University of Jordan

Faculty of Graduate Studies

COMPUTER SIMULATION OF ADAPTIVE CONTROL FOR THERMOPLASTICS INJECTION MOLDING PROCESS

M.Sc. Thesis

By

Nash'at Kamal Abdel-Jalil

SUPERVISORS

Dr. Dib Abu Fara and Dr. Yousef Al-Assaf

Submitted in Partial Fulfillment of the Requirements For the Degree of

Master of Science in Chemical Engineering.

Faculty of Graduate Studies,University of Jordan.

Amman , Jordan

October,1992

The Examining Committee considers this thesis satisfactory and acceptable for the award of the Degree of Master of Science in Chemical Engineering on October 13, 1992.

Dr. Dib Abu Fara : Chairman of the Committee

Chemical Engineering Department

University of Jordan

Dr. Yousef Al-Assaf : Member of the Committee

Industrial Engineering Department

University of Jordan

Dr. No'man Haimour : Member of the Committee

Chemical Engineering Department

University of Jordan

Dr. Sabah Al-Bermani : Member of the Committee

Industrial Engineering Department

University of Jordan

Dr. Naim M. Faqir : Member of the Committee

Chemical Engineering Department

University of Jordan

ملخص

لقد تم في هذا البحث استخدام نظام التحكم المتكيّف للتحكم في عملية تصنيع البلاستيك بالحقن وذلك باستخدام محاكاةالحاسوب . ولقد تم اختيار الضغط داخل قالب الحقن للتحكم به لما له من تأثير مباشر على الخواص النهائية للمنتج وتداخله مع معظم متغيرات عملية الحقن . ولقد تم في هذه الدراسة تطوير النماذج الرياضية المتوفرة لتغير الضغط داخل قالب الحقن لتشمل الخاصية غير الخطية له .

ولقد أظهرة نتائج نظام التحكم المتكيف تحسناً ملحوظاً عن نتائج أنظمة التحكم الاعتيادية.

ABSTRACT

An adaptive control technique, which is the Generalized Predictive Control (GPC), was applied for the control of the injection molding process through simulation. The cavity pressure in the filling stage was chosen as the controlled variable due to its direct effect on the final product quality and its interactions with most of the process variables. A model that represents the dynamic behaviour of cavity pressure related to the control valve which was developed in a previous study was used in the study. This model does not incorporate the nonlinearity and time-variation encountered in the process. Therefore, another model was identified using real input/output data.

The results of GPC were compared to that of the fixed- parameters PID controller. The results showed an improvement over the PID controller.

ACKNOWLEDGEMENTS

It is with deep pleasure I express my thanks to all who helped me during the preparation and completion of this work. Special thanks are indebted to my supervisors, Dr. Dib Abu Fara and Dr. Yousef Al-Assaf , whom without their support, encouragement, and advice, my work could have been more difficult. Also I highly appreciate the patient and support of my family during my long preoccupation with this research, I should not forget the support given to me throughout this work by all my friends, especially; Husam Rasmi, Wae'l Nasrallah, Mohammed Abu-Samak, Monaf Najim Al-din and Mohammed Hammash.

Contents

List of Figures

NOMENCLATURE

A, B, C : Polynomials in the CARIMA model

$\hat{E}$: Vector containing $\hat{e}(t)$ values

E_j, F_j : polynomials in the Diophantine equation

$e(t)$: Error between predicted output and the set point

G : Matrix whose elements are g_i

G_j : Polynomial equals to $E_j B$

$\overline{G}$: Part of G_j associated with unknown terms

$\tilde{G}$: Part of G_j associated with known terms

h : Sampling interval

I : Identity matrix;

J : Value of a quadratic cost function

kd : Process time delay in samples

$K(t)$: Correction gain vector in RLS

K_1 : Ramping slope in the cavity pressure model

K_2 : Process gain in the cavity pressure model

n : Number of estimated parameters

$N1$: Minimum prediction horizon

$N2$: Maximum prediction horizon

NU : The control horizon

$P(t)$: The covariance matrix in the RLS algorithm

Pc : The cavity pressure

t : Time

T : Filtering polynomial

$u(t)$: Control signal input to the process

$\tilde{U}_{opt}$: Vector of future control increments

$w(t)$: Set point for the process

W : Vector of set points in future

$y(t)$: Measured output of the process

Y : Vector of y(t)

$\hat{y}(t)$: predicted output of the process

z^{-1} : Backward shift operator

β : Constant forgetting factor

$\beta(t)$: Variable forgetting factor

Δ : The difference operator$(1\text{-}z^{-1})$

$\hat{\epsilon}(t)$: The error between estimated and true parameters

$\hat{\theta}$: Vector of estimated parameters;

λ : Control weighting factor

$\zeta(t)$: Zero mean white noise

σ : Tuning knob for $\beta(t)$

τ : The process time constant

τ_d : Derivative time for the PID

τ_i : Integral time for the PID

$\phi(t)$: Vector containing input/output data in RLS

$\Phi(t)$: Matrix containing $\phi(t)$ vectors

Chapter 1

INTRODUCTION

1.1 Injection Molding Process

Injection Molding process is one of the most important polymer processing techniques, in which a polymeric material is converted into plastic finished articles [1]. A schematic diagram of an injection molding machine is shown in Figure 1.1 .

According to the nature of the process, it can be considered to be consisting of several consecutive stages, these stages are: melting, filling, packing and cooling. During melting the solid polymer is pushed from the feed hopper and moved forward by the rotating screw through the barrel and towards the nozzle. Electric heaters which surround the barrel are used to melt the polymer. Then the molten polymer is forced through the nozzle and the delivery system(sprue and runners) into the cavity by the application of pressure to the rear of the screw using the hydraulic system. After filling of the cavity, more polymer is forced into the cavity to densify it and to compensate for shrinkage in the part during cooling. After that the mold gate freezes-off and the polymer is cooled to solidification to the cavity shape. Then the article is ejected and the process is repeated [1].

A good indication of what is occurring inside the mold cavity is the cavity pressure measurement. Figure 1.2 shows the cavity pressure profile as a function of time. The

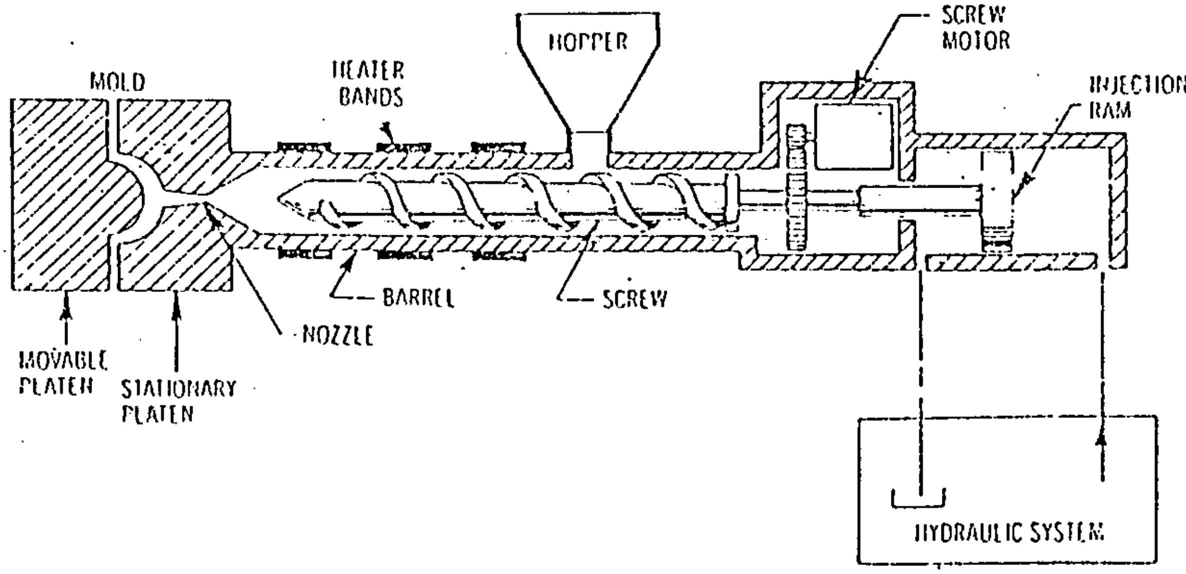

Figure 1.1: A Schematic Diagram of a Reciprocating Screw Injection Molding Machine.

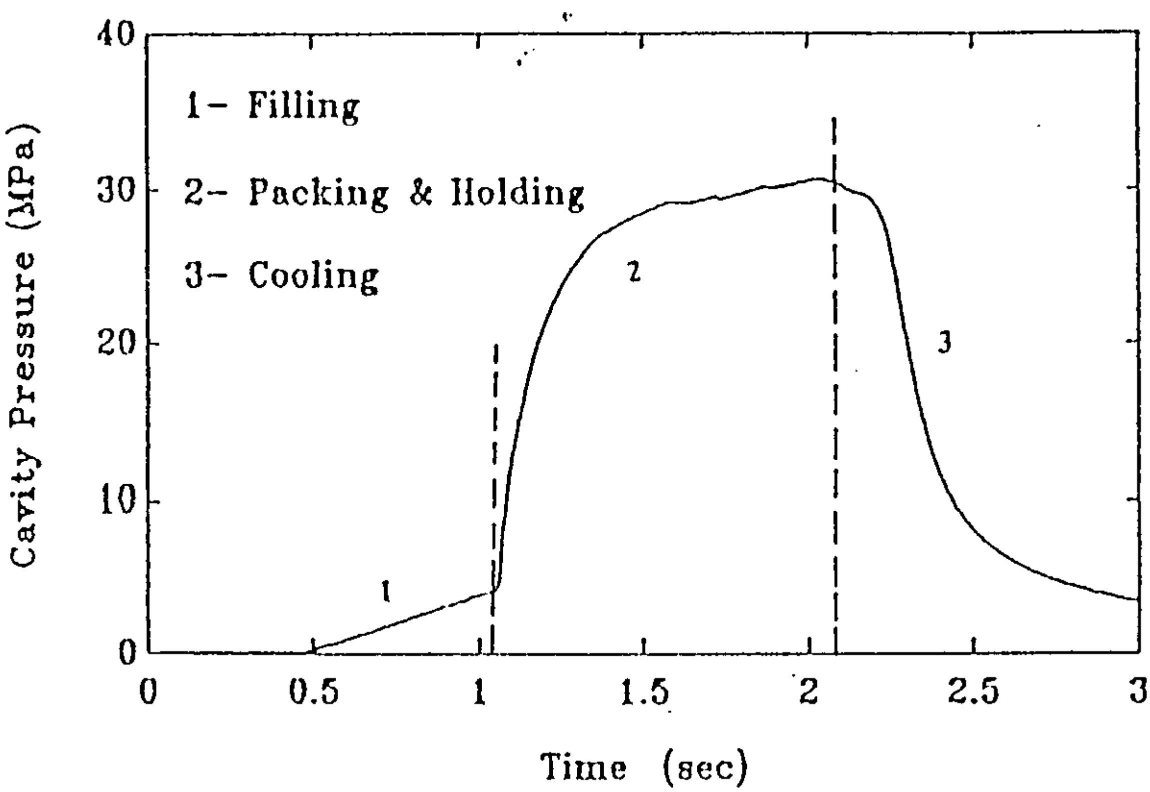

Figure 1.2: Cavity Pressure Profile as a Function of Time. of an Injection Molding Cycle[2].

stages of the process are apparent from the figure.

The quality of the product is specified by its physical and mechanical properties. Each of the above stages and the corresponding variables affect the final product properties [3, 4, 5]. The process variables can be classified as [6]:

1. Input variables: these are the machine input variables such as the barrel temperature and injection speed. External disturbances and material properties are considered as input variables too.

2. Process state variables or processing variables: which are the physical properties of the melt as a function of time, such as melt temperature and pressure.

3. Output variables: which are the physical and mechanical properties of the product.

However, these variables are highly interdependent, some of these variables affect the final product quality more than the others. Cavity gate pressure is one of the most significant variables in the process, due to its interactions with most of the process variables, and its direct effect on the product quality [2, 3, 6].

1.2 Control of the Process

Importance of plastics is increasing, since the plastic articles are used in many applications such as airplanes, houses, cars and others, replacing metals in many applications, and in many cases the product should be of very high quality. So, a control scheme should be applied to the process in order to produce consistent parts of required quality and with minimum losses in time, energy and material.

Closed-loop control based on measurement of the output variables is limited by sensing capabilities. The product characteristics cannot be measured directly. On the other hand, the machine variables are not directly related to the output variables, the relations are complex, not well established and interrupted by the processing variables [2].

Another alternative is the closed-loop control of the processing variables. These variables are directly related to the output variables, so more adequate control of the product quality will be obtained.

Control of a processing variable requires the determination of the desired profile which corresponds to the required quality and a quantitative dynamic model that represents the time-varying and nonlinear nature of the process.

Several studies were done to produce dynamic models relating certain processing variable to the control valve opening as a manipulated variable [7, 8, 9]. Simple dynamic models for the cavity gate pressure during the filling and packing stages were obtained [2]. These models are of much importance for control studies, however they do not incorporate the time- variation and nonlinearity present in the process.

Conventional controllers with fixed settings such as the Proportional - Integral - Derivative (PID) controller can give satisfactory performance provided the settings have been properly tuned [10]. However, the process is characterized by nonlinear behaviour, time- variation, complex interactions between the process variables and the process is affected by noise and disturbances. All of these, make the initial setting and retuning of the PID controller a difficult and time- consuming task. Therefore, another control design procedure which can tune its parameters automatically is needed, such as Self-Tuning Control (STC).

A self-tuning control algorithm is a combination of a control design method and an identification technique [10, 11]. Figure 1.3 shows a block diagram of a general self-tuning

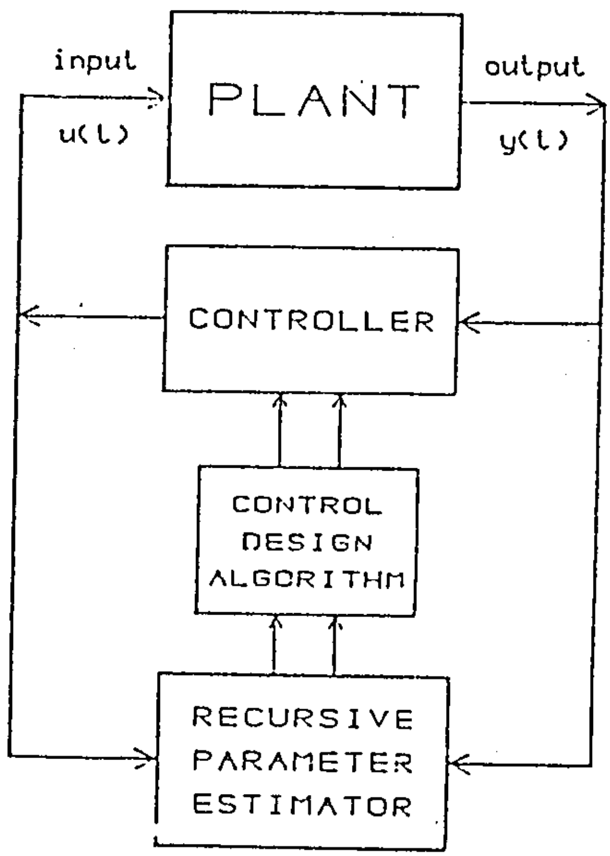

Figure 1.3: Structure of a General Self-Tuning Controller[11].

controller applied to a certain process. Based on the on-line measured input/output data the estimator updates the assumed dynamic model of the process, then the control algorithm calculates the control signal which satisfy a certain cost function such as reducing the error between the actual response and the desired one.

As will be discussed in the next chapter many types of STC are available [12], they could differ in the cost function to be minimized, the type of the identified model or they could have a prediction nature. As will be shown in the next chapter, a self-tuning controller, namely the Generalized Predictive Control (GPC) [13, 14, 16], was proved through various applications to be suitable for general self-tuning purposes. In this study, this controller will be used to control the cavity pressure in the filling stage of the injection molding process.

1.3 Objectives of the Study

The goal of this study is to investigate the application of adaptive control on the injection molding process. It is a step towards solving the complexities involved in the control of the process, such as the nonlinearity and the complex interactions between the process variables.

The study will be performed through simulation, with the following objectives:

1. The development of computer programs that implement the GPC algorithm along with an identification technique, and to investigate the various design parameters.
2. The evaluation of the use of the GPC algorithm in the control of the cavity gate pressure during the filling stage, and to compare the results obtained with those of the PID controller.

Chapter 2

LITERATURE REVIEW

2.1 Injection Molding

Several studies have been reported in the literature concerned with the modelling and control of injection molding process [16 → 27]. Most of these studies were of qualitative nature and did not treat the complex interaction of the process variables. Never the less, these studies comprised the base for more quantitative and detailed studies in the area of the dynamics and control of the injection molding process.

Sanchagrin [3] employed an experimental design method to determine the effect of several input variables on some of the output variables. The results were in time series form representing the relation between input and output parameters. One of the obtained relations is:

$$Pc(k) = 0.872Pc(k-1) + 6.797Ph(k) - 5.66Ph(k-1) \quad (2.1)$$

Where Pc is the maximum cavity pressure at the cycle k, and Ph is the holding pressure. He determined the most important parameters such as the ram velocity and the holding pressure, and examined closed loop control on them.

Shankar and Paul [4] developed a deterministic nonlinear lumped parameter dynamic model by combining analytical models of individual machine elements. The model was of seventh order, it was solved numerically and simulated on a digital computer. The model gave some deviations from the experimental results.

Wang *et al.* [28] have used an approach similar to that of Shankar and Paul, taking into consideration the transient behaviour for the development of a simulation model. They studied the effect of machine variables on ram velocity.

Parnaby and Eissa [29] used finite difference and lumped- parameter techniques to solve the transport equations. The aim was to produce a model reference for on-line computer control. Many assumptions were made to simplify the equations. The model consists of equations in polynomial form for various variables (such as the melt temperature and ram velocity) as function of time. Depending on this paper they [30] constructed a model reference computer control and studied the factors that affect the product quality in both injection and solidification stages. In the injection stage , the ram velocity was controlled using a digital PID controller , based on measurements of linear displacement of the ram . In the solidification stage the controlled variables were the volume shrinkage and the relaxation of elastic strain following cessation of the flow . The interactions between several process variables were studied . The experimental and calculated cavity pressure indicated a remarkable deviation . Also this is the case with the birefingence value (a measure of normal stresses). This decreased the model suitability to control the process.

Wang *et al.* [31] developed a transfer function model relating the ram velocity to the control valve opening. Recursive least squares method was used to estimate the parameters of the discrete version of the model. The discrete model was employed by Agrawal *et al.* [32], who designed a self-tuning regulator for control of Ram velocity by simulation. Generalized minimum variance controller was used, but no experimental evaluation was carried out.

Ricketson and Wang [5] constructed a simple steady state model relating part thickness to some other variables. The model is updated if thickness measurement error exceeds a tolerance value. Least squares method was used to estimate the model coefficients. Statistical experiment designs were used to form the process model, and it was regulated during the process.

Kamal *et al.* [7] developed dynamic models for nozzle and hydraulic pressure, using experimental transfer function approach. Deterministic and stochastic models were obtained relating the variations of hydraulic and nozzle pressure to changes in the control valve opening. Abu Fara *et al.* [33] used the dynamic models developed for hydraulic and nozzle pressure to evaluate the performance of some conventional controllers using computer simulation.

Abu Fara [2] employed the transfer function approach to develop dynamic models for some of the injection molding variables with emphasis on cavity pressure. The models describe the variations of the studied variable to changes in the control valve opening. The dynamic behaviour of cavity gate pressure in the filling and packing stages of the process was studied. In the filling stage both deterministic and stochastic modelling techniques were used to obtain the models.The deterministic model was best modelled by a first order plus time delay superimposed on a ramping component of the form:

$$Pc(t) = K_1 t + K_2(1 - e^{\frac{-(t-td)}{\tau}}) \tag{2.2}$$

where Pc is the cavity gate pressure, K_1 is the ramping slope, K_2 is the process gain, τ is the process time constant, t is time and td is the time delay. Converting this model to the digital form using zero order hold results in [2]:

$$\frac{Pc(t)}{u(t)} = \frac{b_0 z^{-1} - b_1 z^{-2} - b_2 z^{-3}}{1 - a_1 z^{-1} + a_2 z^{-2}} \tag{2.3}$$

Where

$a_1 = 1 + e^{\frac{-h}{\tau}}$

$a_2 = e^{\frac{-h}{\tau}}$

$b_0 = K_1 h$

$b_1 = K_1 h e^{\frac{-h}{\tau}} - K_2(1 - e^{\frac{-h}{\tau}})$

$b_2 = K_2(1 - e^{\frac{-h}{\tau}})$

The numerical values of these parameters were calculated using the following values of the process parameters [2]:

$K_1 = 14.0\ psi/\%.sec$

$K_2 = 3.34\ psi/\%$

$h = .01$, the sampling time in seconds

$\tau = .106$ sec

Thus the resulting model is:

$$\frac{Pc(t)}{u(t)} = \frac{0.14 z^{-1} + 0.17 z^{-2} - 0.3 z^{-3}}{1 - 1.91 z^{-1} + 0.91 z^{-2}} \tag{2.4}$$

In the packing stage, step tests were carried out to develop dynamic models of hydraulic, nozzle and cavity pressures. They were best modelled by a first order plus time delay models.

It was shown that the models are nonlinear and they depend mainly on the amount of polymer in the cavity in both filling and packing stages.

To evaluate the developed models, a simulation study was carried out using: PI, PID and Dahlin controllers. The controllers were tuned according to Integral of Time Weighted Absolute Error (ITAE) criterion. Then the controllers settings were used to control the process experimentally. Due to nonlinearity, a non-satisfactory performance was seen. Some improvement occurred, when a gain scheduling control strategy was employed, where the filling stage was divided into three regions and the packing stage into two.

Recently, Srinivasan *et al.* [34, 35] have employed the developed models by Abu Fara. They carried a simulation study to evaluate a Learning Control scheme applied to the filling stage. The control algorithm makes use of the cyclical nature of the process to improve the effectiveness of the closed- loop control. Error in the overall cycle is used for correction in the next one. The accuracy of the results was improved over that achieved using traditional controllers, but the developed controller did not treat the nonlinear and time-variation of the process.

The application of conventional control strategies for the injection molding process has achieved acceptable results with limitations. It hasn't rectified the problems exhibited in the dynamic response of the process. Thus, detailed stochastic identification and application of modern control theory to the process would be desirable. The rapidly growing availability of powerful and inexpensive computer hardware and software capabilities provide opportunities for achieving on-line process identification and adaptive control techniques, such as self-tuning control (STC).

2.2 Self-Tuning Control

Self-tuning control is a main approach to adaptive control. Its method for obtaining an automatic adjustment mechanism is to identify the system using measured input/output data and then the controller output is decided using the identified model [10].

Another approach is the model reference adaptive controller (MRAC). The main idea in this approach is the existance of a reference model that specify the desired performance of the process. Measured input/output data are used with the reference model to adjust the controller parameters. The overall aim is to force the actual output to correspond to the desired model output [12, 36].

From a design point of view, the self-tuning controllers can be classified into two categories: the first aims to minimize a quadratic cost function such as minimizing the error between the set point and the measured output, and the second attempts to locate the closed-loop poles at specified locations which is known as pole-placement algorithm [12].

The work of Astrom & Wittenmark [37] and Peterka [38] have introduced the minimum variance (MV) controller. The objective of this controller is to minimize the error between the predicted output and the desired set point profile. The MV suffers from some problems: lack of identifiability in closed-loop and instability with nonminimum phase processes due to cancellation of unstable process zeros. Some of these problems were tackled by several researchers [39, 40, 41]. A comprehensive outcome of the research on MV is the introduction of the Generalized Minimum Variance (GMV) controller [11, 42] . The cost function in this control algorithm introduces penalty on the control action as well as minimization of the error between measured output and the desired set point.

It incorporates a variety of design polynomials to ensure stability and correct tracking of the set point. The use of control weighting reduces the control effort and allows for stabilizing certain non-minimum phase processes [11, 42].

Unfortunately, MV and to a lesser extent GMV are sensitive to the choice of the dead time in the process. If the dead time is incorrectly chosen or it varies with time, then the instability of the controller will be expected[10, 16].

Parallel developments to those discussed above have taken place in Pole-placement control [43]. This algorithm can successfully handle a varying dead time or non-minimum phase systems. The incorporation of a desired pole-location allows model-reference like objectives to be reached by the closed-loop.

