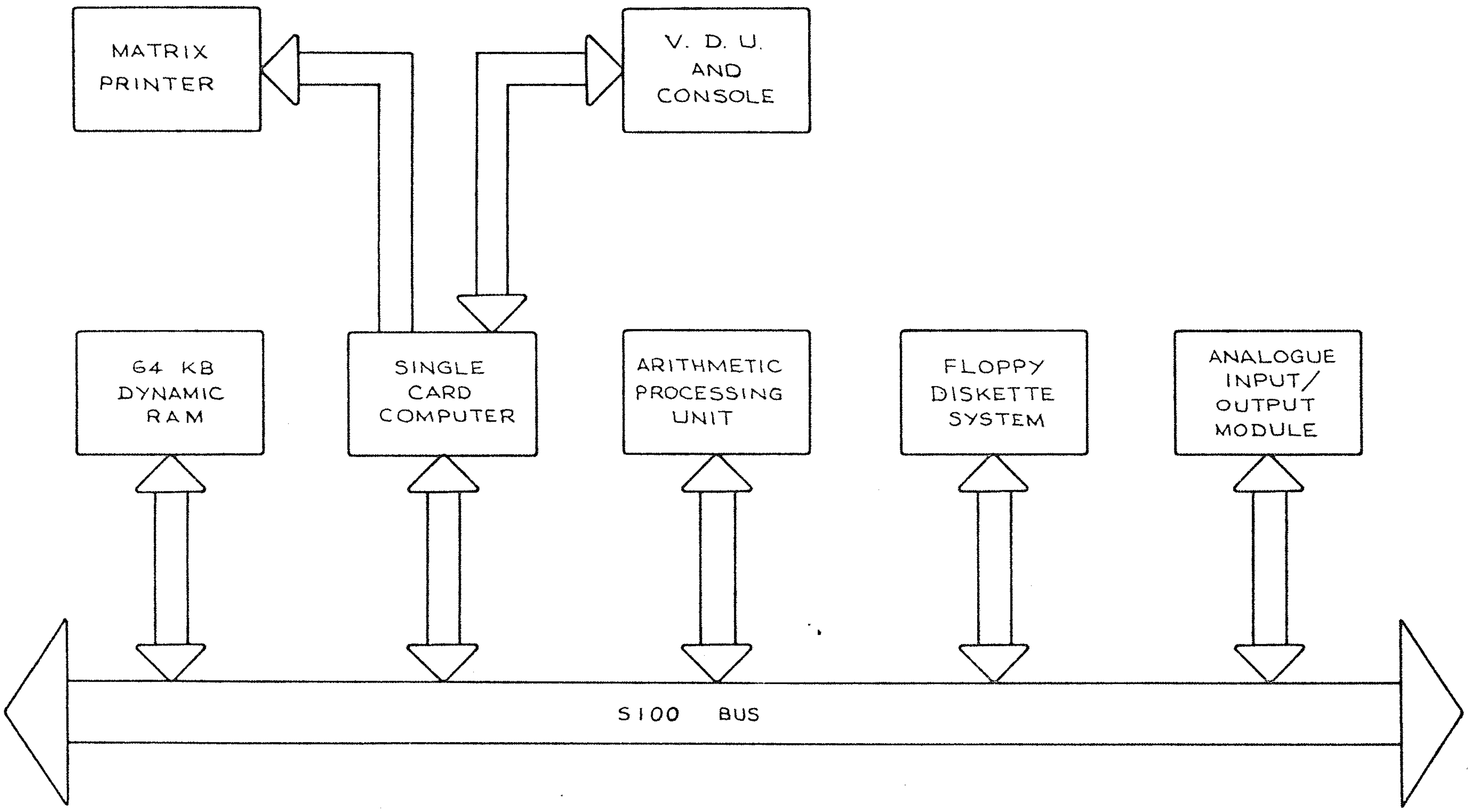


Figure 7.1 Block diagram of microprocessor system

7.3 Driving Unit

The driving unit has one motor simulating the novel high gain tracking motor. Another motor is used as the slew motor. Referring back to Chapter 1, the servomechanism is supposed to be put on board a ship, thus rolling causing the major disturbances. Hence the target and the ship rolling must be simulated in the rig. The ship rolling motion is simulated by placing the whole dual-drive servomechanism on a rocking platform driven by a motor. The general layout of the driving unit is as shown in Figure 7.2

The motors used in the demonstration rig are all permanent magnet printed motors. They are d.c. machines where the wound coils of the conventional cylindrical armatures have been replaced by layers of flat conductors arranged in the form of a non-magnetic disc. The conductors are punched or notched from copper sheets. The conductor in each layer are connected together to form one continuous winding consisting of two air cored coils in parallel. An axial magnetic field is produced by permanent magnet pairs arranged with alternate polarity around the armature. The main advantage of printed motors is the freedom from the effects of iron in the armature. Because of this, the torque produced by current flowing in the armature conductors is not limited by any saturation effects, nor is it affected by changes in load. The use of permanent magnets gives a linear torque-speed characteristic which is highly desirable in servomechanism applications.

The casing of the tracking motor is mounted onto the inertia disc of the slew motor. In order to provide a firm base for the tracking motor to react upon, special consideration was needed in the design of the slew motor control loop. As proposed, the slew motor will be isolated from the system most of the time preventing interference to the tracking motor. Since the slew motor has a lower bandwidth than the tracking motor all torques developed by the high gain motor may not be transmitted through the slew motor. External locking device must therefore be provided to clamp the shaft of the slew motor once it is in position. A failsafe brake was chosen for this purpose. The brake will be off only if a voltage is applied to the coil of the brake. When the slew motor is not active the brake is kept on by not applying any voltage to the brake.

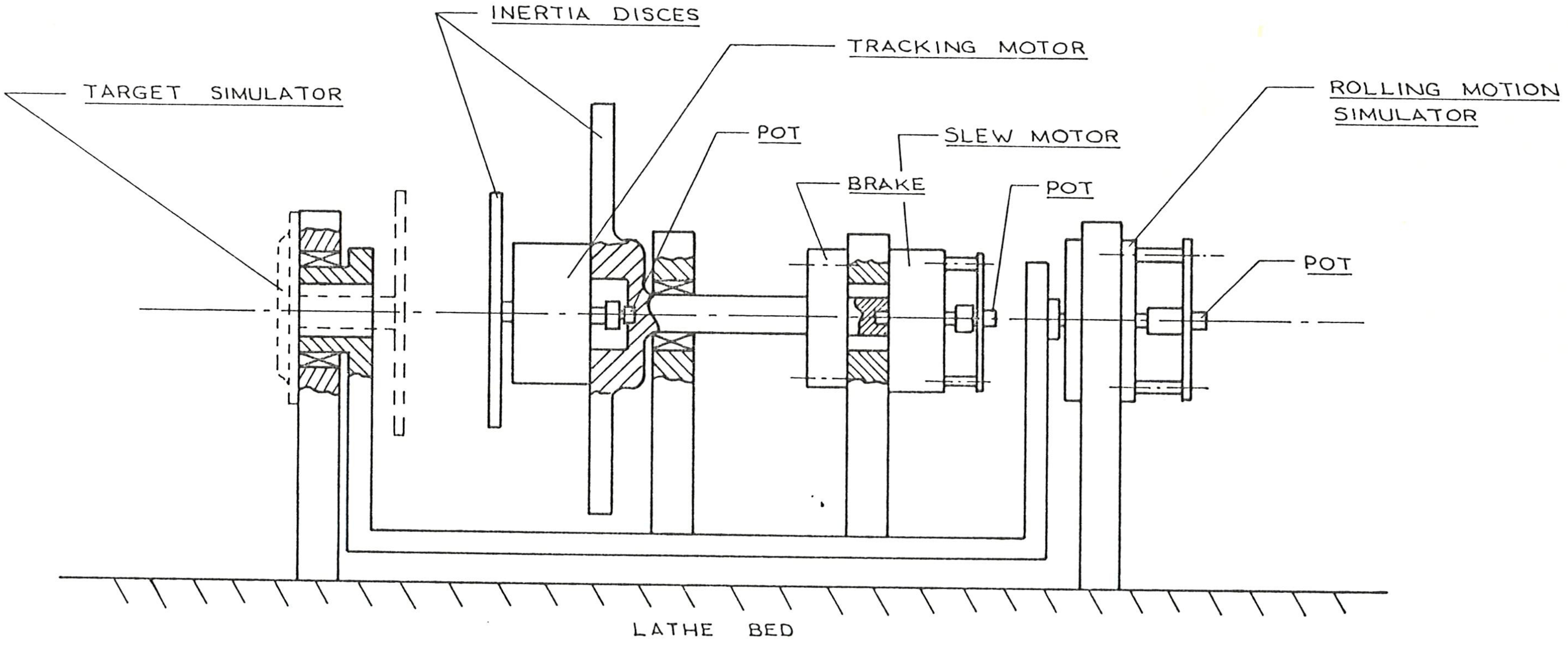


Figure 7.2 Layout of proposed demonstration rig

As shown in Figure 7.2, the motors are directly coupled to their loads such that any gear train is eliminated. In consequence, the non-linearity of back-lash inherited from imperfect gear teeth was not introduced into the rig and therefore stability of the system was not impaired.

All the motors used in the demonstration rig are controlled by their own servo control loop. Mechanically, the slew motor and the tracking motor have the same characteristics. Except the tracking motor has an integral tachometer. It enables the tracking motor control loop to have a higher bandwidth than the slew motor. Within the control loops, the angular positional sensors used are all hybrid rotary potentiometers, model number HRP 11/1E supplied by Penny and Giles. The potentiometers have an electrical angle of 340 degrees which is more than sufficient for the demonstration rig. The hybrid type potentiometers are superior than the wire-wound type, or the plastic type, potentiometers because of their virtually infinite resolution and linearity of ±0.25% In consequence, the signal-to-noise ratio is high. To complete the servo control loops, the EM100B servo amplifiers from McLennon were bought. In order to tune the control loops to the required bandwidth and performance, series compensation networks are used. Three three-term controllers were built for this purpose.

7.4 Discussions

The demonstration rig erected has the proposed dual-drive tracking servomechanism simulated using two printed circuit motors. The system is placed on a rocking platform. The ship rolling motion is to be simulated by feeding signals to a servomechanism which drives the rocking platform. The signals that are being fed to the rocking platform servomechanism may be a record of a typical ship rolling motion time history made on a magnetic tape. Since the time on the demonstration rig is scaled the magnetic tape must be played back at a scaled down speed. At present, there is nothing on the demonstration rig that simulates the target. It was first planned to feed the target position signal directly to the monitor through one of the analogue input ports using a magnetic tape recorder. A pre-determined target trajectory is recorded on the magnetic tape. However, the demonstration rig has been designed that the target may eventually be simulated by a light source driven by a fourth motor as indicated in Figure 7.2. A light sensor array may then be attached onto the inertia disc of the tracking motor to pick up the error signals between the tracking motor and the target.

It was regretted that it was not possible to do any simulation work on the rig due to the expiry of the contract. Hopefully, this work will be carried on in the future. If simulation is going to be performed on the rig all the signals communicating between the driving unit and the monitor must be within the range of 2.5 volts. This is the working limit imposed by the hardware of the analogue input/output device. The performance of the system may be assessed by the error signals obtained from the light sensor array. The simulation may be made more realistic by putting physical stoppers on the tracking motor to restrict its working range.

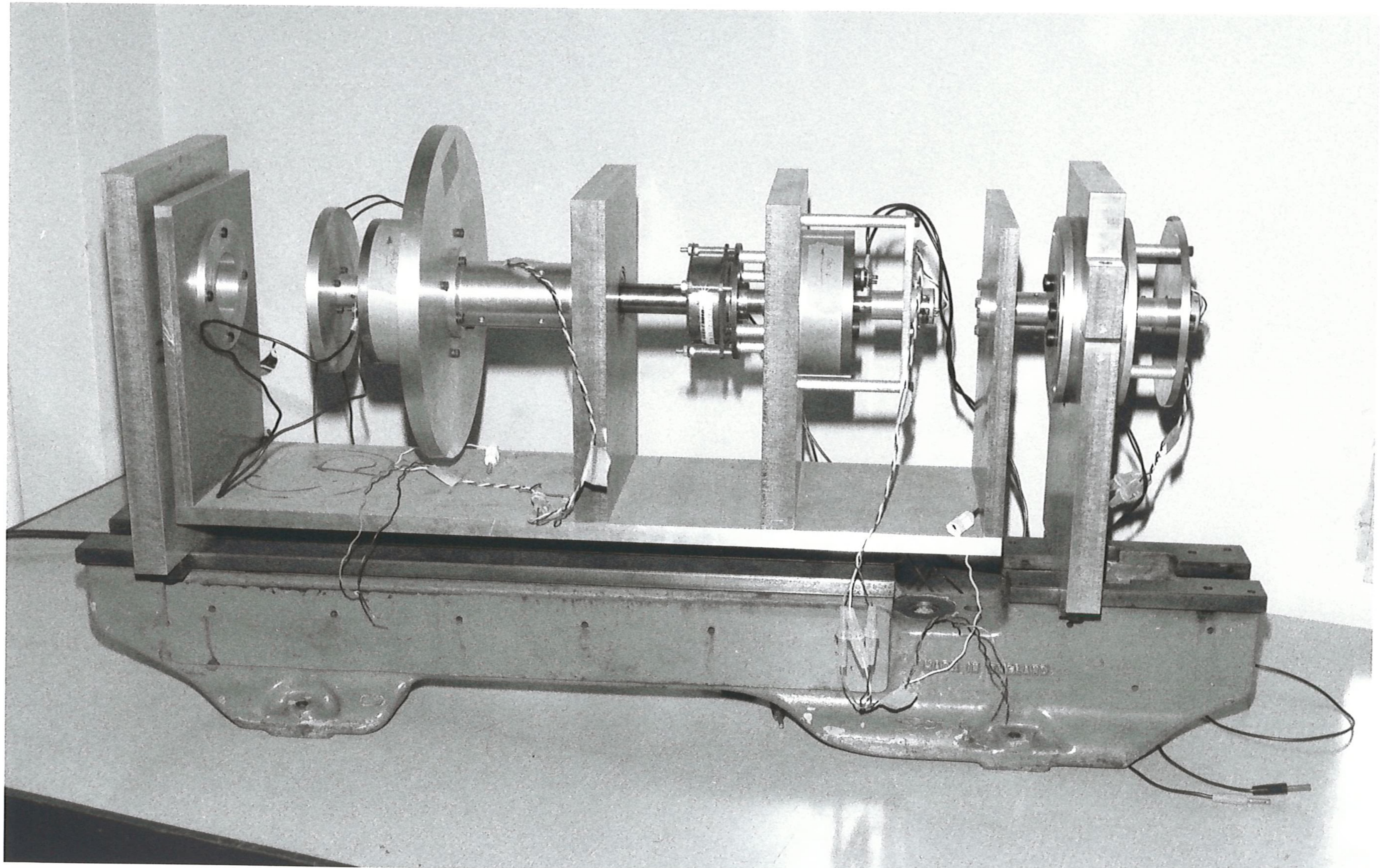


PLATE 7.1 Demonstration Rig

CHAPTER 8

GENERAL CONCLUSIONS

It is generally true that higher accuracy and faster response in servomechanisms can only be achieved by having higher system gain. Research efforts had been devoted to the design of a novel prime mover that gives high torque characteristics. Such a novel drive has been proposed. However, the proposed drive only generates a tremendous torque within a limited rotation range. A novel drive configuration was then further proposed to be applied in tracking systems. One of the possible solutions to solve the non-linearity of the drive is to employ the newly developed drive and a conventional drive in a piggy-back fashion. The former is used as a tracking motor while the latter is used as a slew motor. Such configurations then demand a new control strategy to reduce the coupling effect between the two drives.

The whole philosophy behind the proposed control law in this thesis is to keep the interference from the slew motor to a minimum. A microprocessor-based monitor models the absolute target positions and the ship rolling motion. The trends of the two signals are then predicted using the models built into its memory. The parameters of the models are being updated constantly. A decision is then made whether realignment is necessary to reduce the chance of losing its target in the future. The time and position of the realignment are evaluated by optimising a related cost function.

The whole complex tracking problem was solved by separating the original problem into several smaller, easier and independent problems. Each individual problem was then solved accordingly. The proposed control law then involved the signal modelling, the model parameter estimation, the prediction and the decision making. The implementation of the proposed control strategy involves repeating the model parameter estimation, prediction and decision making continually at intervals. The intervals are physically constrained by the computing speed of the monitor. In real-time applications, the sampling periods can only be equal to, or longer than, these intervals.

The working environment of the proposed tracking servomechanism had been analysed in Chapter 1. It was assumed that the target motion and ship rolling motion corrupted with noise will be the dominant signals. A constant acceleration model, or equivalently a quadratic polynomial, was chosen in modelling the absolute target trajectory. In essence, the target trajectory is being fitted by a quadratic polynomial function. Such a crude model was proposed because the tracking servomechanism is supposed to track all types of targets. A priori information on types of targets is out of question. The only thing that can be certain is the trajectory must be continuous. Modelling the target motion will, however, be appropriate only to a finite time interval of the motion. The ship rolling motion, on the other hand, was proposed to be modelled as an autoregressive stochastic process. This was a subjective proposal. The only knowledge on the ship rolling available to the author is its approximate spectrum. The autoregressive model is also restricted to a finite length of the rolling because the ship rolling is not a truely stationary process in nature. Stationarity can be assumed to a segment of the infinite time series.

It is preferable to employ recursive parameter estimation algorithms rather than batch processing since the estimation is performed on-line and a recursive algorithm is more efficient. Time is a crucial factor. The well-known discrete Kalman filter algorithm is very attractive. In view of the finite nature of the deterministic and stochastic models, the parameters should only be estimated from a finite number of past measurements. Measurements that are too old must be dropped out from the estimates completely. A finite memory version of the algorithm has been derived. The finite memory Kalman filter can be visualised as the estimates obtained from a length of data inside a window which moves along with time.

The finite memory version of the Kalman algorithm actually has potential applications in a broad area. For instance, only a fixed amount of data is available in identifying a plant. If canonical Kalman algorithm is used the first few data points will not be used effectively due to the transient of the filter itself, and also the initial guess of the parameter vector. On the other hand, the finite memory version may re-use the data again to give better estimates. It effectively reduces the error arising from wrong initial guess. The whole finite memory Kalman filter algorithm preserves the basic form of the canonical Kalman filter algorithm. The difference is the introduction of the old measurement to be discarded and the observation matrix at the time of the old measurement, etc. The finite memory version may be performed as a canonical Kalman filter by replacing certain elements of the matrices with zeros. The finite memory algorithm cannot drop out an old data without putting on one new measurement at the same time. Another slightly different algorithm is required for discarding one measurement only. The finite memory Kalman filter algorithm, however, suffers a deficiency: The transition matrix of the system has to be projected backwards to the time that the measurement is being dropped. This may be found to be computationally difficult in some cases, especially in time-variant systems. The algorithm also is required to remember the measurement, and the observation matrices at the times the measurements were made. The algorithm is best used incorporated with systems having an identity matrix as the transition matrix, and with constant observation matrices.

The finite memory Kalman filter has been actually tried out as a parameter estimator. The results were listed in Chapter 3. This showed that by manipulating the window length, unanticipated changes in parameters can be easily accommodated. Short window length allowed fast response to changes. However, too short the window length resulted with high variances. These all agreed with what were expected. The performance of the finite memory Kalman filter may be improved further by adopting an adaptive window. A long window may be used to give low variance on the estimates when no unanticipated change is detected. Once a sudden change in the parameters occurs, the window length may then be reduced to give a faster response to the sudden changes in the parameters.

The canonical Kalman filter in effect has an infinite long window. Therefore, the finite memory Kalman filter gives a faster response than the canonical Kalman filter, when adapting to changes if both use the same system model. However, the canonical Kalman filter can have a faster response to changes if plant noise is introduced in the system model. The characteristic of the derived finite memory Kalman filter is very similar to the canonical Kalman filter working with plant noise. Both algorithms restrict the gain matrix becoming zero, thus the estimates may never be isolated from the measurements. The finite memory Kalman filter without plant noise is better than the canonical Kalman filter in the sense that the window length is easier to understand, than the variance of the plant noise. Furthermore, measurements may be removed from the estimates at will. The ignorance of the correlation between estimation errors and past plant noise prevents the derivation of a finite memory Kalman filter algorithm incorporated with plant noise in the system model.

The cost function originally suggested was to offset the weighted probability of losing the target against the weighted realignments. Both weights were initially proposed to be time variant. The numerical simulation results indicated that the time variant weighting had no significant effect on the performance for the chosen parameters. The results suggested that an uniform weighting is sufficient. This simplifies the problem further by not requiring the knowledge of the duration of an engagement with a target. The important factor is however the ratio between the two weights. Too low a ratio resulted in frequent realignments. On the other hand, the tracker lost its target more often when the ratio was too high. The percentages of misses and realignments both have an exponential relationship with the ratio. The cross-over point of the percentages of misses and realignments occurred between the weight ratio of two to three. Within this range, neither the realignments nor the misses would be excessive. The numerical simulation results also justified the assumption of one realignment, at most, within the prediction period in finding a numerical optimum of the cost function. The above two findings enabled a simpler, and quicker, algorithm to determine the optimal solution of the cost function. The algorithm can then make use of the fact that the amount of realignment is twice the difference between the predicted target position and the slew motor position at the time of realignment. The only task is then to find the time of realignment The determination of the optimal predicted slew motor trajectory becomes a one-variable optimisation problem. The number of variables in the original optimisation problem is twice the number of the time intervals inside the prediction period if the problem is treated as a multi-stage decision process. Otherwise, the problem has unknown numbers of variables if it is treated as a straight forward optimisation problem.

The simulation also revealed that among the chosen ship rolling and target models the dominant motion is the ship rolling. The frequent realignment of the slew motor was mainly due to the fluctuation in the ship motion. The simulated ship rolling motion did not truely reflect the characteristic of the real motion. Occasionally, the fluctuations were too violent. Nevertheless, a sixth order autoregressive model was sufficient to model the ship rolling motion. It is envisaged that the model will perform much better if real ship rolling motion signals were used because of the absence of the violent fluctuations.

In view of the simulation results, the proposed control strategy cannot totally eliminate the possibility of losing its target during an engagement. However, this is the limitation of using the novel high torque, non-linear drive. This is also due to the inevitable noise in the measurements. There may be ways to reduce the possibility further. The microprocessor-based monitor may be used to override the control strategy when the target is going outside the working range of the tracking motor.

Initially, it was planned to validate the proposed control strategy on a demonstration rig. The rig was commissioned and has been erected for action. It is unfortunate that there was no time to implement the control law completely to the rig.

SUGGESTIONS FOR FURTHER WORK

The proposed control strategy for the novel dual drive tracking servomechanism presented in this thesis is not fully developed. It is only showed that the control strategy is feasible. Several areas of work require further development. During the initial development of the control strategy, some interesting points arose and required investigation.

As the simulation results suggested that one realignment is sufficient in the optimisation process. Thus, the optimisation algorithm requires modifications. It can be done more quickly.

The validation of the eventually developed control strategy should be performed on the existing demonstration rig.

In the work of modelling, the effect of the ship rolling motion requires further studies. Simulations using real ship rolling motion signals are essential before the proposed tracking system put to service. The suggestion of a sixth order autoregressive model for the ship motion is based on the simulated ship motion. This must be validated with true motion signals.

The Kalman filter is generally used with a system without any constrain on its states. In real-life, every system has its own limitation. Therefore, it is interesting to see the Kalman filter in association with constraints. This combination should increase the accuracy and the confidence in estimating the states of the process.

As already mentioned in Chapter 3 and Appendix B, the finite memory version of the Kalman filter cannot work with a system model having plant noise. It may be useful if this restriction can be released.

APPENDIX A

MATRIX INVERSION LEMMA

Let S and R be positive definite matrices not necessarily of the same order. Let M be a matrix such that $M^T R^{-1} M$ is of the same order as S.

Then: $$(S^{-1} + M^T R^{-1} M)^{-1} = S - SM^T (R + MSM^T)^{-1} MS \qquad \text{(A1)}$$

Proof:

$$(S^{-1} + M^T R^{-1} M) \left[S - SM^T (R + MSM^T)^{-1} MS\right]$$

$$= S^{-1} \left[S - SM^T (R + MSM^T)^{-1} MS \right]$$

$$+ M^T R^{-1} M \left[S - SM^T (R + MSM^T)^{-1} MS \right]$$

$$= S^{-1} S - S^{-1} SM^T (R + MSM^T)^{-1} MS + M^T R^{-1} MS$$

$$- M^T R^{-1} MSM^T (R + MSM^T)^{-1} MS$$

$$= I - M^T (R + MSM^T)^{-1} MS + M^T R^{-1} MS$$

$$- M^T R^{-1} MSM^T (R + MSM^T)^{-1} MS$$

$$= I + M^T R^{-1} MS - M^T (R + MSM^T)^{-1} MS$$

$$- M^T R^{-1} MSM^T (R + MSM^T)^{-1} MS$$

$$= I + M^T R^{-1} MS - M^T R^{-1} R (R + MSM^T)^{-1} MS$$

$$- M^T R^{-1} MSM^T (R + MSM^T)^{-1} MS$$

$$= I + M^T R^{-1} MS - M^T R^{-1} (R + MSM^T)(R + MSM^T)^{-1} MS$$

$$= I + M^T R^{-1} MS - M^T R^{-1} MS$$

$$= I \qquad \text{(Q.E.D.)}$$

Corollary:

$$(S^{-1} + M^T R^{-1} M)^{-1} M^T R^{-1} = SM^T (R + MSM^T)^{-1} \qquad \text{(A2)}$$

Proof:

$$(S^{-1} + M^T R^{-1} M)^{-1} = S - SM^T (R + MSM^T)^{-1} MS$$

$$M(S^{-1} + M^T R^{-1} M)^{-1} M^T R^{-1} = MSM^T R^{-1} - MSM^T (R + MSM^T)^{-1} MSM^T R^{-1}$$

Premultiplied by $(MSM^T)^{-1}$

$$(MSM^T)^{-1} M (S^{-1} + M^T R^{-1} M)^{-1} M^T R^{-1}$$

$$= R^{-1} - (R + MSM^T)^{-1} MSM^T R^{-1}$$

Premultiplied by $(R + MSM^T)$

$$(R + MSM^T)(MSM^T)^{-1} M (S^{-1} + M^T R^{-1} M)^{-1} M^T R^{-1}$$

$$= (R + MSM^T) R^{-1} - MSM^T R^{-1}$$

$$(R + MSM^T)(MSM^T)^{-1} M(S^{-1} + M^T R^{-1} M)^{-1} M^T R^{-1} = I$$

$$M(S^{-1} + M^T R^{-1} M)^{-1} M^T R^{-1} = MSM^T (R + MSM^T)^{-1}$$

$$(S^{-1} + M^T R^{-1} M)^{-1} M^T R^{-1} = SM^T (R + MSM^T)^{-1} \qquad \text{(Q.E.D.)}$$

APPENDIX B

KALMAN FILTERING

B.1 Model for Random Processes

Given a dynamic model of

$$\underline{x}(k+1) = A(k+1,k)\ \underline{x}(k) + B(k)\ \underline{u}(k) \tag{B1}$$

and the measurement equation is

$$\underline{z}(k+1) = D(k+1)\ \underline{x}(k+1) + \underline{v}(k+1) \tag{B2}$$

where $\underline{x}(.)$ is a nxl state vector, $\underline{u}(.)$ is a pxl input vector, $\underline{z}(.)$ is a mxl measurement vector and $\underline{v}(.)$ is a mxl measurement noise vector. A(k+l,k) is a nxn transition matrix, B(.) is a nxp control matrix and D(.) is a mxn observation matrix. $\underline{v}(.)$ is a zero-mean gaussian random variable. i.e.

$$E\left[\ \underline{v}(k)\right] = \underline{0} \tag{B3}$$

and

$$E\left[\ \underline{v}(k)\ \underline{v}^T(j)\ \right] = R(k)\ \delta_{jk} \tag{B4}$$

where

$$\delta_{jk} = \begin{cases} 0 & ; \quad j \neq k \\ 1 & ; \quad j = k \end{cases} \tag{B5}$$

The measurement noise, $\underline{v}(.)$ is assumed to be uncorrelated with the state vector, $\underline{x}(.)$

B.2 Inclusion of One New Measurement

Before the derivation, it is necessary to assume that an estimate of $\underline{x}(k)$, $\hat{\underline{x}}(k/k,\ell)$, and its covariance matrix, $P(k/k,\ell)$, are available. The notation of $\hat{\underline{x}}(k/k,\ell)$ means an estimate of the state $\underline{x}$ at time interval k based on the measurements from time intervals $k-\ell$ to k inclusive.

B.2.1 Derivation from Linear Minimum Variance Approach.

Given the estimate $\hat{\underline{x}}(k/k,\ell)$ and its covariance matrix $P(k/k,\ell)$ the new estimate $\hat{\underline{x}}(k+1/k+1,\ell+1)$ with one new measurement added is assumed to be:

$$\hat{\underline{x}}(k+1/k+1,\ell+1) = \hat{\underline{x}}(k+1/k,\ell) + K(k+1) \;\tilde{\underline{z}}(k+1/k,\ell) \qquad (B6)$$

where

$\tilde{\underline{z}}(k+1/k,\ell)$ is the innovation

$$\tilde{\underline{z}}(k+1/k,\ell) = \underline{z}(k+1) - \hat{\underline{z}}(k+1/k,\ell) \qquad (B7)$$

The estimate $\hat{\underline{x}}(k+1/k+1,\ell+1)$ must satisfy the criterion that the estimation error variance is minimum, i.e. the cost of minimisation is:

$$J = P(k+1/k+1,\ell+1)$$

$$= E\left[\tilde{\underline{x}}(k+1/k+1,\ell+1) \;\tilde{\underline{x}}^T(k+1/k+1,\ell+1)\right] \qquad (B8)$$

Since the best estimate of $\underline{z}(k+1)$ is

$$\hat{\underline{z}}(k+1/k,\ell) = D(k+1) \;\hat{\underline{x}}(k+1/k,\ell) \qquad (B9)$$

(B7) becomes:

$$\tilde{\underline{z}}(k+1/k,\ell) = D(k+1) \;\underline{x}(k+1) + \underline{v}(k+1) - D(k+1) \;\hat{\underline{x}}(k+1/k,\ell)$$

$$= D(k+1) \;\tilde{\underline{x}}(k+1/k,\ell) + \underline{v}(k+1) \qquad (B10)$$

Therefore:

$$\tilde{\underline{x}}(k+1/k+1,\ell+1) = \underline{x}(k+1) - \hat{\underline{x}}(k+1/k+1,\ell+1)$$

$$= \underline{x}(k+1) - \hat{\underline{x}}(k+1/k,\ell) - K(k+1) \quad \tilde{\underline{z}}(k+1/k,\ell)$$

$$= \tilde{\underline{x}}(k+1/k,\ell) - K(k+1)\ D(k+1)\ \tilde{\underline{x}}(k+1/k,\ell) - K(k+1)\ \underline{v}(k+1)$$

$$= \left[\ I - K(k+1)\ D(k+1)\ \right] \tilde{\underline{x}}(k+1/k,\ell) - K(k+1)\ \underline{v}(k+1)$$

(B11)

Hence, with the fact that $\underline{x}(k+1/k,\ell)$ and $\underline{v}(k+1)$ are uncorrelated

$$P(k+1/k+1,\ell+1) = E\left[\ \tilde{\underline{x}}(k+1/k+1,\ell+1) \quad \tilde{\underline{x}}^T(k+1/k+1,\ell+1)\right]$$

$$= \left[I - K(k+1)\ D(k+1)\right] P(k+1/k,\ell) \left[I - K(k+1)\ D(k+1)\right]^T$$

$$+ \ K(k+1)\ R(k+1)\ K^T(k+1) \qquad (B12)$$

where

$$P(k+1/k,\ell) = A(k+1,k)\ P(k/k,\ell)\ A^T(k+1,k) \qquad (B13)$$

expanding (B12)

$$P(k+1/k+1,\ell+1) = P(k+1/k,\ell) - K(k+1)\ D(k+1)\ P(k+1/k,\ell)$$

$$- \ P(k+1/k,\ell) \quad D^T(k+1)\ K^T(k+1)$$

$$+ \ K(k+1)\left[\ D(k+1)\ P(k+1/k,\ell)\ D^T(k+1) + R(k+1)\right] K^T(k+1)$$

(B14)

Completing the square gives:

$$P(k+1/k+1,\ell+1) =$$

$$\left[K(k+1) - V\right] \left[D(k+1)\ P(k+1/k,\ell)\ D^T(k+1) + R(k+1)\right] \left[K(k+1) - V\right]^T$$

$$- \ V\left[\ D(k+1)\ P(k+1/k,\ell)\ D^T(k+1) + R(k+1)\right]V^T + P(k+1/k,\ell) \qquad (B15)$$

where V must satisfy the expression:

$$V \left[D(k+1)\ P(k+1/k,\ell)\ D^T(k+1) + R(k+1)\right] = P(k+1/k,\ell)\ D^T(k+1) \tag{B16}$$

From (B15), it will be minimum if K(k+1) = V and hence:

$$K(k+1) = P(k+1/k,\ell)\ D^T(k+1)\ \left[D(k+1)\ P(k+1/k,\ell)\ D^T(k+1) + R(k+1)\right]^{-1} \tag{B17}$$

and

$$P(k+1/k+1,\ell+1) = \left[I - K(k+1)\ D^T(k+1)\right]\ P(k+1/k,\ell) \tag{B18}$$

Together with the fact that:

$$\hat{\underline{x}}(k+1/k,\ell) = A(k+1,k)\ \hat{\underline{x}}(k/k,\ell) + B(k)\ \underline{u}(k) \tag{B19}$$

(B6), (B13), (B17), (B18) and (B19) form a complte algorithm.

B.2.2 Derivation from weighted Least Squares Approach

From the model (B1)

$$\underline{x}(k+1) = A(k+1,k)\ \underline{x}(k) + B(k)\ \underline{u}(k)$$

after rearranging gives

$$\underline{x}(k) = A(k,k+1)\ \underline{x}(k+1) - A(k,k+1)\ B(k)\ \underline{u}(k) \tag{B20}$$

Extending the time interval backward to $k-\ell$

$$\underline{x}(k-\ell) = A(k-\ell,k+1)\ \underline{x}(k+1) - \beta(k-\ell,k+1) \tag{B21}$$

where

$$\beta(k-\ell,k+1) = \sum_{j=k-\ell}^{k} A(k-\ell,j+1)\ B(j)\ \underline{u}(j) \tag{B22}$$

From the measurement equation (B2)

$$\underline{z}(k-\ell) = D(k-\ell)\ A(k-\ell,k)\ \underline{x}(k) - D(k-\ell)\ \beta(k-\ell,k) + \underline{v}(k-\ell) \tag{B23}$$

Consider the measurements from interval k-ℓ to k inclusive, let

$$Z(k) = \begin{pmatrix} \underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k) \\ \underline{z}(k-\ell+1) + D(k-\ell+1)\ \beta(k-\ell+1,k) \\ \cdot \\ \cdot \\ \cdot \\ \underline{z}(k-1) + D(k-1)\ \beta(k-1,k) \\ \underline{z}(k) \end{pmatrix}$$

$$N(k) = \begin{pmatrix} D(k-\ell)\ A(k-\ell,k) \\ D(k-\ell+1)\ A(k-\ell+1,k) \\ \cdot \\ \cdot \\ \cdot \\ D(k-1)\ A(k-1,k) \\ D(k) \end{pmatrix}$$

$$V(k) = \begin{pmatrix} \underline{v}(k-\ell) \\ \underline{v}(k-\ell+1) \\ \cdot \\ \cdot \\ \cdot \\ \underline{v}(k-1) \\ \underline{v}(k) \end{pmatrix}$$

and

$$Q(k) = \begin{pmatrix} R(k-\ell) & & & & & \\ & R(k-\ell+1) & & & 0 & \\ & & \cdot & & & \\ & & & \cdot & & \\ & 0 & & & \cdot & \\ & & & & R(k-1) & \\ & & & & & R(k) \end{pmatrix}$$

Then

$$Z(k) \;=\; N(k)\;\underline{x}(k) \;+\; V(k) \tag{B24}$$

In the case of weighted Least Squares estimation the cost function is:

$$J \;=\; \left[Z(k) - N(k)\;\hat{\underline{x}}(k/k,\ell)\right]^T Q^{-1}(k)\;\left[Z(k) - N(k)\;\hat{\underline{x}}(k/k,\ell)\right] \tag{B25}$$

The estimator that minimises the cost is [35]

$$\hat{\underline{x}}(k/k,\ell) \;=\; P(k/k,\ell)\;N^T(k)\;\;Q^{-1}(k)\;\;Z(k) \tag{B26}$$

and

$$P(k/k,\ell) \;=\; \left[N^T(k)\;\;Q^{-1}(k)\;\;N(k)\right]^{-1} \tag{B27}$$

When one new measurement is available, (B24) is augmented to:

$$\begin{pmatrix} Z(k) + N(k)\;A(k,k+1)\;B(k)\;\underline{u}(k) \\ z(k) \end{pmatrix} = \begin{pmatrix} N(k)\;A(k,k+1) \\ D(k+1) \end{pmatrix} \;\underline{x}(k+1) \;+\; \begin{pmatrix} V(k) \\ \underline{v}(k+1) \end{pmatrix} \tag{B28}$$

The weighted Least Squares estimate of $\underline{x}(k+1)$ is then:

$$\begin{aligned} &\hat{\underline{x}}(k+1/k+1,\ell+1) \\ &= P(k+1/k+1,\ell+1)\;\Bigg[\left(A^T(k,k+1)\;N^T(k)\qquad D^T(k+1)\right)\;\begin{pmatrix} Q^{-1}(k) & 0 \\ 0 & R^{-1}(k+1) \end{pmatrix} \\ &\qquad\qquad \times\;\begin{pmatrix} Z(k) + N(k)\;A(k,k+1)\;B(k)\;\underline{u}(k) \\ z(k+1) \end{pmatrix}\Bigg] \\ &= P(k+1/k+1,\ell+1)\;\left[A^T(k,k+1)\;N^T(k)\;Q^{-1}(k)\;Z(k)\right. \\ &+\; A^T(k,k+1)\;N^T(k)\;Q^{-1}(k)\;N(k)\;A(k,k+1)\;B(k)\;\underline{u}(k) \\ &+\; \left. D^T(k+1)\;\;R^{-1}(k+1)\;\;\underline{z}(k+1)\right] \end{aligned} \tag{B29}$$

where

$P(k+1/k+1,\ell+1)$

$$= \left[\left(A^T(k,k+1)\ N^T(k) \quad D^T(k+1) \right) \begin{pmatrix} Q^{-1}(k) & 0 \\ 0 & R^{-1}(k+1) \end{pmatrix} \begin{pmatrix} N(k) & A(k,k+1) \\ & D(k+1) \end{pmatrix} \right]^{-1}$$

$$= \left[A^T(k,k+1)\ N^T(k)\ Q^{-1}(k)\ N(k)\ A(k,k+1) \ + \ D^T(k+1)\ R^{-1}(k+1)\ D(k+1) \right]^{-1} \tag{B30}$$

Substitute (B26) and (B27) into (B29) gives

$$\hat{\underline{x}}(k+1/k+1,\ell+1)$$

$$= P(k+1/k+1,\ell+1)\ \left[A^T(k,k+1)\ P^{-1}(k/k,\ell)\ \hat{\underline{x}}(k/k,\ell)\right.$$

$$+ \ A^T(k,k+1)\ P^{-1}(k/k,\ell)\ A(k,k+1)\ B(k)\ \underline{u}(k)$$

$$\left. + \ D^T(k+1)\ \ R^{-1}(k+1)\ \ \underline{z}(k+1)\ \right]$$

$$= P(k+1/k+1,\ell+1)\ \left[A^T(k,k+1)\ P^{-1}(k/k,\ell)\ A(k,k+1)(A(k+1,k)\ \hat{\underline{x}}(k/k,\ell)\right.$$

$$+ \ B(k)\ \underline{u}(k))$$

$$\left. + \ D^T(k+1)\ R^{-1}(k+1)\,\underline{z}(k+1)\ \right]$$

$$= P(k+1/k+1,\ell+1)\ \left[P^{-1}(k+1/k,\ell)\ \hat{\underline{x}}(k+1/k,\ell) \ + \ D^T(k+1)\,R^{-1}(k+1)\ \underline{z}(k+1)\right] \tag{B31}$$

Substitute (B27) into (B30) gives:

$$P(k+1/k+1,\ell+1)$$

$$= \left[A^T(k,k+1)\ P^{-1}(k/k,\ell)\ A(k,k+1) + D^T(k+1)\ R^{-1}(k+1)\ D(k+1)\right]^{-1}$$

$$= \left[P^{-1}(k+1/k,\ell) \ + \ D^T(k+1)\ R^{-1}(k+1)\ D(k+1)\right]^{-1} \tag{B32}$$

Applying the Inversion lemma, Appendix A, to (B32):

$$P(k+1/k+1,\ell+1) \ = \ P(k+1/k,\ell)$$

$$-P(k+1/k,\ell)\ D^T(k+1)\left[\ D(k+1)\ P(k+1/k,\ell)\ D^T(k+1)\ + R(k+1)\right]^{-1}$$

$$x\ D(k+1)\ P(k+1/k,\ell) \tag{B33}$$

Let $K(k+1) \;=\; P(k+1/k+1,\ell+1)\; D^T(k+1)\; R^{-1}(k+1)$ (B34)

(B31) becomes:

$$\underline{\hat{x}}(k+1/k+1,\ell+1) \;=\; P(k+1/k+1,\ell+1)\; P^{-1}(k+1/k,\ell)\; \underline{\hat{x}}(k+1/k,\ell) + K(k+1)\; \underline{z}(k+1)$$

$$= \left[I - K(k+1)\; D(k+1)\right]\; \underline{\hat{x}}(k+1/k,\ell) + K(k+1)\; \underline{z}(k+1) \tag{B35}$$

From the Inversion lemma again, it can be shown that:

$$\left[P^{-1}(k+1/k,\ell) + D^T(k+1)\; R^{-1}(k+1)\; D(k+1)\right]^{-1} D^T(k+1)\; R^{-1}(k+1)$$

$$= P(k+1/k,\ell)\; D^T(k+1)\; \left[D(k+1)\; P(k+1/k,\ell)\; D^T(k+1) + R(k+1)\right]^{-1} \tag{B36}$$

Hence;

$$K(k+1) = P(k+1/k,\ell)\; D^T(k+1)\; \left[D(k+1\; P(k+1/k,\ell)\; D^T(k+1) + R(k+1)\right]^{-1} \tag{B37}$$

and

$$P(k+1/k+1,\ell+1) \;=\; \left[I - K(k+1)\; D(k+1)\right]\; P(k+1/k,\ell) \tag{B38}$$

Therefore, (B6), (B13), (B19), (B37) and (B38) form the complete algorithm which corresponds to the algorithm in the last section.

B.2.3 Summary of the Algorithm for Adding One New Measurement

$$P(k+1/k,\ell) \;=\; A(k+1,k)\; P(k/k,\ell)\; A^T(k+1,k) \tag{B39}$$

$$\underline{\hat{x}}(k+1/k,\ell) \;=\; A(k+1,k)\; \underline{\hat{x}}(k/k,\ell) + B(k)\; \underline{u}(k) \tag{B40}$$

$$K(k+1) \;=\; P(k+1/k,\ell)\; D^T(k+1)\; \left[D(k+1)\; P(k+1/k,\ell)\; D^T(k+1) + R(k+1)\right]^{-1} \tag{B41}$$

$$\underline{\hat{x}}(k+1/k+1,\ell+1) \;=\; \left[I - K(k+1)\; D(k+1)\right]\; \underline{\hat{x}}(k+1/k,\ell) + K(k+1)\; \underline{z}(k+1) \tag{B42}$$

$$P(k+1) \;=\; \left[I - K(k+1)\; D(k+1)\right]\; P(k+1/k,\ell) \tag{B43}$$

B.3 Dropping of One Old Measurement

It is assumed that an estimate of $\underline{x}(k)$, $\hat{\underline{x}}(k/k,\ell)$, and its covariance matrix, $P(k/k,\ell)$, are available. The objective is to develop the algorithm for the removal of the data at time $k-\ell$ from the estimate.

B.3.1 Derivation from Linear Minimum Variance Approach

Given the estimate $\hat{\underline{x}}(k/k,\ell)$ and its covariance matrix $P(k/k,\ell)$, the new estimate $\hat{\underline{x}}(k/k,\ell-1)$ with one old measurement at time $k-\ell$ removed is assumed to be:

$$\hat{\underline{x}}(k/k,\ell-1) = \hat{\underline{x}}(k/k,\ell) - K(k-\ell)\ \tilde{z}(k-\ell/k,\ell) \qquad \text{(B44)}$$

where

$$\tilde{\underline{z}}(k-\ell/k,\ell) = \underline{z}(k-\ell) - \hat{\underline{z}}(k-\ell/k,\ell) \qquad \text{(B45)}$$

The estimate $\hat{\underline{x}}(k/k,\ell-1)$ must satisfy the criterion that the estimation error variance is minimum, i.e. the cost for minimisation is:

$$J = P(k/k,\ell-1)$$

$$= E\left[\tilde{\underline{x}}(k/k,\ell-1)\ \tilde{\underline{x}}^T(k/k,\ell-1)\ \right] \qquad \text{(B46)}$$

Substitute (B2) and (B21) into (B45):

$$\tilde{\underline{z}}(k-\ell/k,\ell) = \underline{z}(k-\ell) - \hat{\underline{z}}(k-\ell/k,\ell)$$

$$= D(k-\ell)\ \underline{x}(k-\ell) + \underline{v}(k-\ell) - D(k-\ell)\ \hat{\underline{x}}(k-\ell/k,\ell)$$

$$= D(k-\ell)\ A(k-\ell,k)\ \underline{x}(k) - D(k-\ell)\ \beta(k-\ell,k)$$

$$+ \underline{v}(k-\ell) - D(k-\ell)\ A(k-\ell,k)\ \hat{\underline{x}}(k/k,\ell)$$

$$+ D(k-\ell)\ \beta(k-\ell,k)$$

$$= D(k-\ell)\ A(k-\ell,k)\ \tilde{\underline{x}}(k/k,\ell) + \underline{v}(k-\ell) \qquad \text{(B47)}$$

Hence

$$\hat{\underline{x}}(k/k,\ell-1) \;=\; \hat{\underline{x}}(k/k,\ell) - K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\; \tilde{\underline{x}}(k/k,\ell)$$

$$-\; K(k-\ell)\; \underline{v}(k-\ell)$$

Or

$$\underline{x}(k) - \hat{\underline{x}}(k/k,\ell-1) \;=\; \underline{x}(k) - \hat{\underline{x}}(k/k,\ell) + K(k-\ell)\; \underline{v}(k-\ell)$$

$$+\; K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\; \tilde{\underline{x}}(k/k,\ell)$$

$$\tilde{\underline{x}}(k/k,\ell-1) \;=\; \left[\, I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]\; \tilde{\underline{x}}(k/k,\ell) + K(k-\ell)\; \underline{v}(k-\ell) \tag{B48}$$

Since the estimate $\hat{\underline{x}}(k/k,\ell)$ is based on the measurements from interval $k-\ell$ to k, the estimation error $\tilde{\underline{x}}(k/k,\ell)$ is correlated with the measurement noise $\underline{v}(k-\ell)$. However the estimate $\hat{\underline{x}}(k/k,\ell-1)$ does not involve the measurement $z(k-\ell)$ and $\tilde{\underline{x}}(k/k,\ell-1)$; therefore has no relation with $v(k-\ell)$. Thus:

$$E\left[\tilde{\underline{x}}(k/k,\ell-1)\; \tilde{\underline{x}}^T(k/k,\ell-1)\right] \;+\; E\left[K(k-\ell)\; \underline{v}(k-\ell)\; \underline{v}^T(k-\ell)\; K^T(k-\ell)\right]$$

$$= \left[I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]\; P(k/k,\ell)\; \left[I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]^T$$

$$P(k/k,\ell-1) + K(k-\ell)\; R(k-\ell)\; K^T(k-\ell)$$

$$= \left[I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]\; P(k/k,\ell)\; \left[I + K(K-\ell)\; D(k-\ell)\; A(k-\ell,k)\;\right]^T$$

$$P(k/k,\ell-1)$$

$$= \left[I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]\; P(k/k,\ell)\; \left[I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]^T$$

$$-\; K(k-\ell)\; R(k-\ell)\; K^T(k-\ell) \tag{B49}$$

Completing the square gives:

$$P(k/k,\ell-1)$$

$$= \left[K(k-\ell) + V\right]\; \left[D(k-\ell)\; A(k-\ell,k)\; P(k/k,\ell)\; A^T(k-\ell,k)\; D^T(k-\ell) - R(k-\ell)\,\right]$$

$$\times\; \left[K(k-\ell) + V\right]^T$$

$$-V\ \left[D(k-\ell)\ A(k-\ell,k)\ P(k/k,\ell)\ A^T(k-\ell,k)\ D^T(k-\ell)\ -\ R(k-\ell)\right]\ V^T$$

$$+\ \ P(k/k,\ell) \qquad \text{(B50)}$$

where V must satisfy:

$$V\ \left[D(k-\ell)\ A(k-\ell,k)\ P(k/k,\ell)\ A^T(k-\ell,k)\ D^T(k-\ell)\ -\ R(k-\ell)\right]$$

$$=\ \ P(k/k,\ell)\ A^T(k-\ell,k)\ D^T(k-\ell) \qquad \text{(B51)}$$

From (B50) then, it will be minimum if $K(k-\ell) = -V$. Thus:

$$K(k-\ell)\ \ =\ \ -P(k/k,\ell)\ A^T(k-\ell,k)\ D^T(k-\ell)$$

$$x\ \left[D(k-\ell)\ A(k-\ell,k)\ P(k/k,\ell)\ A^T(k-\ell,k)\ D^T(k-\ell)\ -\ R(k-\ell)\right]^{-1} \qquad \text{(B52)}$$

and

$$P(k/k,\ell-1)\ \ =\ \left[\ I\ +\ K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right]\ P(k/k,\ell) \qquad \text{(B53)}$$

and (B44) can be rewritten as:

$$\hat{\underline{x}}(k/k,\ell-1)\ \ =\ \left[\ I\ +\ K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right]\ \hat{\underline{x}}(k/k,\ell)$$

$$-\ \ K(k-\ell)\ (\underline{z}(k-\ell)\ +\ D(k-\ell)\ \beta(k-\ell,k)) \qquad \text{(B54)}$$

Thus (B52), (B53), and (B54), together with the system and measurement equations form a complete algorithm for removing one measurement at time $k-\ell$.