However, this controller suffers from some problems too. The algorithm may fail if there are model/process mismatch (i.e. it depends on the correct choice of the model order).

A further development in self-tuning control is the Generalized Predictive Control (GPC) [13, 14]. The algorithm predicts a series of future outputs and decide a series of controls at each sample to set the actual output at these samples to the desired set point. A Controlled -Auto-Regressive-Integrated-Moving-Average(CARIMA) model is used in the controller design which introduces an integral effect, so the offsets in the response are eliminated. In addition to that the algorithm contains a lot of design parameters (knobs), upon their choice the performance of the controller can be altered to meet the application control specifications.

In contrast to the other mentioned algorithms, the GPC can be robust even with varying process dead time and model order. In addition to the capability to control non-minimum phase processes [16].

GPC was applied to various practical applications: to a cement grinding mill [44], to a spray drying tower [45] which uses a multiloop version of GPC and to a compliant robot arm [46]. These applications have shown that GPC is an attractive contender for general self-tuning applications, especially when the process is non-linear and time-variant, which is a feature of the injection molding process.

All of the above self-tuning methods require a parameter estimation algorithm to provide the controller with on-line estimates. One of the on- line parameter estimation techniques is the Recursive Least Squares (RLS) algorithm. The estimated parameters are used in controller calculations as if they were exact, leading to a certainty equivalent adaptive controller [12]. Needless to say that basic RLS technique has to be modified in order to be used for nonlinear and time-varying processes. These modifications with simulation examples will be discussed in the next chapter.

Chapter 3

PARAMETER ESTIMATION

To control a process, dynamic models which describe the behaviour of the process are needed. Usually these models are not known. Methods for obtaining the models can be divided into theoritical and experimental. Theoritical models are obtained by using transport equations together with the necessary equations of state and constitutive relations. However, the resulting differential equations are complex and demand a large computational effort [2, 7]. Experimental modelling, called identification or parameter estimation, can be classified as off-line or on-line. The first is used for a complete set (batch) of data using a fitting criterion. The second fits a dynamic model of an assumed structure using running input/output data and a recursive criterion. On-line identification is a basic step in self-tuning control. Its main function is to provide parameter estimates of the process model for the controller at each time instant. A locally linearized model is identified, where changes in the dynamics are transformed into parameter changes.

A simple and good on-line identification method is the Recursive Least Squares (RLS) [36]. This method gives fast convergence especially for a deterministic model. On the other hand, the gradient methods [36] are characterized by slow convergence and hence poor adaptation. Other methods such as the Maximum Likelihood and Extended Least

Squares are used for identification of stochastic models. Therefore, and since that the GPC depends on a deterministic model [13], the RLS method is adopted. It will be discussed here and will be used as an estimator with the GPC algorithm.

The first step in identification is the assumption of the model form. In this chapter a CARIMA(Controlled-Auto-Regressive-Integrated- Moving-Average) model is considered. However, other model forms can be used. Figure 3.1 shows a process and disturbance schematic diagram assuming that the process is represented by the CARIMA model. The CARIMA model is given by [13]:

$$A(z^{-1})y(t) = z^{-kd}B(z^{-1})u(t-1) + \frac{C(z^{-1})\zeta(t)}{\Delta} \tag{3.1}$$

where y,u, andζ are the measured output, control input, and uncorrelated random sequence respectively. Δ is the difference operator $(1 - z^{-1})$ and kd is the time delay in samples. A,B and C are polynomials in the backward shift operator z^{-1} with degrees na , nb and nc respectively:

$$A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{na} z^{-na}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_{nb} z^{-nb}$$

$$C(z^{-1}) = c_0 + c_1 z^{-1} + \cdots + c_{nc} z^{-nc}$$

The parameters of these polynomials are those to be estimated. The model can be written in a compact form, using vector notation:

$$\Delta y(t) = \phi^T \hat{\theta} \tag{3.2}$$

where:

$$\hat{\theta} = [a_1, \cdots, a_{na}, b_0, b_1, \cdots, b_{nb}, c_0, c_1, \cdots, c_{nc}]^T$$

and

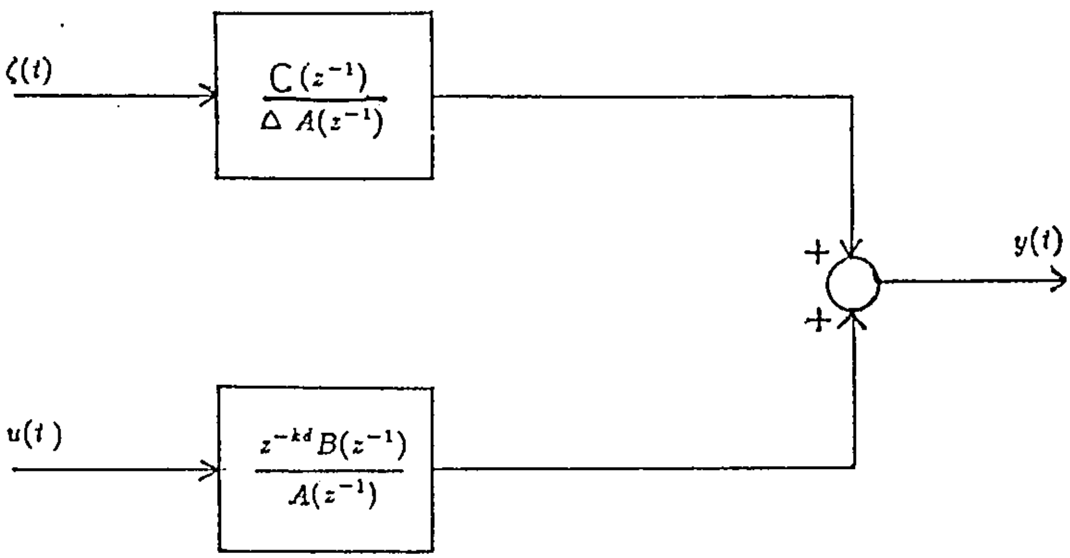

Figure 3.1: Process and Disturbance Schematic Diagram. Assuming a CARIMA Model.

$$\phi = [-\Delta y(t-1), -\Delta y(t-2), \cdots, -\Delta y(t-na), \Delta u(t-kd-1),$$

$$, \cdots, \Delta u(t-kd-nb), e(t-1), e(t-2), \cdots, e(t-nc)]$$

The parameters of the above model denoted by $\hat{\theta}$ can be identified for a set of data using the Standard Least Squares (SLS) method.

3.1 Standard Least Squares (SLS) Algorithm

The complete derivation of the algorithm is present in reference [36]. It is reviewed in Appendix A. The algorithm is based on minimizing the following cost function which represents the sum of sqaures of errors:

$$J = \sum_{t=1}^{N} \hat{\epsilon}^2(t) = \hat{E}^T \hat{E} \tag{3.3}$$

Where $\epsilon(t)$ is the difference between measured and estimated output at time t (i.e. the modelling error). It is defined by:

$$\hat{\epsilon}(t) = \Delta y(t) - \hat{y}(t) = \Delta y(t) - \phi^T(t)\hat{\theta} \tag{3.4}$$

$\hat{E}$ is a column vector containing $\hat{\epsilon}(t)$ values:

$$\hat{E} = Y - \hat{Y} = Y - \Phi^T \hat{\theta}$$

where :

$$Y = [\Delta y(1), \Delta y(2), \cdots, \Delta y(N)]^T$$

Φ is a matrix of size $N \times (na + nb + nc)$, where N is the number of data points. This matrix is defined as:

$$\Phi = [\phi(1), \phi(2), \cdots, \phi(N)]^T$$

it contains input/output data up to time t=N.

Rewriting equation (3.3) :

$$J = (Y - \Phi^T \hat{\theta})^T (Y - \Phi^T \hat{\theta}) \tag{3.5}$$

By minimizing J with respect to $\hat{\theta}$, the least squares estimation of $\hat{\theta}$ parameters is given by the following equation [36]:

$$\hat{\theta} = [\Phi^T \Phi]^{-1} [\Phi^T Y] \tag{3.6}$$

This equation gives the least squares estimates of the model parameters for a set of data. So it can be used for off-line identification to obtain a fixed model. But for self-tuning control purposes, an on-line identification method is needed. So, a recursive form of the SLS known as Recursive Least Squares (RLS) is considered.

3.2 Recursive Least Squares

The idea in the recursive algorithm is to utilize the new input/output data and the past estimates to decide the new estimates at each sampling instant [36]. Figure 3.2 shows a schematic diagram of the RLS method. This algorithm is based on minimizing the same cost function in the SLS algorithm given by Equation(3.3).

The identification procedure consists of the following steps:

1. calculation of the prediction error

$$\hat{\epsilon}(t) = \Delta y(t) - \phi^T(t)\hat{\theta}(t-1) \tag{3.7}$$

2. calculation of the gain adjustment vector or the Kalman gain

$$K(t) = \frac{P(t-1)\phi^T(t)}{1 + \phi^T(t)P(t-1)\phi(t)} \tag{3.8}$$

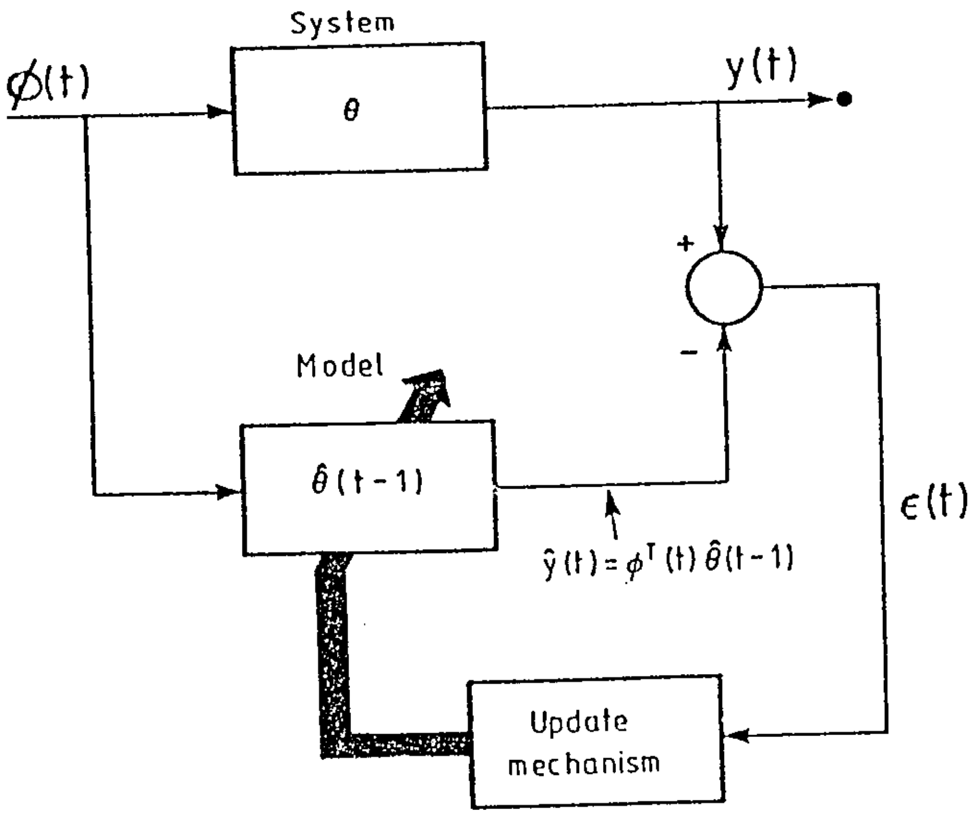

Figure 3.2: A Schematic Diagram of the RLS Identification Method.

3. the parameters are then updated through

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t) * \epsilon(t) \quad (3.9)$$

4. then the covariance matrix is updated

$$P(t) = P(t-1) - K(t)[P(t-1)\phi(t)]^T \quad (3.10)$$

P(t), the covariance matrix, is a diagonal matrix. In the SLS method this matrix is defined by:

$$P(t) = [\Phi^T \Phi]^{-1} \quad (3.11)$$

the covariance matrix has a great effect on the performance of the RLS algorithm, as will be discussed later.

3.3 Using the RLS Algorithm

A developed FORTRAN program, presented in Appendix B, was used to simulate the RLS algorithm. Now, to operate this algorithm, the model order must be assumed, then the initial parameter values are chosen. Another important object is the covariance matrix mentioned previously, an initial value of it is required. P(t) is a square matrix of dimension $n \times n$,where n is the number of parameters to be estimated. The size of this matrix is related to the uncertainty in the present model, where a large value of P(t) means that the current model is not an adequate representation of the process and hence if new rich data is received then the algorithm will be motivated, and therefore the model is updated. Usually the P(t) matrix is chosen initially to be diagonal of the form:

$$P(0) = mI \quad (3.12)$$

Where m is a positive scalar and I is the unit matrix.

3.3.1 Initial Covariance Matrix Size

The following model will be considered to see the effect of initial covariance matrix size on the performance of the algorithm:

$$y(t) = -a_1 y(t-1) + b_0 u(t-1) + b_1 u(t-2) \tag{3.13}$$

where $a_1 = -0.5$, $b_0 = 1.6$ and $b_1 = 1$ are the actual values of the parameters. The input U(t) was a zero mean white noise and the initial parameters vector was chosen to be: $\hat{\theta}(0) = [0., 0., 1.]$. The effect of initial P matrix size on the estimator behavior is seen in Figures 3.3 and 3.4 where the effect of various choices of P(0) is shown. It is evident that a larger initial value of P(0) leads to quick correct estimation, this is shown in Figure 3.3 where P(0) was chosen to be: $P(0) = 100I$. Figure 3.4 shows the use of a small covariance size: $P(0) = 0.1I$. The estimator fails to estimate correctly. The reason can be explained using Equation (3.9) . In this equation, and in order to identify new θ, values of $\epsilon(t)$ and K(t) must not be small. But, K(t) is directly proportional to the size of P(t), so a higher size of P(t) enhances the estimator to identify provided that there is an error $\epsilon(t)$

Not only the P(t) affects the value of the gain vector K(t), but also the level of excitation contained in $\Phi(t)$. This is discussed in the following section.

3.3.2 Level of Excitation

In order for the estimator to work properly the *input/output* data should be excited (*i.e. control signals should be changing and large enough to produce a noticeable effect*

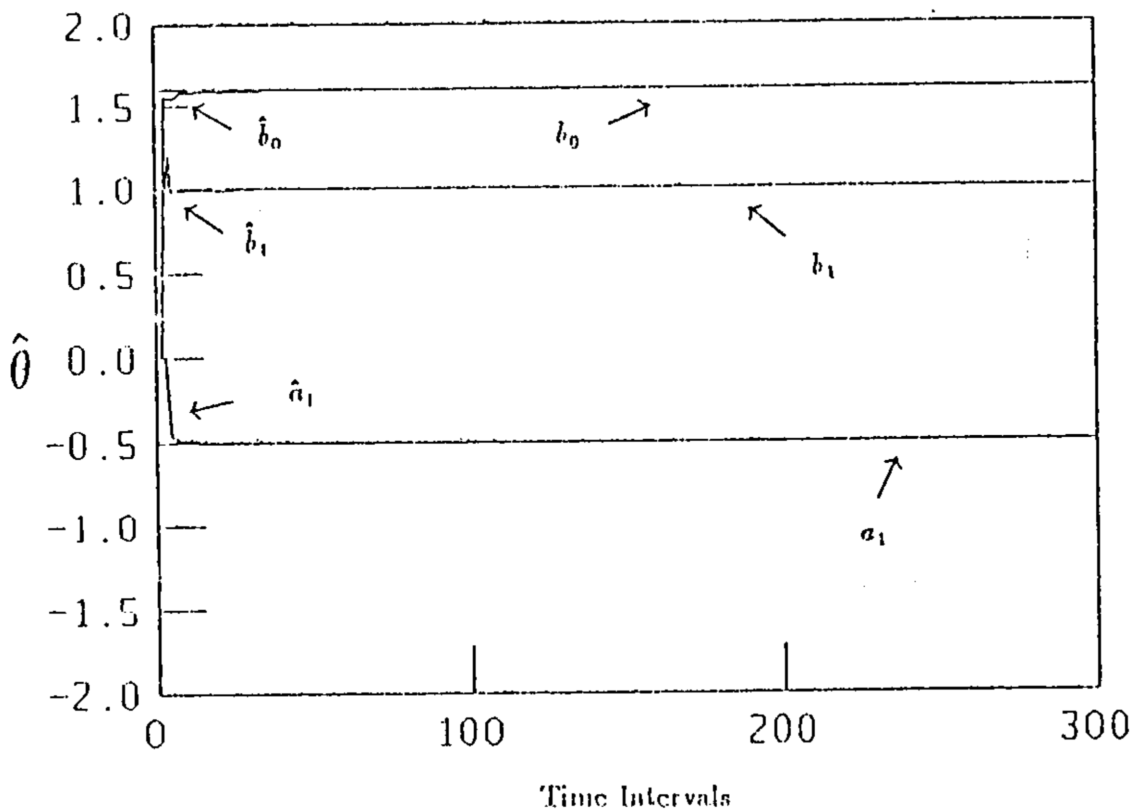

Figure 3.3: Effect of Initial Covariance Matrix Size(P(0)=100 I).

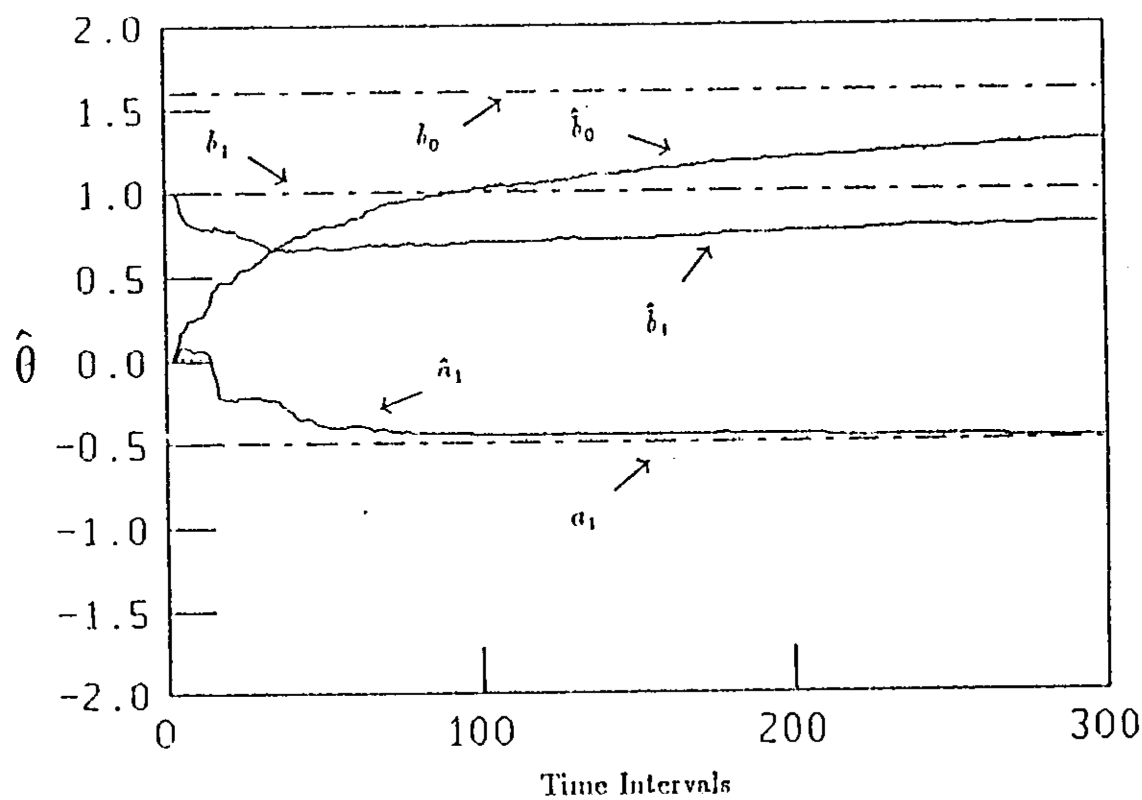

Figure 3.4: Effect of Initial Covariance Matrix Size(P(0)=0.1 I).

on the output). Poor data leads to bad estimation or failure in converging to the correct parameters values. This is obvious from Figures 3.5 and 3.6 . The same previous model was estimated with the same initial conditions. However Figure 3.5 shows the behaviour of the estimator with U(t) chosen to be constant : $U(t) = 0.1$. It is obvious that Low excitation leads to failure in identifying the model correctly. In Figure 3.6 the input was a unit variance noise. It is seen that in few samples the correct model was identified. Mathematically this can be explained using Equation (3.9) where the value of the K(t) vector due to poor data will tend to zero in the first case. Therefore the estimated parameters will not be updated, even if there is a large error in parameters.

Level of data excitation affects the size of the covariance matrix during operation, where a high excited data leads to a decrease in the P(t) size each time a new rich data is received. So, with time, the effect of K(t) on Equation (3.9) will be less effective. And, if the dynamics of the process change, the estimator will not have the ability to track the changing parameters. To overcome this deficiency, the algorithm is modified as will be discussed in the following section.

3.3.3 Change in Process Dynamics

A required property of an estimator is the ability to track parameter changes which reflects a change in the dynamics of the process. The algorithm in its given shape will not do the required function, because the covariance matrix will keep decreasing in size with time and sometimes one or more of the diagonal elements may become zero valued and will never move from there. Therefore the algorithm must be modified to prevent P(t) from becoming too small. This is achieved by the introduction of the exponential forgetting factor (β). The main objective of this factor is to give more consideration to

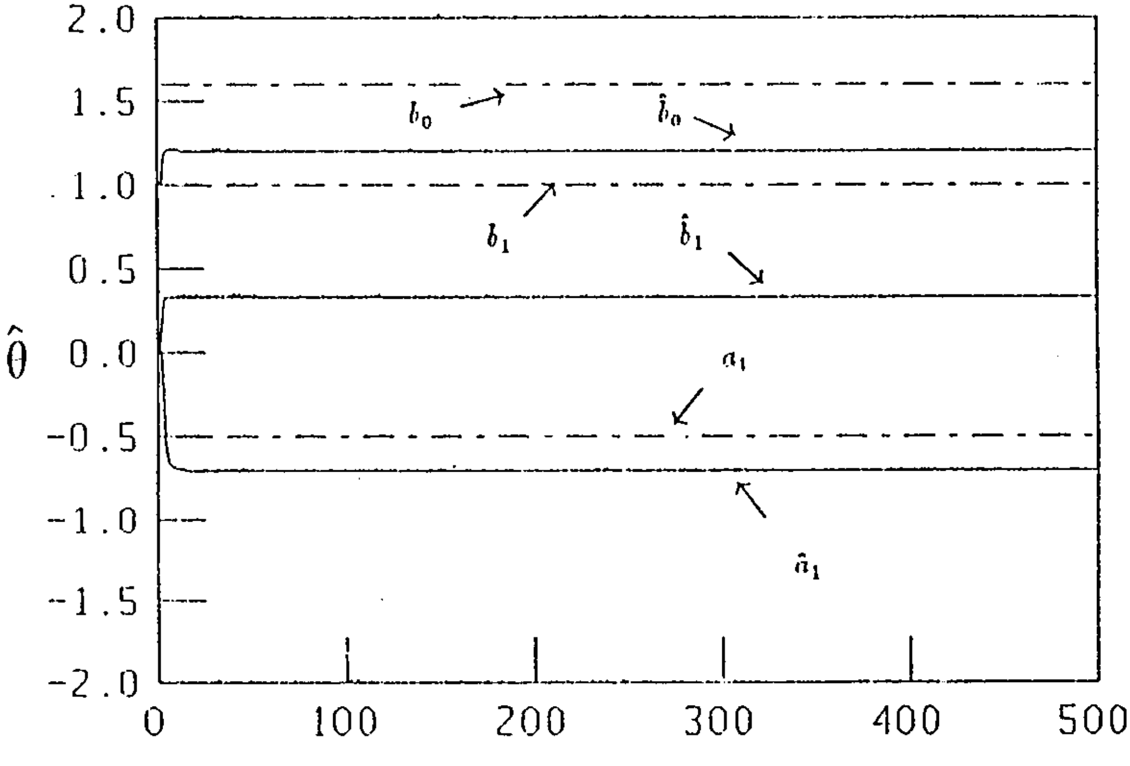

Figure 3.6: Effect of Level of Excitation (u(t)=0.1).

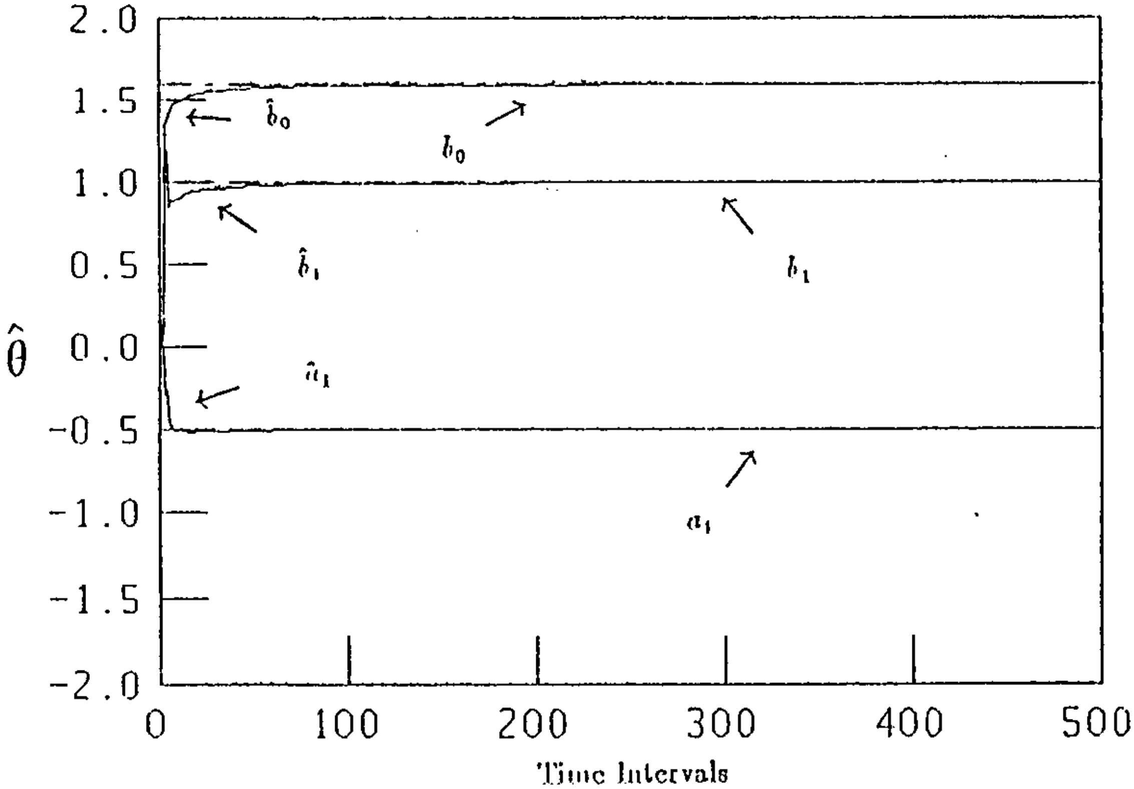

Figure 3.6: Effect of Level of Excitation (u(t)=White Noise).

new data (i.e. to forget past data). This factor changes the algorithm slightly.

The new cost function equation to be minimized is given by:

$$J = \sum_{t=1}^{N} \beta^{N-t} \epsilon^2(t) \tag{3.14}$$

Where β is the forgetting factor. The effect of forgetting is clear in the above equation, as time increases, past data is multiplied by a smaller factor(β^{N-t}), therefore giving the new data more importance. Equations (3.8) and (3.10) respectively change to:

$$K(t) = \frac{P(t-1)\phi^T(t)}{\beta + \phi^T(t)P(t-1)\phi(t)} \tag{3.15}$$

and

$$P(t) = (P(t-1) - K(t)[P(t-1)\phi(t)]^T)/\beta \tag{3.16}$$

the value of β usually is chosen to be between 0 and 1, its role is obvious from the last equation, where it helps in increasing the size of the covariance matrix since each element in P(t) is divided by a fraction, which will improve the estimation efficiency. To demonstrate the effect of the forgetting factor the following model is considered:

$$y(t) = -a_1 y(t-1) + b_0 u(t-1) + b_1 u(t-2) \tag{3.17}$$

where $a_1 = -1., b_0 = 0.75$ and $b_1 = 0.5$ are the actual values. The parameters were changed as follows:at sample 100 a_1 was changed to -0.7 and at sample 200 b_0 was changed to 1.3. Initial θ was set to $[0., 0., 1.]$ and $P(0) = 100I$.

Figures 3.7, 3.8 and 3.9 show the effect of various choices of forgetting factor, it can be seen that by decreasing the value of β below 1 (i.e. more forgetting), better estimation and tracking is obtained. in the case where $\beta = 1$ (i.e. no forgetting), see Figure 3.7, the estimator is not capable of tracking to parameter changes. This is due to the very small size of P(t).

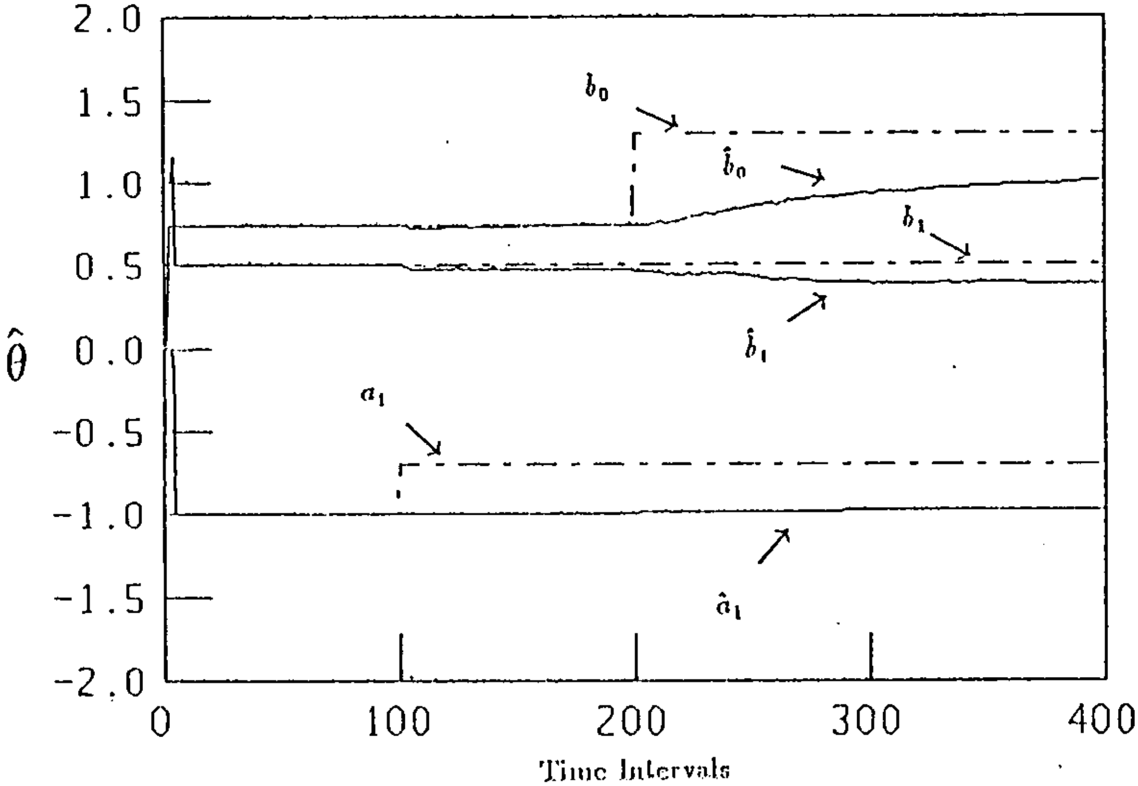

Figure 3.7: Constant Forgetting Factor (β=1).

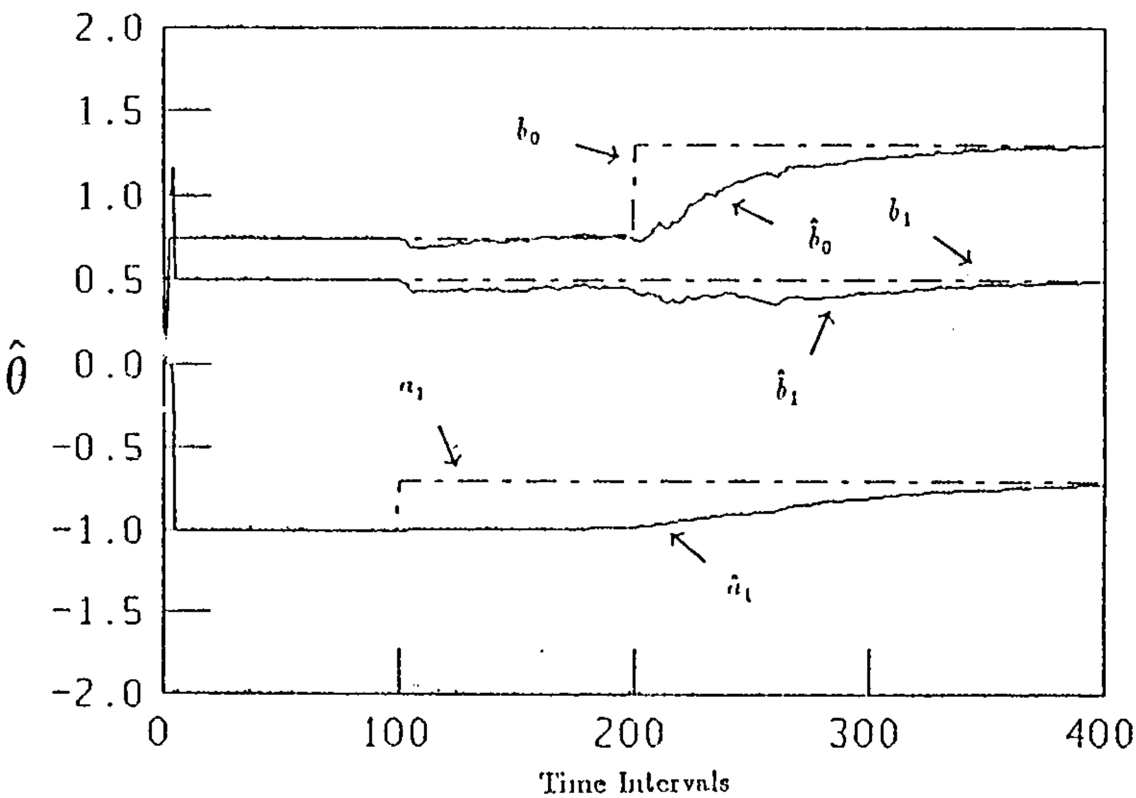

Figure 3.8: Constant Forgetting Factor (β=0.98).

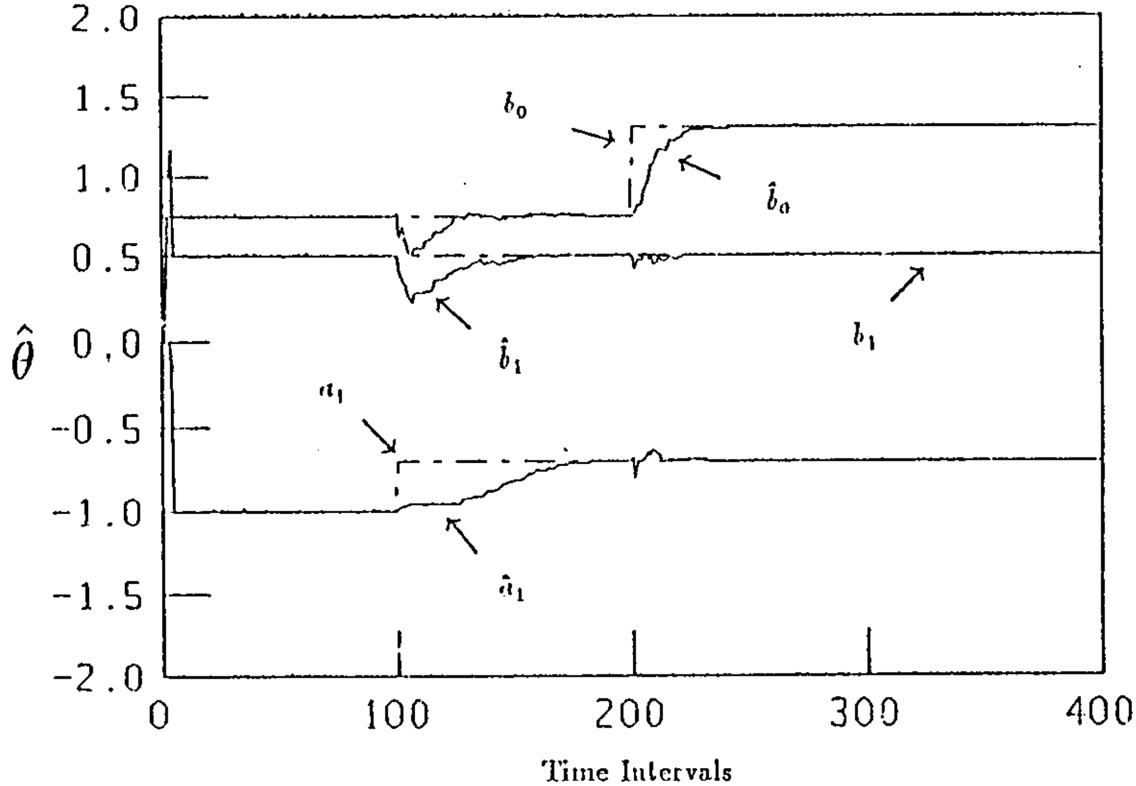

Figure 3.9: Constant Forgetting Factor (β=0.9).

The forgetting factor use is not always fruitful. Under certain conditions it leads to the phenomenon known as estimator blow-up.

3.3.4 Estimator Blow-Up

A problem arises when dealing with constant forgetting factors, with a process suffering from periods of poor data, for example during regulation or at steady state, it is known as the blow-up of the estimator [47]. The reason for this can be seen by considering Equation (3.16) where the second term on the right hand side becomes zero or close to zero (i.e. when the data is not excited) and the equation becomes:

$$P(t) = P(t-1)/\beta \tag{3.18}$$

This means that the covariance matrix will be divided each time by a fraction, and increases without limit successively. After some time the estimator becomes sensitive due to the large size of P(t) leading to rapid movement of the estimated parameters when new rich data become available. This phenomenon may lead to failure or instability in a self-tuning scheme.

To illustrate this phenomenon a process given by the model is estimated:

$$y(t) = -a_1 y(t-1) + b_0 u(t-1) + b_1 u(t-2) + e(t) \tag{3.19}$$

where $a_1 = -1., b_0 = .75$ and $b_1 = .5$. Where at sample 200, b_0 was changed to 1.3. Up to sample 400 the input signal was a white noise, but after that it was set to $U(t) = 0.1$. The blow-up phenomenon is shown in Figure 3.10, it is seen that the estimator performs well initially, and the forgetting factor helped in parameter tracking, after a long time with the low excitation input, the covariance becomes very large due to successive increase, it results in an abrupt change in the parameter estimates in an uncontrolled way.

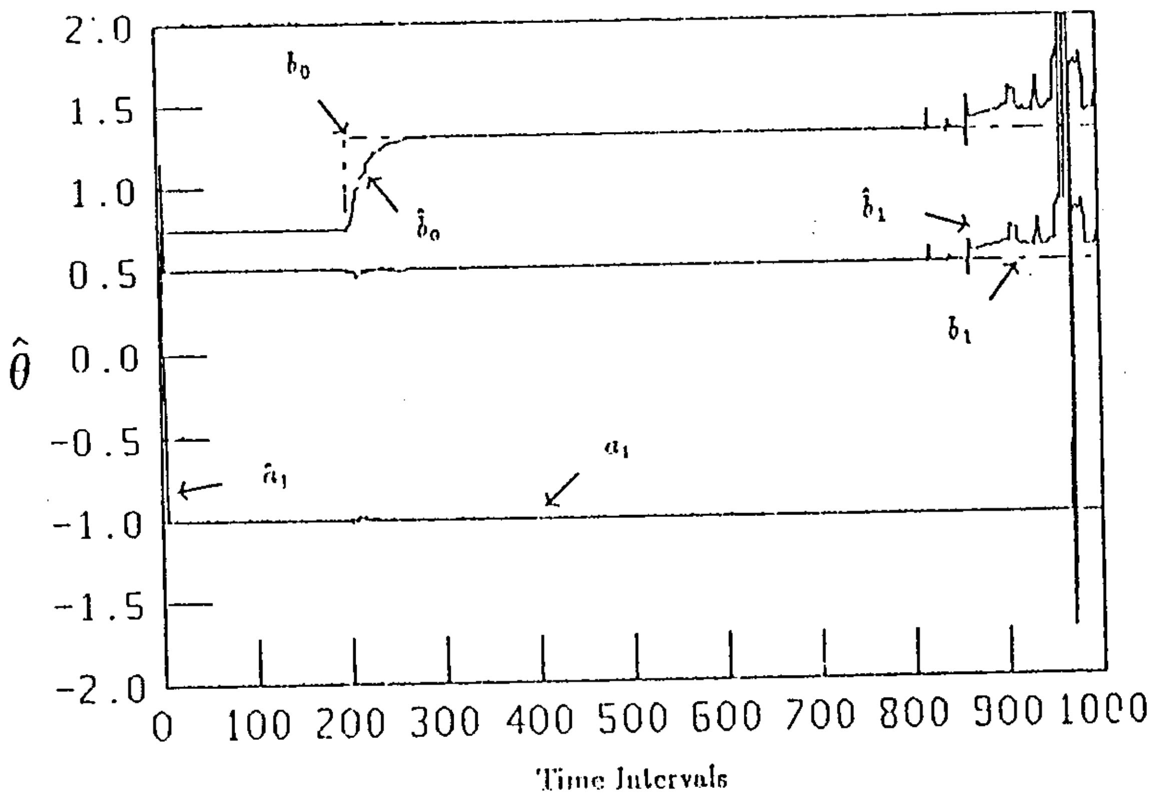

Figure 3.10: Blow-up Phenomenon.

To solve the problem of blow-up, a variable forgetting factor scheme is adopted.

3.3.5 Variable Forgetting

Another modification is made to the estimation algorithm, which will prevent the covariance matrix from becoming enlarged at periods of low excitation. The equation which govern this is given by Fortescue [48] as follows:

$$\beta(t) = 1 - \frac{[1 - K(t-1)\phi(t)^T]\epsilon^2(t)}{\sigma} \tag{3.20}$$

where σ is a design parameter chosen according to operating conditions of the process, mainly the noise content. Where the form of the equation was chosen to give a forgetting factor value close to 1.0 when the excitation in data is poor. Furthermore, the forgetting factor depends also on the error between estimated and true parameters, a higher error leads to decreasing the forgetting factor value. σ is used in the equation to cope with the noise content in the process (i.e. to decrease the sensitivity of the forgetting factor to the noise effect). Use of this form of forgetting factor will prevent the blow-up of the estimator in addition to improving the tracking capability of the estimator provided that a proper choice of σ is set. It can be deduced from equation (3.20) that at periods of low excitation the value of $\beta(t)$ will be close to 1, so this will not allow the covariance matrix from increasing in size at that period.

The effect of introducing the variable forgetting factor is illustrated in Figure 3.11, where the same example which was used in the blow-up section is employed here again. By comparison with Figure 3.10, it is seen that the blow-up problem is eliminated in the poor data region, due to preventing the covariance matrix form being large. Moreover, the estimator was able to track the parameter changes at the initial stage of the experiment

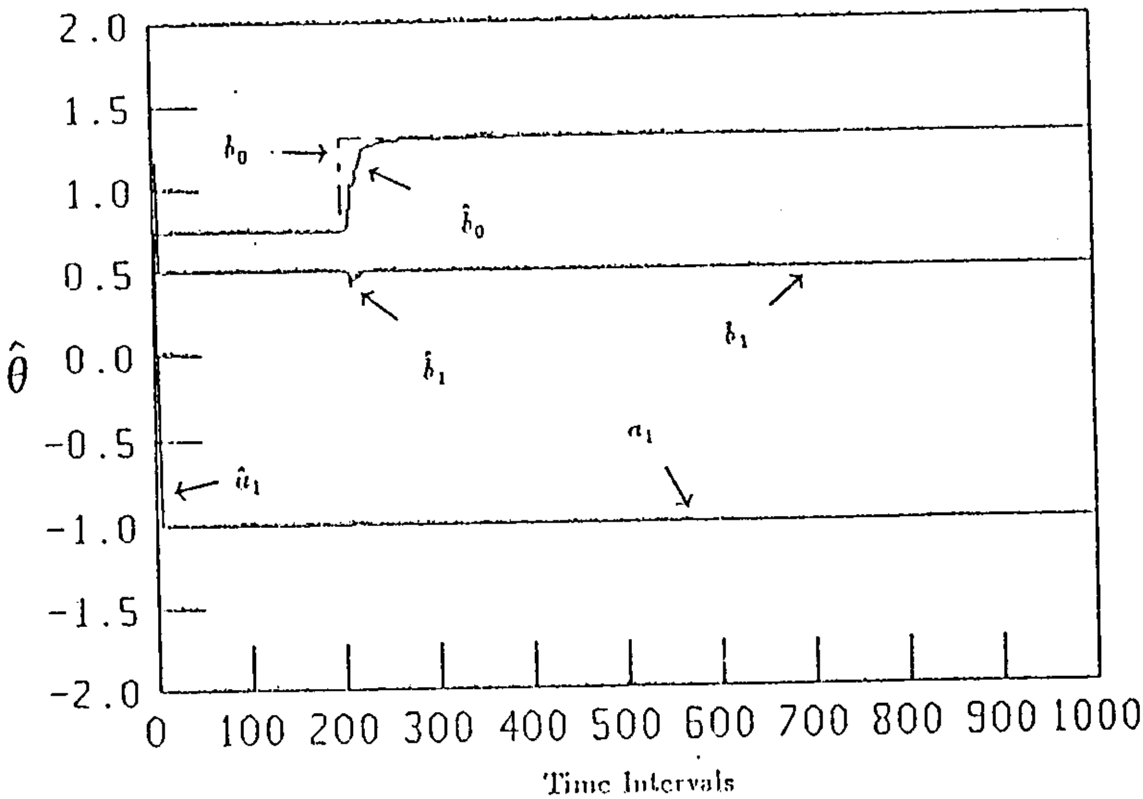

Figure 3.11: Use of Variable Forgetting Factor.

due to presence of data excitation. A value of (.1) was used for σ in this example, however a lower value of σ will increase the sensitivity of the forgetting factor.

This scheme must be applied with care in presence of noise, since $\epsilon(t)$ will have a remarkable value, this error will not be distinguished from the true error. It will result in decreasing the forgetting factor value and therefore increasing the size of the covariance matrix, resulting in large parameter update every sample. The problem can be handled through suitable choice of σ, where a higher value will reduce the sensitivity of the covariance matrix to changes in the input.

3.3.6 Noise Effect

For a deterministic model the estimator converges to the correct values in few samples provided that the data is sufficiently excited. In the case when the process is corrupted by noise, the estimator will not converge to the correct values. This fact is clear by considering Equation (3.9) where noise will be included in the $\epsilon(t)$ term which well affect the model update at each sampling instant. Increasing the noise variance will increases the deviations of parameters, this is shown in Figures 3.12 and 3.13 . Where in Figure 3.12, the noise variance was (1) , and in Figure 3.13 the variance was (3.)

Noise effect can be reduced by decreasing the sensitivity of the estimator, this is achieved by :choosing a small covariance matrix initially, bounding the forgetting factor close to 1.0 or in the case of using the variable forgetting factor scheme by choosing a high value of σ.