B.3.2 Derivation from weighted Least Squares Approach

With (B21) and (B22), let

$$Z(k)\ \ =\ \ \begin{pmatrix} \underline{z}(k-\ell+1)\ +\ D(k-\ell+1)\ \beta(k-\ell+1,k) \\ \underline{z}(k-\ell+2)\ +\ D(k-\ell+2)\ \beta(k-\ell+2,k) \\ \vdots \\ \underline{z}(k-1)\ +\ D(k-1)\ \beta(k-1,k) \\ \underline{z}(k) \end{pmatrix}$$

$$N(k) \quad = \quad \begin{pmatrix} D(k-\ell+1)\ A(k-\ell+1,k) \\ D(k-\ell+2)\ A(k-\ell+2,k) \\ \cdot \\ \cdot \\ \cdot \\ D(k-1)\ A(k-1,k) \\ D(k) \end{pmatrix}$$

$$V(k) \quad = \quad \begin{pmatrix} \underline{v}(k-\ell+1) \\ \underline{v}(k-\ell+2) \\ , \\ \cdot \\ \cdot \\ \underline{v}(k-1) \\ \underline{v}(k) \end{pmatrix}$$

and

$$Q(k) = \begin{pmatrix} R(k-\ell+1) & & & & & 0 \\ & R(k-\ell+2) & & & & \\ & & \cdot & & & \\ & 0 & & \cdot & & \\ & & & & \cdot & \\ & & & & R(k-1) & \\ & & & & & R(k) \end{pmatrix}$$

Then

$$Z(k) \quad = \quad N(k)\ \underline{x}(k)\ +\ V(k) \qquad \text{(B55)}$$

With the cost

$$J \quad = \quad \left[Z(k)\ -\ N(k)\ \hat{\underline{x}}(k/k,\ell-1)\right]^T\ Q^{-1}(k)\ \left[Z(k)\ -\ N(k)\ \hat{\underline{x}}(k/k,\ell-1)\right] \qquad \text{(B56)}$$

The weighted Least Squares estimator that minimises (B56) is [35]

$$\hat{\underline{x}}(k/k,\ell-1) = P(k/k,\ell-1)\ N^T(k)\ Q^{-1}(k)\ Z(k) \tag{B57}$$

where

$$P(k/k,\ell-1) = \left[\ N^T(k)\ Q^{-1}(k)\ N(k)\right]^{-1} \tag{B58}$$

Together with the old measurement at time interval k-ℓ

$$\begin{pmatrix} \underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k) \\ Z(k) \end{pmatrix} = \begin{pmatrix} D(k-\ell)\ A(k-\ell,k) \\ N(k) \end{pmatrix} \underline{x}(k) + \begin{pmatrix} \underline{v}(k-\ell) \\ V(k) \end{pmatrix} \tag{B59}$$

The corresponding weighted least squares estimate is:

$$\hat{\underline{x}}(k/k,\ell) = P(k/k,\ell) \left[(A^T(k-\ell,k)\ D^T(k-\ell) \qquad N^T(k)) \begin{pmatrix} R^{-1}(k-\ell) & 0 \\ 0 & Q^{-1}(k) \end{pmatrix} \right.$$

$$\left. \times \begin{pmatrix} \underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k) \\ Z(k) \end{pmatrix} \right]$$

$$= P(k/k,\ell)\ \left[A^T(k-\ell,k)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ (\underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k))\right.$$

$$\left. +\ N^T(k)\ Q^{-1}(k)\ Z(k)\right] \tag{B60}$$

and

$$P(k/k,\ell) = \left[(A^T(k-\ell,k)\ D^T(k-\ell) \quad N^T(k)) \begin{pmatrix} R^{-1}(k-\ell) & 0 \\ 0 & Q^{-1}(k) \end{pmatrix} \begin{pmatrix} D(k-\ell)\ A(k-\ell,k) \\ N(k) \end{pmatrix} \right]^{-1}$$

$$= \left[\ A^T(k-\ell,k)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right.$$

$$\left. +\ N^T(k)\ \ Q^{-1}(k)\ \ N(k)\ \right]^{-1} \tag{B61}$$

Substituting (B58) into (B61) gives:

$$P(k/k,\ell-1) \;=\; \left[P^{-1}(k/k,\ell) - A^{T}(k-\ell,k)\; D^{T}(k-\ell)\; R^{-1}(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]^{-1} \tag{B62}$$

Applying the Inversion Lemma in Appendix A to (B62):

$$\begin{aligned} P(k/k,\ell-1) \;=\; & P(k/k-\ell) \\ - \;& P(k/k,\ell)\; A^{T}(k-\ell,k)\; D^{T}(k-\ell)\; \left[D(k-\ell)\; A(k-\ell)\; P(k/k,\ell)\; A^{T}(k-\ell,k)\; D^{T}(k-\ell)\right. \\ - \;& \left. R(k-\ell)\right]^{-1}\; D(k-\ell)\; A(k-\ell,k)\; P(k/k,\ell) \end{aligned} \tag{B63}$$

Let

$$\begin{aligned} K(k-\ell) \;=\; & -P(k/k,\ell)\; A^{T}(k-\ell,k)\; D^{T}(k-\ell) \\ \times \;& \left[D(k-\ell)\; A(k-\ell,k)\; P(k/k,\ell)\; A^{T}(k-\ell,k)\; D^{T}(k-\ell) - R(k-\ell)\right]^{-1} \end{aligned} \tag{B64}$$

Thus

$$P(k/k,\ell-1) \;=\; \left[I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]\; P(k/k,\ell) \tag{B65}$$

Also from the Inversion Lemma

$$\begin{aligned} & \left[P^{-1}(k/k,\ell) - A^{T}(k-\ell,k)\; D^{T}(k-\ell)\; R^{-1}(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]^{-1} \\ & \times\; A^{T}(k-\ell,k)\; D^{T}(k-\ell)\; R^{-1}(k-\ell) \\ & = \; -P(k/k,\ell)\; A^{T}(k-\ell,k)\; D^{T}(k-\ell) \\ & \times\; \left[D(k-\ell)\; A(k-\ell,k)\; P(k/k,\ell)\; A^{T}(k-\ell,k)\; D^{T}(k-\ell) - R(k-\ell)\right]^{-1} \end{aligned} \tag{B66}$$

together with (B57)

$$\begin{aligned} \underline{\hat{x}}(k/k,\ell) \;=\; & P(k/k,\ell-1)\; \left[P^{-1}(k/k,\ell)\; \underline{\hat{x}}(k/k,\ell) - A^{T}(k-\ell,k)\; D^{T}(k-\ell)\right. \\ & \qquad \left. R^{-1}(k-\ell)\; (z(k-\ell) + D(k-\ell)\beta(k-\;,k))\right] \\ = \;& \left[I + K(k-\ell)\; D(k-\ell)\; A(k-\ell,k)\right]\; \underline{\hat{x}}(k/k,\ell) \\ - \;& K(k-\ell)\; (\underline{z}(k-\ell) + D(k-\ell)\; \beta(k-\ell,k)) \end{aligned} \tag{B67}$$

Therefore (B64), (B65), (B67) together with the system and measurement equations corresponds to the algorithm developed from the minimum variance approach.

B.3.3. Summary of Algorithm for Removing Old Measurement

$$K(k-\ell) = -P(k/k,\ell)\ A^T(k-\ell,k)\ D^T(k-\ell)$$

$$\times \left[D(k-\ell)\ A(k-\ell,k)\ P(k/k,\ell)\ A^T(k-\ell,k)\ D^T(k-\ell) - R(k-\ell)\right]^{-1} \quad (B68)$$

$$P(k/k,\ell-1) = \left[I + K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right] P(k/k,\ell) \quad (B69)$$

$$\hat{\underline{x}}(k/k,\ell-1) = \left[I + K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right] \hat{\underline{x}}(k/k,\ell)$$

$$-\ K(k-\ell)\ (\underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k)) \quad (B70)$$

B.4 Moving Window

It is proposed to use a data window of $\ell+1$ measurements. The estimate of $\underline{x}(k+1)$, $\hat{\underline{x}}(k+1/k+1,\ell)$, is based on the measurements from time interval $k-\ell+1$ upto $k+1$ inclusive. When one new measurement is available, the estimate of the state is up-dated, but at the same time the oldest measurement is dropped.

B.4.1 Derivation from Linear Minimum Variance Approach

For the estimate to be a linear combination of the measurement, the new estimate will have the form:

$$\hat{\underline{x}}(k+1/k+1,\ell) = \hat{\underline{x}}(k+1/k,\ell) + K(k+1)\ \tilde{\underline{z}}(k+1/k,\ell) - K(k-\ell)\ \tilde{\underline{z}}(k-\ell/k,\ell) \quad (B71)$$

in which

$$\tilde{\underline{z}}(k+1/k,\ell) = z(k+1) - \hat{\underline{z}}(k+1/k,\ell)$$

$$= z(k+1) - D(k+1)\hat{\underline{x}}(k+1/k,\ell)$$

$$= D(k+1)\ \tilde{\underline{x}}(k+1/k,\ell) + \underline{v}(k+1) \quad (B72)$$

and

$$\tilde{\underline{z}}(k-\ell/k,\ell) = z(k-\ell) - \hat{\underline{z}}(k-\ell/k,\ell)$$

$$= z(k-\ell) - D(k-\ell)\ A(k-\ell,k+1)\ \hat{\underline{x}}(k+1/k,\ell)$$

$$+ D(k-\ell)\ \beta(k-\ell,k+1)$$

$$= D(k-\ell)\ A(k-\ell,k+1)\ \tilde{\underline{x}}(k+1/k,\ell) + \underline{v}(k-\ell) \qquad \text{(B73)}$$

Therefore;

$$\tilde{\underline{x}}(k+1/k+1,\ell) = \underline{x}(k+1) - \hat{\underline{x}}(k+1/k+1,\ell)$$

$$= \tilde{\underline{x}}(k+1/k,\ell) - K(k+1)\ \tilde{\underline{z}}(k+1/k,\ell)$$

$$+ K(k-\ell)\ \tilde{\underline{z}}(k-\ell/k,\ell)$$

$$= \left[I - K(k+1)\ D(k+1) + K(k-\ell)\ D(k-\ell)\ A(k-\ell,k+1)\right]$$

$$\tilde{\underline{x}}(k+1/k,\ell)$$

$$- K(k+1)\ \underline{v}(k+1) + K(k-\ell)\ \underline{v}(k-\ell) \qquad \text{(B74)}$$

In (B74), $\tilde{\underline{x}}(k+1/k,\ell)$ is the estimation error based on the measurements from $k-\ell$ to k. Thus, it is correlated with the measurement noise at $k-\ell$. However, the estimation error $\tilde{\underline{x}}(k+1/k+1,\ell)$ is based on the measurements from $k-\ell+1$ to $k+1$. The estimate $\hat{\underline{x}}(k+1/k+1,\ell)$ is totally unrelated with the measurements made before the interval $k-\ell+1$. Thus:

$$E\left[\tilde{\underline{x}}(k+1/k+1,\ell)\ \ \tilde{\underline{x}}^T(k+1/k+1,\ell)\right] + K(k-\ell)\ E\left[\ \underline{v}(k-\ell)\ \ \underline{v}^T(k-\ell)\right]\ K^T(k-\ell)$$

$$= \left[I - K(k+1)\ D(k+1) + K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right]\ E\left[\ \tilde{\underline{x}}(k+1/k,\ell)\ \ \tilde{\underline{x}}^T(k+1/k,\ell)\right]$$

$$x\ \left[I - K(k+1)\ D(k+1) + K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right]^T$$

$$+\ K(k+1)\quad E\left[\underline{v}(k+1)\ \ \underline{v}^T(k+1)\right]\ K^T(k+1)$$

$$P(k+1/k+1,\ell)\ \ +\ \ K(k-\ell)\ R(k-\ell)\ K^T(k-\ell)$$

$$= \left[I - K(k+1)\ D(k+1) + K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right]\ P(k+1/k,\ell)$$

$$\times \left[I - K(k+1)\ D(k+1) + K(k-\ell)\ D(k-\ell)\ A(k-\ell,k)\right]^T$$

$$+ \ K(k+1)\ R(k+1)\ K^T(k+1) \qquad (B75)$$

Let $G = (K(k+1) \quad -K(k-\ell))$ (B76)

$$M = \begin{pmatrix} D(k+1) \\ D(k+1)\ A(k-\ell,k) \end{pmatrix} \qquad (B77)$$

$$U = \begin{pmatrix} \underline{z}(k+1) \\ \underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k+1) \end{pmatrix} \qquad (B78)$$

and $S = \begin{pmatrix} R(k+1) & 0 \\ 0 & -R(k-\ell) \end{pmatrix}$ (B79)

(B71) and (B75) can be simplified as:

$$\hat{\underline{x}}(k+1/k+1,\ell) = \left[I - GM\right] \hat{\underline{x}}(k+1/k,\ell) - GU \qquad (B80)$$

and

$$P(k+1/k+1,\ell) = \left[I - GM\right] P(k+1/k,\ell) \left[I - GM\right]^T + GSG^T \qquad (B81)$$

Completing the square of (B81)

$$P(k+1/k+1,\ell) = \left[G - V\right] \left[M\ P(k+1/k,\ell)\ M^T + S\right] \left[G - V\right]^T$$

$$- V \left[M\ P(k+1/k,\ell)\ M^T + S\right] V^T + P(k+1/k,\ell) \qquad (B82)$$

With V satisfying the expression:

$$V \left[M\ P(k+1/k,\ell)\ M^T + S\right] = P(k+1/k,\ell)\ M^T \qquad (B83)$$

From (B82), $P(k+1/k+1,\ell)$ is minimum if $G = V$.

Thus:

$$G = P(k+1/k,\ell)\ M^T \left[M\ P(k+1/k,\ell)\ M^T + S\right]^{-1} \qquad \text{(B84)}$$

and

$$P(k+1/k+1,\ell) = \left[I - GM\right] P(k+1/k,\ell) \qquad \text{(B85)}$$

B.4.2 Derivation from Weighted Least Squares Approach

With (B21) and (B22), let:

$$Z(k) = \begin{pmatrix} \underline{z}(k-\ell+1) + D(k-\ell+1)\ \beta(k-\ell+1,k) \\ \underline{z}(k-\ell+2) + D(k-\ell+2)\ \beta(k-\ell+2,k) \\ \cdot \\ \cdot \\ \cdot \\ \underline{z}(k-2) + D(k-2)\ \beta(k-2,k) \\ \underline{z}(k-1) + D(k-1)\ \beta(k-1,k) \end{pmatrix}$$

$$N(k) = \begin{pmatrix} D(k-\ell+1)\ A(k-\ell+1,k) \\ D(k-\ell+2)\ A(k-\ell+2,k) \\ \cdot \\ \cdot \\ \cdot \\ D(k-2)\ A(k-2,k) \\ D(k-1)\ A(k-1,k) \end{pmatrix}$$

$$V(k) = \begin{pmatrix} \underline{v}(k-\ell+1) \\ \underline{v}(k-\ell+2) \\ \cdot \\ \cdot \\ \cdot \\ \underline{v}(k-2) \\ \underline{v}(k-1) \end{pmatrix}$$

and

$$Q(k) = \begin{pmatrix} R(k-\ell+1) & & & & & \\ & R(k-\ell+2) & & & 0 & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & 0 & & & R(k-2) & \\ & & & & & R(k-1) \end{pmatrix}$$

Therefore:

$$\begin{pmatrix} Z(k) + N(k)\ A(k,k+1)\ B(k)\ \underline{u}(k) \\ \underline{z}(k) \end{pmatrix} = \begin{pmatrix} N(k)\ A(k,k+1) \\ D(k+1) \end{pmatrix} \underline{x}(k+1)$$

$$+ \begin{pmatrix} V(k) \\ \underline{v}(k+1) \end{pmatrix} \qquad \text{(B86)}$$

and

$$\begin{pmatrix} \underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k) \\ Z(k) \end{pmatrix} = \begin{pmatrix} D(k-\ell)\ A(k-\ell,k) \\ N(k) \end{pmatrix} \underline{x}(k)$$

$$+ \begin{pmatrix} \underline{v}(k-\ell) \\ V(k) \end{pmatrix} \qquad \text{(B87)}$$

From (B86), the weighted least squares estimate is:

$$\hat{\underline{x}}(k+1/k+1,\ell) = P(k+1/k+1,\ell) \left[(A^T(k,k+1)\ N^T(k) \quad D^T(k+1)) \begin{pmatrix} Q^{-1}(k) & 0 \\ 0 & R^{-1}(k+1) \end{pmatrix} \right.$$

$$\left. x \begin{pmatrix} Z(k) + N(k)\ A(k,k+1)\ B(k)\ \underline{u}(k) \\ \underline{z}(k+1) \end{pmatrix} \right]$$

$$= P(k+1/k+1,\ell) \quad \left[A^T(k,k+1)\ N^T(k)\ Q^{-1}(k)\ Z(k)\right.$$

$$+ \quad D^T(k+1) \; R^{-1}(k+1) \; \underline{z}(k+1)$$

$$+ \quad A^T(k,k+1) \; N^T(k) \; Q^{-1}(k) \; N(k) \; A(k,k+1) \; B(k) \; \underline{u}(k)\Big] \tag{B88}$$

With:

$$P(k+1/k+1,\ell) \quad = \quad \Bigg[\; (A^T(k,k+1) \; N^T(k) \quad D^T(k+1)) \begin{pmatrix} Q^{-1}(k) & 0 \\ 0 & R^{-1}(k+1) \end{pmatrix}$$

$$\times \quad \begin{pmatrix} N(k) \; A(k,k+1) \\ D(k+1) \end{pmatrix} \Bigg]^{-1}$$

$$= \quad \Big[A^T(k,k+1) \; N^T(k) \; Q^{-1}(k) \; N(k) \; A(k,k+1)$$

$$+ \quad D^T(k+1) \; R^{-1}(k+1) \; D(k+1)\Big]^{-1} \tag{B89}$$

From (B87), the corresponding weighted least squares estimate is:

$$\underline{\hat{x}}(k/k,\ell) \quad = \quad P(k/k,\ell) \; \Bigg[\; (A^T(k-\ell,k) \; D^T(k-\ell) \quad N^T(k)) \begin{pmatrix} R^{-1}(k-\ell) & 0 \\ 0 & Q^{-1}(k) \end{pmatrix}$$

$$\times \quad \begin{pmatrix} \underline{z}(k-\ell) \quad + \quad D(k-\ell) \;\; \beta(k-\ell,k) \\ Z(k) \end{pmatrix} \Bigg]$$

$$= \quad P(k/k,\ell) \; \Big[A^T(k-\ell,k) \; D^T(k-\ell) \; R^{-1}(k-\ell)$$

$$\times \quad (\underline{z}(k-\ell) + D(k-\ell) \; \beta(k-\ell,k))$$

$$+ \quad N^T(k) \quad Q^{-1}(k) \quad Z(k) \; \Big] \tag{B90}$$

With

$$P(k/k,\ell\,) \quad = \quad \Bigg[\; (A^T(k-\ell,k) \; D^T(k-\ell) \quad N^T(k)) \quad \begin{pmatrix} R^{-1}(k-\ell) & 0 \\ 0 & Q^{-1}(k) \end{pmatrix}$$

$$\times \quad \begin{pmatrix} D(k-\ell) \;\; A(k-\ell,k) \\ N(k) \end{pmatrix} \Bigg]^{-1}$$

$$= \Big[A^T(k-\ell,k)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ D(k-\ell)\ A(k-\ell,k)$$

$$+ N^T(k)\ Q^{-1}(k)\ N(k) \Big]^{-1} \qquad \text{(B91)}$$

From (B90)

$$N^T(k)\ Q^{-1}(k)\ Z(k) = P^{-1}(k/k,\ell)\ \hat{\underline{x}}(k/k,\ell) - D(k-\ell)\ \beta(k-\ell,k)$$

$$- A^T(k-\ell,k)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ \underline{z}(k-\ell) \qquad \text{(B92)}$$

Substituting (B21) into (B92) gives:

$$N^T(k)\ Q^{-1}(k)\ Z(k) = P^{-1}(k/k,\ell)\ A(k,k+1)\ \hat{\underline{x}}(k+1/k,\ell)$$

$$- N^T(k)\ Q^{-1}(k)\ N(k)\ A(k,k+1)\ B(k)\ \underline{u}(k)$$

$$- A^T(k-\ell,k)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ \underline{z}(k-\ell)$$

$$- A^T(k-\ell,k)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ D(k-\ell)\ \beta(k-\ell,k+1) \qquad \text{(B93)}$$

Substituting (B93) into (B88)

$$\hat{\underline{x}}(k+1/k+1,\ell) = P(k+1/k+1,\ell)\ \Big[P^{-1}(k+1/k,\ell)\ \hat{\underline{x}}(k+1/k,\ell)$$

$$+ D^T(k+1)\ R^{-1}(k+1)\ \underline{z}(k+1)$$

$$- A^T(k-\ell,k+1)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ (\underline{z}(k-\ell)$$

$$+ D(k-\ell)\ \beta(k-\ell,k+1))\ \Big] \qquad \text{(B94)}$$

Adding and subtracting the terms $(D^T(k+1)\ R^{-1}(k+1)\ D(k+1)\ \hat{\underline{x}}(k+1/k,\ell))$ and $(A^T(k-\ell,k+1)\ D^T(k-\ell)\ R^{-1}(k-\ell)\ D(k-\ell)\ A(k-\ell,k+1)\ \hat{\underline{x}}(k+1/k,\ell))$ in the right hand side of (B94), and together with the introduction of:

$$M = \begin{pmatrix} D(k-\ell)\ A(k-\ell,k+1) \\ D(k+1) \end{pmatrix}$$

$$S \quad = \quad \begin{pmatrix} -R(k-\ell) & 0 \\ 0 & R(k+1) \end{pmatrix}$$

and

$$U \quad = \quad \begin{pmatrix} \underline{z}(k-\ell) + D(k-\ell) \; \beta(k-\ell,k+1) \\ \underline{z}(k+1) \end{pmatrix}$$

(B94) becomes:

$$\hat{\underline{x}}(k+1/k+1,\ell) \quad = \quad P(k+1/k+1,\ell) \quad \left[(P^{-1}(k+1/k,\ell) + M^TS^{-1}M) \; \hat{\underline{x}}(k+1/k,\ell) \right.$$
$$\left. + \; M^TS^{-1} \; (U - M \; \hat{\underline{x}}(k+1/k,\ell)) \right] \qquad (B95)$$

From (B91)

$$N^T(k) \; Q^{-1}(k) \; N(k) \quad = \quad P^{-1}(k/k,\ell)$$
$$- \; A^T(k-\ell,k) \; D^T(k-\ell) \; R^{-1}(k-\ell) \; D(k-\ell) \; A(k-\ell,k) \qquad (B96)$$

Substituting (B96) into (B89) gives:

$$P(k+1/k+1,\ell) \quad = \quad \left[P^{-1} \; (k+1/k,\ell) \; + \; M^TS^{-1}M\right]^{-1} \qquad (B97)$$

Thus:

$$\hat{\underline{x}}(k+1/k+1,\ell) \quad = \quad \hat{\underline{x}}(k+1/k,\ell)$$
$$+ \; P(k+1/k+1,\ell) \; M^TS^{-1} \; (U - M \; \hat{\underline{x}}(k+1/k,\ell)) \qquad (B98)$$

Let $G \quad = \quad P(k+1/k+1,\ell) \; M^TS^{-1}$ (B99)

(B98) becomes:

$$\hat{\underline{x}}(k+1/k+1,\ell) \quad = \quad \left[I - GM\right] \quad \hat{\underline{x}}(k+1/k,\ell) \; + \; GU \qquad (B100)$$

Applying the Inversion Lemma to (B97) gives:

$$P(k+1/k+1,\ell) \quad = \quad P(k+1/k,\ell)$$
$$- \; P(k+1/k,\ell) \; M^T(M \; P(k+1/k,\ell)M^T \; + \; S)^{-1}$$
$$\times \; M \; P(k+1/k,\ell) \qquad (B101)$$

From the Inversion Lemma, using the relationship:

$$(P^{-1}(k+1/k,\ell) + M^T S^{-1} M)^{-1} M^T S^{-1}$$

$$= P(k+1/k,\ell)\ M^T\ (M\ P(k+1/k,\ell)\ M^T + S)^{-1} \qquad \text{(B102)}$$

(B99) and (B101) can be rewritten as:

$$G = P(k+1/k,\ell)\ M^T \left[M\ \ P(k+1/k,\ell)\ M^T + S\right]^{-1} \qquad \text{(B103)}$$

and

$$P(k+1/k+1,\ell) = \left[I - GM\right]\ P(k+1/k,\ell) \qquad \text{(B104)}$$

Therefore (B98) can be rewritten as:

$$\hat{\underline{x}}(k+1/k+1,\ell) = \left[I - GM\right]\ \hat{\underline{x}}(k+1/k,\ell) + GU \qquad \text{(B105)}$$

Hence (B13), (B19), (B103), (B104) and (B105) form the algorithm corresponding to the one derived from the minimum variance approach.