A further procedure to reduce the effect of noise on the estimator is to employ estimator jacketting. In this scheme the estimator is switched off when the error between the true output and the set point is less than a certain small value, where data available

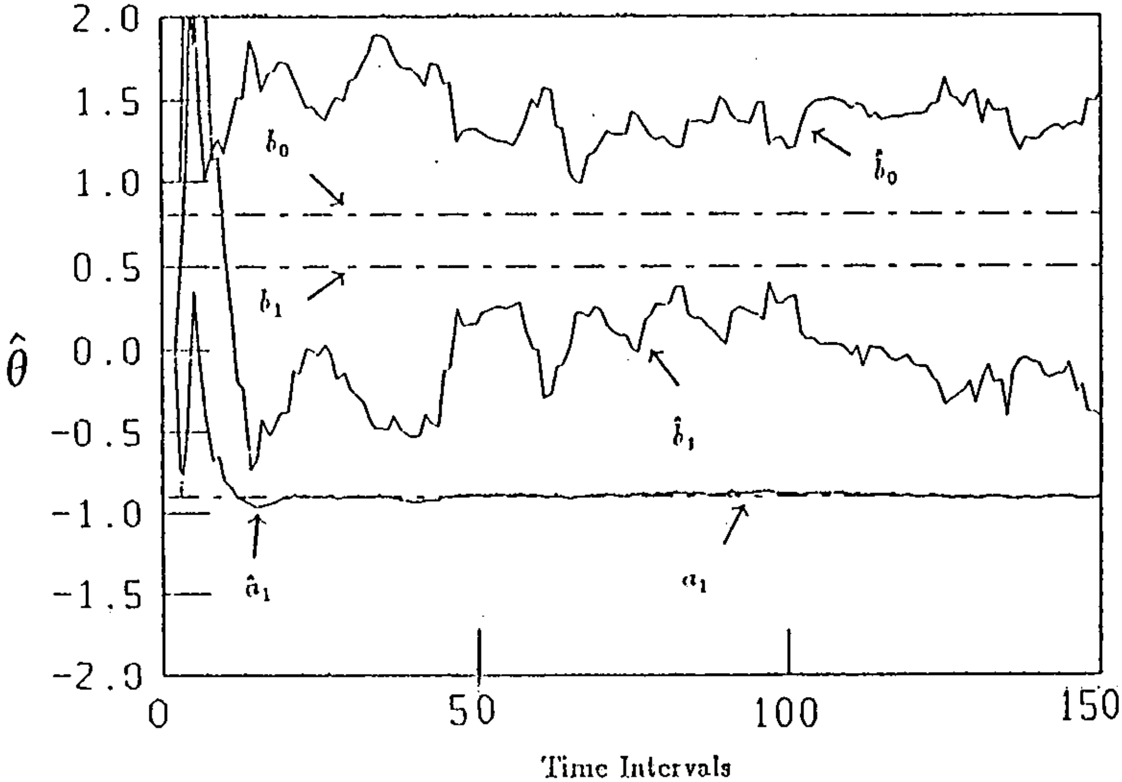

Figure 3.12: Effect of Noise(Variance=1).

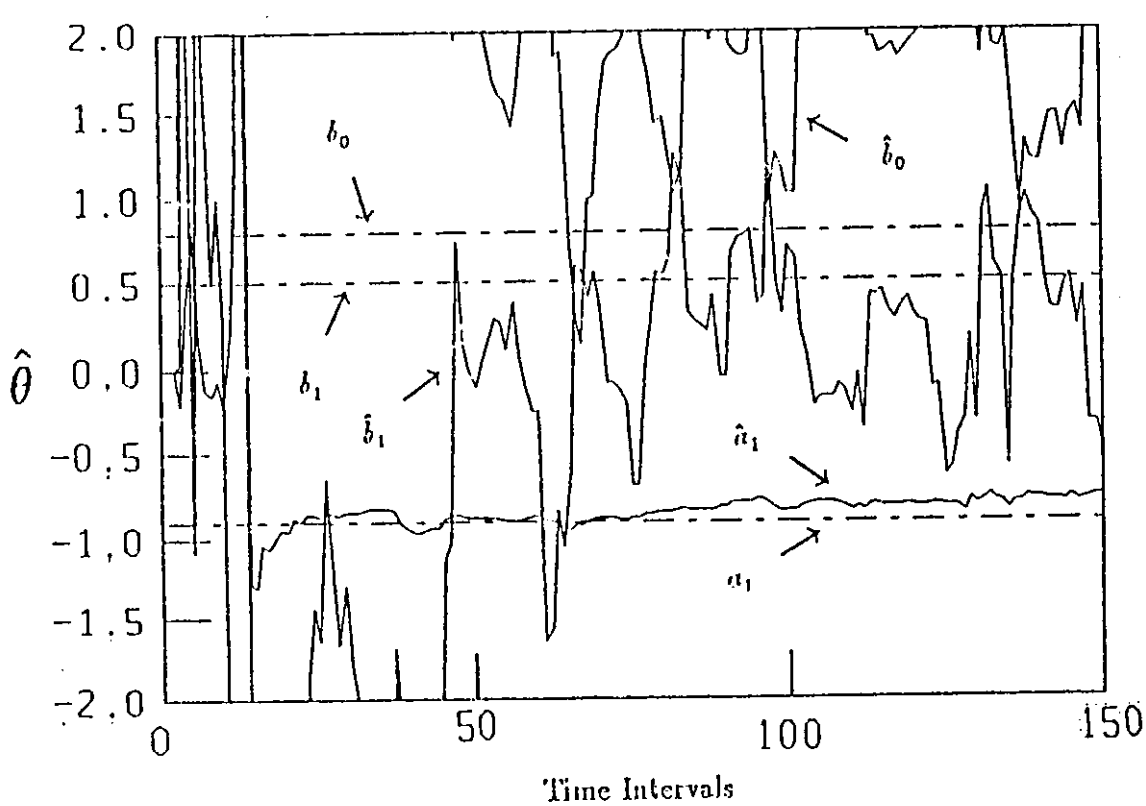

Figure 3.13: Effect of Noise(Variance=3).

at this region is mainly due to noise [44].

3.4 Conclusions

The RLS method includes a lot of design parameters. Choice of these parameters depends on the controlled process nature. Injection molding is characterized by nonlinearity and time-variation, in addition to noise corruption. So, when applying the GPC to Injection molding process, proper choice of the RLS parameters is required to provide the GPC with a good model, otherwise the controller may fail in its function.

Chapter 4

GENERALIZED PREDICTIVE CONTROL (GPC)

As mentioned before, the GPC is an effective control algorithm, it has many tuning knobs which give the controller its flexibility in dealing with various practical situations [13, 14, 15, 16].

In this chapter, the algorithm will be briefly reviewed, where the details are present in references [13, 14, 16]. Simulation examples are included to show the properties of the controller.

4.1 Process Model

One of the important features of the GPC algorithm, is the assumption of a CARIMA model of the process. This model builds an integral action into the feedback loop which ensures rejection of load disturbances and leads to zero offset in the output [13, 44]. The CARIMA model was given in the previous chapter as:

$$A(z^{-1})y(t) = z^{-kd}B(z^{-1})u(t-1) + \frac{C(z^{-1})\zeta(t)}{\Delta} \tag{4.1}$$

Where the equation parameters and polynomials are as defined in Chapter 3.

In the above model, and in order to make it a deterministic one , the order of the C

polynomial is set to be 1, therefore the model becomes [13]:

$$A(z^{-1})y(t) = z^{-kd}B(z^{-1})u(t-1) + \frac{\zeta(t)}{\Delta} \tag{4.2}$$

where the effect of C polynomial is absorbed in A and B polynomials.

4.2 Long Range Prediction

To derive a j-step ahead predictor based on the CARIMA model , the following Diophantine identity is considered [13]:

$$1 = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \tag{4.3}$$

Where E_j and F_j are polynomials of order j-1 and na respectively, which can be determined given the $A(z^{-1})$ polynomial. Combining equations (4.2) and (4.3) gives the prediction equation [13]:

$$y(t+j|_t) = G_j\Delta u(t+j-1) + F_jy(t) + E_j\zeta(t+j) \tag{4.4}$$

Where $y(t+j|_t)$ is the predicted output after j time samples in future and based on data up to time t and $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$, a Polynomial of order $nb+j-1$. The last term in the prediction equation will be in future and unknown in the prediction range since the degree of E_j is $j-1$. The optimum prediction of y is obtained by assuming that this terms is zero [13]:

$$\hat{y}(t+j|_t) = G_j\Delta u(t+j-1) + F_jy(t) \tag{4.5}$$

In the GPC algorithm,the prediction is extended to become a multi-step prediction, where a set of future outputs is predicted in the range: 1 to N , based on recursion of equation (4.3) over that range. The result is a set of future predicted outputs with a form identical to that of equation (4.5). The first term on the RHS of this equation is divided into two parts at each prediction step, separating between known and future terms according to the following identity:

$$G_j(z^{-1}) = \overline{G}_j(z^{-1}) + z^{-j}\tilde{G}_j(z^{-1}) \qquad (4.6)$$

where: $\overline{G}_j$ is a polynomial corresponding to the future terms, and $\tilde{G}_j$ is another polynomial that corresponds to the known terms.

let:

$$f(t+j) = \tilde{G}_j \Delta u(t-1) + F_j Y(t) \qquad (4.7)$$

which represents the known terms.

The predicted outputs for all future steps in the prediction range can be combined in vector form as follows:

$$Y = [y(t+1), y(t+2), \cdots, y(t+N)]^T \qquad (4.8)$$

Similarly, the other terms are written as:

$$\tilde{U} = [\Delta u(t), \Delta u(t+1), \cdots, \Delta u(t+N-1)]^T \qquad (4.9)$$

$$f = [f(t+1), f(t+2), \cdots, f(t+N)]^T \qquad (4.10)$$

and consider a set point vector W, over the prediction range:

$$W = [w(t+1), w(t+2), \cdots, w(t+N)]^T \qquad (4.11)$$

so equation (4.5) in vector form is given by:

$$Y = G\tilde{U} + f \qquad (4.12)$$

where G is a matrix of dimension $N \times N$, it is a lower triangular matrix of the following form:

$$G = \begin{bmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & \dots & g_1 \end{bmatrix}$$

4.3 The Controller

GPC is a controller based on optimizing a cost function. This cost function is given by:

$$J(N1, N2, \lambda) = E[\sum_{j=N1}^{N2} e^2(t+j) + \lambda \sum_{j=1}^{N2} \Delta u^2(t+j-1)] \tag{4.13}$$

where E means expectation over the prediction range, which is conditioned on data up to time t, e is the difference between set point and the predicted output defined by:

$$e(t+j) = w(t+j) - \hat{y}(t+j) \tag{4.14}$$

N1 is the minimum prediction horizon, N2 is the maximum prediction horizon, λ is the control weighting scalar factor and w(t+j) is the set point in future, which may be known a priori or assumed equal to w(t).

In the above equation, prediction is performed in the range $N1 \rightarrow N2$, and a set of projected control increments is decided. While the control signal is weighted by a positive scalar λ. Combining equations (4.12) and (4.13) [13] gives :

$$J = (W - G\tilde{U} - f)^T (W - G\tilde{U} - f) + \lambda \tilde{U}^T \tilde{U} \tag{4.15}$$

minimizing this equation with respect to the control increments vector gives the projected control increment vector $\tilde{U}$, where the first element in this vector is applied to the process [13]:

$$\tilde{U}_{opt} = (G^T G + \lambda I)^{-1} G^T (W - f) \tag{4.16}$$

where I is the unit matrix, $G^T G$ is a matrix of order $N2 \times N2$, so at every sample a matrix of order $N2 \times N2$ will be inverted. This needs a large computational capacity especially for large values of N2. Therefore, the use of the control horizon will decrease the computational burden, as will be discussed in the next section.

4.4 The Control Horizon

Use of the control horizon is an interesting feature of the GPC, it means that after an interval NU, the control increments ΔU will be assumed to be zero. This equals applying an infinite control weighting on future control increments. In the cost function, this is achieved by replacing N2 by NU in the second summation on the right hand side. So, the cost function becomes:

$$J(N1, N2, NU, \lambda) = E[\sum_{j=N1}^{N2} e^2(t+j) + \lambda \sum_{j=1}^{NU} \Delta u^2(t+j-1)] \tag{4.17}$$

This assumption will simplify the controller calculations, since the order of the G matrix, mentioned previously, becomes $(N2 - N1 + 1 \times NU)$ as follows [16]:

$$G = \begin{bmatrix} g_{N1} & g_{N1-1} & \cdots & 0 \\ g_{N1+1} & g_{N1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N2} & g_{N2-1} & \cdots & g_{N2-NU+1} \end{bmatrix}$$

In particular if NU=1, this means that equation (4.16) will no longer include matrix inversion, since the order of $G^T G$ will be 1×1. Moreover, choice of NU value has an effect on the output of the process, where a small value of NU will give sluggish response, but choosing higher values a more active control is obtained.

It was shown that the objective of the controller is to force the process output to follow a specified set point profile. However, at each sample, the GPC performs the following steps:

1. The parameters of the process model are identified using the RLS method

2. The predicted outputs in the prediction range are calculated

3. The cost function with the specified parameters is minimized with respect to future control actions, and hence the optimal future control vector $\tilde{U}_{opt}$ is obtained.

4. The first element in $\tilde{U}_{opt}$ vector is used as an input to the process, then the data vectors are shifted and the calculations are repeated at the next sample

4.5 Design Parameters

N1,N2 and λ form the design knobs of the controller in addition to the control horizon (NU). The minimum prediction horizon (N1) is related to the time delay, kd, of the process, if kd is known, then N1 is set to be equal to kd or more.

N2, the maximum prediction horizon, is chosen to include all the transient response which is affected by the current control signal (At least: N2 $\geq 2n_b - 1$), and it will be better if chosen up to the rise time of the process. However, a value of 10 is usually sufficient to lead to stable control [13].

λ, the control weighting factor is used where control action is important. It is used as a fine tuner for the control signal and the response. Moreover, it may help in avoiding the inversion of a singular matrix in equation (4.16).

Certain choices of these parameters produces control performances similar to that of other control algorithms such as Pole- placement, GMV and others [13, 15, 16].

4.6 Data Filtering

In the model considered in the GPC algorithm, equation (4.1), it is usually not easy to estimate the parameters of the polynomial representing the noise components ($C(z^{-1})$) since noise is unknown and time-varying [11, 42, 49]. However by replacing ($C(z^{-1})$) by a fixed filtering polynomial ($T(z^{-1})$) will help in detuning high frequency components dominated by noise, load disturbances, nonlinearities and unmodelled dynamics [44]. Rewriting equation (4.1) as:

$$A(z^{-1})y(t) = z^{-kd}B(z^{-1})u(t-1) + \frac{T(z^{-1})\zeta(t)}{\Delta} \tag{4.18}$$

Including the T-filter will change the algorithm slightly: The Diophantine equation corresponding to the new model will be [13]:

$$T(z^{-1}) = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \tag{4.19}$$

the predicted output becomes:

$$\hat{y}(t+j|_t) = G_j\Delta u^f(t+j-1) + F_jy^f(t) \tag{4.20}$$

where u^f andy^f, means filtered quantities. The partition of G_j will be done according to the following identity:

$$G_j(z^{-1}) = T(z^{-1})\overline{G}_j(z^{-1}) + z^{-j}\tilde{G}_j(z^{-1}) \tag{4.21}$$

the remainder is the same as that of the previous case.

This improves the performance of the GPC, since the filter cuts the high frequency components outside the bandwidth of the process [46]. This leads to a noticeable less active controller output. Furthermore, it affects the behaviour of the estimator, because the estimator receives attenuated data, therefore better estimation is obtained.

Choice of the T-polynomial has no general rule, however in previous applications [13, 45, 49] a first order polynomial was adequate.

Tuffs [49] made the following choice:

$$T(z^{-1}) = 1 + t_1 z^{-1} \tag{4.22}$$

where $t_1 = -(1 - h/tr) = -0.8$, where h is the sampling time and tr is the number of main samples in which to reject 63 percent of a step-like change in load on the process.

Alassaf [44] made the choice of the T-polynomial as a low-pass filter having a cut-off frequency related to the open-loop bandwidth as:

$$\frac{1}{T(s)} = \frac{1}{\frac{2}{wn}s + 1} \tag{4.23}$$

where T(s) is the continuous version of the T-filter and wn is the underdamped process natural frequency.

The corresponding discrete representation of the T-filter using Zero Order Hold (ZOH) is:

$$\frac{1}{T(z^{-1})} = \frac{1 - e^{\frac{-hwn}{2}}}{1 - e^{\frac{-hwn}{2}} z^{-1}} \tag{4.24}$$

in this study the T-polynomial will be chosen to be a first order polynomial .

4.7 Simulation Examples

The aim of these simulation examples is to show the behaviour of the GPC, its design parameters role and the T-filter role . Simulation was done using a FORTRAN program written as part of this research. A list of this program is present in appendix C.

In all simulation examples the set point was a square wave of amplitudes 30 to 70 with a period of 50 sampling times. The control signal was bounded between -100 and 100. The default settings of the controller are chosen to be: [.5,1,10,1] for [λ,N1,N2,NU].

To demonstrate the effect of GPC parameters on the performance and not to confuse it with the effect of the identification technique, the estimator was not employed in the experiments except the experiment which deals with a process with changes in its dynamics.

- Role of λ

A model of the form:

$$\frac{y(t)}{u(t)} = \frac{.1 + .2z^{-1}}{1 - .9z^{-1}} \tag{4.25}$$

was used in this experiment to see the effect of control weighting. Figure 4.1 shows the performance of GPC without control weighting (i.e λ=0). The control output is active especially at the points of changing amplitude. But, tight tracking of the set point is obvious. Figure 4.2 shows the performance when using λ (λ=0.5), it is obvious that the control output is less active than the previous case, though at the cost of a significant oscillations in the response. Increasing the value of λ will give less active control and more slower response.

- Role of NU

A model of third order was used to illustrate the role of the control horizon. The model considered is:

$$\frac{y(t)}{u(t)} = \frac{.5 + .3z^{-1} - .15z^{-2} + .4z^{-3}}{1 + .1z^{-1} - .5z^{-2} - .3z^{-3}} \tag{4.26}$$

Several values of the control horizon were used as follows: at samples [1,75,175,275,375] the control horizon was set to [1,2,3,5,10] respectively . Where the other controller settings are chosen to be:[0.5,1,10)] for [λ,N1,N2]. Figure 4.3 shows that by increasing the control horizon NU a more active control is achieved. This is true due to

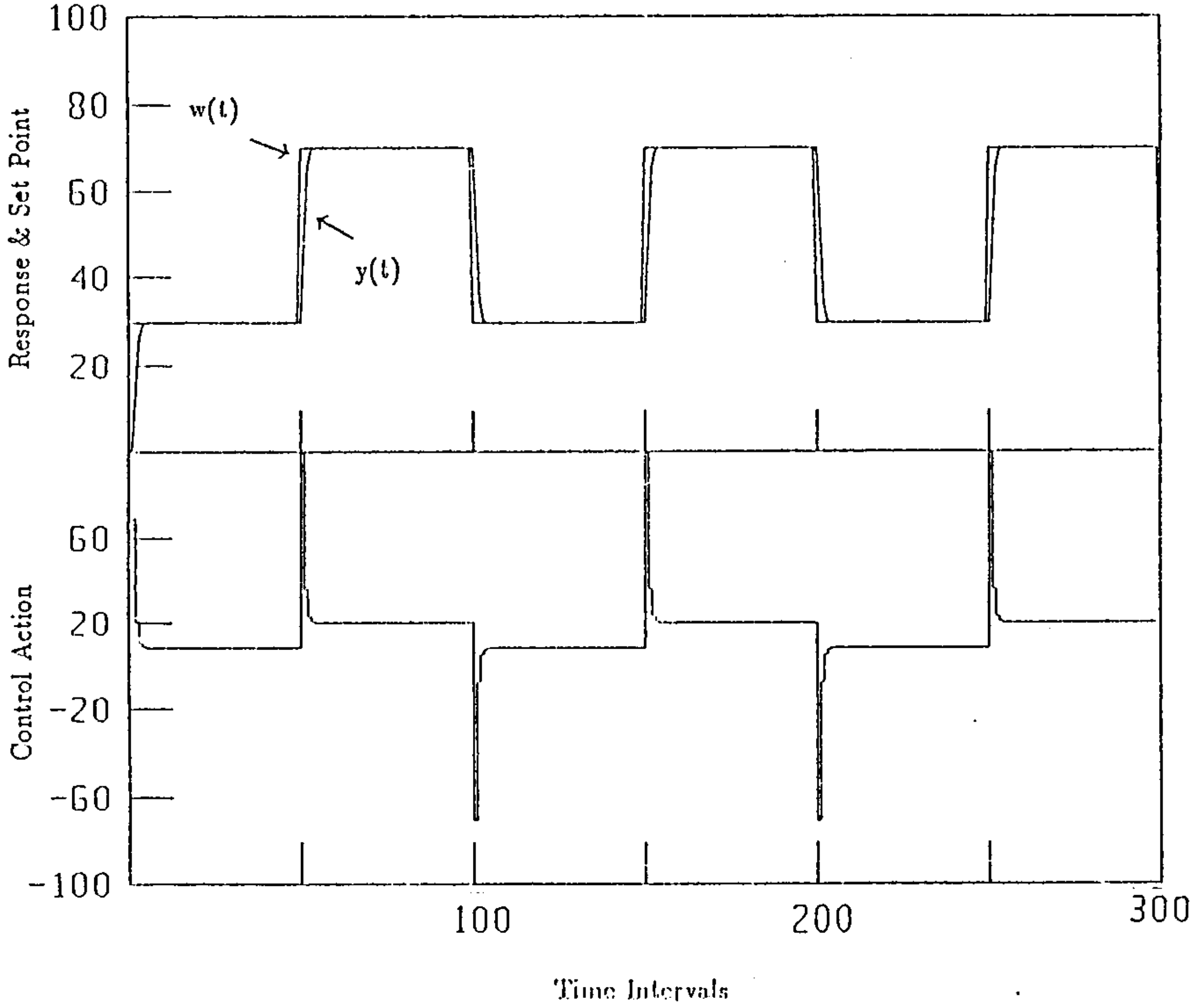

Figure 4.1: GPC Without Control Weighting (λ=0).

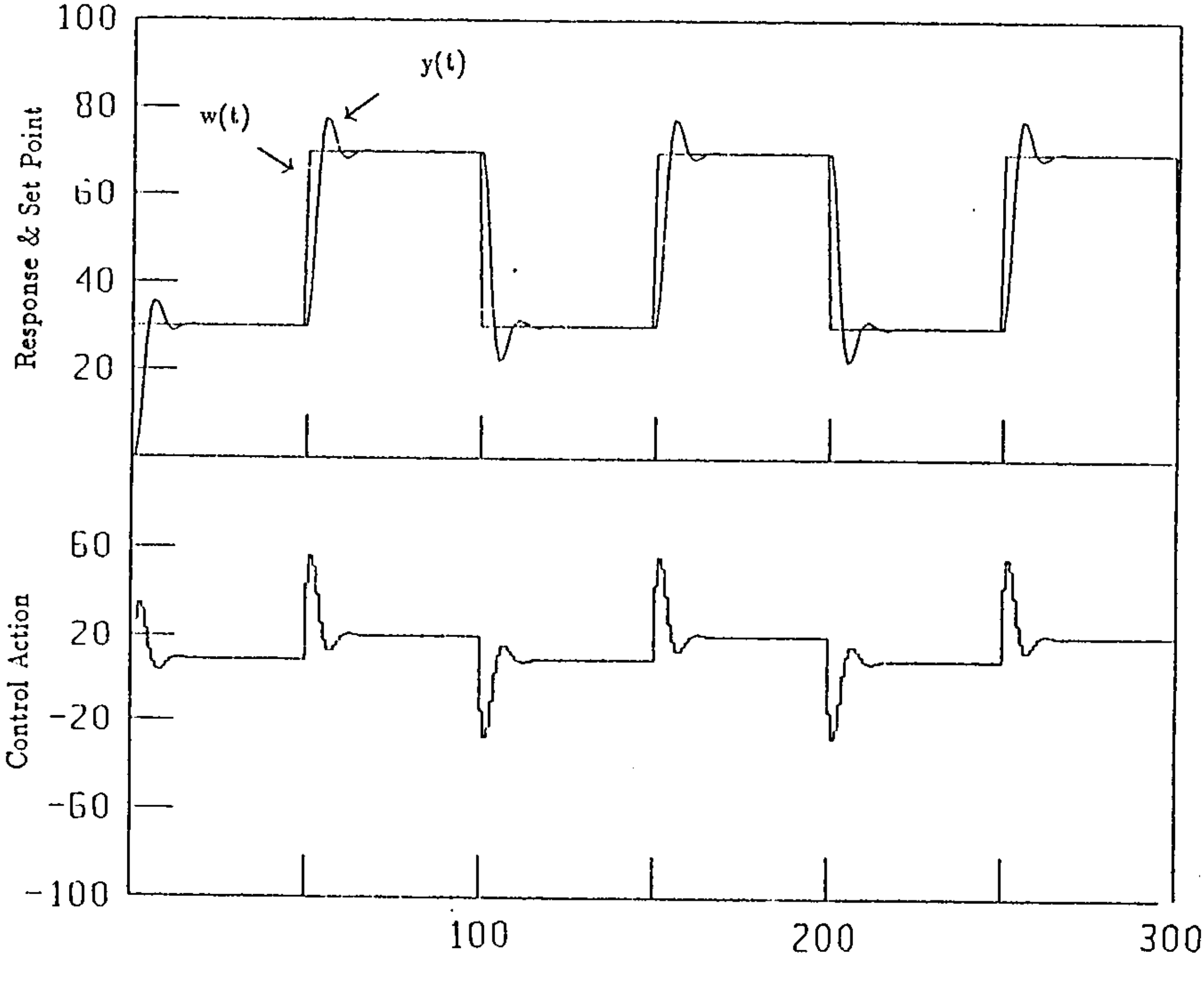

Figure 4.2: GPC With Control Weighting (λ=0.5).