B.4.3 Summary of Algorithm for Moving Window

$$\hat{\underline{x}}(k+1/k,\ell) = A(k+1,k)\ \ \hat{\underline{x}}(k/k,\ell) + B(k)\ \underline{u}(k) \qquad \text{(B106)}$$

$$P(k+1/k,\ell) = A(k+1,k)\ P(k/k,\ell)\ A^T(k+1,k) \qquad \text{(B107)}$$

$$G = P(k+1/k,\ell)\ M^T \left[M\ P(k+1/k,\ell)\ M^T + S\right]^{-1} \qquad \text{(B108)}$$

$$P(k+1/k+1,\ell) = \left[I - GM\right]\ P(k+1/k,\ell) \qquad \text{(B109)}$$

$$\hat{\underline{x}}(k+1/k+1,\ell) = \left[I - GM\right]\ \hat{\underline{x}}(k+1/k,\ell) + GU \qquad \text{(B110)}$$

with

$$M = \begin{pmatrix} & D(k+1) \\ D(k-\ell) & A(k-\ell,k+1) \end{pmatrix}$$

$$S = \begin{pmatrix} R(k+1) & 0 \\ 0 & -R(k-\ell) \end{pmatrix}$$

$$U = \begin{pmatrix} \underline{z}(k+1) \\ \underline{z}(k-\ell) + D(k-\ell)\ \beta(k-\ell,k+1) \end{pmatrix}$$

and

$$G = (K(k+1) \quad -K(k-\ell)).$$

B.5 Gains for Moving Window

In (B108) the term $\left[MP(k+1/k,\ell)\ M^T + S\right]^{-1}$ can be expanded as:

$$\left[M\ P\ (k+1/k,\ell)\ M^T + S\right]^{-1}$$

$$= \left[\begin{pmatrix} D(k+1) \\ D(k-\ell)\ A(k-\ell,k+1) \end{pmatrix} \quad P(k+1/k,\ell)\ (D^T(k+1) \quad A^T(k-\ell,k+1)\ D^T(k-\ell)) \right.$$

$$\left. + \quad \begin{pmatrix} R(k+1) & 0 \\ 0 & -R(k-\ell) \end{pmatrix} \right]^{-1}$$

$$= \begin{bmatrix} a & c \\ b & d \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} e & g \\ f & h \end{bmatrix}$$

where;

$$a = D(k+1)\ P(k+1/k,\ell)\ D^T(k+1) + R(k+1) \qquad (B112)$$

$$b = D(k-\ell)\ A(k-\ell,k+1)\ P(k+1/k,\ell)\ D^T(k+1) \qquad (B113)$$

$$c = D(k+1)\ P(k+1/k,\ell)\ A^T(k-\ell,k+1)\ D^T(k-\ell) \qquad (B114)$$

$$d = D(k-\ell)\ A(k-\ell,k+1)\ P(k+1/k,\ell)\ A^T(k-\ell,k+1)\ D^T(k-\ell) - R(k-\ell) \qquad (B115)$$

$$e = \left[a - cd^{-1}b\right]^{-1} \qquad (B116)$$

$$f = -d^{-1}b\left[a - cd^{-1}b\right]^{-1} \qquad (B117)$$

$$g = -a^{-1}c\left[d - ba^{-1}c\right]^{-1} \qquad (B118)$$

$$h = \left[d - ba^{-1}c\right]^{-1} \qquad (B119)$$

Therefore (B108) becomes:

$$K(k+1) = P(k+1/k,\ell)\left[D^T(k+1)\ e + A^T(k-\ell,k+1)\ D^T(k-\ell)\ f\right] \qquad (B120)$$

$$-K(k-\ell) = P(k+1/k,\ell)\left[D^T(k+1)\ g + A^T(k-\ell,k+1)\ D^T(k-\ell)\ h\right] \qquad (B121)$$

(B120) and (B121) can be rewritten as:

$$K(k+1) = P(k+1/k,\ell)\ D^T(k+1)\left[D(k+1) P(k+1/k,\ell)\ D^T(k+1) + R(k+1)\right]^{-1}$$

$$+\ K(k-\ell)\ D(k-\ell)\ A(k-\ell,k+1)\ P(k+1/k,\ell)\ D^T(k+1)$$

$$x\left[D(k+1)\ P(k+1/k,\ell)\ D^T(k+1) + R(k+1)\right]^{-1} \qquad (B122)$$

$$-\ K(k-\ell) = P(k+1/k,\ell)\ A^T(k-\ell,k+1)\ D^T(k-\ell)$$

$$x\left[D(k-\ell)\ A(k-\ell,k+1)\ P(k+1/k,\ell)\ A^T(k-\ell,k+1)\ D^T(k-\ell) - R(k-\ell)\right]^{-1}$$

$$-\ K(k+1)\ D(k+1)\ P(k+1/k,\ell)\ A^T(k-\ell,k+1)\ D^T(k-\ell)$$

$$x\left[D(k-\ell)\ A(k-\ell,k+1)\ P(k+1/k,\ell)\ A^T(k-\ell,k+1)\ D^T(k-\ell) - R(k-\ell)\right]^{-1} \qquad (B123)$$

In (B122), the first term on the right hand side is the gain of adding one new measurement, (B41). In (B123), the first term on the right hand side is the gain of removing one old measurement, (B68). Thus, the gains of moving window are related to, but are more complicated than, the gains of adding and removing measurements separately.

B.6 Model with Plant Noise

Occasionally, due to the existence of disturbances in the system or the inadequate knowledge of the complete system, model presented in Section B.1 becomes inappropriate. The model is modified with the introduction of a plant noise, $\underline{\omega}(k)$.

$$\underline{x}(k+1) = A(k+1,k)\ \underline{x}(k) + B(k)\ \underline{u}(k) + C(k)\ \underline{\omega}(k) \qquad (B124)$$

$$\underline{z}(k+1) = D(k+1)\ \underline{x}(k+1) + \underline{v}(k+1) \qquad (B125)$$

The plant noise, $\underline{\omega}(k)$ is generally assumed to be a zero mean gaussian random process, uncorrelated with the measurement noise.

i.e. $$E\left[\underline{\omega}(k)\right] = \underline{0} \qquad (B126)$$

$$E\left[\underline{\omega}(k)\ \underline{\omega}^T(j)\right] = Q(k)\ \delta_{jk} \qquad (B127)$$

The model is illustrated in Figure B-1.

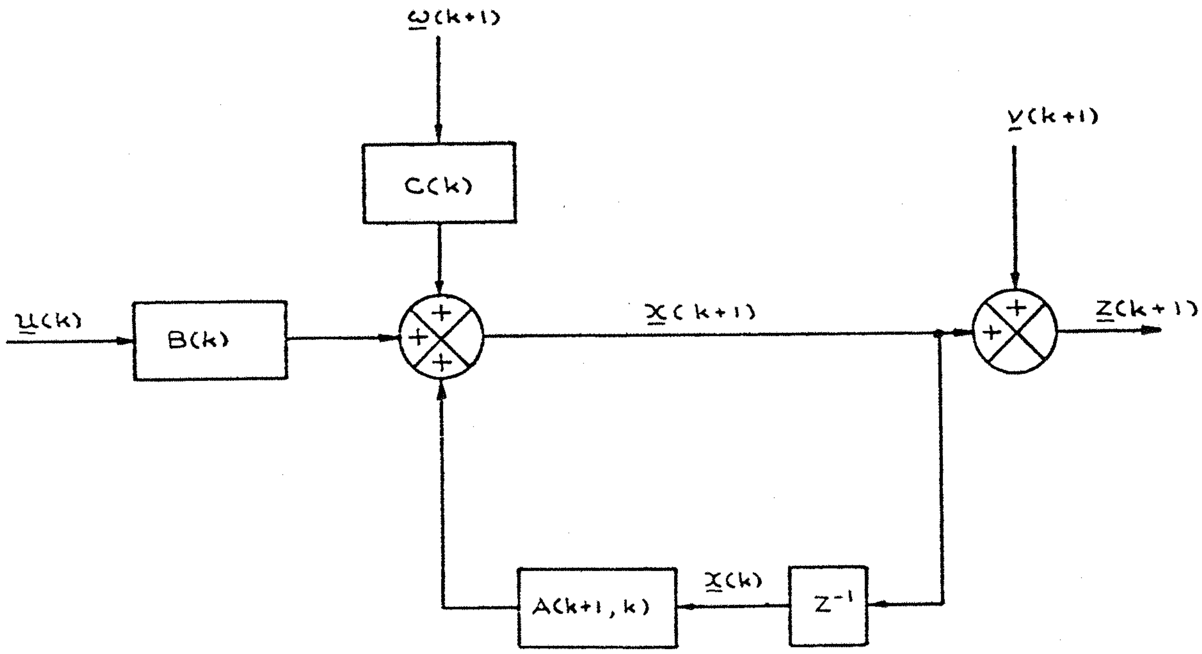


Figure B-1 Model With Plant Noise.

B.6.1 Finite Memory Filtering Algorithm for Models with Plant Noise

It is assumed that an estimate of $\underline{x}(k)$, $\hat{\underline{x}}(k/k,\ell)$ and its covariance, $P(k/k,\ell)$, are available a priori. The finite memory filtering involves the removal of old measurement at $k-\ell$, it is therefore necessary to project $\underline{x}(k)$ and $\hat{\underline{x}}(k/k,\ell)$ back to the time interval $k-\ell$. Re-arranging (B124)

$$\underline{x}(k) = A(k,k+1)\ \underline{x}(k+1) - A(k,k+1)\ B(k)\ \underline{u}(k)$$
$$- A(k,k+1)\ C(k)\ \underline{\omega}(k) \qquad (B128)$$

Extending the time interval backward to $k-\ell$

$$\underline{x}(k-\ell) = A(k-\ell,k+1)\ \underline{x}(k+1) - \beta(k-\ell,k+1)$$
$$- \beta_{\omega}(k-\ell,k+1) \qquad (B129)$$

where $\beta(k-\ell,k+1) = \sum_{j=k-\ell}^{k} A(k-\ell,j+1)\ B(j)\ \underline{u}(j)$

and $\beta_{\omega}(k-\ell,k+1) = \sum_{j=k-\ell}^{k} A(k-\ell,j+1)\ C(j)\ \underline{\omega}(j)$

Since $\omega(j)$ is a random process, the best projection of $\hat{\underline{x}}(k/k,\ell)$ back to $k-\ell$, $\hat{\underline{x}}(k-\ell/k,\ell)$, is given as:

$$\hat{\underline{x}}(k-\ell/k,\ell) = A(k-\ell,k+1)\ \hat{\underline{x}}(k+1/k,\ell) - \beta(k-\ell,k+1) \qquad (B130)$$

From the measurement equation (B125) the actual and estimated measurements at $k-\ell$ are:

$$\underline{z}(k-\ell) = D(k-\ell)\ A(k-\ell,k)\ \underline{x}(k) - D(k-\ell)\ \beta(k-\ell,k)$$
$$- D(k-\ell)\ \beta_{\omega}(k-\ell,k) + \underline{v}(k-\ell) \qquad (B131)$$

$$\hat{\underline{z}}(k-\ell/k,\ell) = D(k-\ell)\ A(k-\ell,k)\ \hat{\underline{x}}(k-\ell/k,\ell) - D(k-\ell)\ \beta(k-\ell,k) \qquad (B132)$$

Assuming, the estimate of the new states is of the form as (B71)

$$\hat{\underline{x}}(k+1/k+1,\ell) \;=\; \hat{\underline{x}}(k+1/k,\ell) + K(k+1)\;\tilde{\underline{z}}(k+1/k,\ell)$$

$$-\;\; K(k-\ell)\;\tilde{\underline{z}}(k-\ell/k,\ell) \qquad \text{(B133)}$$

in which

$$\tilde{\underline{z}}(k+1/k,\ell) \;=\; \underline{z}(k+1) - \hat{\underline{z}}(k+1/k,\ell)$$

$$=\; D(k+1)\,\tilde{\underline{x}}(k+1/k,\ell) + \underline{v}(k+1) \qquad \text{(B134)}$$

and

$$\tilde{\underline{z}}(k-\ell/k,\ell) \;=\; z(k-\ell) - \hat{\underline{z}}(k-\ell/k,\ell)$$

$$=\; D(k-\ell)\;A(k-\ell,k+1)\;\tilde{\underline{x}}(k+1/k,\ell)$$

$$-\;\; D(k-\ell)\;\beta_{\omega}(k-\ell,k+1) + \underline{v}(k-\ell) \qquad \text{(B135)}$$

Therefore:

$$\tilde{\underline{x}}(k+1/k+1,\ell) \;=\; \underline{x}(k+1) - \hat{\underline{x}}(k+1/k+1,\ell)$$

$$=\; \tilde{\underline{x}}(k+1/k,\ell) - K(k+1)\,\tilde{\underline{z}}(k+1/k,\ell)$$

$$+\;\; K(k-\ell)\;\tilde{\underline{z}}(k-1/k,\ell)$$

$$=\; \left[I - K(k+1)\;D(k+1) + K(k-\ell)\;D(k-\ell)\;A(k-\ell,k+1)\right]\tilde{\underline{x}}(k+1/k,\ell)$$

$$-\;\; K(k+1)\;\underline{v}(k+1) + K(k-\ell)\;\underline{v}(k-\ell)$$

$$-\;\; K(k-\ell)\;D(k-\ell)\;\beta_{\omega}(k-\ell,k) \qquad \text{(B136)}$$

In order to find the minimum variance estimate of $\underline{x}(k+1)$, the covariance must be minimised. The covariance matrix of $\hat{\underline{x}}(k+1/k+1,\ell)$ is:

$$P(k+1/k+1,\ell) \;=\; E\left[\tilde{\underline{x}}(k+1/k+1,\ell)\;\tilde{\underline{x}}^{T}(k+1/k+1,\ell)\right] \qquad \text{(B137)}$$

It involves the correlation between the past plant noise and the estimation error $\underline{\tilde{x}}(k+1/k,\ell)$.

It is the ignorance of this correlation that hinders the derivation of the algorithm. It was attempted to assume that they are uncorrelated. Such an assumption was not justified and the derived algorithm diverged when tested with a model with plant noise.

From the Kalman filtering point of view, the introduction of a plant noise in the model is to prevent the gain going to zero. Once the gain is zero, the estimation process is totally independent from the measurements. This phenomenon is referred as data saturation. The finite memory Kalman filter, on the other hand, keeps the gain at a constant value even without the introduction of the plant noise. This is because the finite memory Kalman filter can be treated as a weighted least squares estimator. Its covariance matrix and gain depend on the length of the data window only. Therefore, it is unnecessary to introduce a plant noise in the model if a finite memory Kalman filter is used. However, if the introduction of the plant noise is inevitable and a finite memory Kalman filter is called for, the system may then be remodelled as a combination of two systems, a deterministic model based on the measurable inputs and a stochastic model to cover the random disturbances. Both models will then have no plant noise and the finite memory Kalman filter can be used. This is very similar to the work presented in this thesis.

APPENDIX C

DIGITAL SIMULATION RESULTS

Figure	Target Motion	Ship Motion	Weight ratio (x)	Reciprocal of Decay time (a)
C.1	1	1	1.0	0.1
C.2	1	1	2.0	0.1
C.3	1	1	5.0	0.1
C.4	1	1	2.0	0.025
C.5	1	1	2.0	0.0025
C.6	2	1	2.0	0.025
C.7	3	1	2.0	0.025
C.8	1	2	2.0	0.025
C.9	1	3	2.0	0.025

Note:-

1) The parameters of the target motion is referred to Table 6-2.

2) The three ship motions are generated by passing three different random sequence to three parallel filters described in Table 6-1.

3) The weight ratio, x, is defined as:

$$x = \frac{\text{weight on realignments}}{\text{weight on probability of target loss}}$$

4) The weight profile used is of the form:

$$w(t) = 1.0 - 0.99 \times \text{Exp}(-Axt)$$

where the origin of t is at the initial engagement of a target.

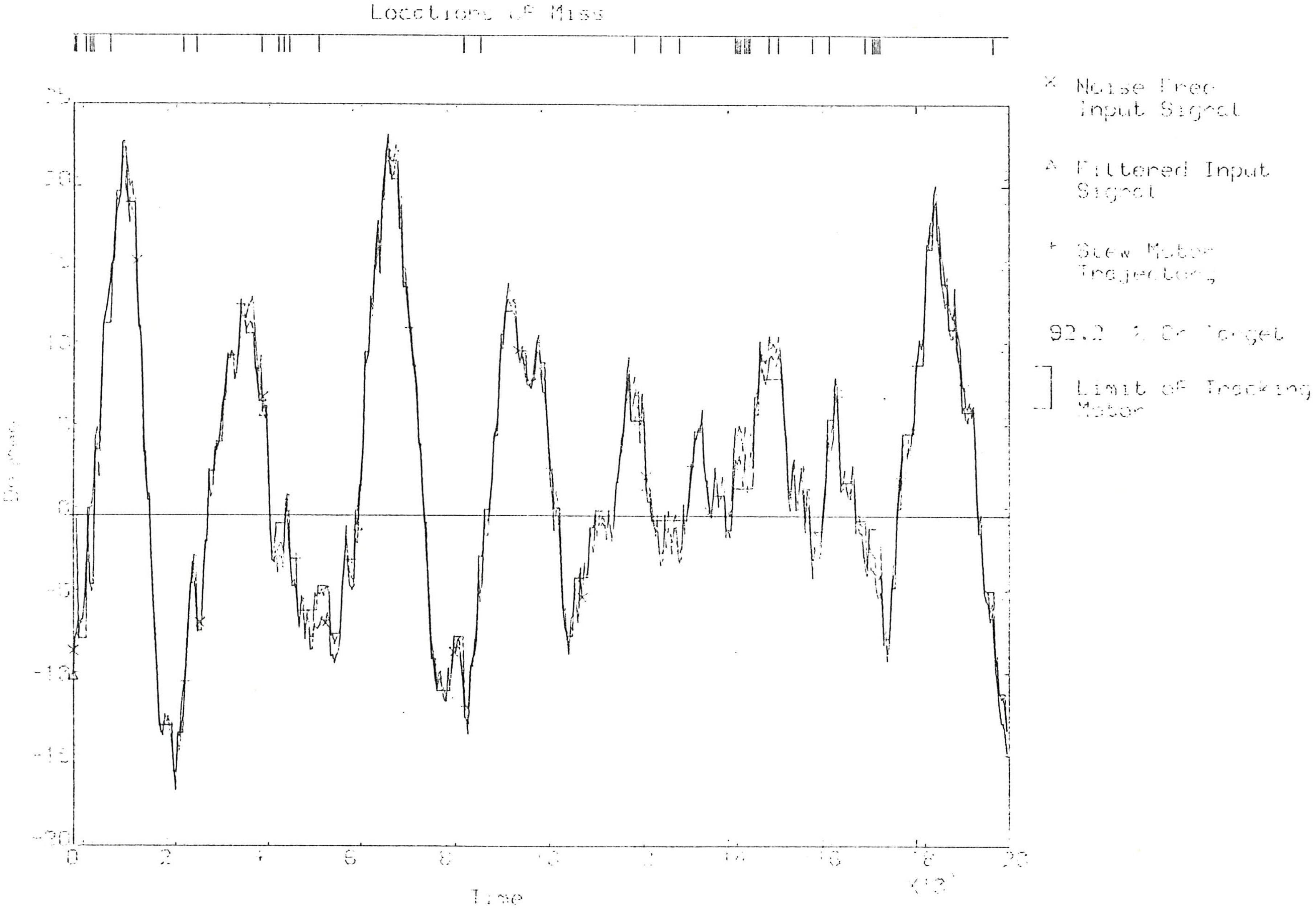


Figure C.1a History of simulated engagement

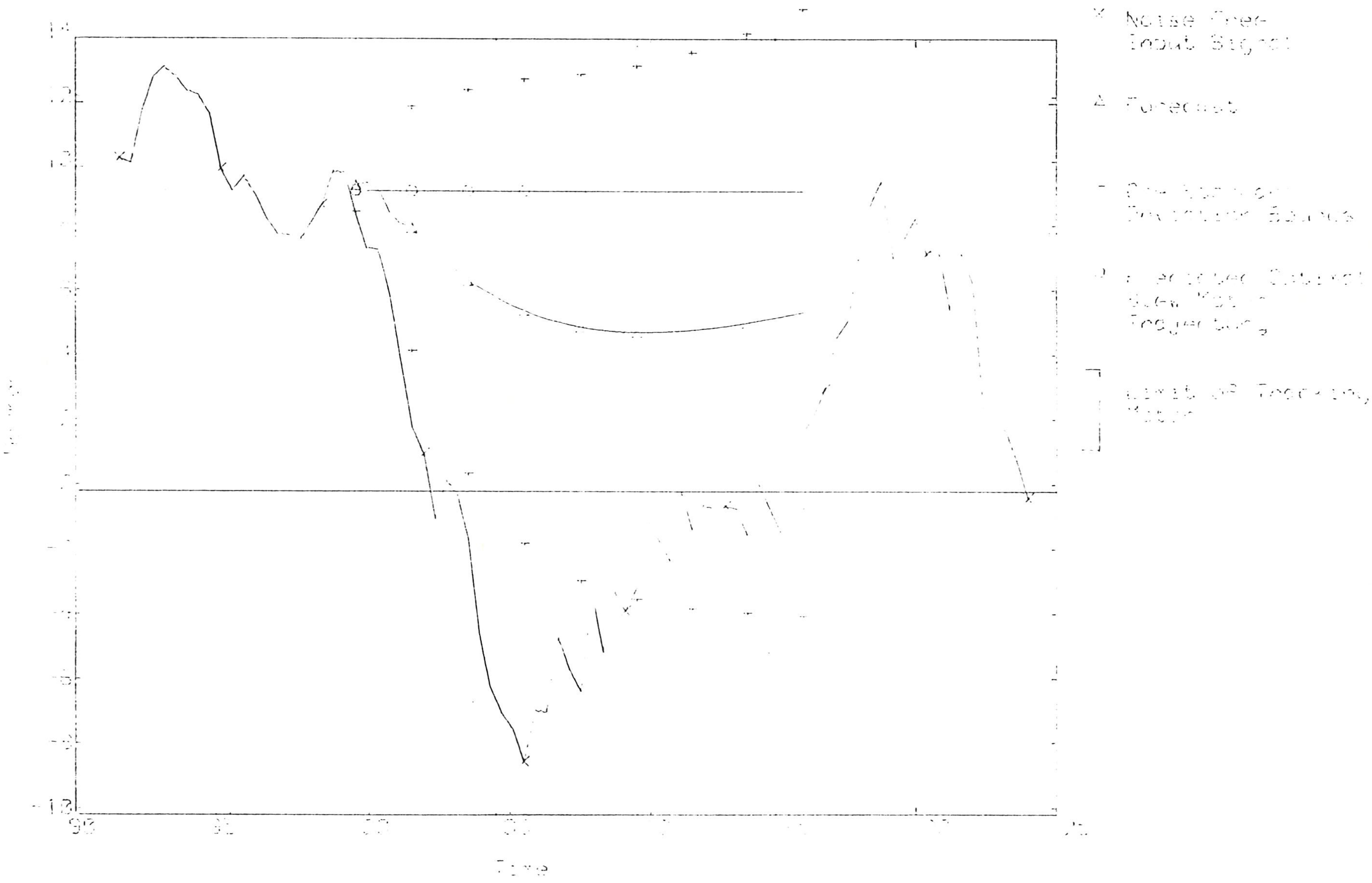


Figure C.1b Forecasts and predicted optimal slew motor trajectory at time interval 100