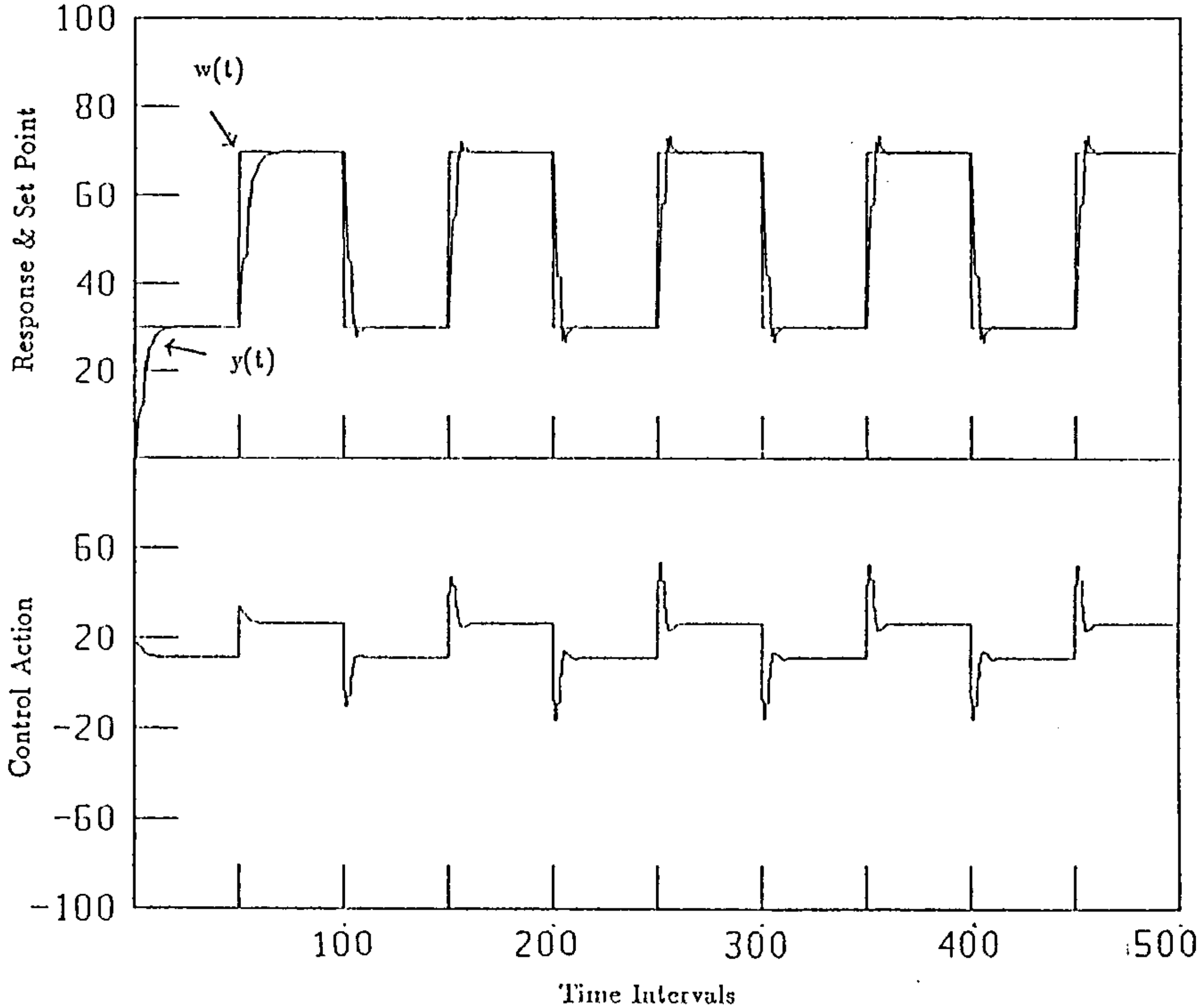

Figure 4.3: Effect of the Control Horizon(NU).

the fact that when the value of NU is increased then the constraints on the control signal are decreased. Therefore, the control effort will be higher. Increasing NU more than 3 makes no apparent difference on the closed-loop behaviour.

If the model order is not correct, and $NU > 1$, this will lead to instability [13, 44].

- Role of N2

  To show the effect of N2, a first order model was applied, the model is given by:

$$\frac{Y(z^{-1})}{U(z^{-1})} = \frac{.1 + .2z^{-1}}{1 - .9z^{-1}} \tag{4.27}$$

At samples [1,75,175,275,375] N2 was set to [1,2,5,10,20] respectively. The other parameters were chosen to be [0.1,1,10] for [λ,N1,NU]. Figure 4.4 shows that increasing the value of N2, a more slower response is obtained. At the beginning of the run when $N2 = 1$, the role of λ was dominant which causes the oscillations in the output. Increasing the prediction horizon gives the controller more time to correct the present error, and this is the reason for the more sluggish response.

- Role of T-filter

  Figure 4.5 declares the important behaviour of the T-filter. The same previous model was simulated, noise of variance (5) was added to the output all the time. Data filtering was not performed during the first 250 samples (T=1). After that the filter was put on, choosing the T-polynomial to be ($T = 1 - .8z^{-1}$). The result is evident: attenuated control action. The T-filter did not affect the set point tracking response of the process.

- Time-variant process

  In most practical applications the dynamics of the process are not stationary

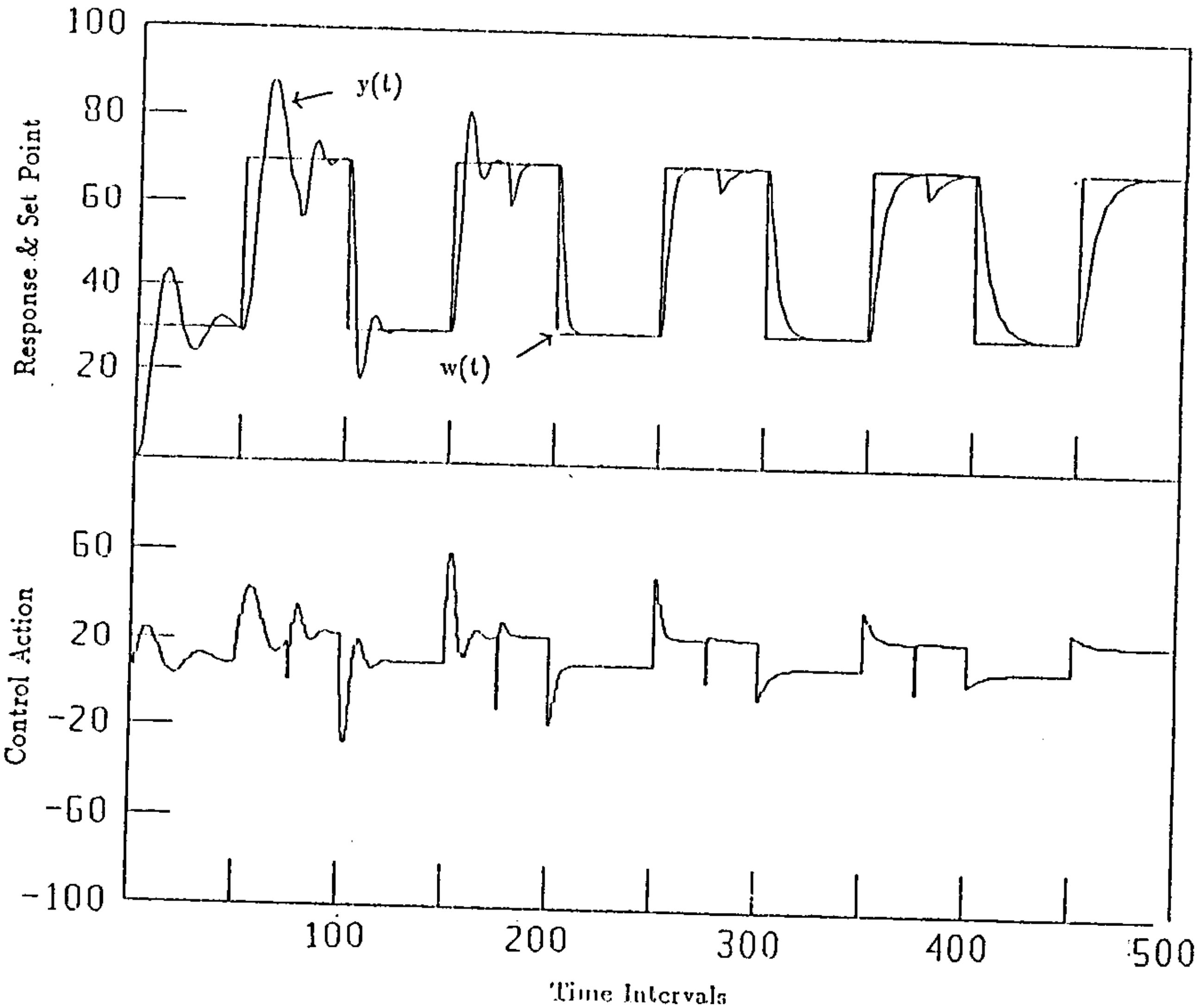

Figure 4.4: Effect of the Prediction Horizon(N2) .

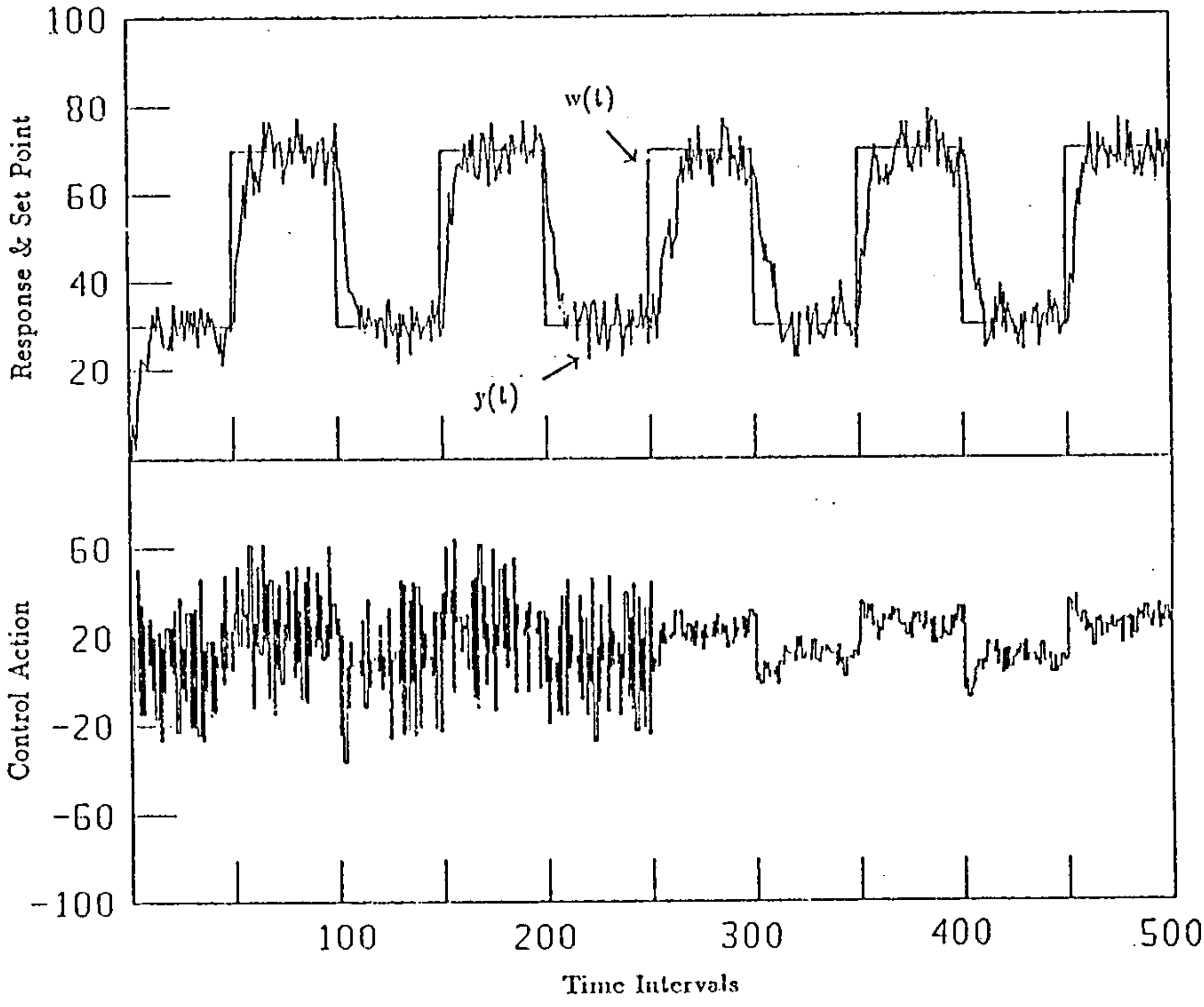

Figure 4.5: Effect of the T-Polynomial.

(i.e. change with time). GPC needs an accurate model of the process in order to do its function, therefore the estimator must track these changes correctly. In this example the estimation was performed on-line with control. The estimator was initialized by $\hat{\theta}_0 = [0, 0, \cdots, 1]$, the covariance matrix was set to 100 initially. The variable forgetting factor is used with $\sigma = 0.1$. A second order model was employed:

$$\frac{y(t)}{u(t)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{4.28}$$

where $a_1 = -0.8, a_2 = -0.1, b_0 = 0.3, b_1 = 0.5$ and $b_2 = -0.2$. The parameters were changed as follows: at sample $80, b_1 = .8$, at sample $170, a_2 = .1$, at sample $210, b_2 = -.3$ and at sample $260, a_1 = -.9$. Figure 4.6 shows the performance of GPC and its capability of handling a time-varying process with on-line estimation. At points of changing the parameters a noticeable error in the response is seen. This is due to changing the dynamic gain of the process. The estimator behaviour is shown in Figure 4.7 . It is clear that the estimator performance is fair, the estimator was able to track the parameters variation with small errors.

Time-varying nature is a property of injection molding process, and more complex than the above example due to nonlinearity and other factors. Therefore, it will be a true test of the adaptivity of the GPC algorithm.

4.8 Conclusions

These examples show some of the practically desired properties of the GPC controller, in which GPC proved its capability in handling several problems , provided that the design knobs are chosen appropriately. The data filter included improves

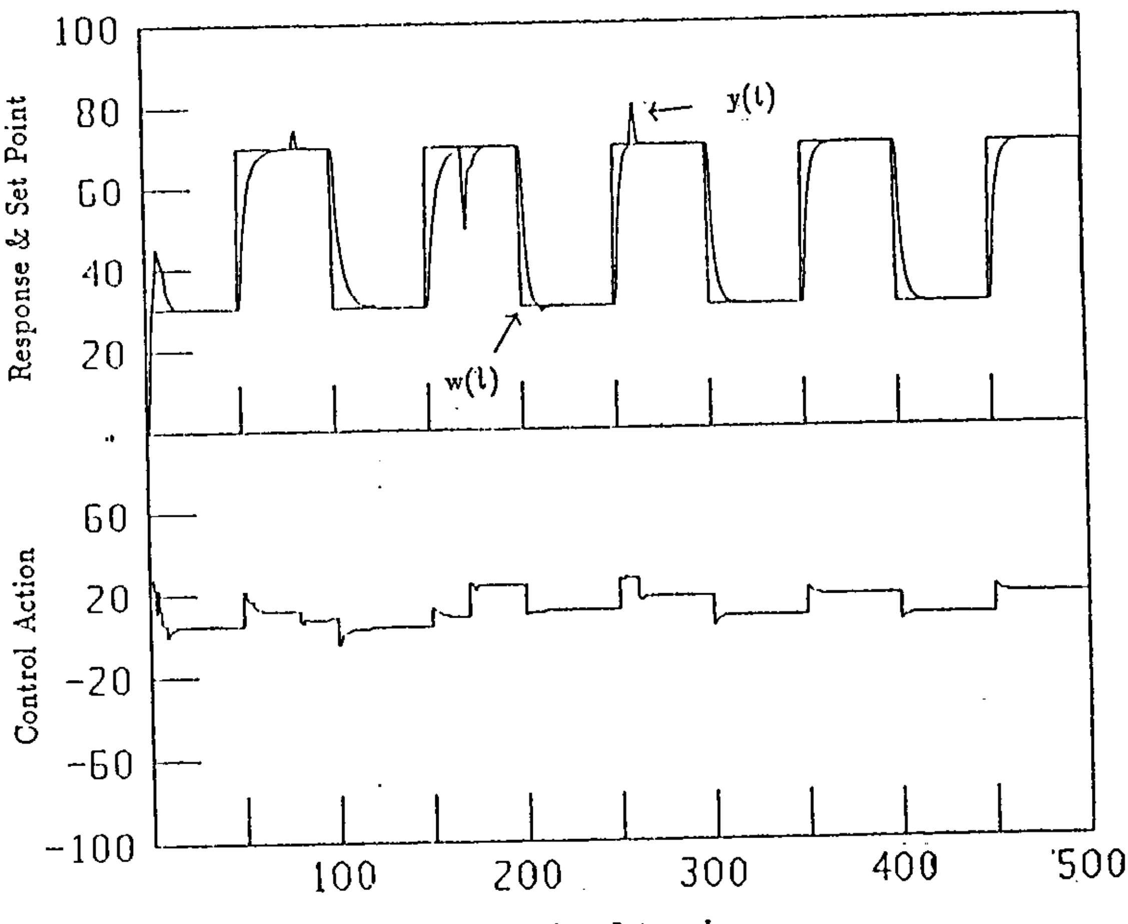

Figure 4.6: GPC With a Time-Variable System.

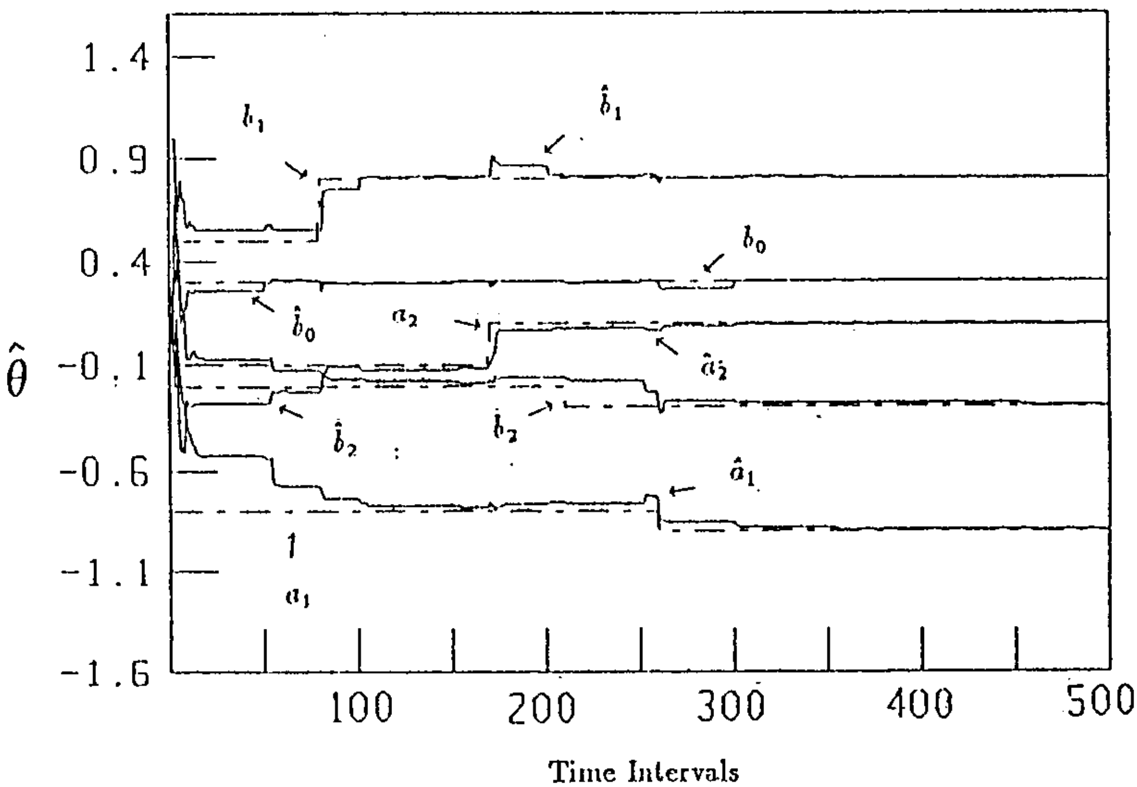

Figure 4.7: The Parameters Estimates For the Time Variable System.

the controller's behaviour in rejecting noise. In addition, GPC has the ability to deal with a time varying process based on the identification of the process model recursively.

In the following chapter,this controller is to be examined by applying it to control the cavity pressure in the injection molding process through simulation. And care should be taken in choosing the tuning knobs of the GPC and the RLS estimator, due to the complex nature of the process.

Chapter 5

APPLICATION OF GPC TO THE INJECTION MOLDING PROCESS

This chapter presents the application of GPC to control the filling stage of the injection molding process using cavity gate pressure. The deterministic model of cavity pressure that was developed by Abu Fara [2] will be used firstly in the simulation. However, this model suffers from some deficiencies, since it does not reflect the nonlinearity and time-variation of the process.

Therefore, and as a part of this study, a dynamic model of cavity pressure in the filling stage is obtained through identification. The model derivation is based on real data of cavity pressure and the corresponding control valve opening values taken from the work of Abu Fara [2]. The parameters of this model are not fixed, they are changing with time. Which shows some of the nonlinear and time-varying behaviour of the process. This model is to be used to evaluate the GPC in control of the cavity pressure.

To make the model represents the actual process more adequately, time delay and noise in the cavity pressure model is added and the response of GPC is studied.

The results of the GPC application are compared with that of a digital PID controller.

In the following section, we start with the application of the controllers to the fixed model.

5.1 Application to the Fixed Model

The deterministic model which was developed by Abu Fara [2] is considered here to represent the dynamics of cavity gate pressure related to the control valve opening. The model was introduced in chapter 2, and represented by the equation :

$$\frac{Pc(t)}{u(t)} = \frac{0.14z^{-1} + 0.17z^{-2} - 0.3z^{-3}}{1 - 1.91z^{-1} + 0.91z^{-2}} \tag{5.1}$$

This model was used in a simulation study to evaluate the performance of some conventional controllers such as the PID by Abu Fara [2] in control of cavity pressure. This controller will be examined here.

The continuous form of the PID controller is given by [10]:

$$u(t) = K_c[e(t) + \frac{1}{\tau_i}\int e(t)dt + \tau_d\frac{de}{dt}(t)] \tag{5.2}$$

where:

U(t) is the controller output at time t.

K_c is the controller gain

e(t) is the error (w(t)-y(t)), the difference between the set-point and the measured output.

τ_i,τ_d are the integral and derivatives times, respectively.

A discrete form of the PID controller can be obtained by replacing the integral in the continuous form by a rectangular integration and replacing the derivative by a finite difference [2, 10]. Thus, if the control action is digitally calculated every h seconds then a discrete form of the PID controller is given by:

$$\frac{u(t)}{e(t)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - z^{-1}} \tag{5.3}$$

where:

$a_0 = K_c[1 + \frac{h}{\tau_i} + \frac{\tau_d}{h}]$

$a_1 = -K_c[1 + 2\frac{\tau_d}{h}]$

$a_2 = K_c\frac{\tau_d}{h}$

The values of the PID controller parameters used are the optimum settings which were calculated for the same model and given in reference [2] as follows:

$K_c = 0.782$ $\%controlvalveopening/psi$

$\tau_i = 0.062$ sec

$\tau_d = 0.009$ sec.

On the other hand, the experiments of GPC employ the RLS estimation algorithm with variable forgetting factor to obtain on-line estimates of the model parameters. The initial settings were:

$\hat{\theta}(0) = [0, 0, 0, 1]$

$P(0) = 10I$

$\beta_{min} = 0.9$.

Four parameters were estimated instead of five, since the denominator of the model equation contains a difference operator ($\Delta = 1 - z^{-1}$) due to the ramping of the cavity pressure, the process model can be written as:

$$\frac{Pc(t)}{u(t)} = \frac{0.14z^{-1} + 0.17z^{-2} - 0.3z^{-3}}{\Delta(1 - .91z^{-1})} \tag{5.4}$$

As a first application, a ramp followed by a step-change and another ramp set-point profile is chosen to show the difference between the PID and GPC performances, and to compare with the results of Abu Fara [2]. Figure 5.1 shows the response of cavity pressure under PID control. The sampling time is chosen to be $.01 sec$. The Figure shows the filling stage of the process excluding the early filling period in which the delivery system is filled. It is seen in the figure that the PID performed well, since the suitable parameters were chosen in [2], except at the initial stage and the step region, where the PID controller needs a long time to track the set point correctly.

Figure 5.2 shows the cavity pressure response using the GPC. The parameters of the GPC are chosen to be: [0.1,1,4,2] for [λ,N1,N2,NU]. The GPC performance is much better than that of the PID. The initial stage performance is much better than that of the PID. Moreover , the step-change behaviour of the GPC is completely different from that of the PID. The GPC started to compensate for the step change before its occurring, with a small error, due to the long-range prediction property of the GPC.

A performance index which measures the optimality of the control algorithm is adopted to serve as a comparison formula between the controllers. This performance index is chosen as follows:

$$PQ = \sqrt{\sum_{i=1}^{N}[(y-w)^2 + \Delta u^2]} \tag{5.5}$$

Where:y,w,Δu are the system response, the set-point and the control increment respectively. The first term in the sum represents the error between the response and the set-point, where the second term corresponds to the control signal activity. A smaller value of the performance index means better controller performance.

This performance index is calculated for both controllers, the values are shown along with the figures. The PQ value of GPC is lower than that of the PID. This shows that

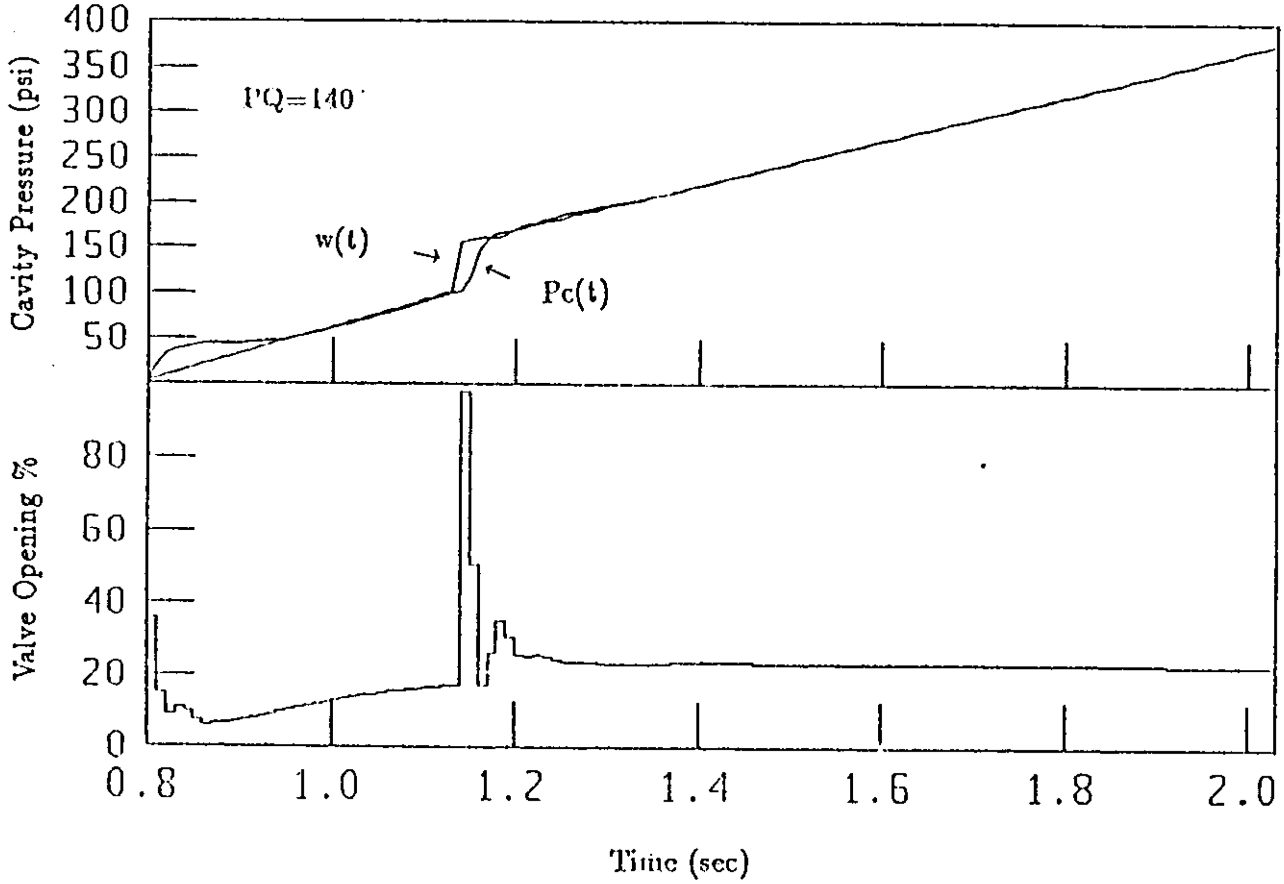

Figure 5.1: Response of Cavity Pressure to a Ramp -Step-Ramp Set Point Profile With the PID Controller.

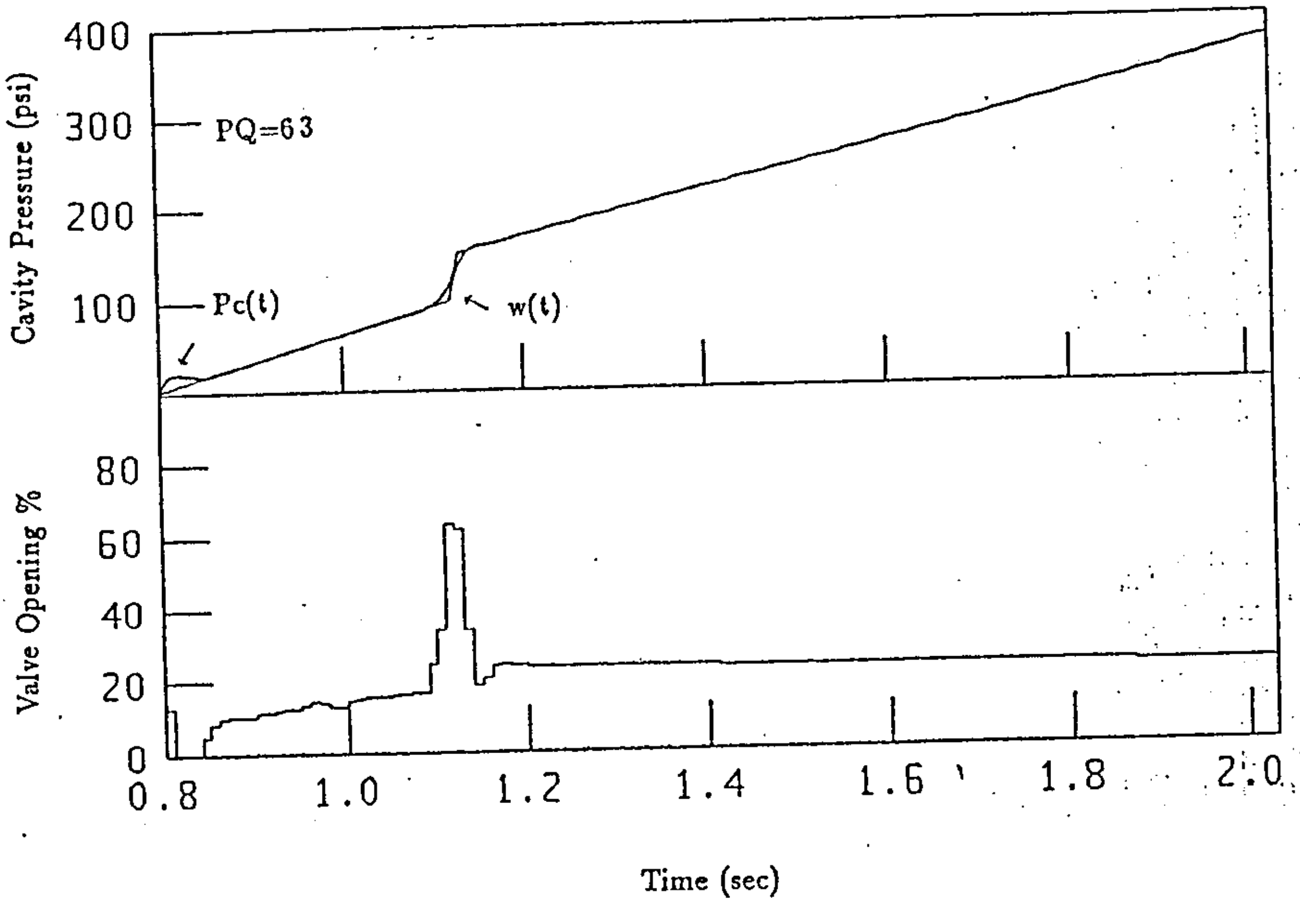

Figure 5.2: Response of Cavity Pressure to a Ramp -Step-Ramp Set Point Profile With the GPC Controller.

the performance of GPC is better than the PID.

The set-point profile used was chosen for testing and comparison purposes . But in real application the pressure profile determines the quality of the injection molding articles . A set-point profile identical to the true trend of cavity pressure in an actual operation is adopted , it is assumed that this set-point profile will give the desired quality of the plastic articles . Figure 5.3 shows the performance of the PID controller with this set-point profile, it adequately follows this profile, in addition to the smooth control signal variations. This performance of the PID is adequate because of that the process model used to represent the cavity pressure is fixed. Furthermore, the optimum settings of the PID controller are used. The results of GPC application are shown in Figure 5.4 with the controller settings values are [0.5,1,3,1] for [λ,N1,N2,NU]. It is as good as that of the PID controller except at the initial stage. The initial excursions in the control signal are due to the poor estimation of parameters initially, but after that the control signal and the response are satisfactory.

Performance index values indicate that the PID is slightly better than the GPC due to the mentioned reasons. However, this is an advantage for the GPC, because it gave the same performance of a well-tuned PID controller, without any prior knowledge of the process.

Figures. 5.5 and 5.6 shows the simulation results of PID and GPC respectively assuming a time delay of two sampling periods [2]. The PID results show an oscillatory behaviour in the control signal and in the response. This is attributed to the presence of the time delay which adds complexity to the dynamic behaviour of the process, leading to detuning the PID controller.

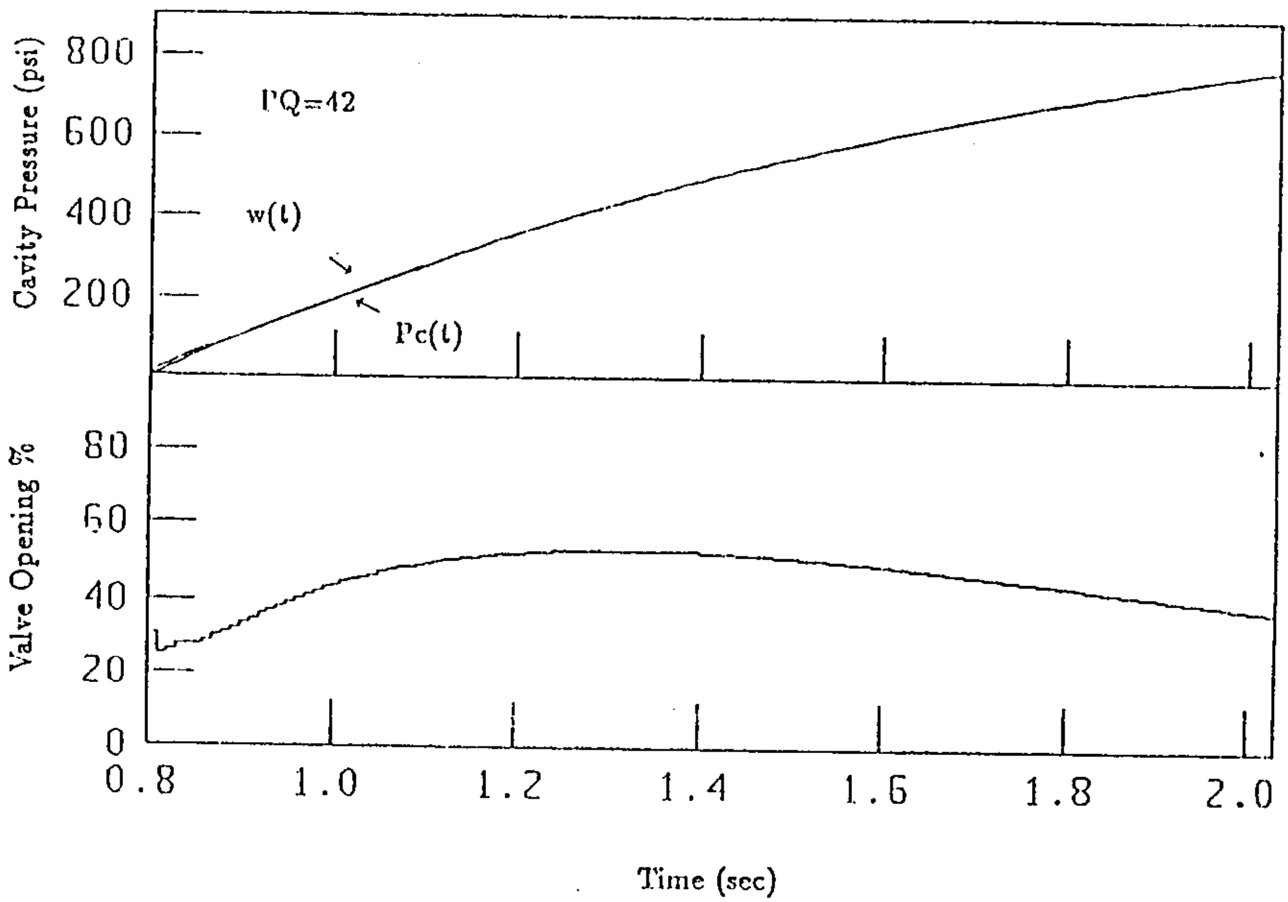

Figure 5.3: Response to an Exponential Set Point Profile With the PID Controller.

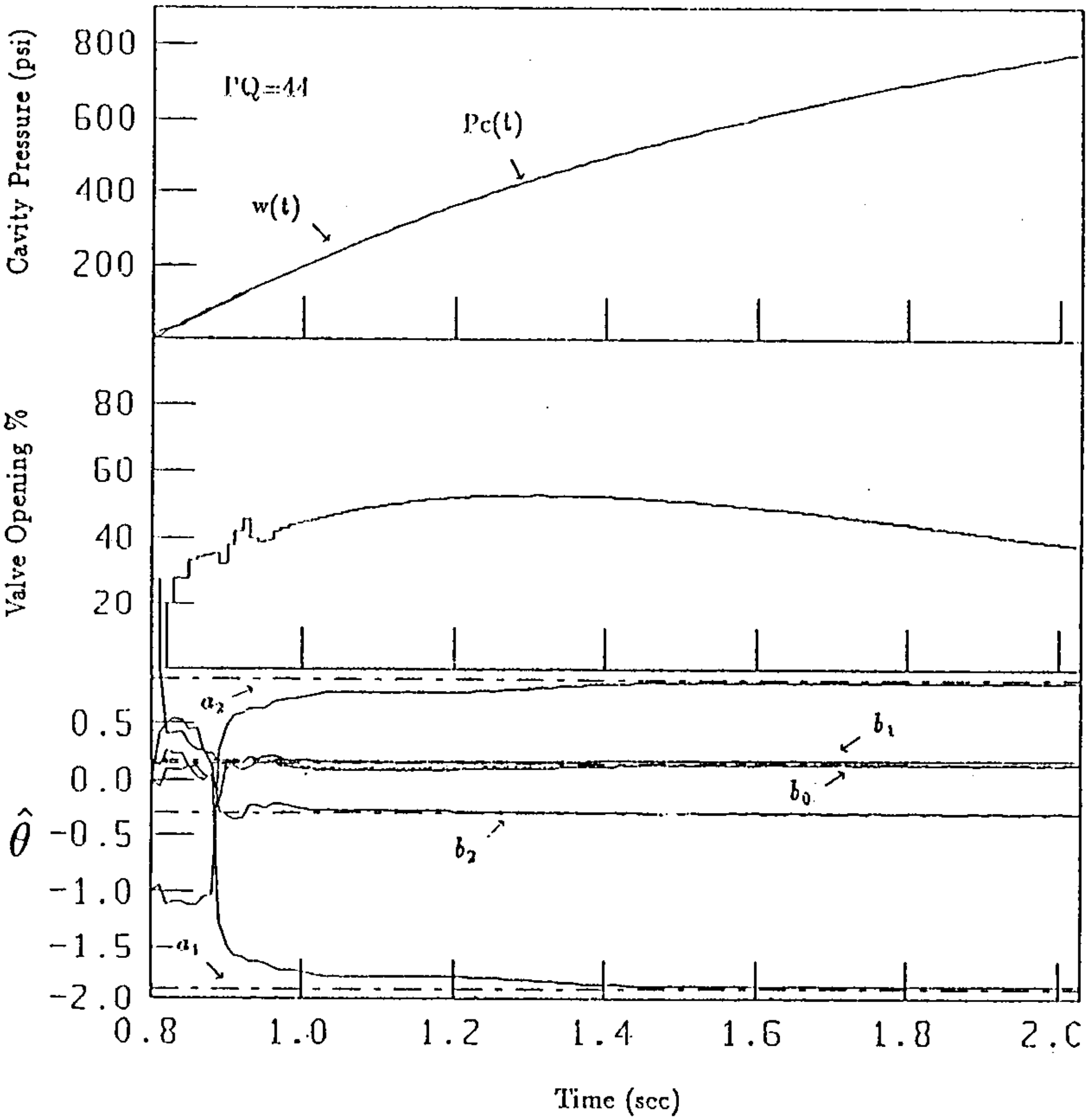

Figure 5.4: Response to an Exponential Set Point Profile With the GPC Controller.

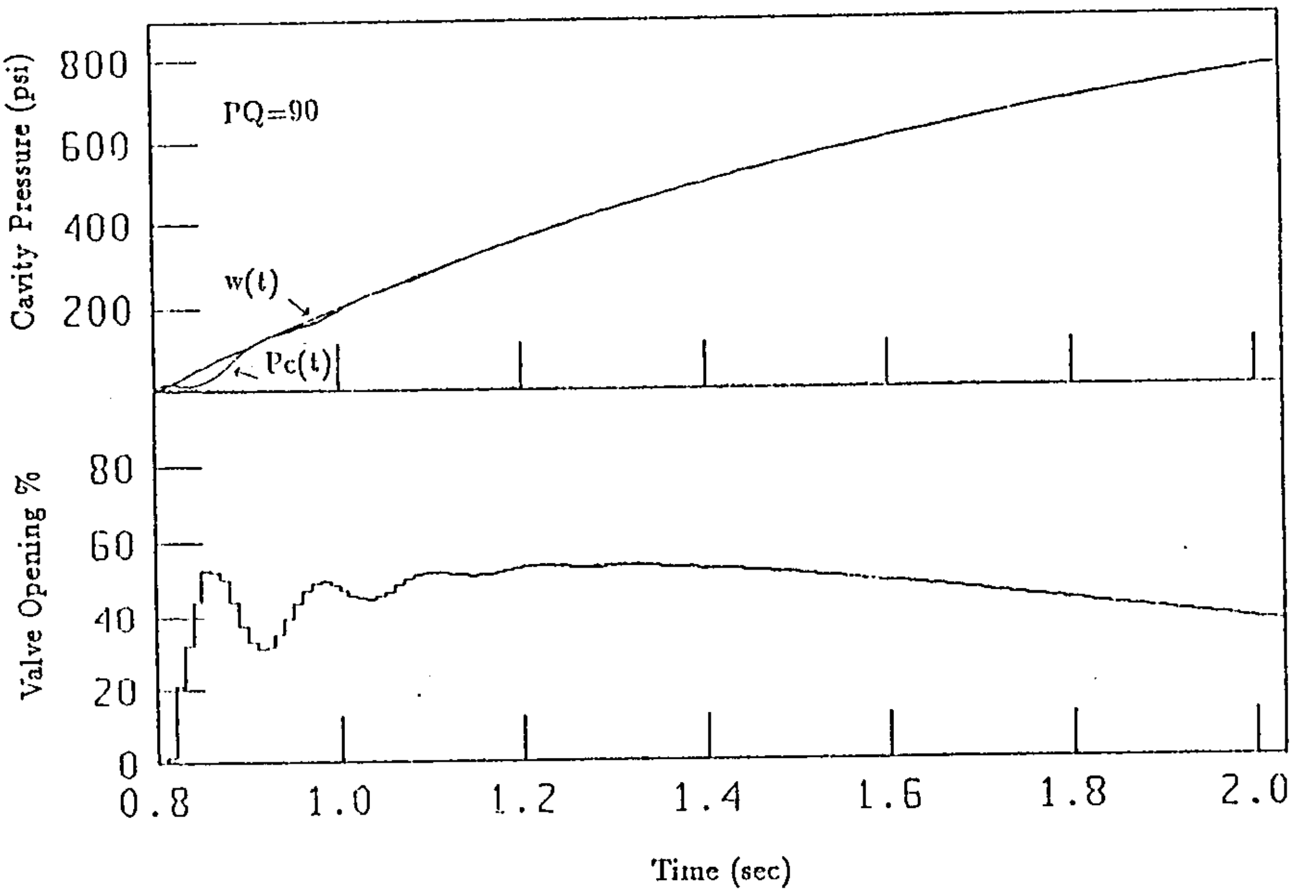

Figure 5.5: Effect of Time Delay With the PID Controller.

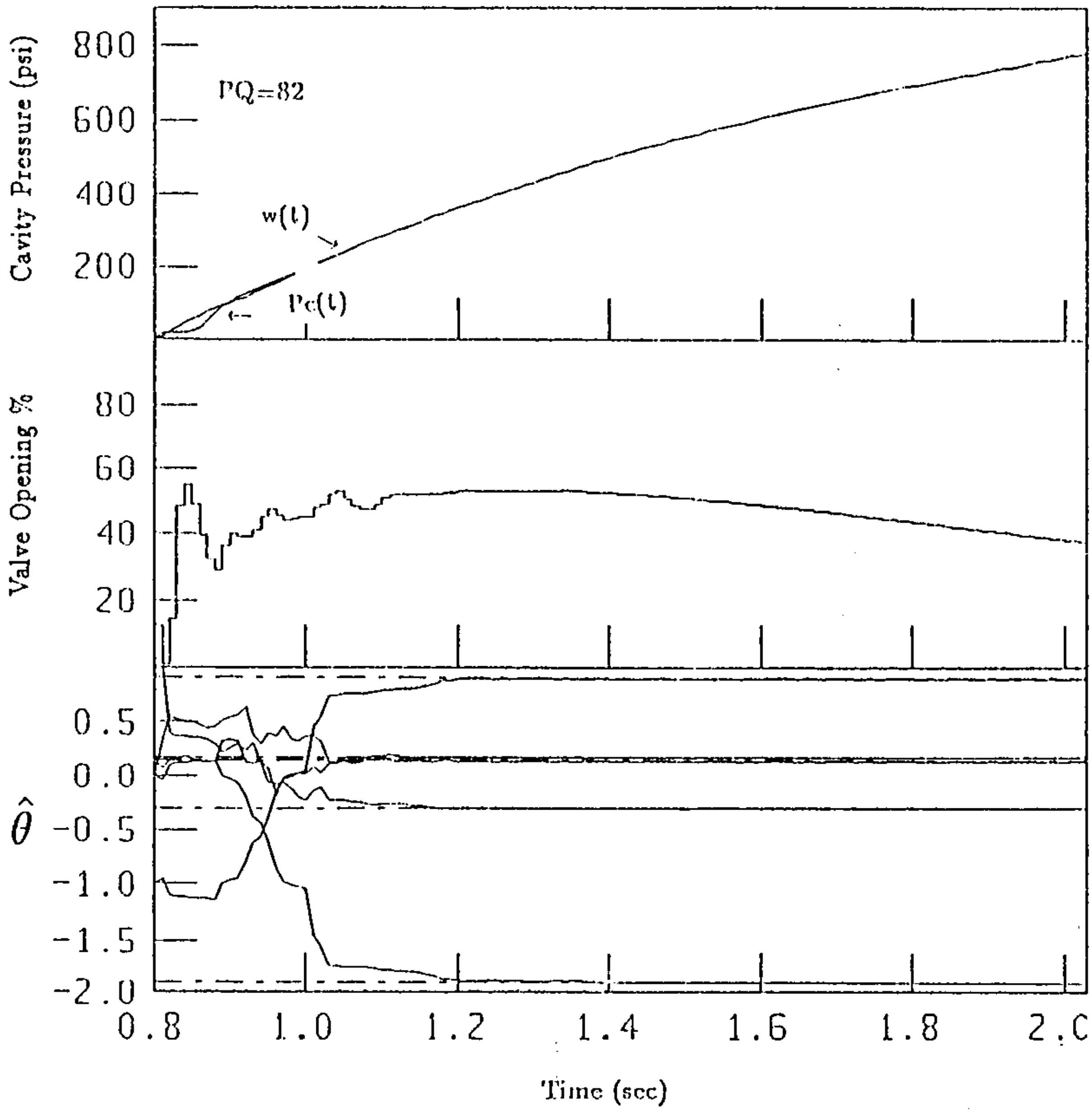

Figure 5.6: Effect of Time Delay With the GPC Controller.