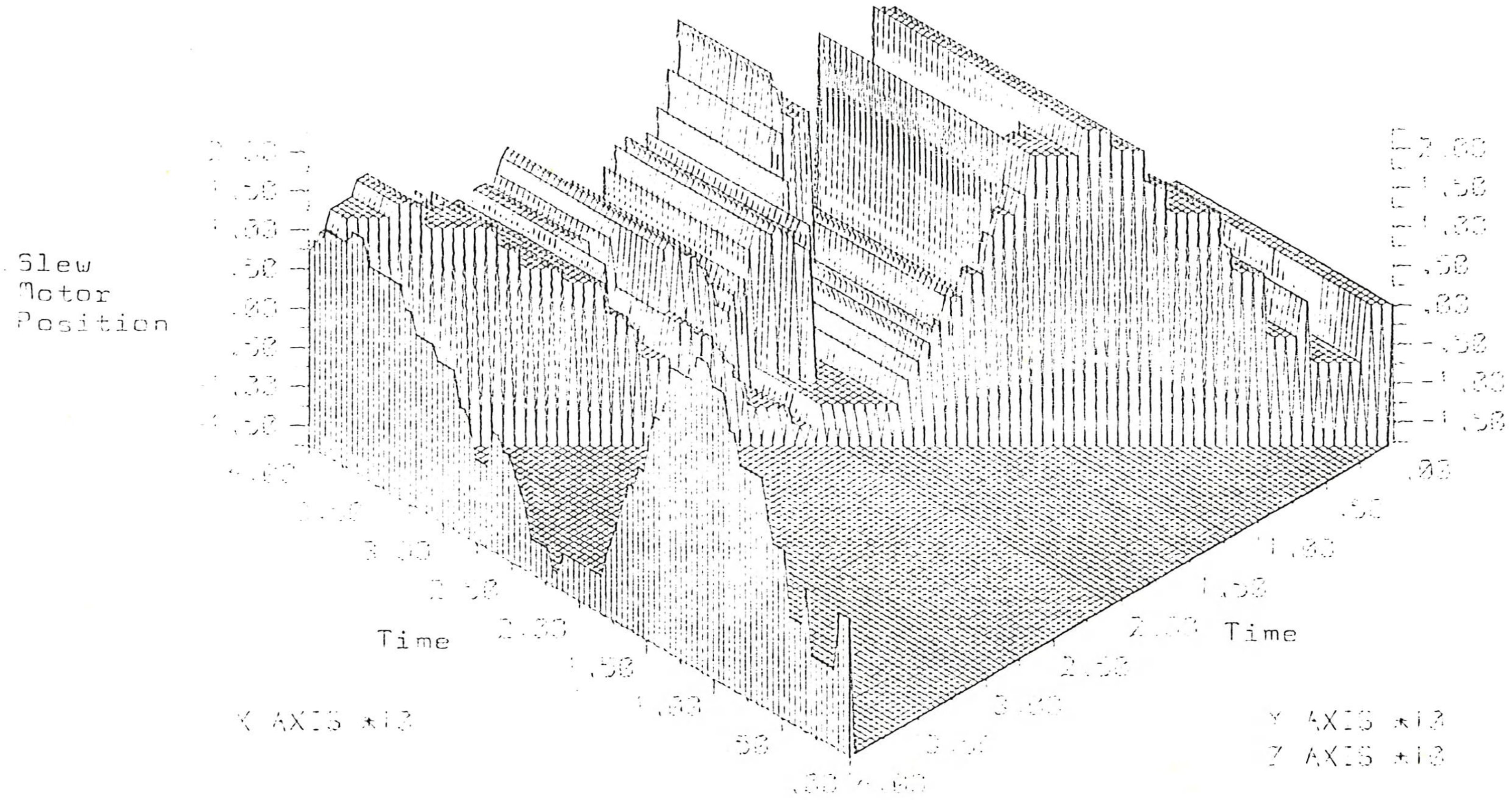


Figure C.1c Predicted optimal slew motor trajectories at the first 40 intervals

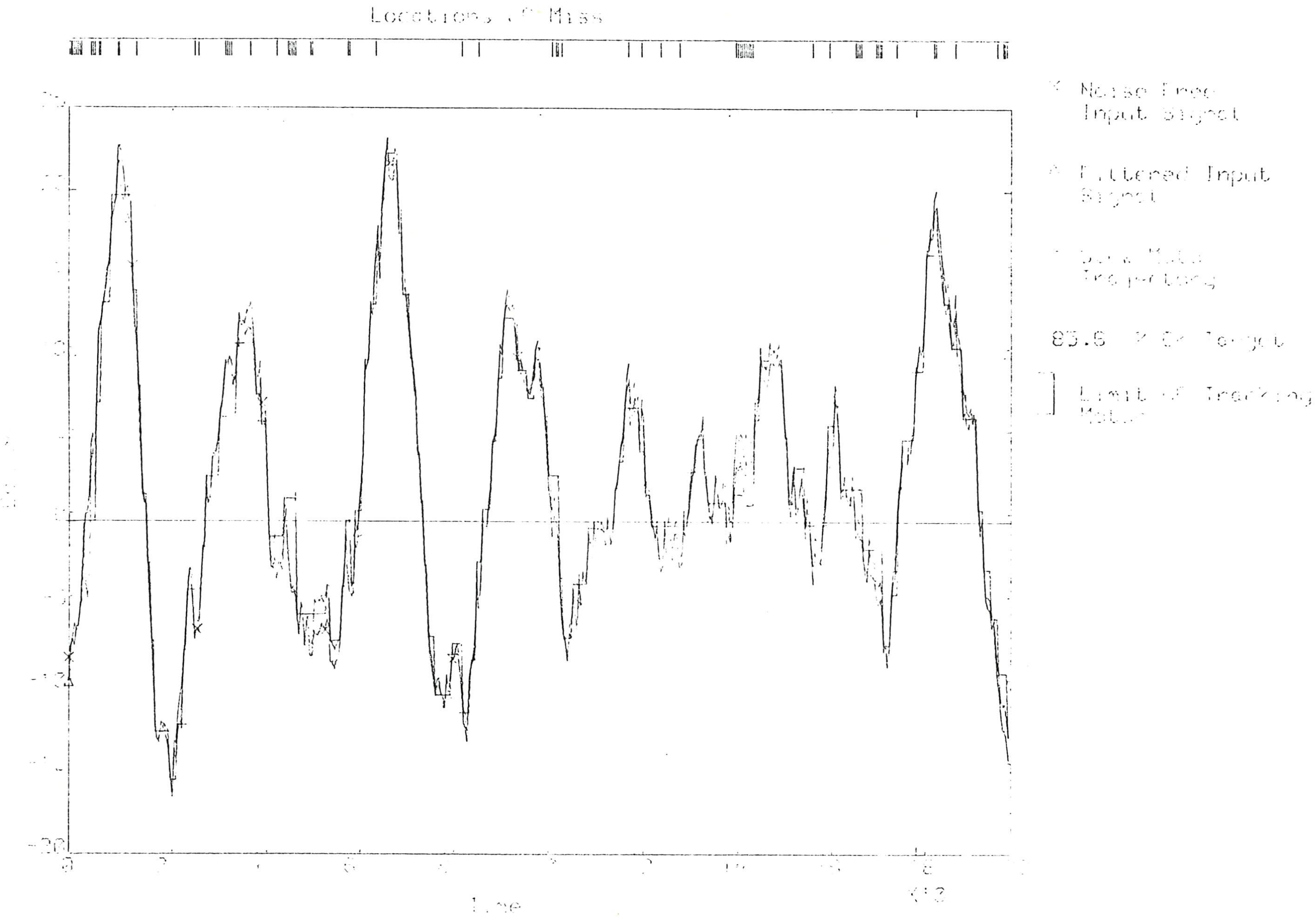


Figure C.2a History of simulated engagement

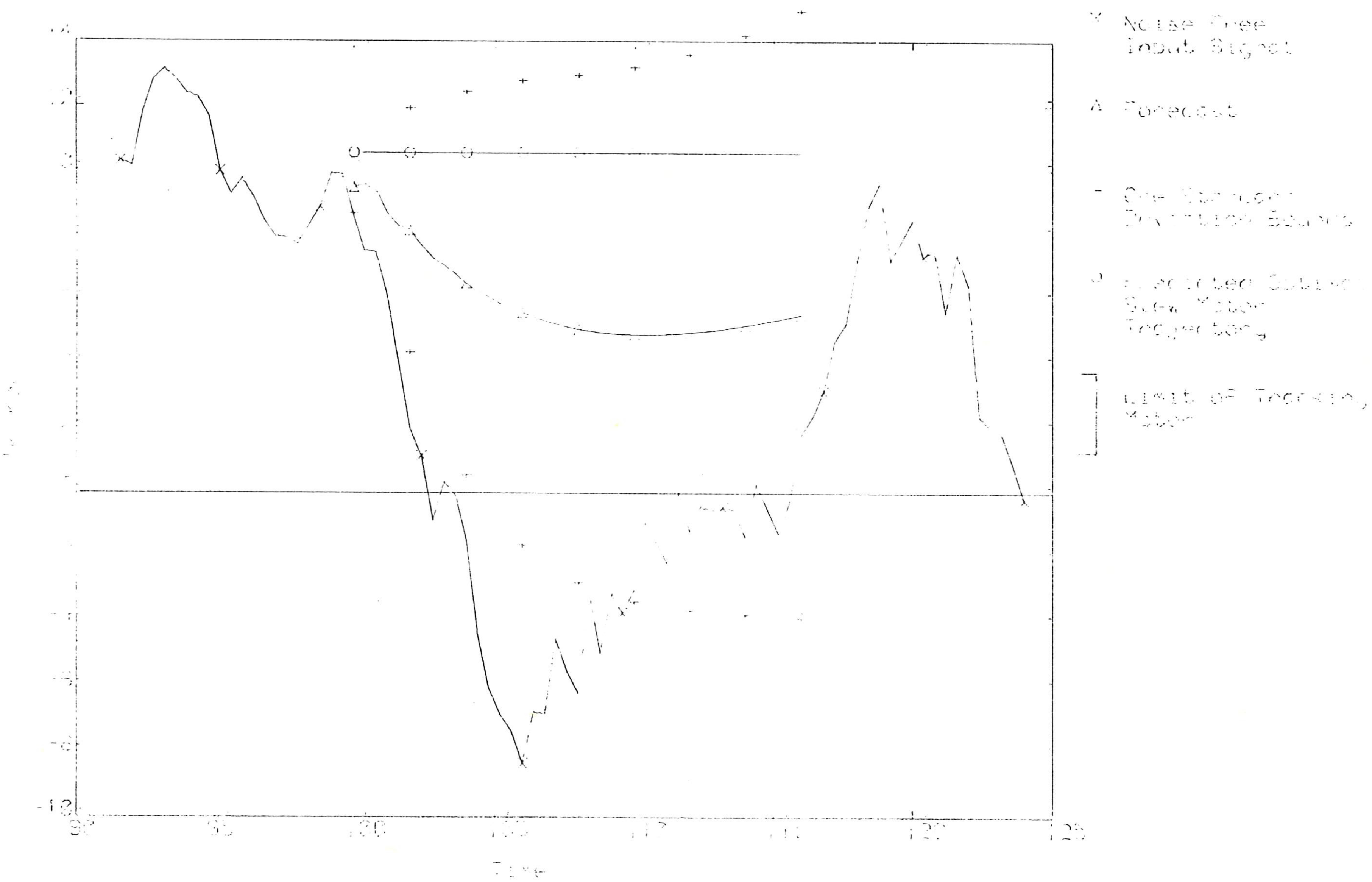


Figure C.2b Forecasts and predicted optimal slew motor trajectory at time interval 100

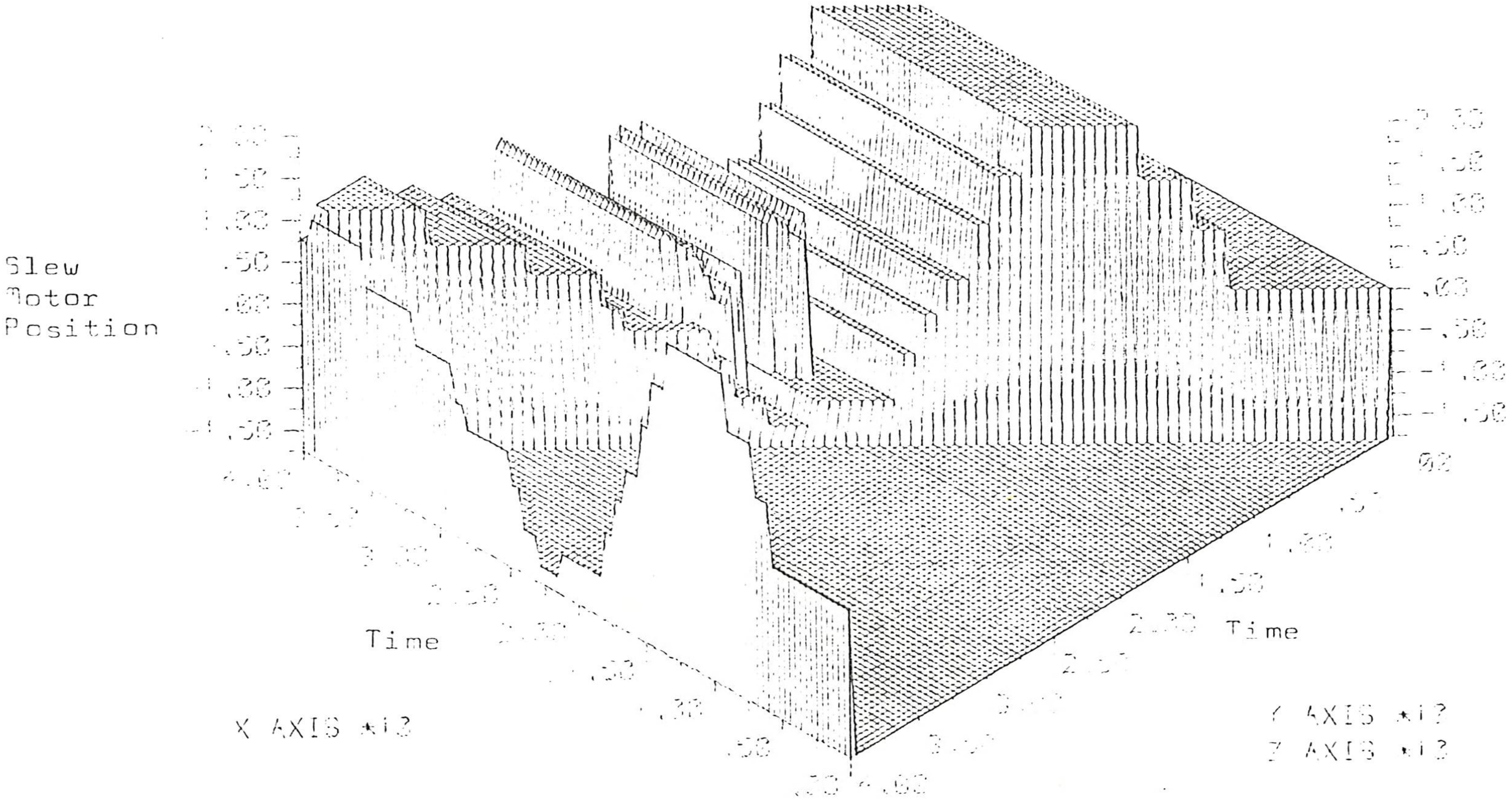


Figure C.2c Predicted optimal slew motor trajectories at the first 40 intervals

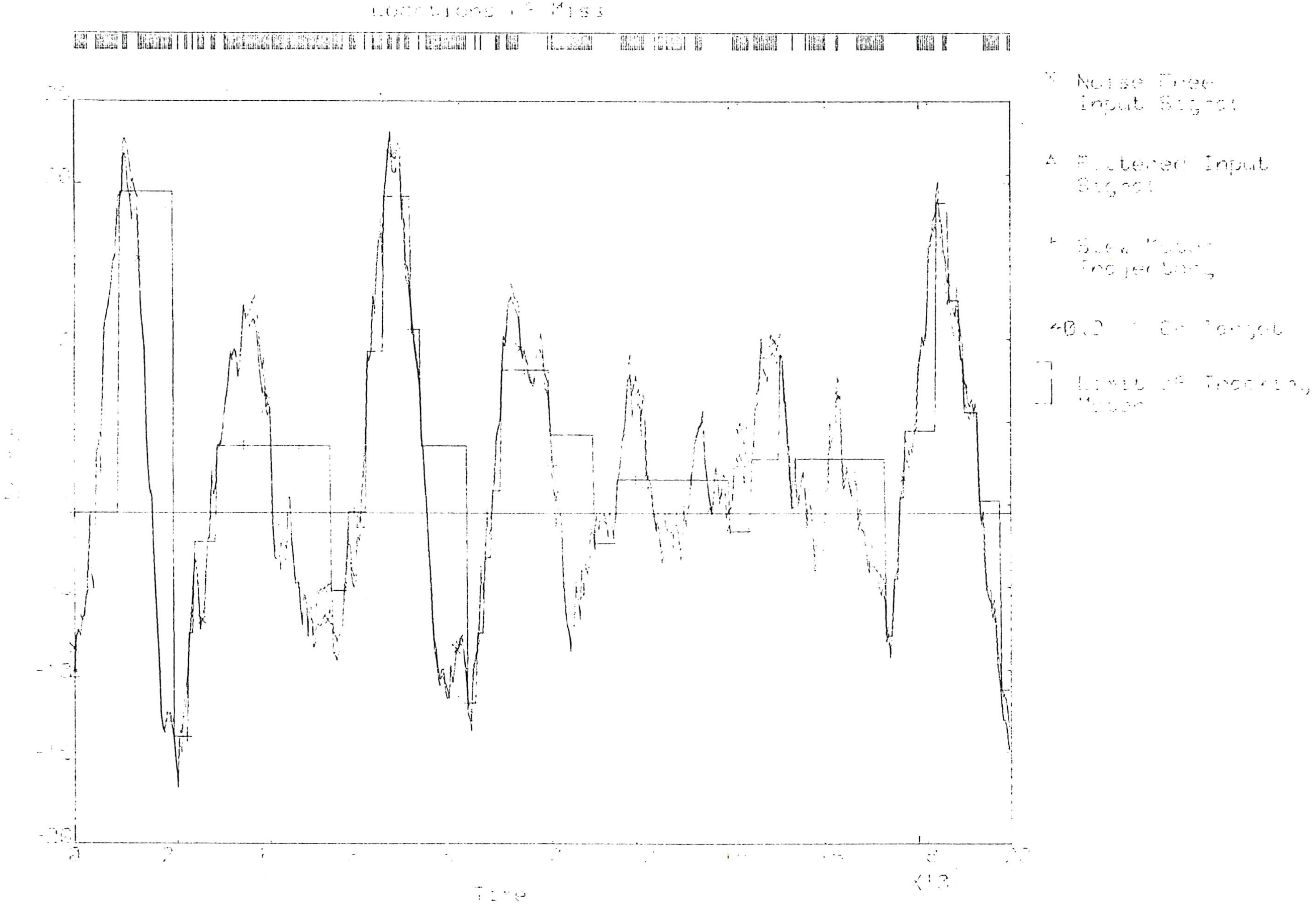


Figure C.3a History of simulated engagement

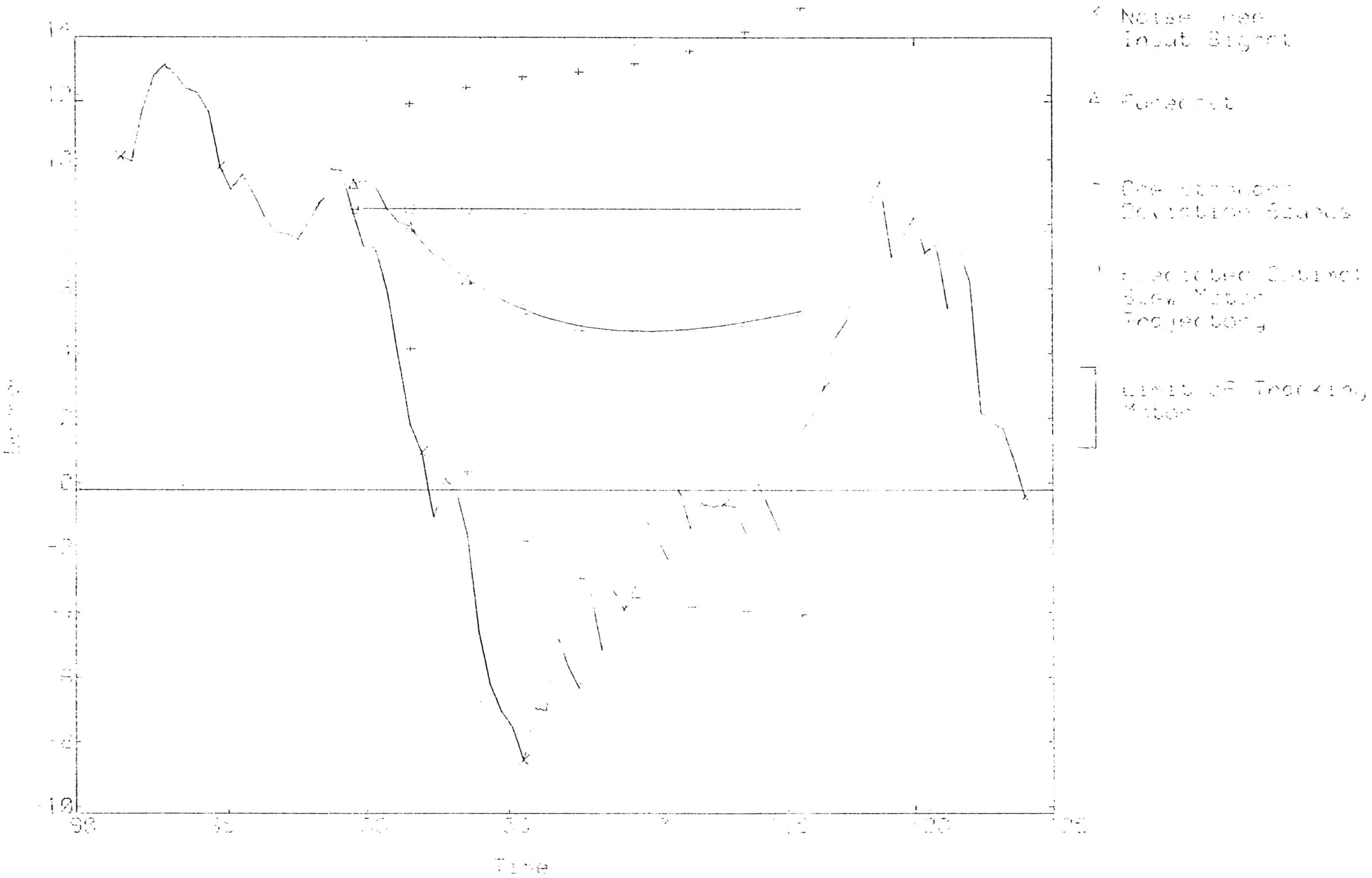


Figure C.3b Forecasts and predicted optimal slew motor trajectory at time interval 100