The GPC simulation results for the same case are shown in Figure 5.6. The parameters are chosen to be [0.5,3,4,1] for [λ,N1,N2,NU]. The set-point tracking is similar to that of the PID. At the initial stage the control output is slightly active due to poor estimation.

Performance index value of GPC is smaller than the PID value indicating a better performance. However, the performance of the GPC can be altered by manipulating the tuning knobs.

Figure 5.7 shows that with increasing the prediction horizon (N2), setting its value to 10 and keeping [λ,N1,NU] at [0.5,3,1], a smoother control output is obtained, but the set point tracking at the initial stage is not tight. However, the performance index value shows in digits the less suitable control obtained due to increasing the control horizon.

Figure 5.8 shows the effect of increasing the control horizon (NU). The values of [λ,N1,N2,NU] are chosen to be [0.5,3,10,2]. In this case, the set pint tracking is better than the previous case due to the more active control output. The performance index value ensures the modification.

To show the effect of control weighting the following values [0,3,4,1] were chosen for the controller parameters [λ,N1,N2,NU]. Figure 5.9 shows thar the decrease in the value of the control weighting gives more active control(compared to Figure 5.6). This is clear from the figure, and leads to increasing the performance index (PQ) value remarkably.

The performance index value can help in deciding, which set of the controller parameters should be chosen.

The noisy nature of the injection molding process inspired the addition of noise with variance (1) to the system output at each time instant. Figure 5.10 shows the results of PID controller application. The PID controller do not rely on an on-line model estimation,

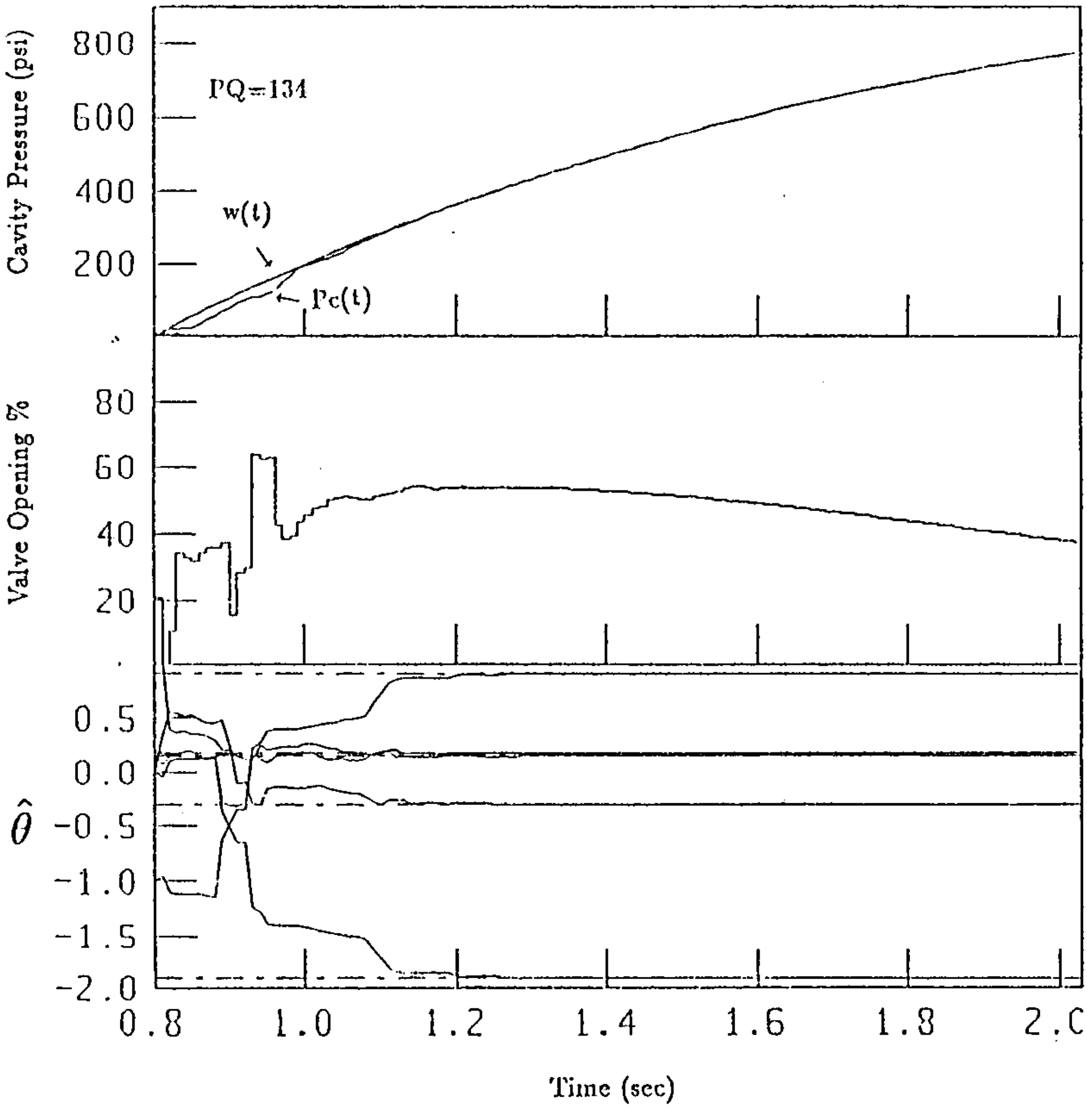

Figure 5.7: Effect of the Prediction Horizon.

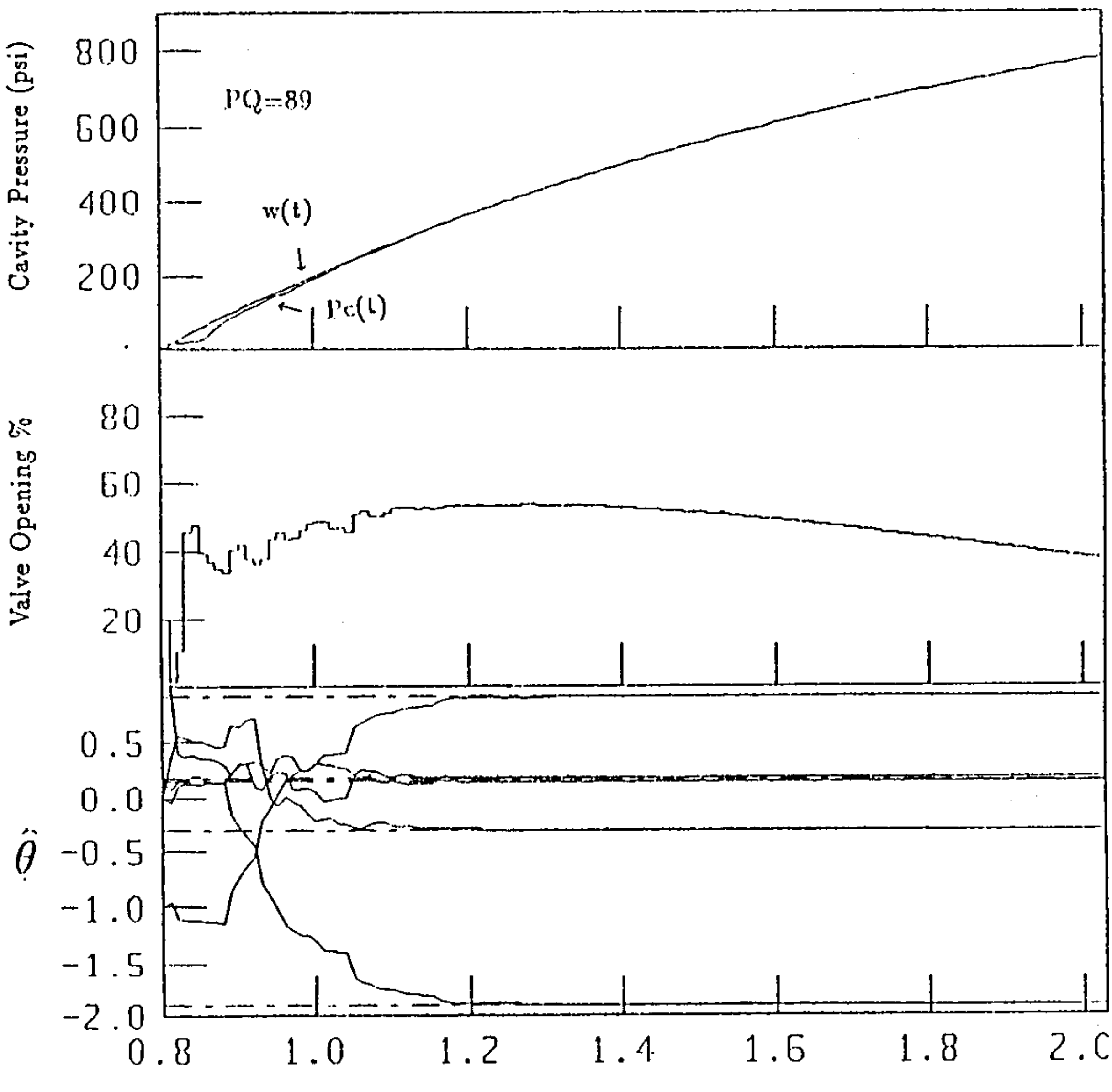

Figure 5.8: Effect of the Control Horizon.

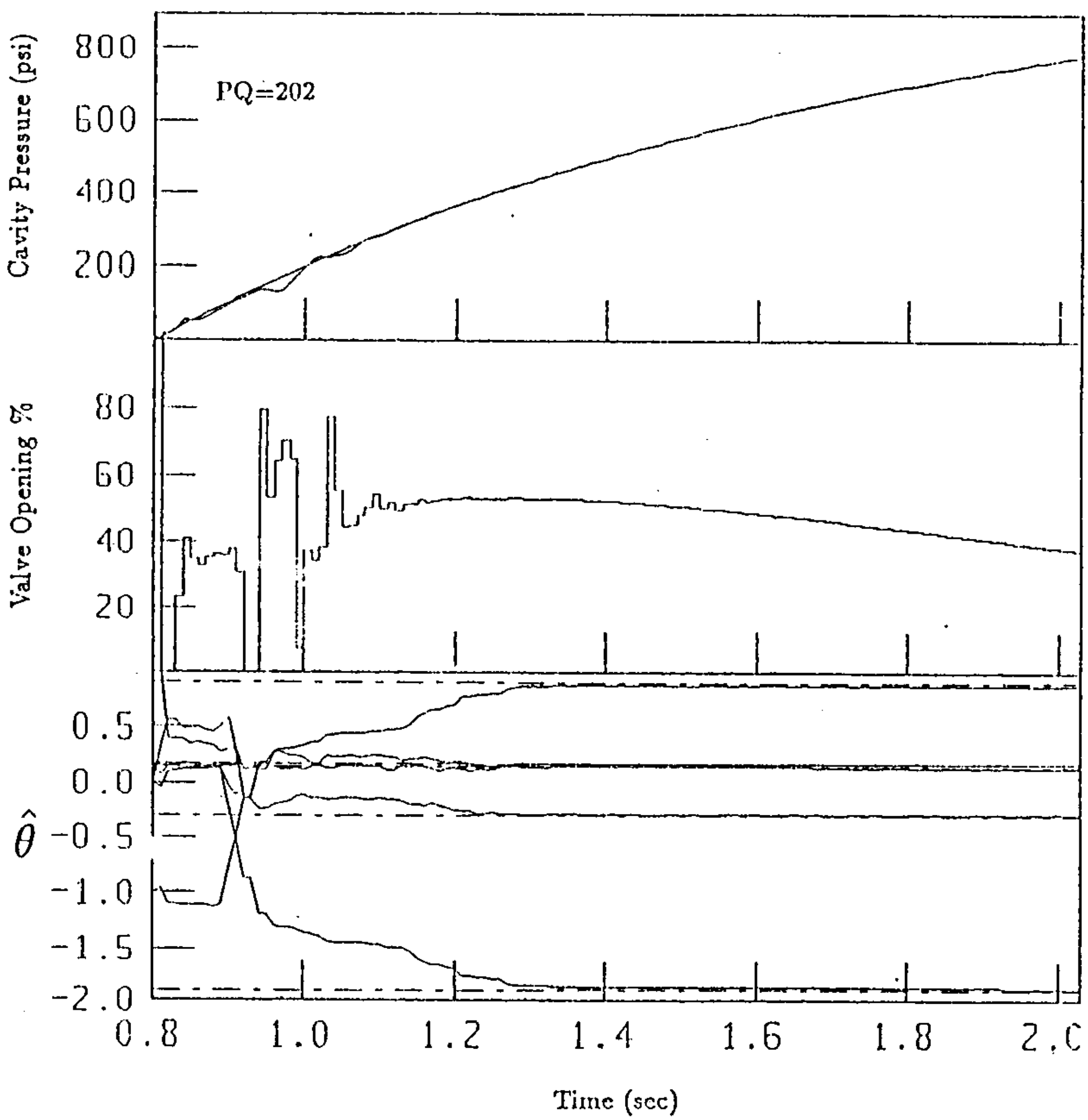

Figure 5.9: Effect of Control Weighting.

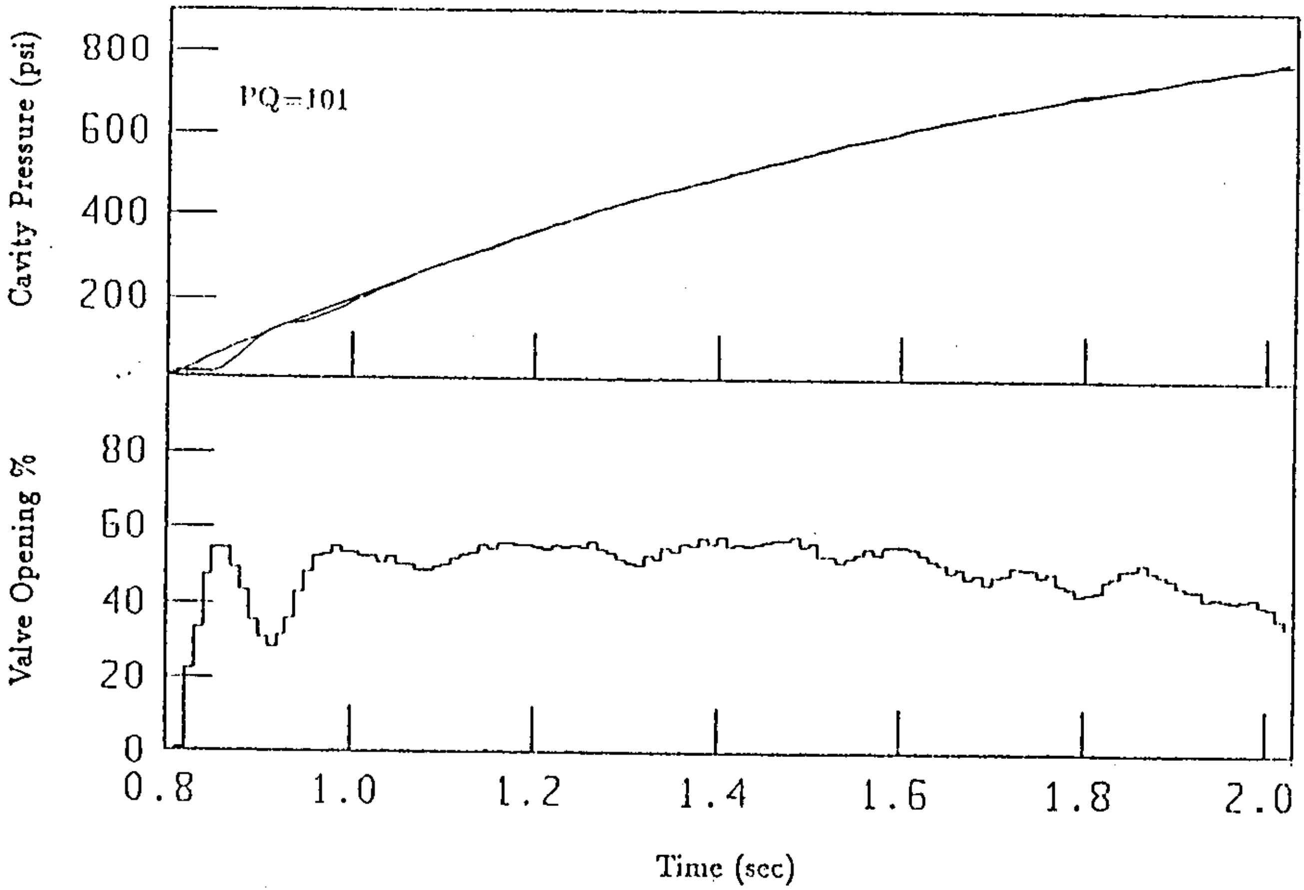

Figure 5.10: Effect of Noise With the PID Controller.

it considers the noise as an error that to be corrected. Figure 5.11 shows the application of the GPC controller without using data filtering. The sensitivity of the estimator was decreased by setting: P(0)=.1 and $\lambda_{min} = 0.995$, to minimize the effect of noise. But the effect of noise on the estimator is remarkable as seen from the figure. The dependency of GPC on the quality of the estimated model causes some excursions in the control signal, since the estimation is not effective in the presence of noise, this was discussed in chapter 3.

The performance of the GPC is improved by use of the data filter. This is obvious from Figure 5.12 . The use of the T-filter improved the estimation quality which affects the GPC.

The performance index values show the improvement in GPC performance with data filtering compared to the case without filtering. In addition, the PID was not better too much than the GPC (without filtering) as indicated by the PQ values.

The deterministic model used was useful in examining the controllers and in describing part of the dynamic behaviour of cavity pressure. But it can't be considered as a true or a complete representation of the dynamic behaviour of the cavity pressure. Because it do not reflect the nonlinearity and time-variation. So, a more realistic model is needed, which gives more information about the dynamic behaviour of the process. In the following section, the development of such a model is discussed.

5.2 Identification of a Time-Varying Model for Cavity Pressure

To evaluate the GPC with a more realistic model than the deterministic one, a group of data files containing real readings of cavity gate pressure and control valve opening

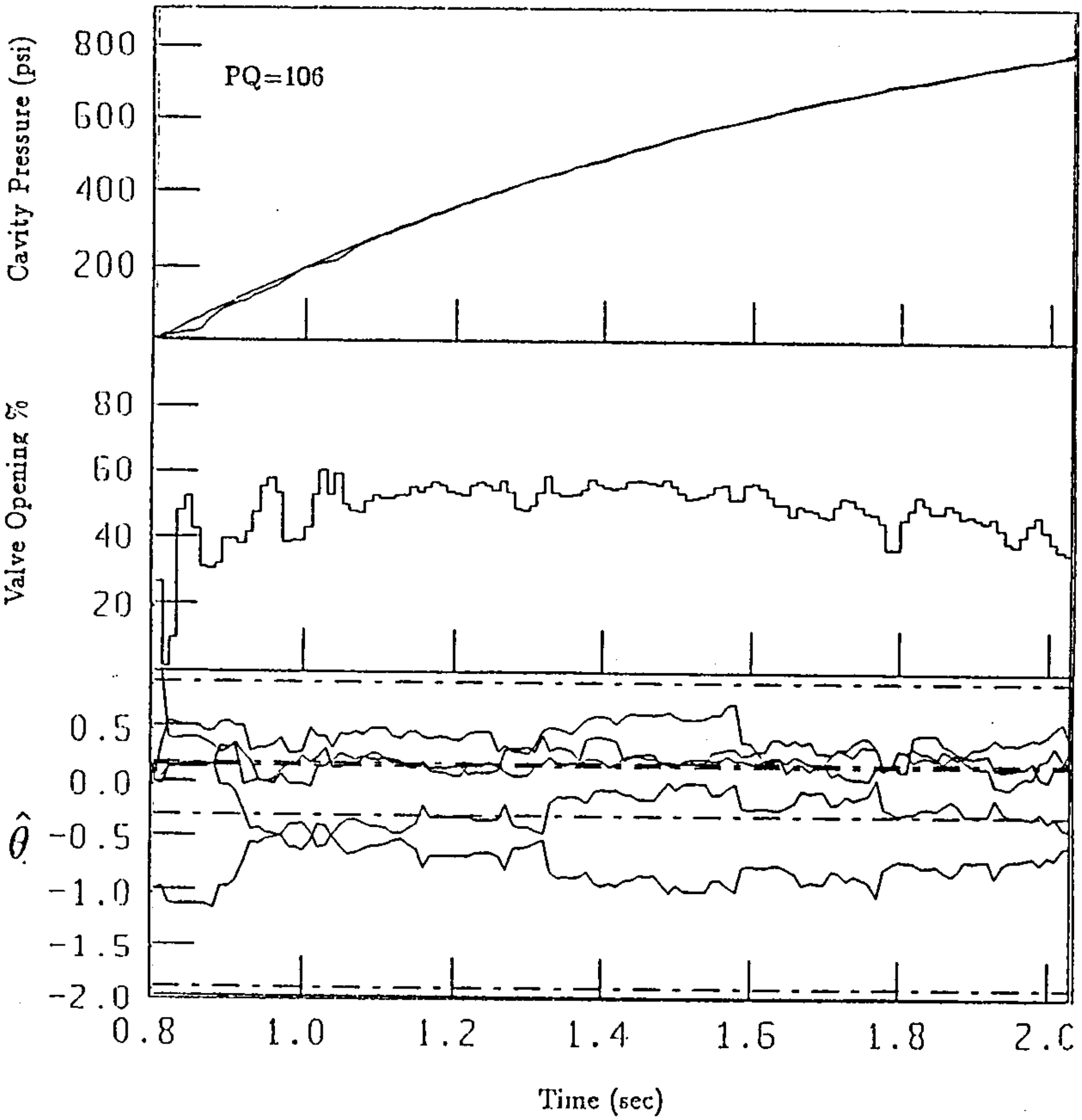

Figure 5.11: Effect of Noise With the GPC Controller Without Data Filtering(T=1).