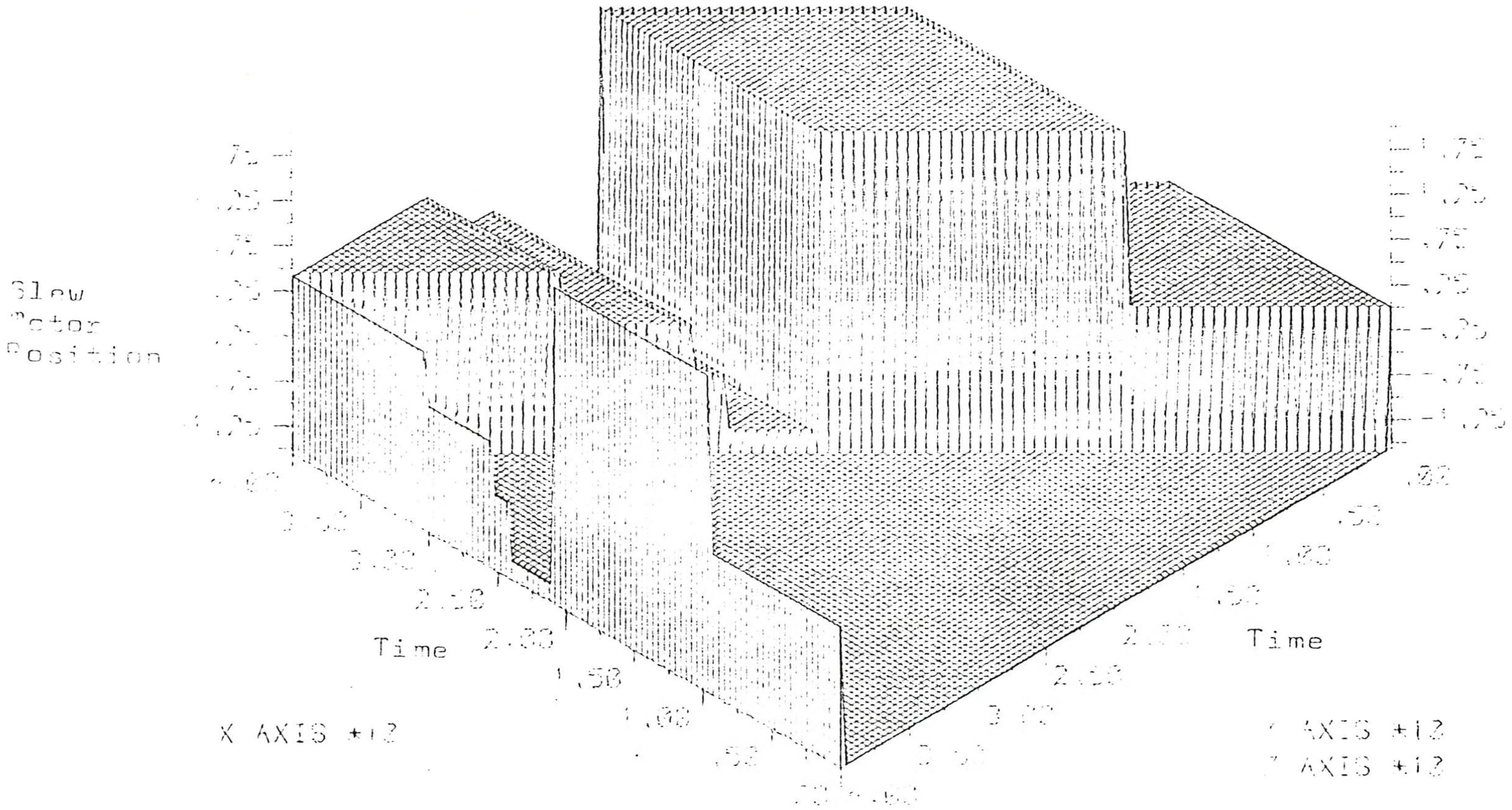


Figure C.3c Predicted optimal slew motor trajectories at the first 40 intervals

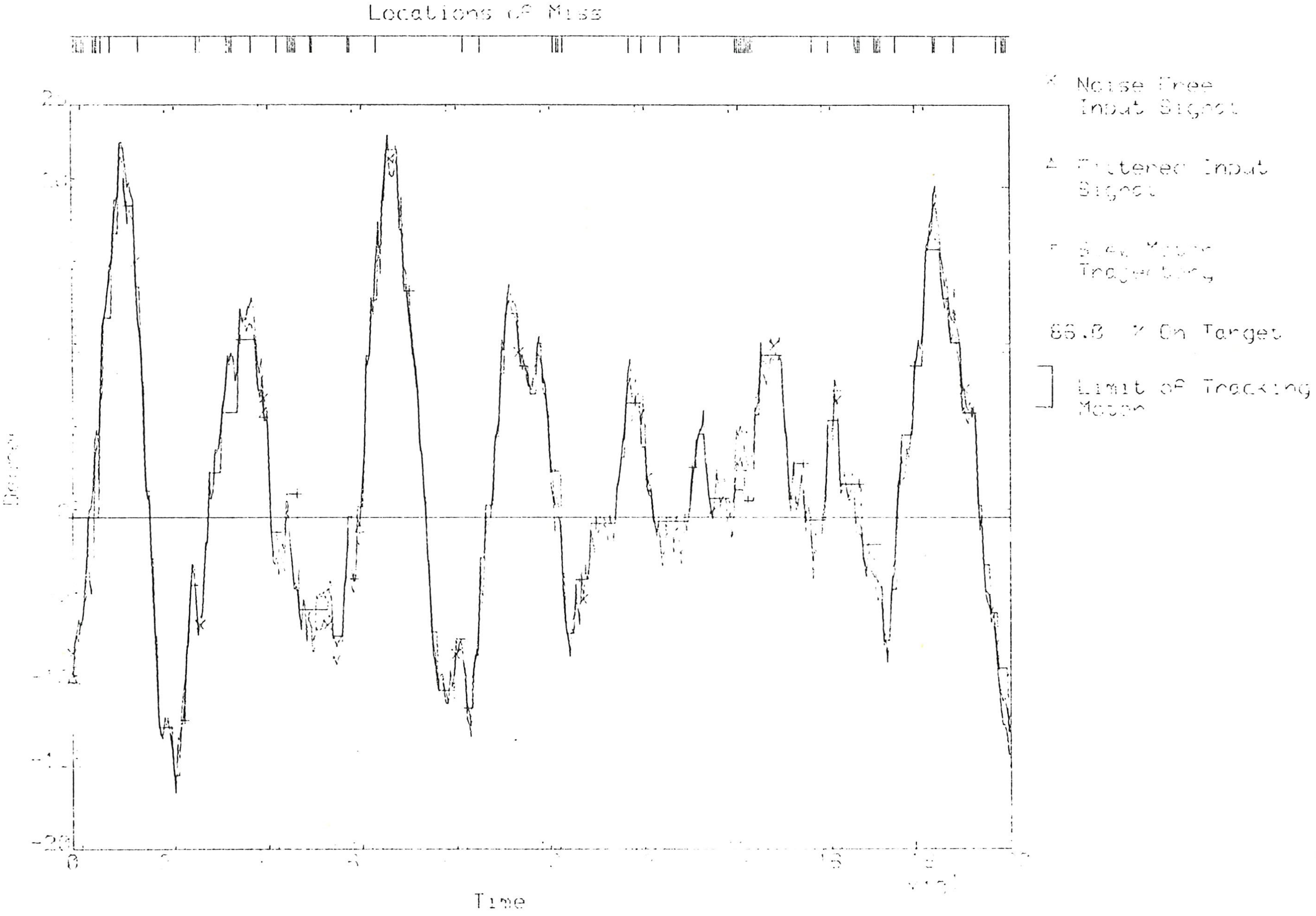


Figure C.4a History of simulated engagement

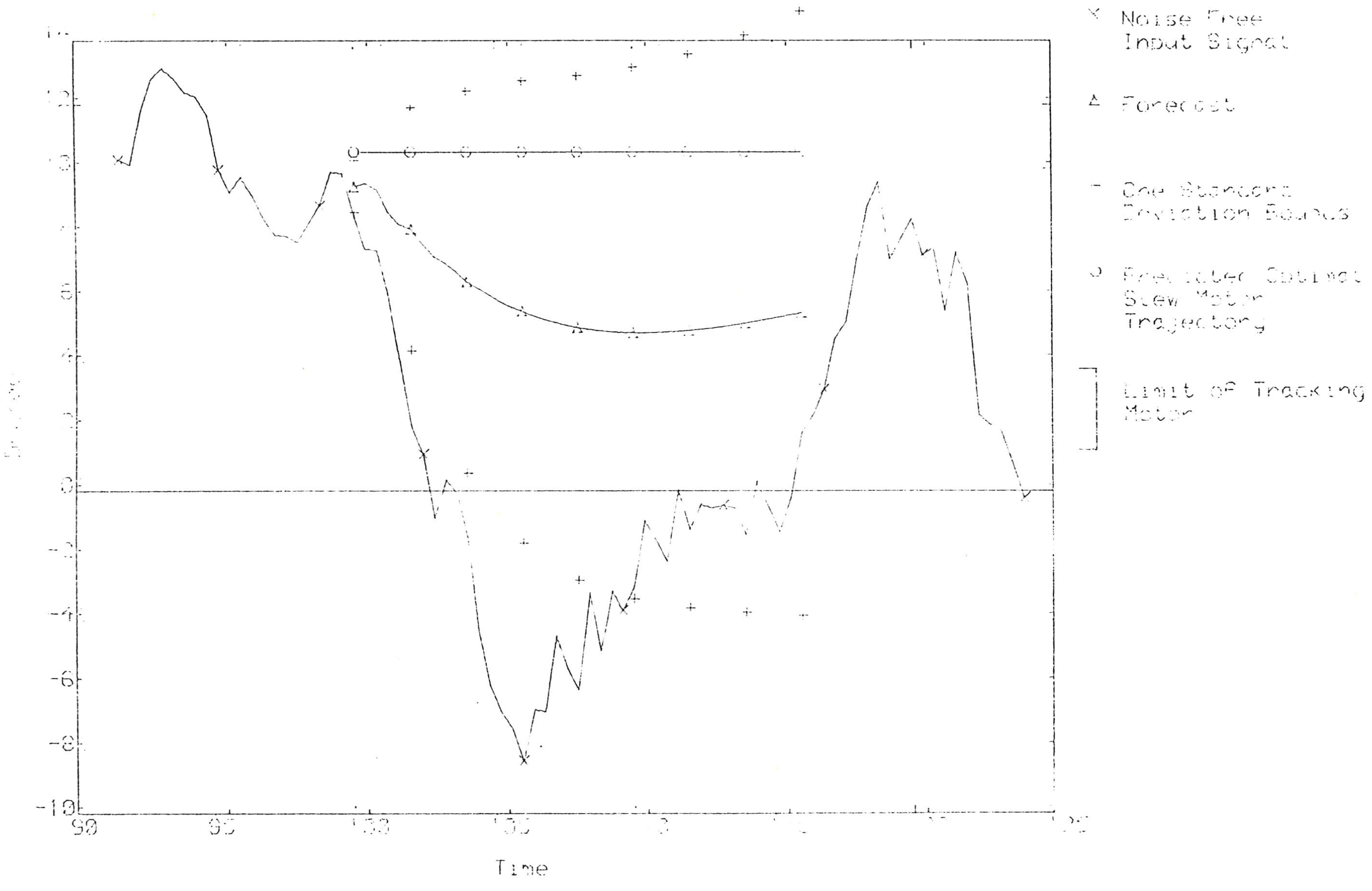


Figure C.4b Forecasts and predicted optimal slew motor trajectory at time interval 100

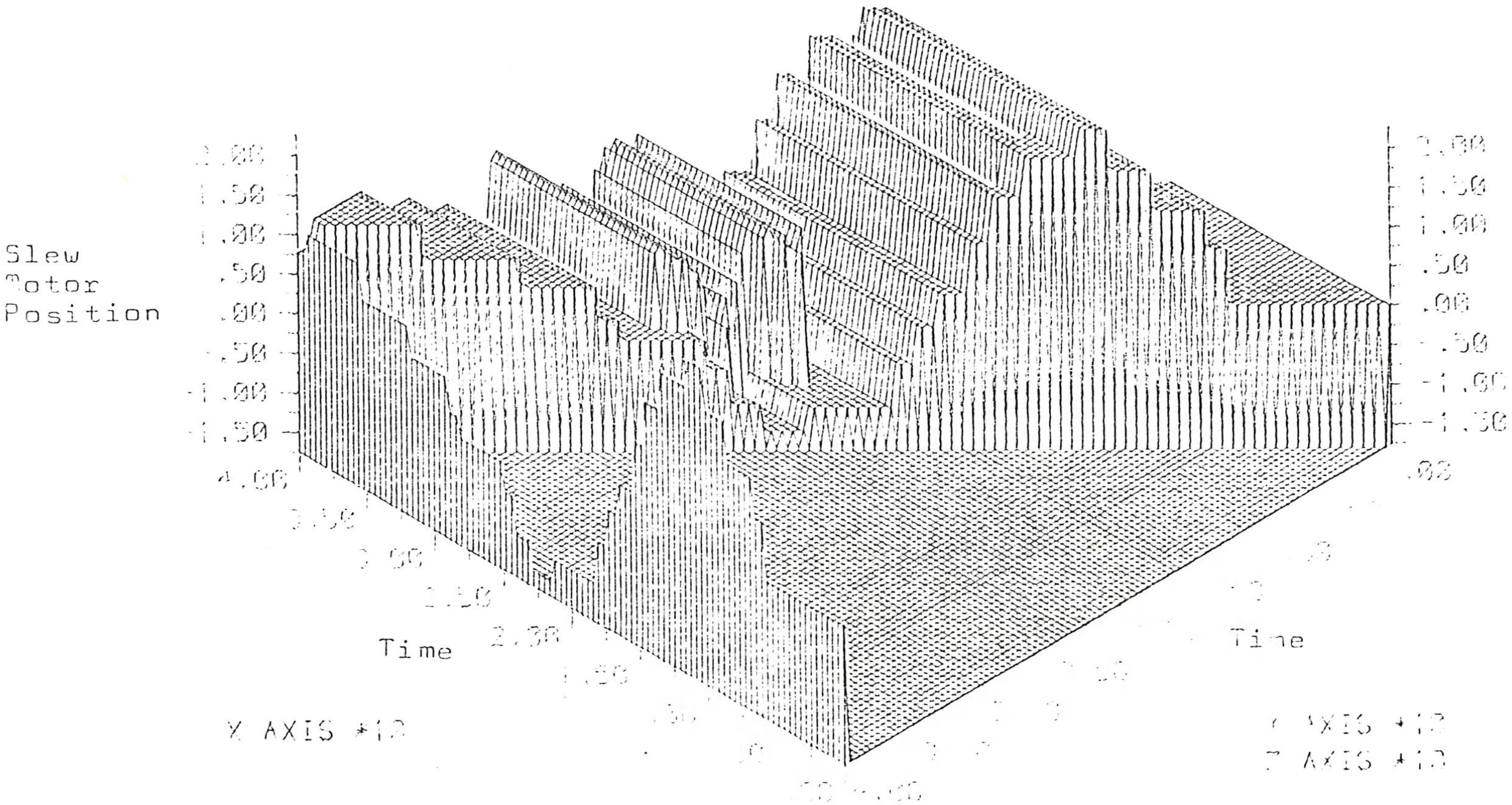


Figure C.4c Predicted optimal slew motor trajectories at the first 40 intervals

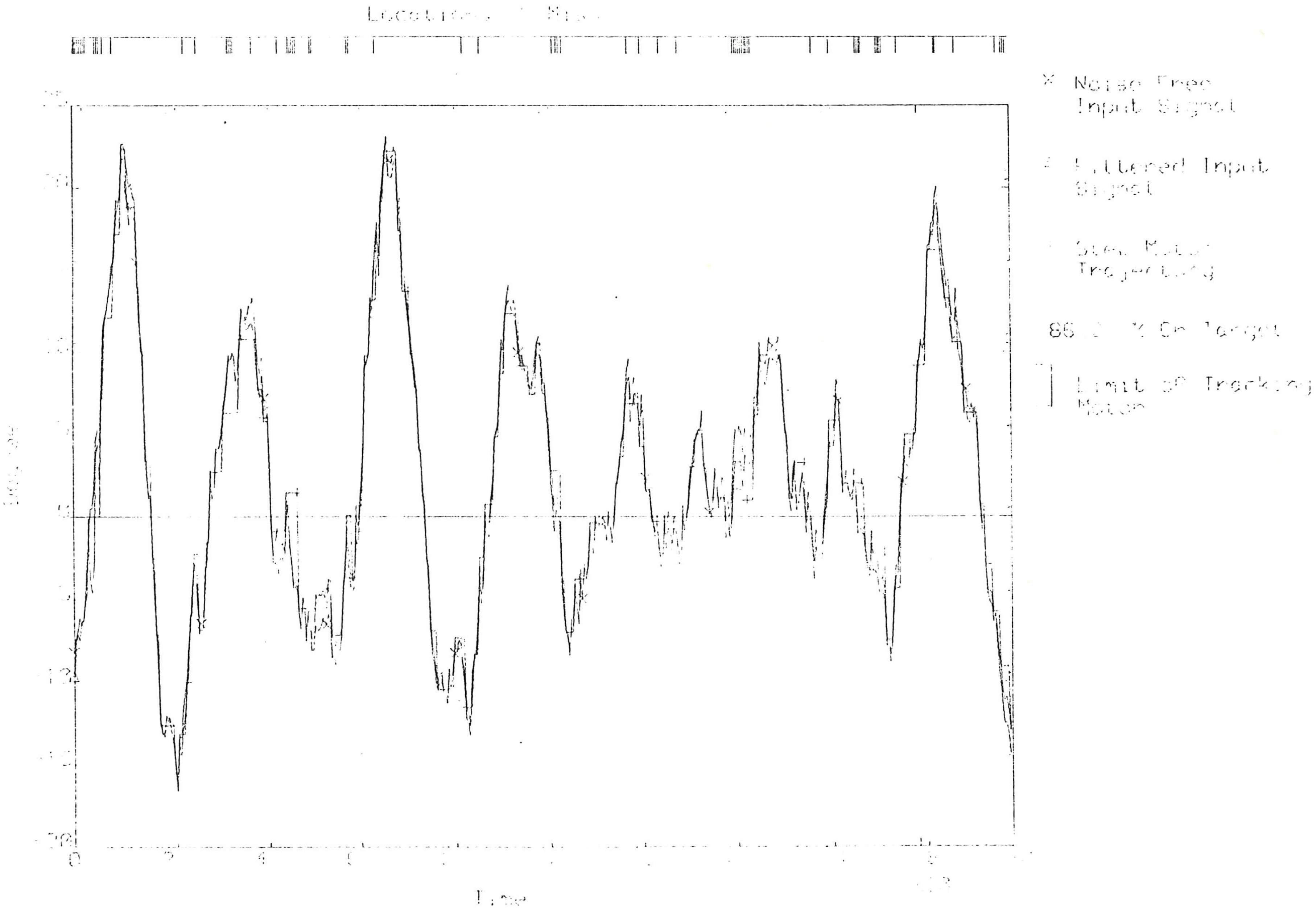


Figure C.5a History of simulated engagement

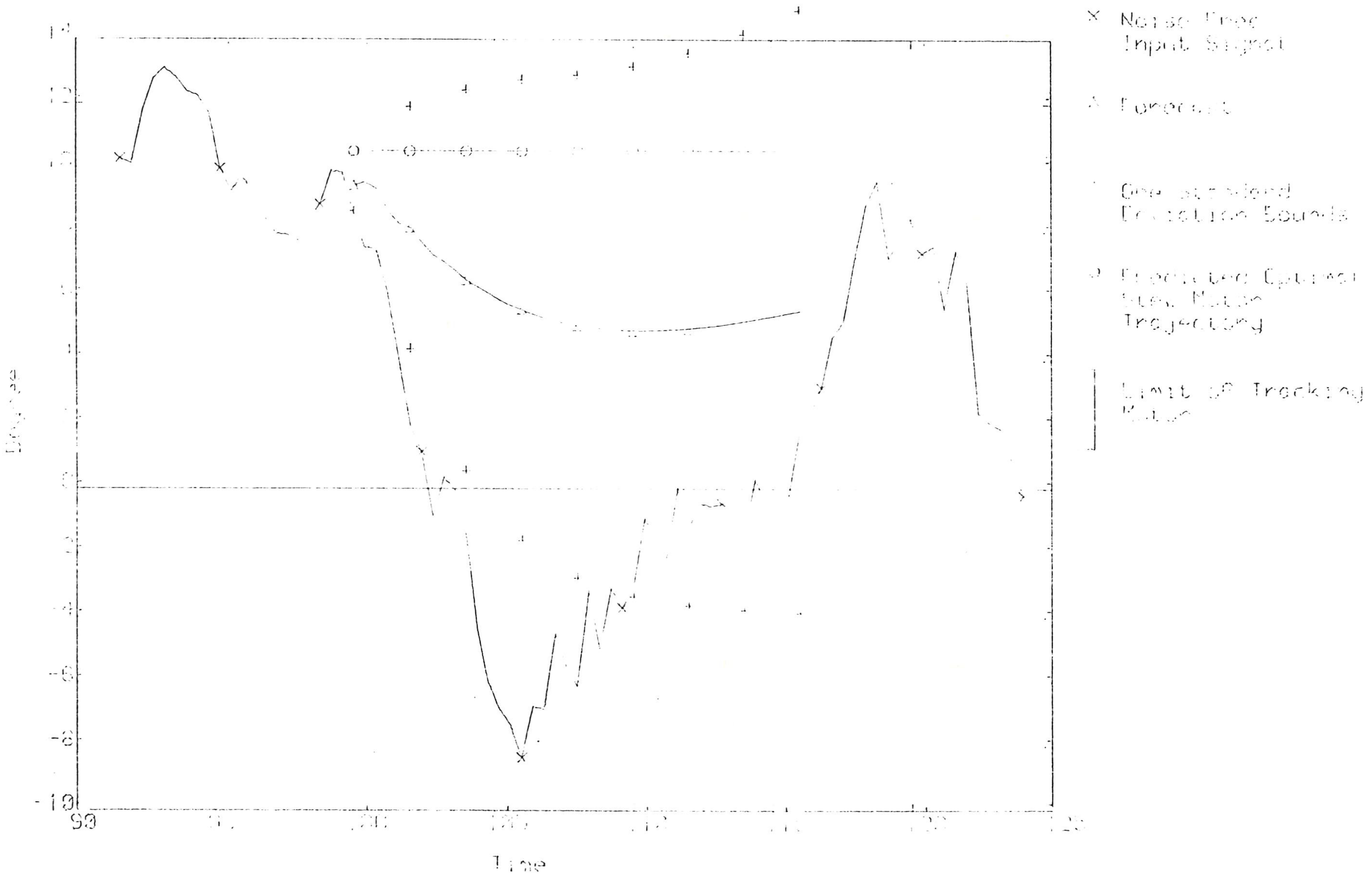


Figure C.5b Forecasts and predicted optimal slew motor trajectory at time interval 100

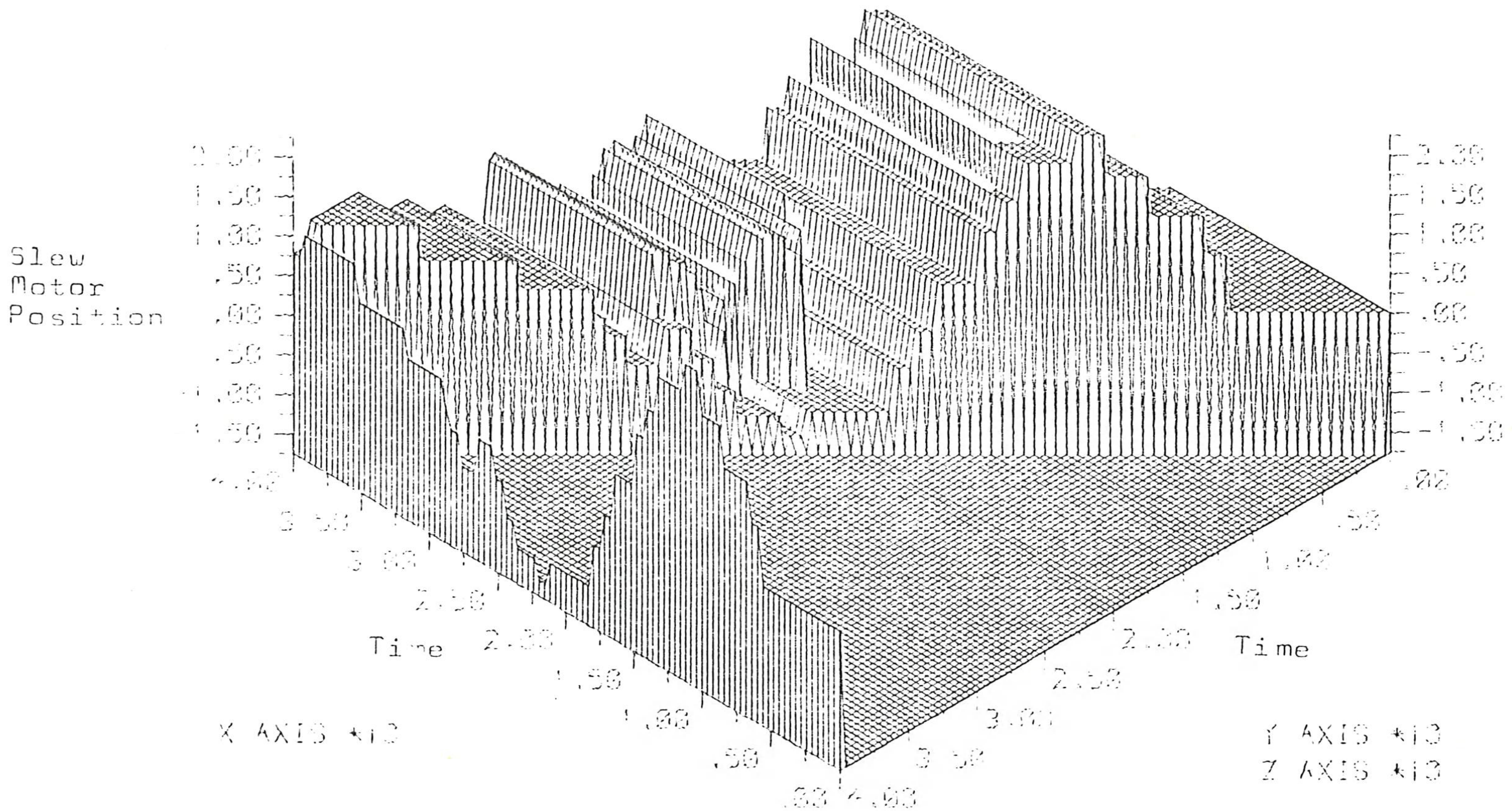


Figure C.5c Predicted optimal slew motor trajectories at the first 40 intervals

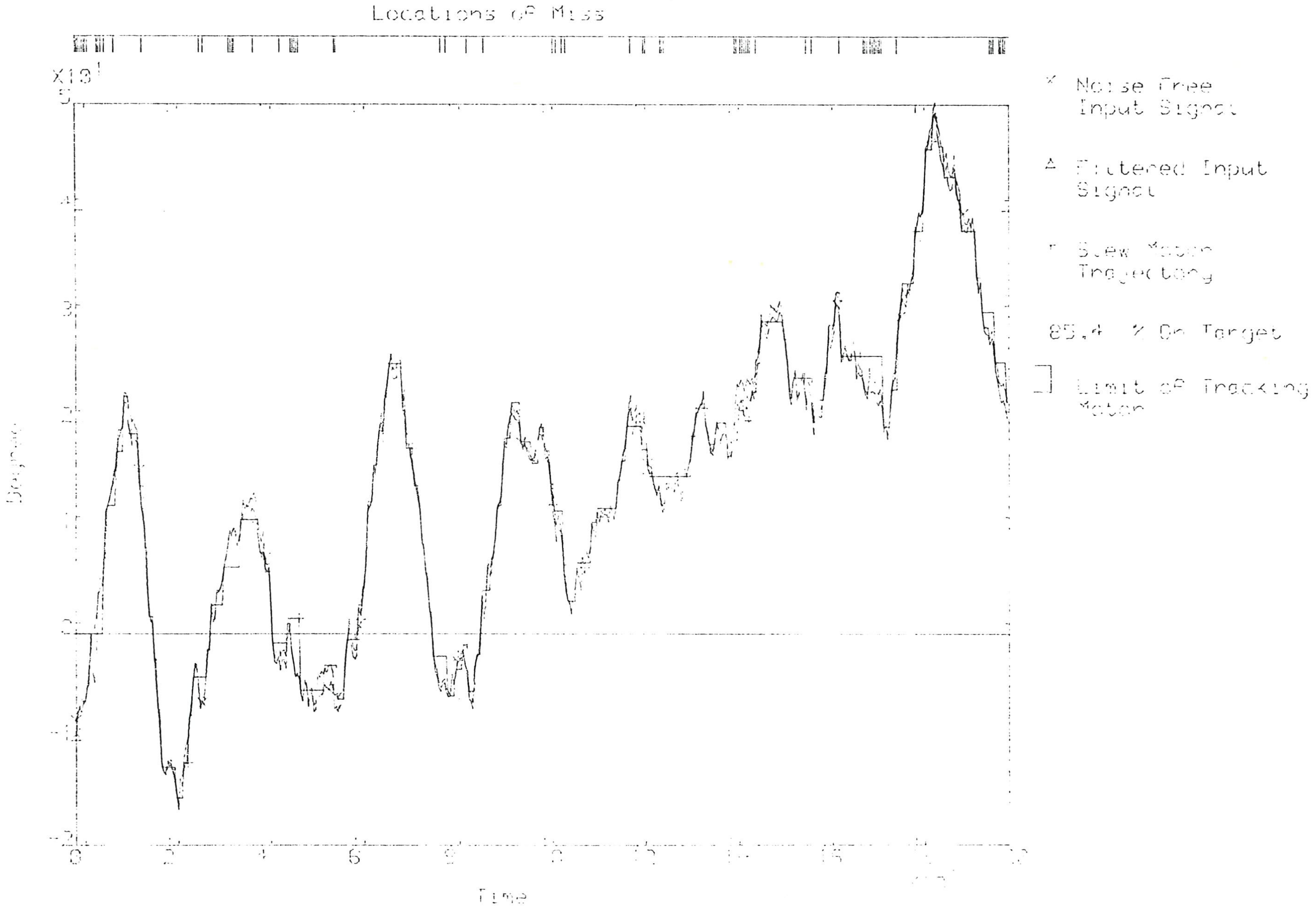


Figure C.6a History of simulated engagement

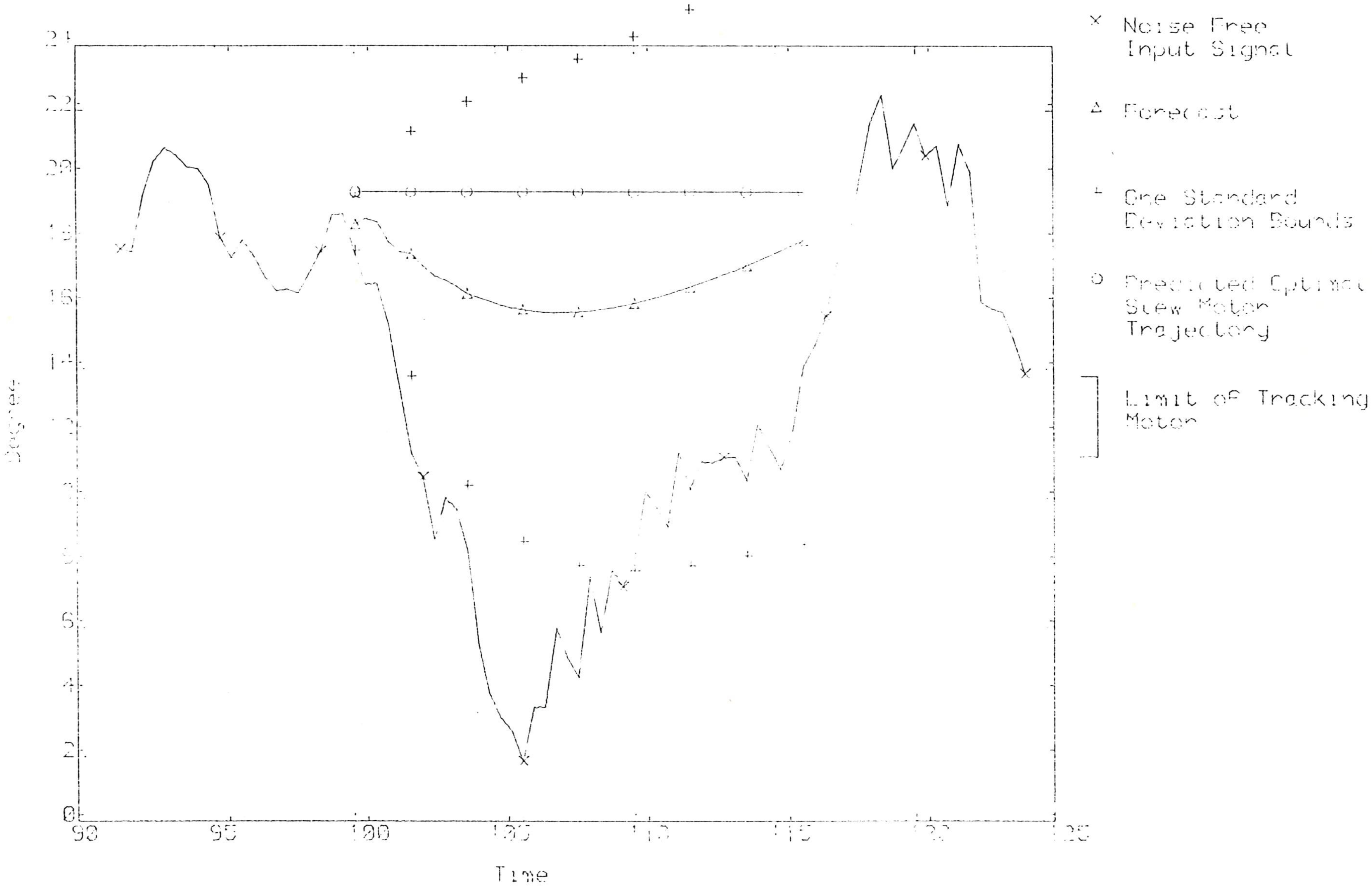


Figure C.6b Forecasts and predicted optimal slew motor trajectory at time interval 100

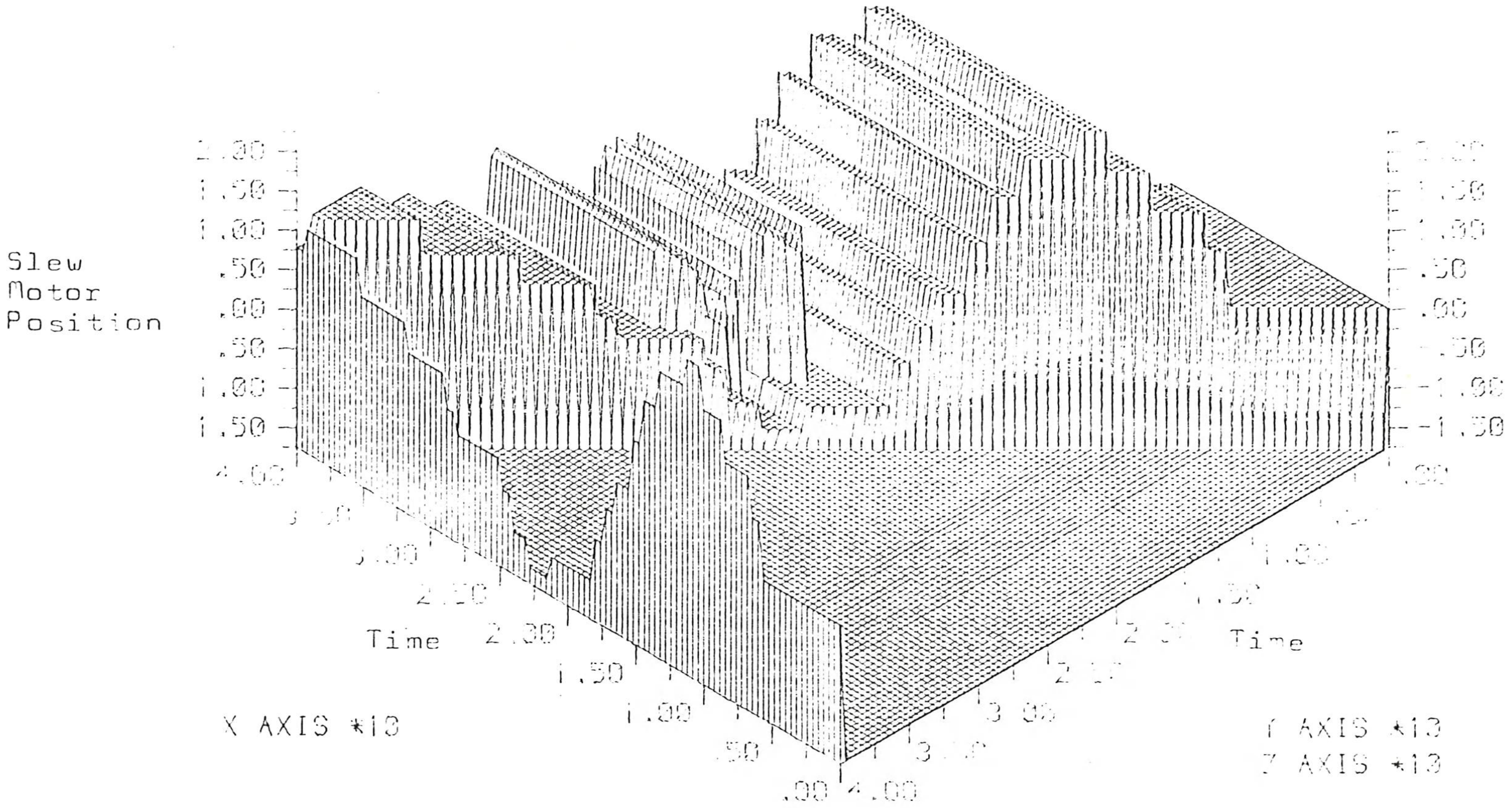


Figure C.6c Predicted optimal slew motor trajectories at the first 40 intervals

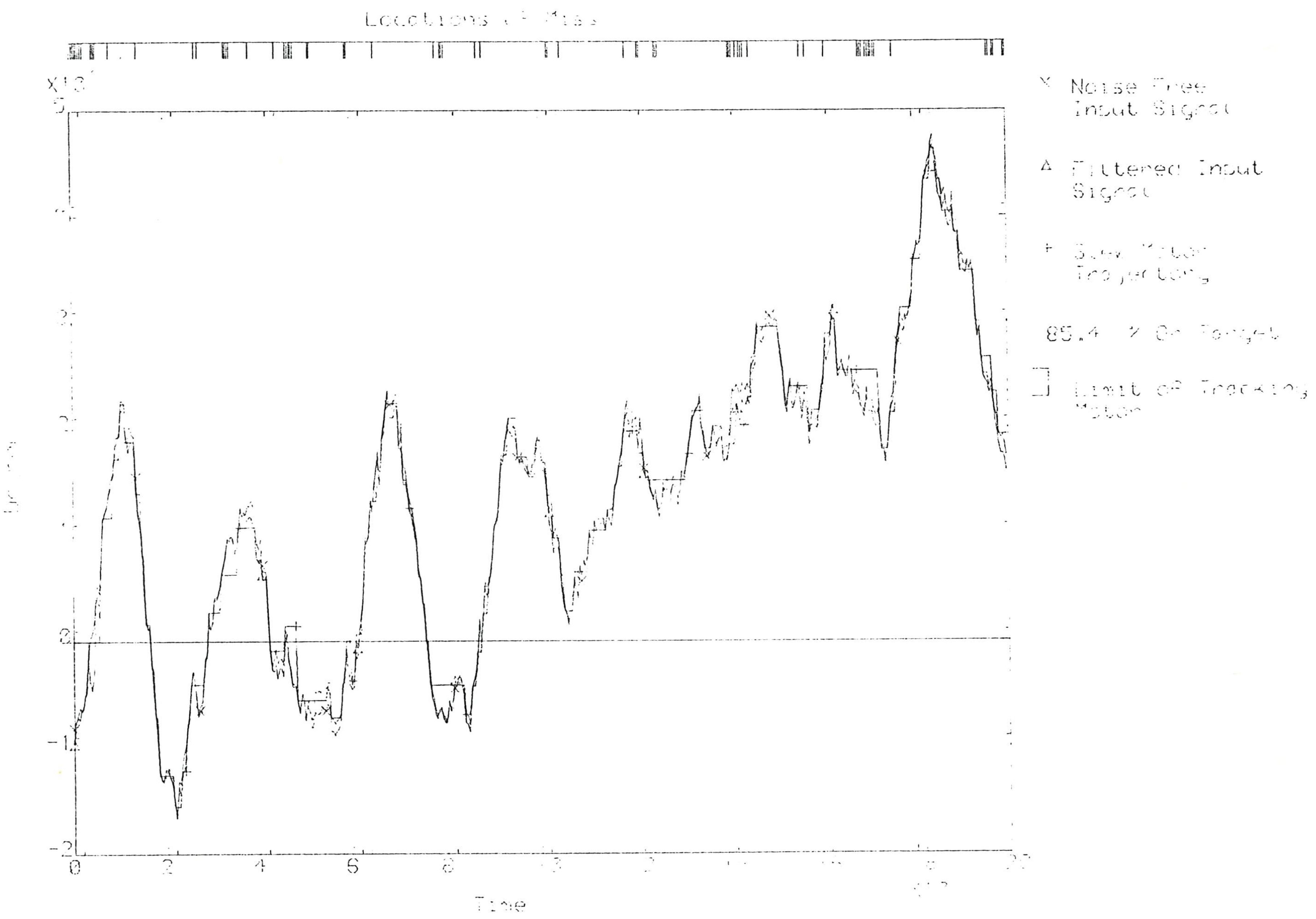


Figure C.7a History of simulated engagement

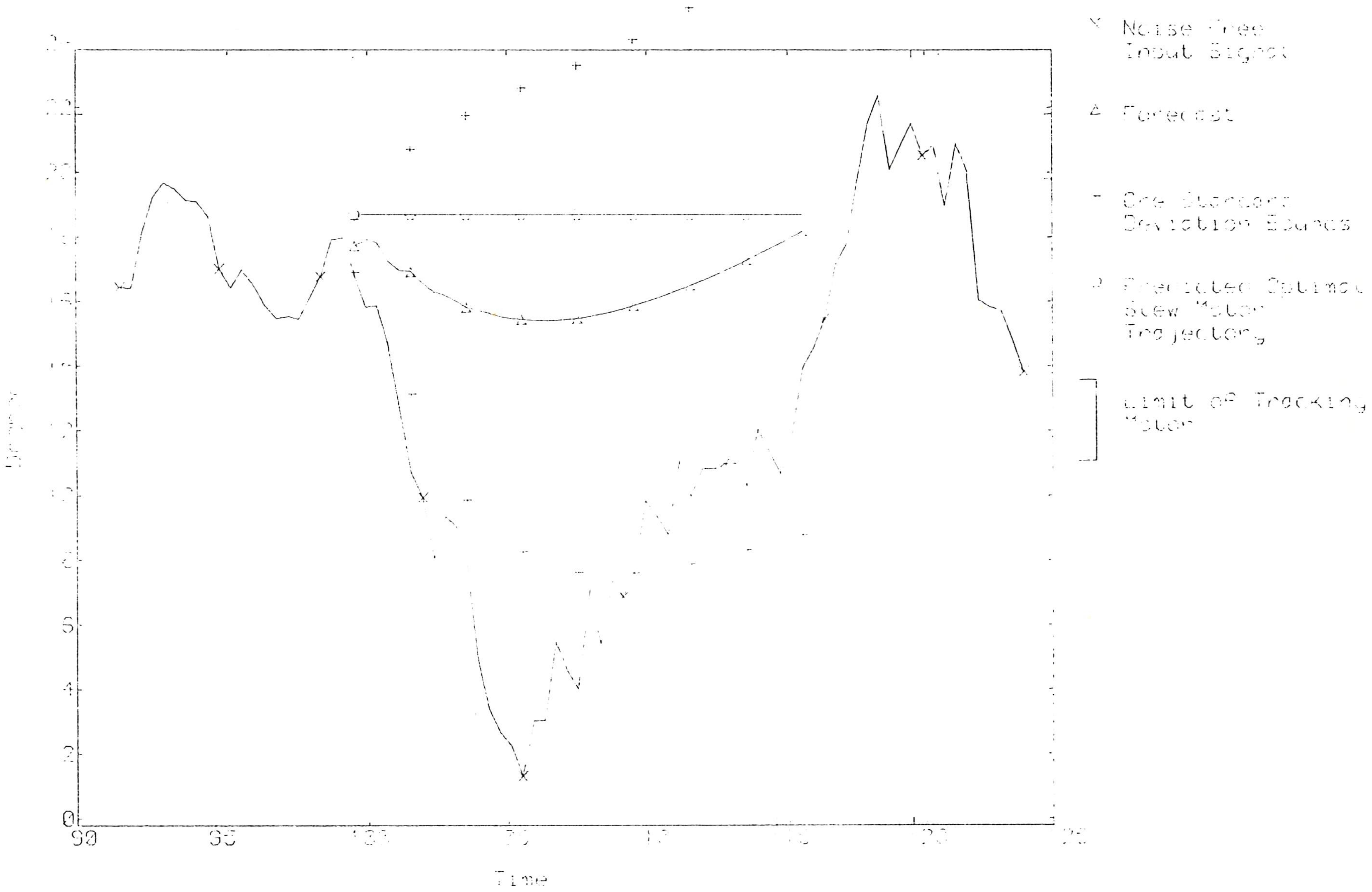


Figure C.7b Forecasts and predicted optimal slew motor trajectory at time intervals 100

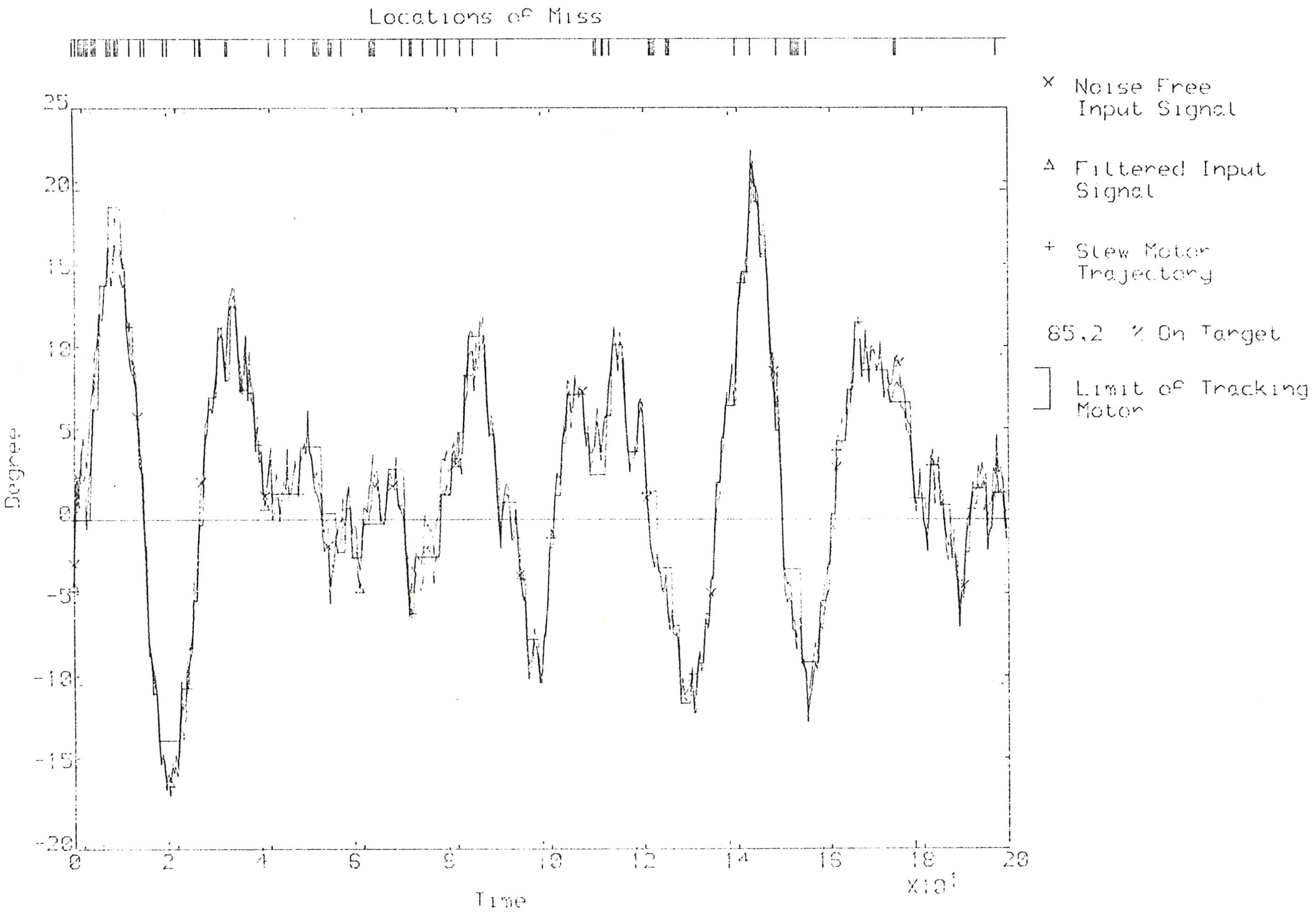


Figure C.8a History of simulated engagement

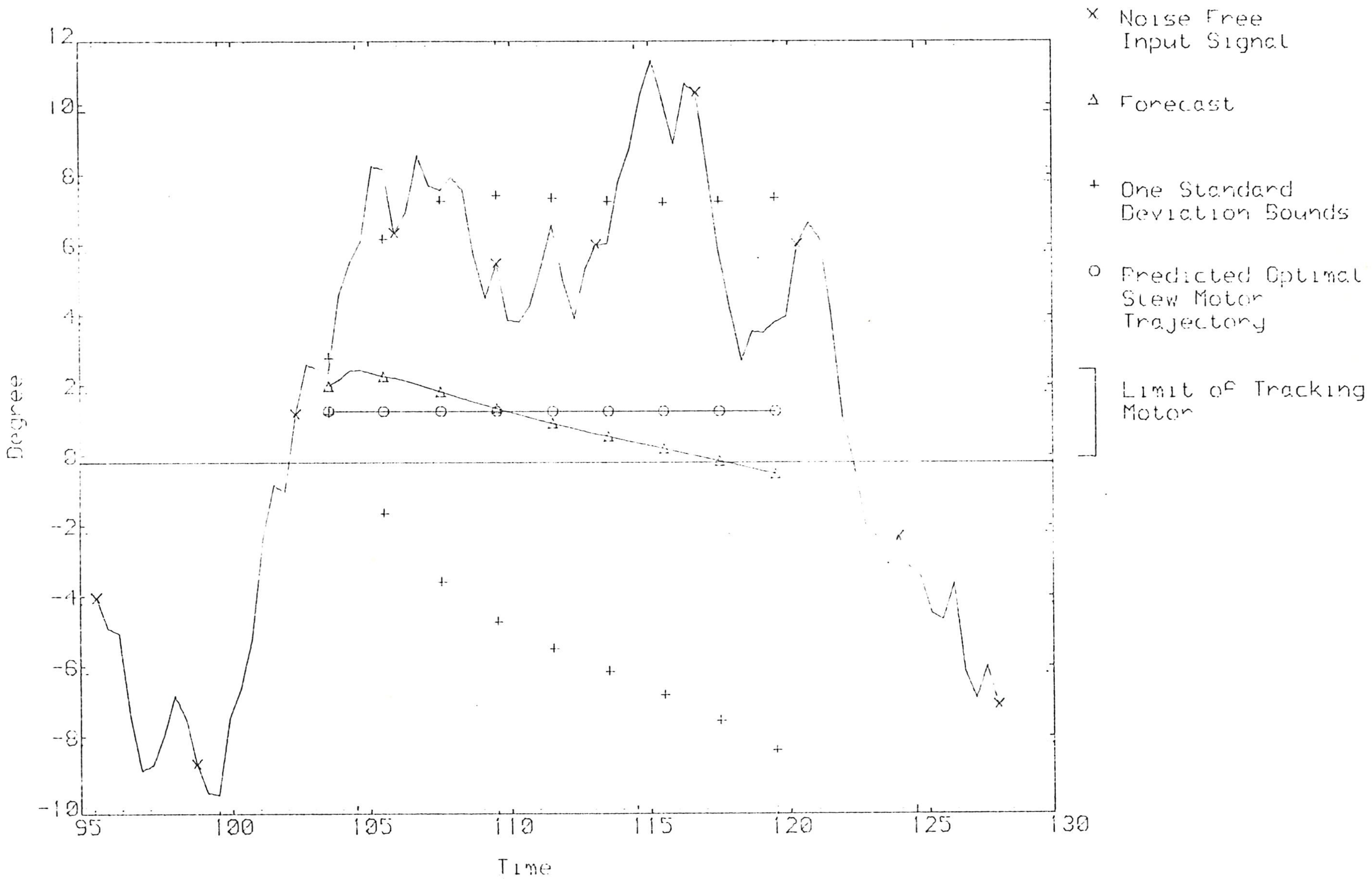


Figure C.8b Forecasts and predicted optimal slew motor trajectory at time intervals 100

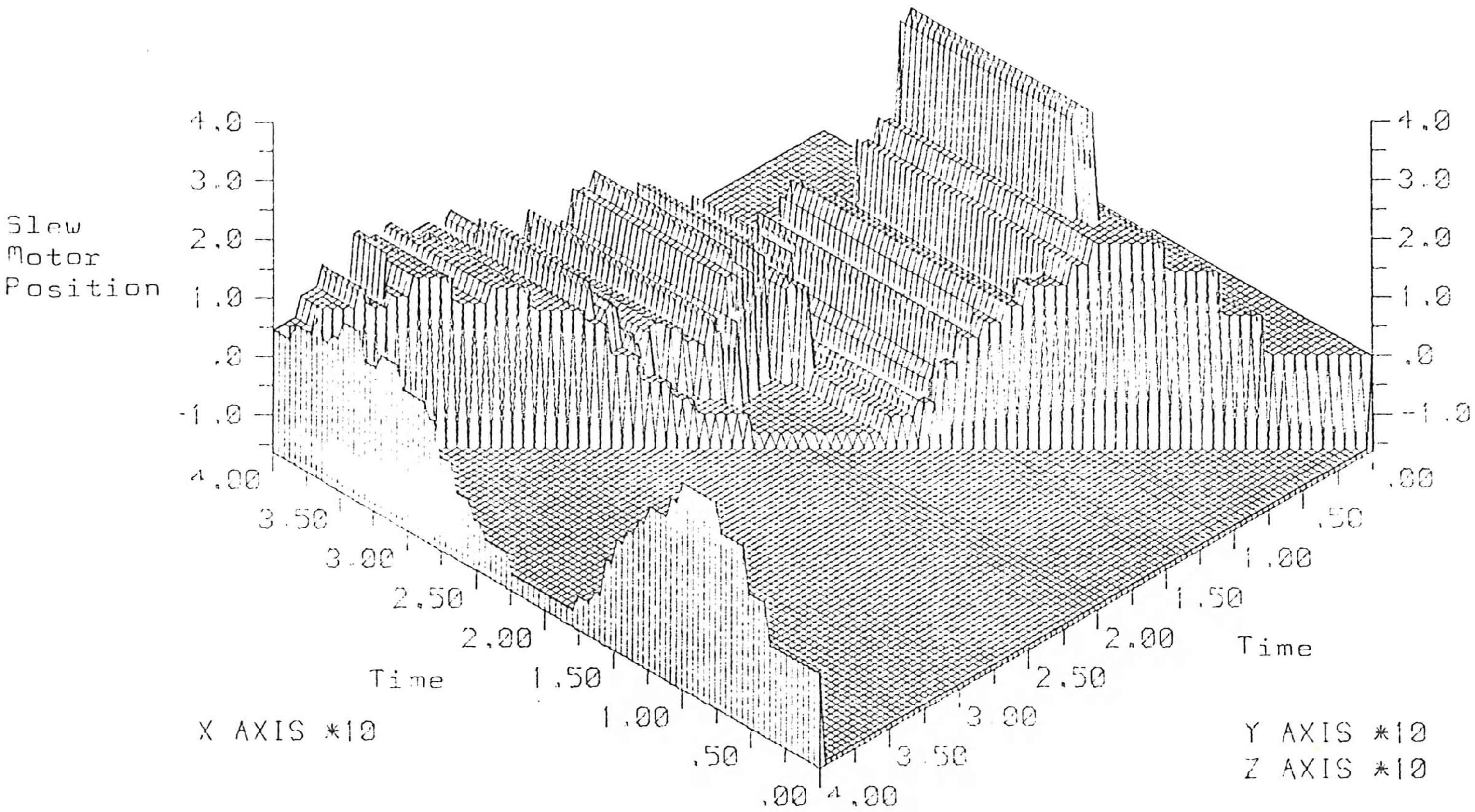


Figure C.8c Predicted optimal slew motor trajectories at the first 40 intervals

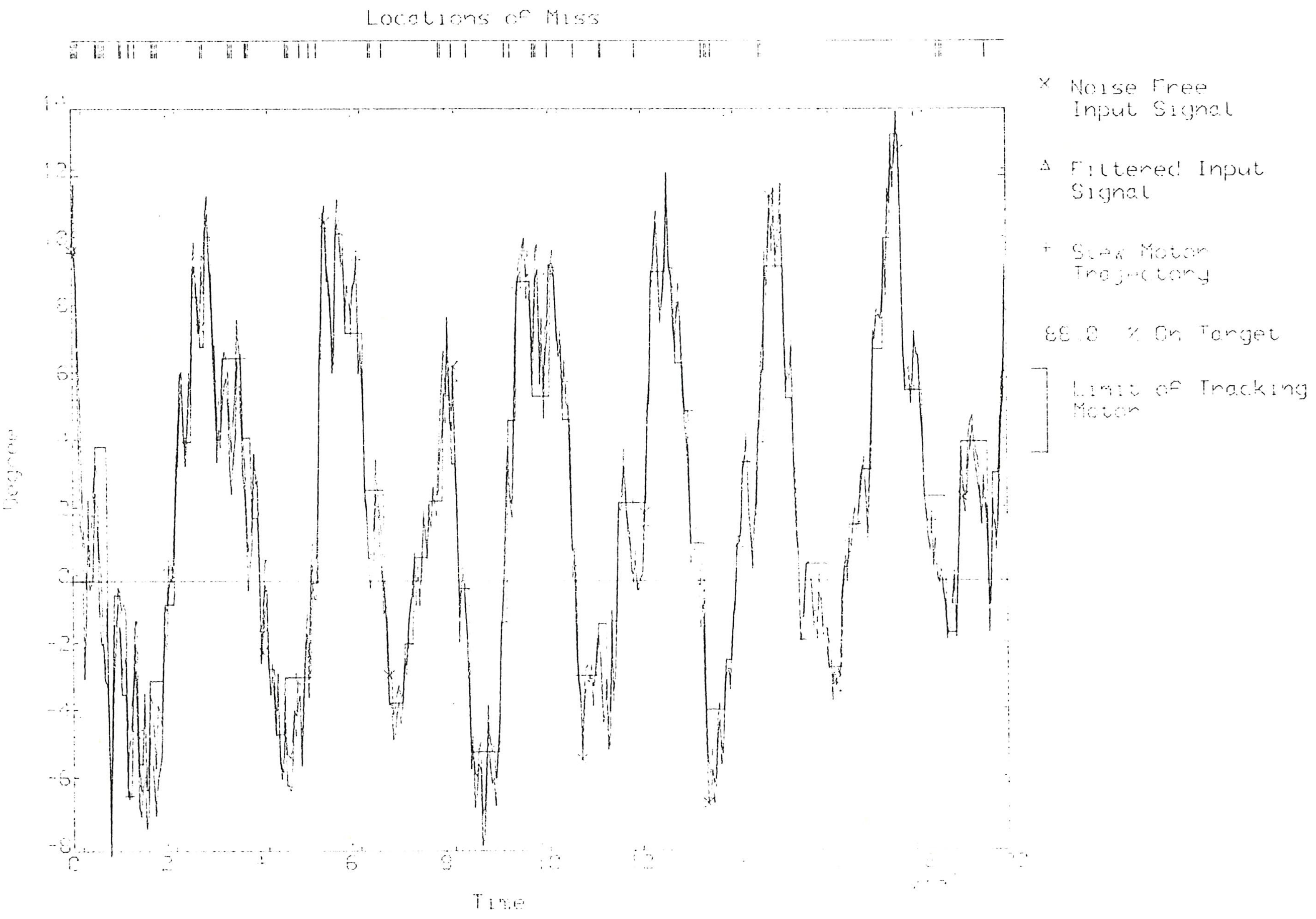


Figure C.9a History of simulated engagement

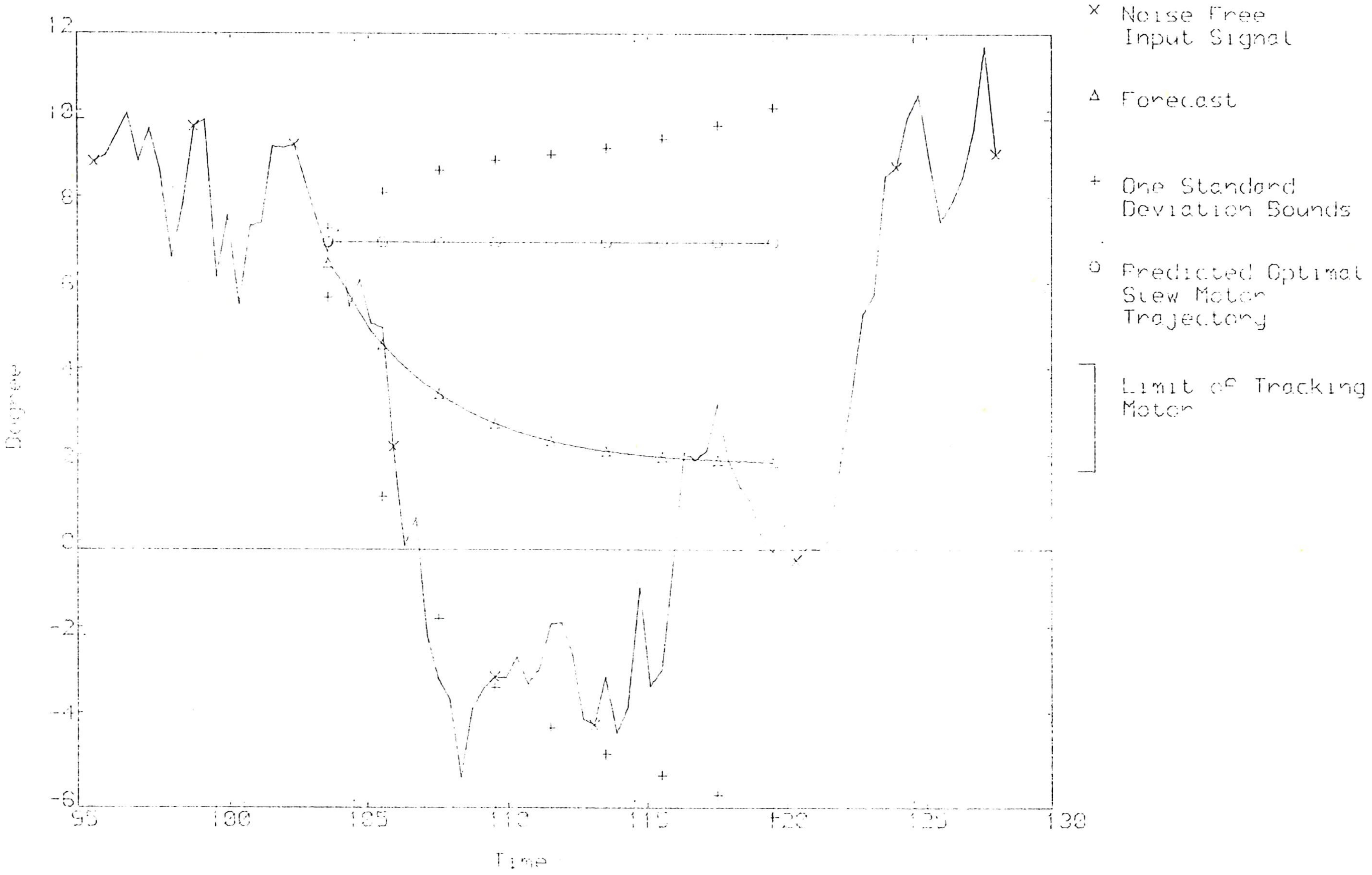


Figure C.9b Forecasts and predicted optimal slew motor trajectory at time interval 100

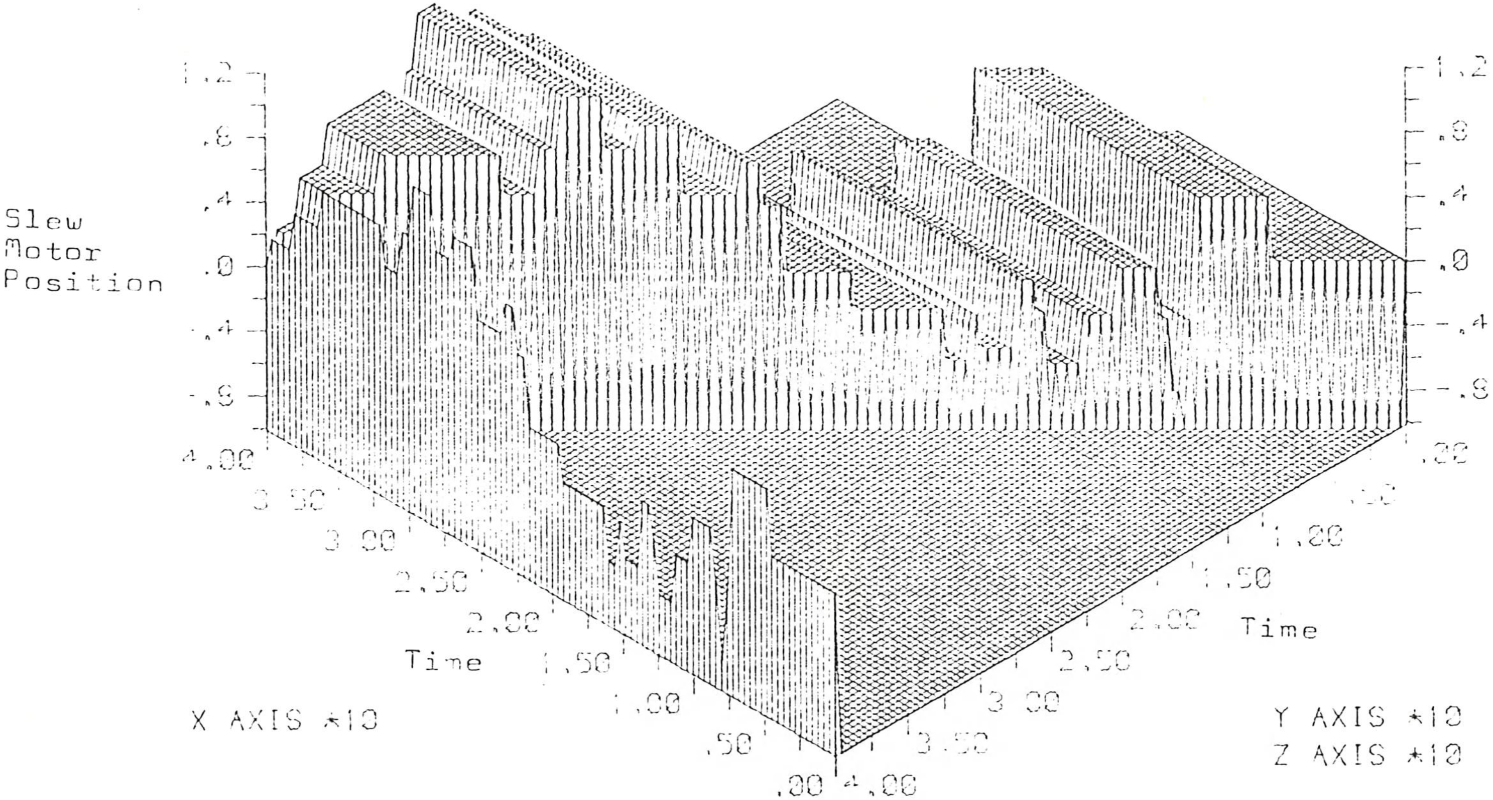


Figure C.9c Predicted optimal slew motor trajectories at the first 40 intervals

BIBLIOGRAPHY

1) Andrews, A. A Square Root Formulation of the Kalman Covariance Equations, AIAA J., Vol 6, No. 6, pp 1165-1166, 1968.

2) Athans, M.; Falb. P.L. Optimal Control - An Introduction to the Theory and its Applications. McGraw-hill N.Y. 1966.

3) Bellantoni, J.F.; Dodge, K.W. A Square Root Formulation of the Kalman Schmidt Filter, AIAA J., Vol 5, No. 7, pp 1309 - 1314, 1967.

4) Bellman, R.E. Dynamic Programming. Princeton University Press, Princeton, New Jersey, 1957.

5) Bhagavan, B.K.; Polge, R.J. Performance of the g-h filter for Tracking Manoeuvring Targets. IEEE Trans. Aerosp. Electr. Syst. Vol AES-10, No. 6, pp 864-866, 1974.

6) Bierman, G.J. Fixed Memory Least Squares Filtering. IEEE Trans. Inf. Theory. Vol 21., No. 6, pp 690-692, 1975

7) Billings, S.A. Introduction to Kalman Filters, IEE Digest No. 1980/64, 1980.

8) Box, G.E.P.; Jenkins, G.M. Time Series Analysis - Forecasting and Control. Holden-Day, 1970.

9) Brunk, H.D. An Introduction to Mathematical Statistics. 2nd Edition, Blaisdell Publishing Co. N.Y. 1965.

10) Chan, Y.T.; Hu, A.G.C.; Plant, J.B. A Kalman Filter Based Tracking Scheme with Input Estimation. IEEE Trans. Aerosp. Electro. Syst. Vol AES-15, No. 2, pp 237-244, 1979.

11) Chatfield, C. Statistics for Technology. 2nd Edition, Chapman and Hall, 1978.

12) Daniel, C.J. High Bandwidth Positional Servo Systems, UWIST under-graduate report, 1979.

13) Derusso, P.M.; Roy, R.J.; Close, C.M. State Variables for Engineers. Wiley, N.Y. 1965.

14) Evans, R.J.; Hewett, C.R.; Barker, F. Radar System Design for Track-While-Scan, IEEE Trans. Aerosp. Electro. Syste., Vol AES-15, No. 1, pp 125-133, 1979.

15) Fiacco, A.V.; McCormick, G.P. Nonlinear Programming - Sequential Unconstrained Minimisation Techniques, Wiley N.Y. 1968.

16) Fletcher, R.; Reeves, C.M. Function Minimisation by Conjugate Gradients, Computer Journal, Vol 7, No. 2, pp 149-154, 1964.

17) Fletcher, R.; Powell, M.J.D., A Rapidly Convergent Desent Method for Minimisation, Computer Journal, Vol. 6, No. 2, pp 162-168, 1963.

18) Garnell, P. Guided Weapon Control Systems, 2nd Edition, Pergamon Press, Oxford, England, 1980.

19) Gersch, J.W. Estimation of the Autoregressive Parameters of a Mixed Autoregressive Moving-Average Time Series, IEEE Trans. on Automatic Control, Vol AC-15, pp 583-588, 1970.

20) Graupe, D. On Convergence of Least-Squares Identifiers of Autoregressive Models having Stable and Unstable Roots, IEEE Trans. on Automatic Control, Vol AC-25, No. 5, pp 999-1002, 1980.

21) Graupe, D.; Krance, D.J.; Moore, J.B. Identification of Autoregressive Moving-Average Parameters of Time Series, IEEE Trans. on Automatic Control, Vol AC-20, No. 1, pp 104-107, 1975.

22) Graupe, D.; Perl, J. Stochastic Approximation Algorithm for Identifying ARMA Process, Int. J. System. Sci. Vol 5, No. 11, pp 1025-1028, 1974.

23) Healey, M. Optimum State Estimation, S.R.C. Vacation Course Notes.

24) Jazwinski, A.H. Stochastic Processes and Filtering Theory, Academic Press, .N.Y. 1970.

25) Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems, Trans. ASME, J Basic Eng. Vol 82D, pp35-45, 1960.

26) Kalman, R.E.; Busy, R.S. New Results in Linear Filtering and Prediction Theory. Trans. ASME, J. Basic Eng., Vol 83D, pp 95-108, 1961.

27) Kashyop, R.L. Estimation of Parameters in a Partially whitened representation of a Stochastic Process, IEEE Trans. on Automatic Control, Vol AC-19, No. 1, pp 13-21, 1974.

28) Krause, D.J.; Graupe, D. Identification of Autoregressive Moving-Average Predictor Models, Proc. 2nd Symp. Nonlinear Estimation Theory, San Diego, Calif. 1971.

29) Lee, R.C.K. Optimal Estimation,Identification and Control, Research Monograph No. 28, The M.I.T. Press, Cambridge, Massach setts.

30) Leitmann, G. Optimisation Techniques with Applications to Aerospace Systems, Academic Press, N.Y. 1962.

31) McAulay, R.J.; Denlinger, E. A Decision-Directed Adaptive Tracker, IEEE Trans. Electr. Syste. Vol AES-9, No. 2, pp 229-236, 1973.

32) McGarty, T.P. Stochastic Systems and State Estimation, Wiley N.Y. 1974.

33) Mehra, R.K. On-line Identification of Linear Dynamic Systems with Applications to Kalman Filtering, IEEE Trans. on Automatic Control, Vol AC-16, No. 1, pp 12-21, 1971.

34) Morgan, D.R. A Target Trajectory Noise Model for Kalman Trackers. IEEE Trans. Aerosp. Electr. Syst. Vol AES-12, No. 3, pp 405-408, 1976.

35) Morrison, N. Introduction to Sequential Smoothing and Prediction. McGraw-Hill, N.Y. 1969.

36) Papoulis, A. Probability, Random Variables and Stochastic Processes, McGraw-Hill, N.Y. 1965.

37) Potter, J.E.; Fraser, D.C. A Formula for Updating the Determinant of the Covariance Matrix, AIAA J., Vol 5, No. 7, pp 1352-1354, 1967.

38) Rao, S.S. Optimisation Theory and Application, Wiley Eastern Ltd., New Delhi, 1979.

39) Rome, H.J. Finite Memory Batch Processing Smoother, IEEE Trans. Aerosp. Eletr. Syst. Vol AES-7, No. 5, pp 968-973, 1971.

40) Sage, A.P.; Melsa, J.L. Estimation Theory with Applications to communications and Control. McGraw-Hill, N.Y. 1971.

41) Schultz, D.G.; Melsa, J.L. State Functions and Linear Control Systems. McGraw-Hill, N.Y. 1967.

42) Seilly, A.H. Hellenoid Actuators - A New Concept in Extremely Fast Acting Solenoids. Lucas Eng. Review, Vol 1, No. 4, pp 112-120, 1979.

43) Singer, R.A. Estimating Optimal Tracking Filter Performance for Manned Manoeuvring Targets, IEEE Trans. Aerosp. Electro. Syst., Vol AES-6, No. 4, pp 473-483, 1970.

44) Singer, R.A.; Behnke, K.W. Real-Time Tracking Filter Evaluation and Selection for Tactical Applications, IEEE Trans. Aerosp. Electr. Syst., Vol AES-7, No. 1, 1971.

45) Speedy, C.B.; Brown, R.F.; Goodwin, G.C. Control Theory - Identification and Optimal Control. Oliver and Boyd, Edinburgh, 1970.

46) Srinath, M.D.; Rajasekaran, P.K. An Introduction to Statistical Signal Processing with Applications, Wiley, N.Y. 1979.

47) Van Tree, H.L. Detection, Estimation and Modulation Theory, Part 1. Wiley N.Y. 1968.

48) Young, P.C. An Instrumental Variable Method for Real-Time Identification of a Noisy Process, Automatica, Vol 6, pp 271-287, 1970.

49) Yule, G.U. On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's sunspot numbers, Phil. Trans. Vol A226 pp 267, 1927.