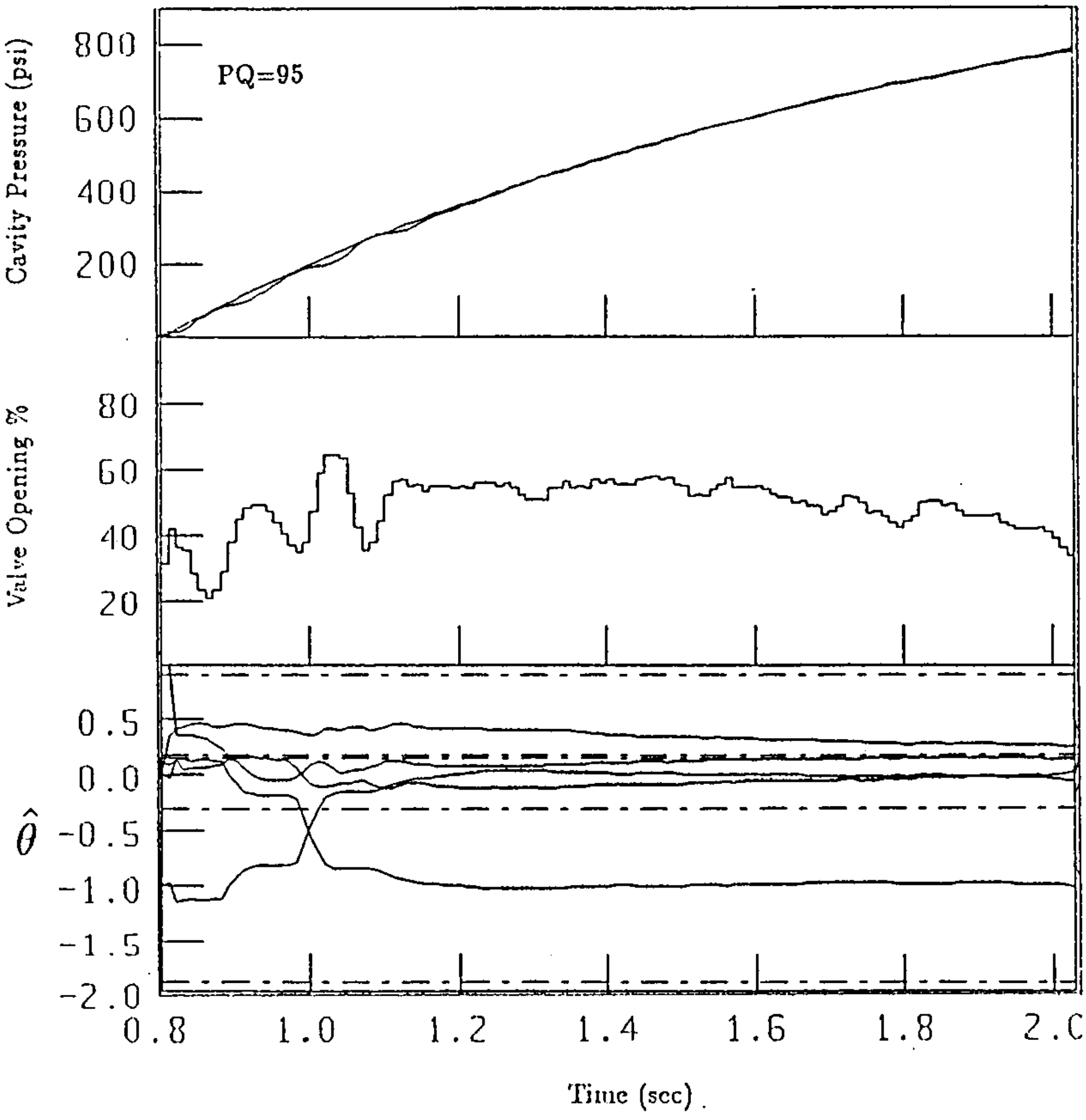

Figure 5.12: Effect of Noise With the GPC Controller

With Data Filtering($T=1-0.8z^{-1}$).

were taken from the work of Abu Fara [2]. These data files were used for stochastic identification of cavity pressure [2]. The input signals to the control valve opening were Pseudo-Random-Binary-Sequences (PRBS). This type of signals is popular for the identification of system dynamics, due to the advantage of these signals in exploring the dynamic modes of the process.

The identified model structure is identical to the form of the deterministic model, given by:

$$\frac{Pc(t)}{u(t)} = \frac{b_0 z^{-1} + b_1 z^{-2} + b_2 z^{-3}}{\Delta(1 + az^{-1})} = \frac{b_0 z^{-1} + b_1 z^{-2} + b_2 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{5.6}$$

Where: $a_2 = -a$ and $a_1 = a - 1$.

The parameters of this model were identified recursively using the RLS estimation algorithm, for several data files representing different test runs. The identification results of one of the data files are shown in Figure 5.13 . The other experiments show the same trend for parameters estimates.

Considering Figure 5.13, the parameters of $B(z^{-1})$ polynomial (i.e . b0,b1 and b2) can be approximated into two intervals as follows: in the interval t=.8 to t=1.14 seconds the $B(z^{-1})$ parameters are:

$b_0 = b_1 = b_2 = 0.1$

and in the interval t=1.14 to t=2.10 seconds these parameters are as follows:

$b_0 = 0.25 \quad b_1 = -0.14$

$b_2 = 0.05$

averaging is used in these two intervals for the $B(z^{-1})$ parameters.

The parameter a is changing continuously with time, it can be approximated by a first order line with the following form:

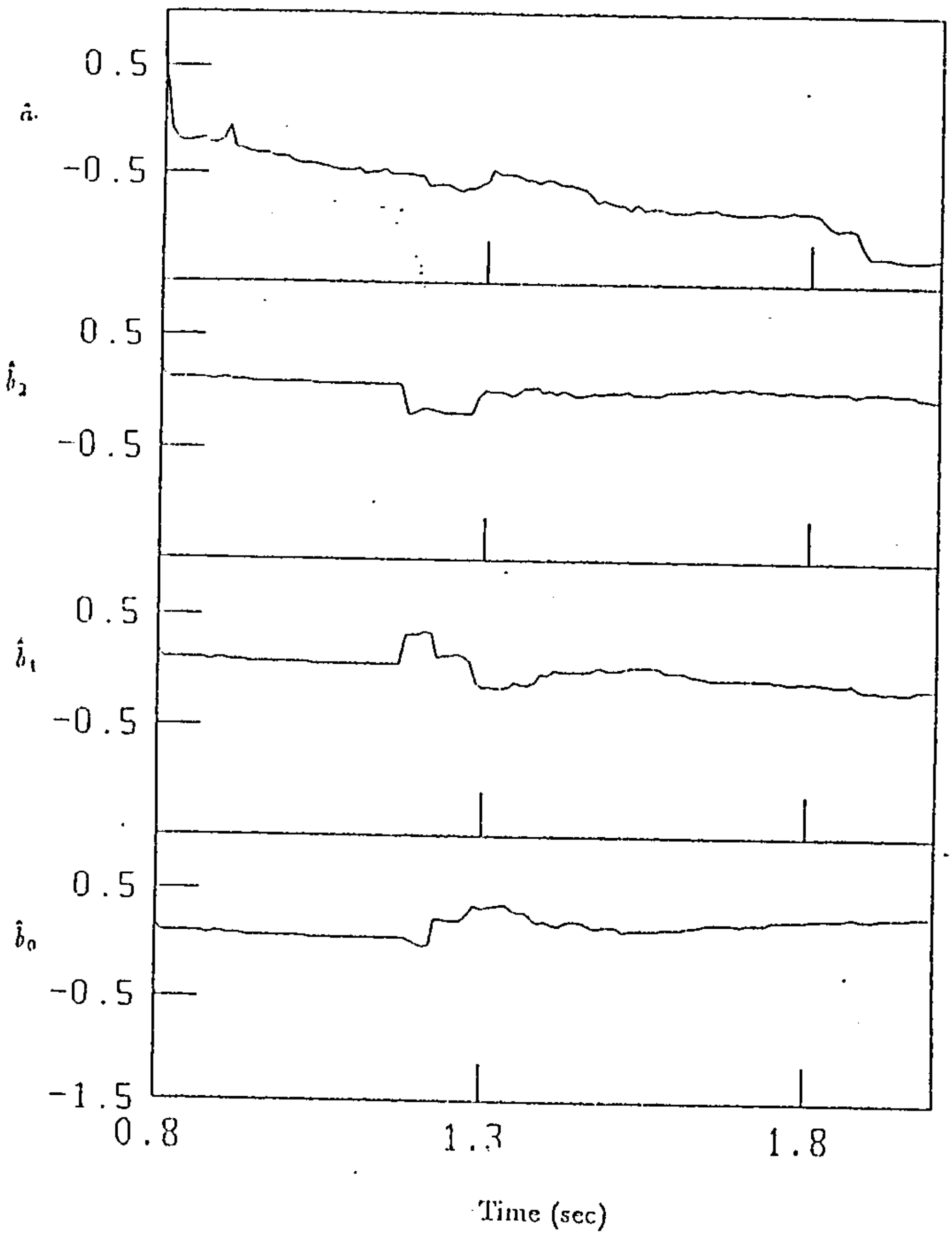

Figure 5.13: The Identified Parameters of the Cavity Pressure Model as a Function of Time.

$a = 1.42 - 0.81t$.

To check the model accuracy, the cavity pressure values were reconstructed using the model developed with the same input signals (PRBS) present in the data file. Figure 5.14 shows the good agreement between the real cavity pressure values and the reconstructed ones.

5.3 Application to the Identified Model

The PID controller is applied to control the cavity pressure using the identified model. The parameters of the PID are the same parameters used in the previous section. Figure 5.15 displays the results of PID application. It is evident that the PID with its fixed parameters can't track the set-point profile as required. Oscillations are seen in the response and in the control signal too. The performance of the PID controller means that the controller settings are not the optimal ones for all the operating conditions of the process, i.e. the controller needs to be retuned for each operating condition.

Gain scheduling for the PID can be used by choosing certain controller parameters for each operating condition. Abu Fara [2] used this scheme by dividing the filling stage into three parts. In each part, the PID parameters are different from the other parts. Better performance than the fixed PID was obtained.

GPC behaviour is shown in Figure 5.16 with the parameters [λ, N1,N2,NU] are chosen as [0.5,3,10,2]., the exact tracking of the set point is obvious, except at the beginning of the experiment and in the region of changing the gain of the process (at t=1.14 sec. the $B(z^{-1})$ parameters change) due to poor estimation of the model. The Figure shows the model parameters and the estimated ones, $B(z^{-1})$ parameters are identified almost

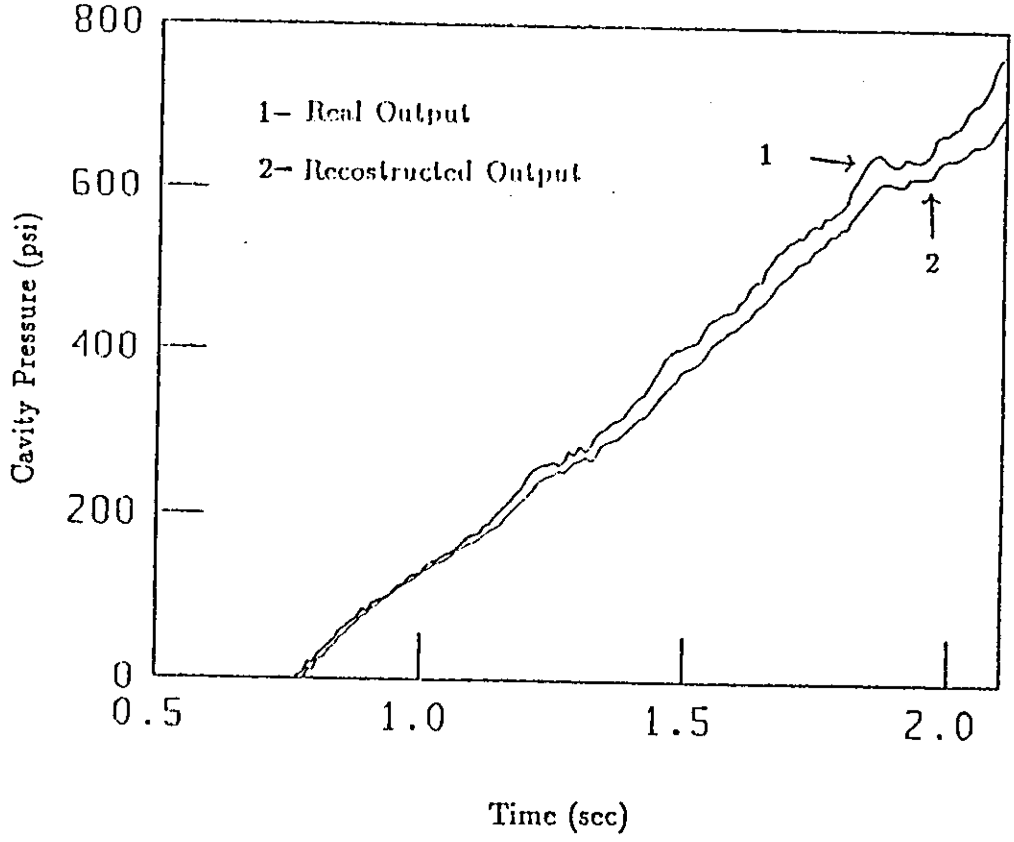

Figure 5.14: Comparison Between the Real Output From the Data File and the Output From the Identified Model Using the Same Input Signals.

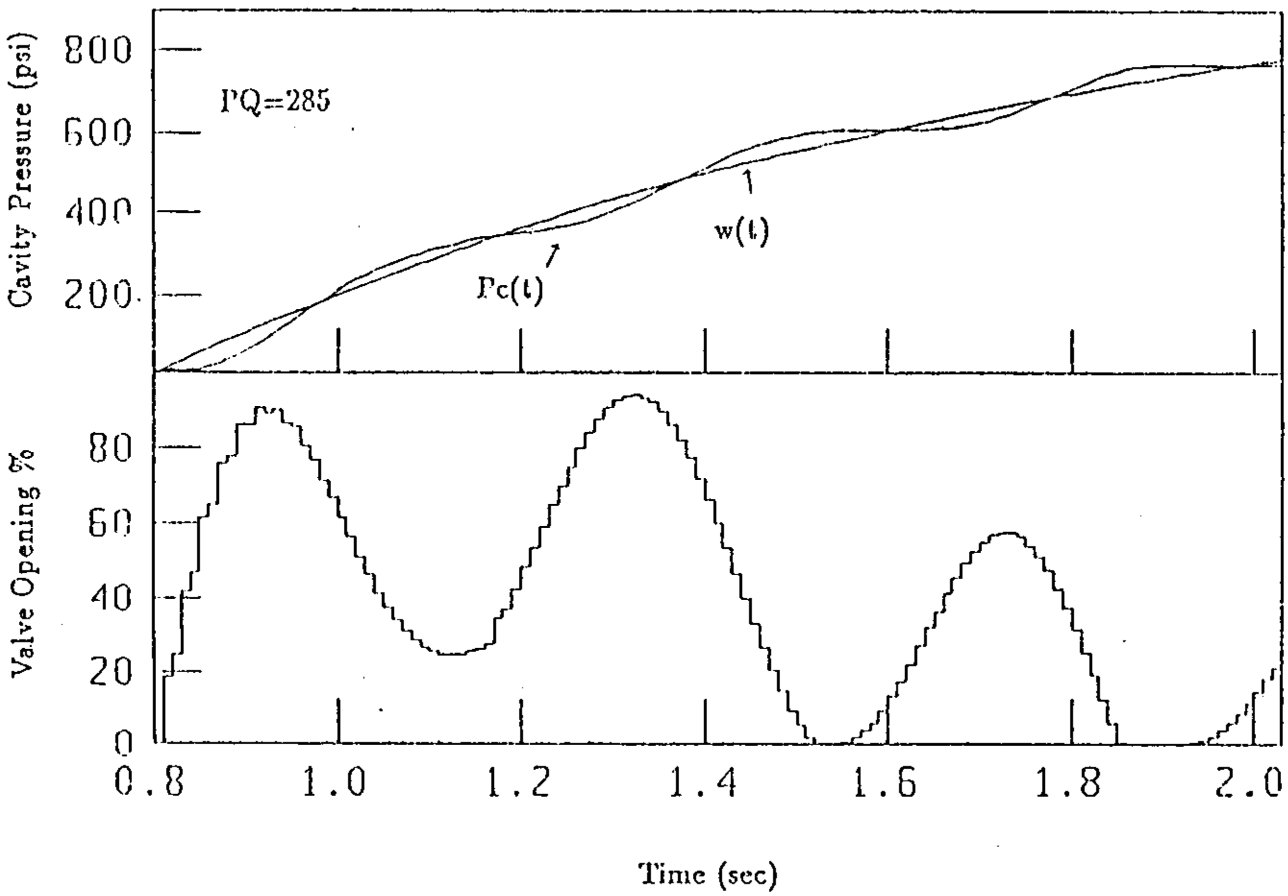

Figure 5.15: Application of the PID Controller to the Identified Model (With Averaging to the Parameters).

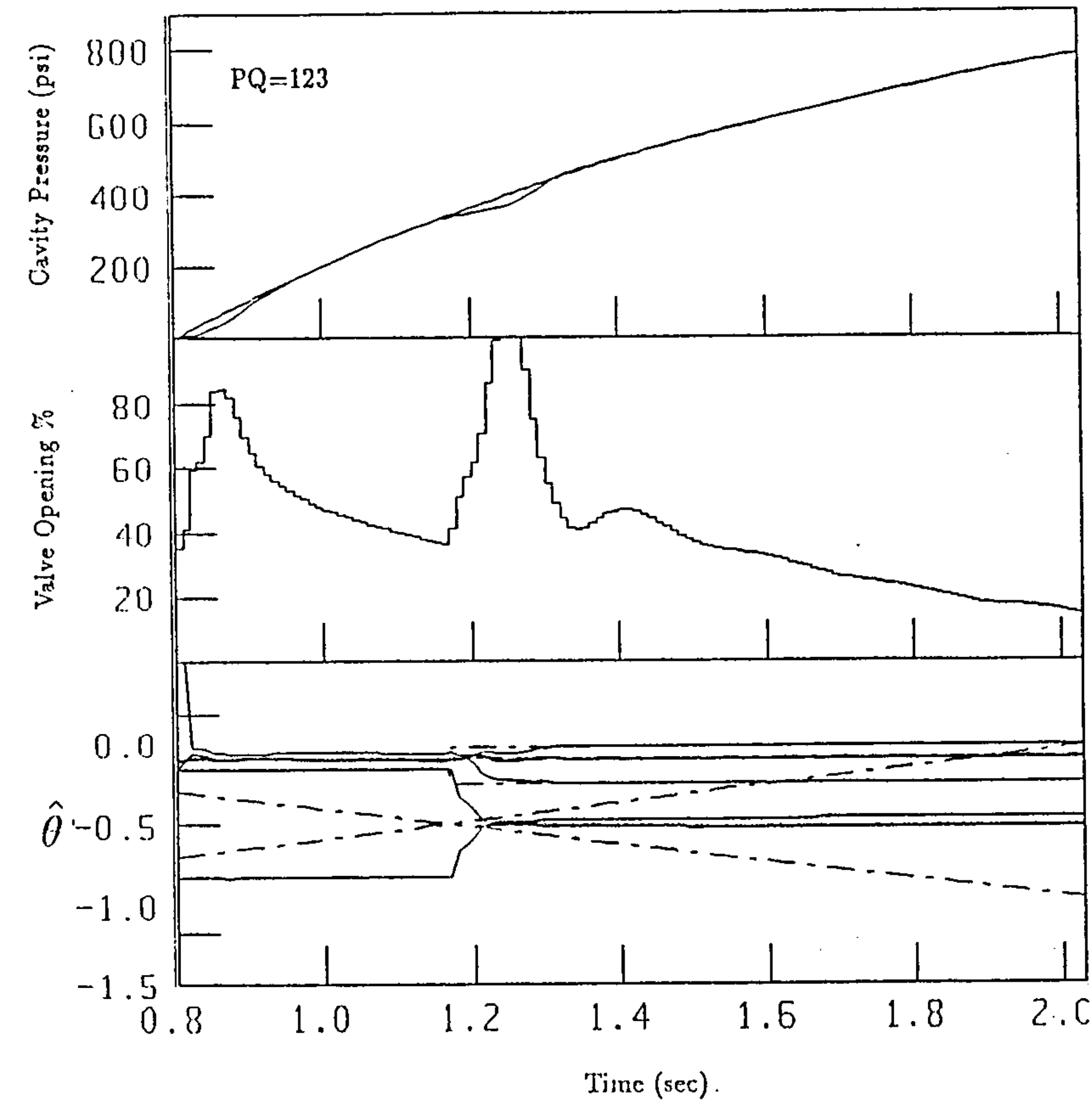

Figure 5.16: Application of the GPC Controller to the Identified Model (With Averaging to the Parameters).

exactly, but $A(z^{-1})$ (a_1 and a_2 are plotted) parameters are not, since they are changing with time continuously, and data excitation is not sufficient to activate the estimator enough. But, with non correct estimation of the $A(z^{-1})$ parameters the GPC was able to function adequately. This is due to the presence of the Δ operator in the model which decrease the effect of $A(z^{-1})$ parameters on the estimated model. Moreover, it is usually sufficient to estimate a model that approximately represent the process.

Performance index values show the wide difference between the two controllers.

To see the effect of noise on the performance of the controllers with the developed model, noise with variance (1) was added to the system output. The behaviour of the PID for this case is shown in Figure 5.17, it is obvious that the PID produces an active control output to follow the required set-point in addition to the oscillatory behaviour due to presence of several factors: changing dynamics, time delay and noise. On the other hand, the GPC was applied for the same case with data filtering, the performance was satisfactory as what can be seen in Figure 5.18, the output follows the set point much better than the PID. The noise effect was decreased by using data filtering. The T-polynomial was chosen to be ($T = 1-0.8z^{-1}$) , leading to better estimation and control action.

The final application of the controllers is done using the identified parameters shown in Figure 5.13, i.e. the identified parameters at each instant were used without approximation or averaging. Figure 5.19 shows PID application to this model, it is obvious that the controller was unable to track the set-point at the initial stage and produces large oscillations. But with time, the performance of the PID is improved indicating that the controller settings are adequate for those operating conditions. The performance of GPC

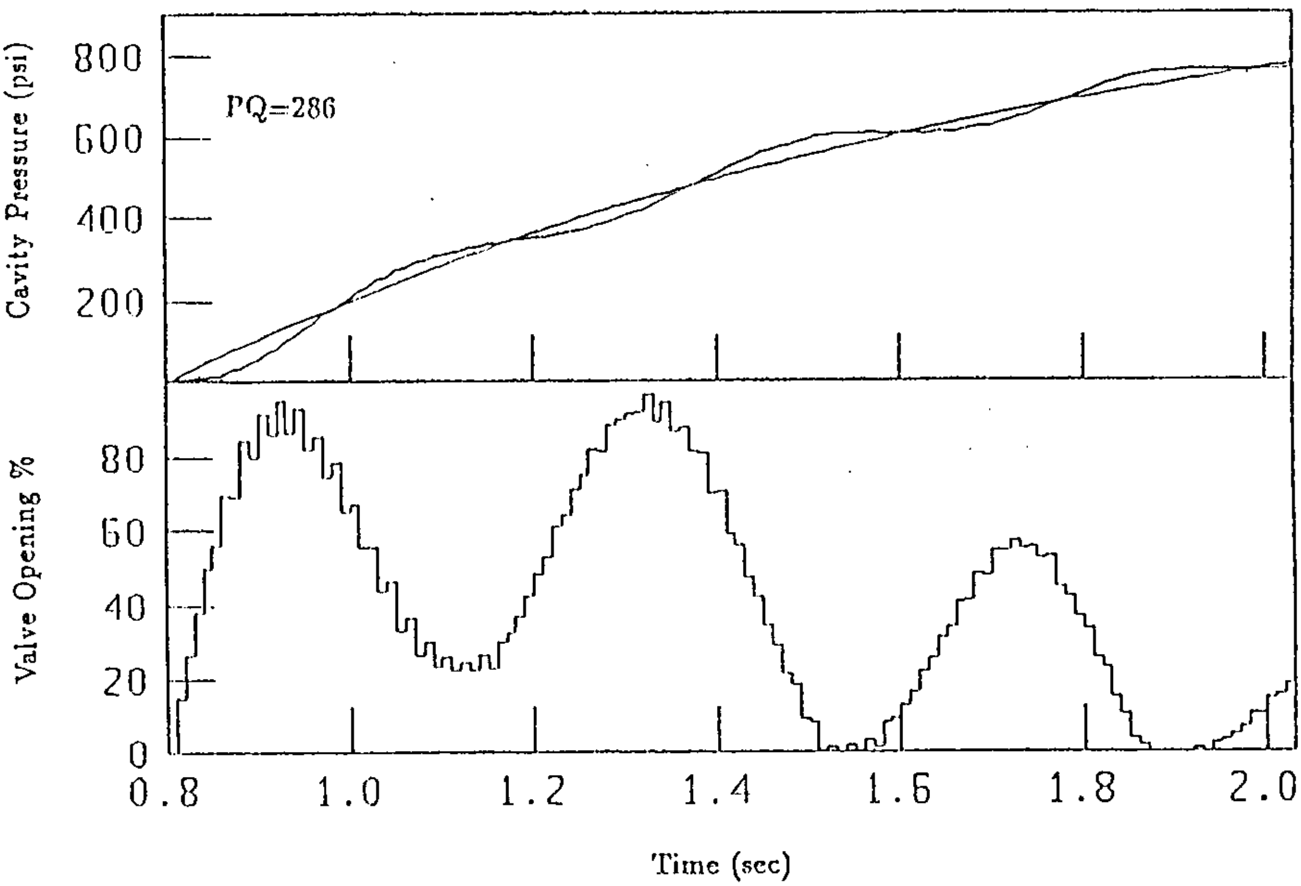

Figure 5.17: Effect of Noise on the Identified Model With the PID Controller.

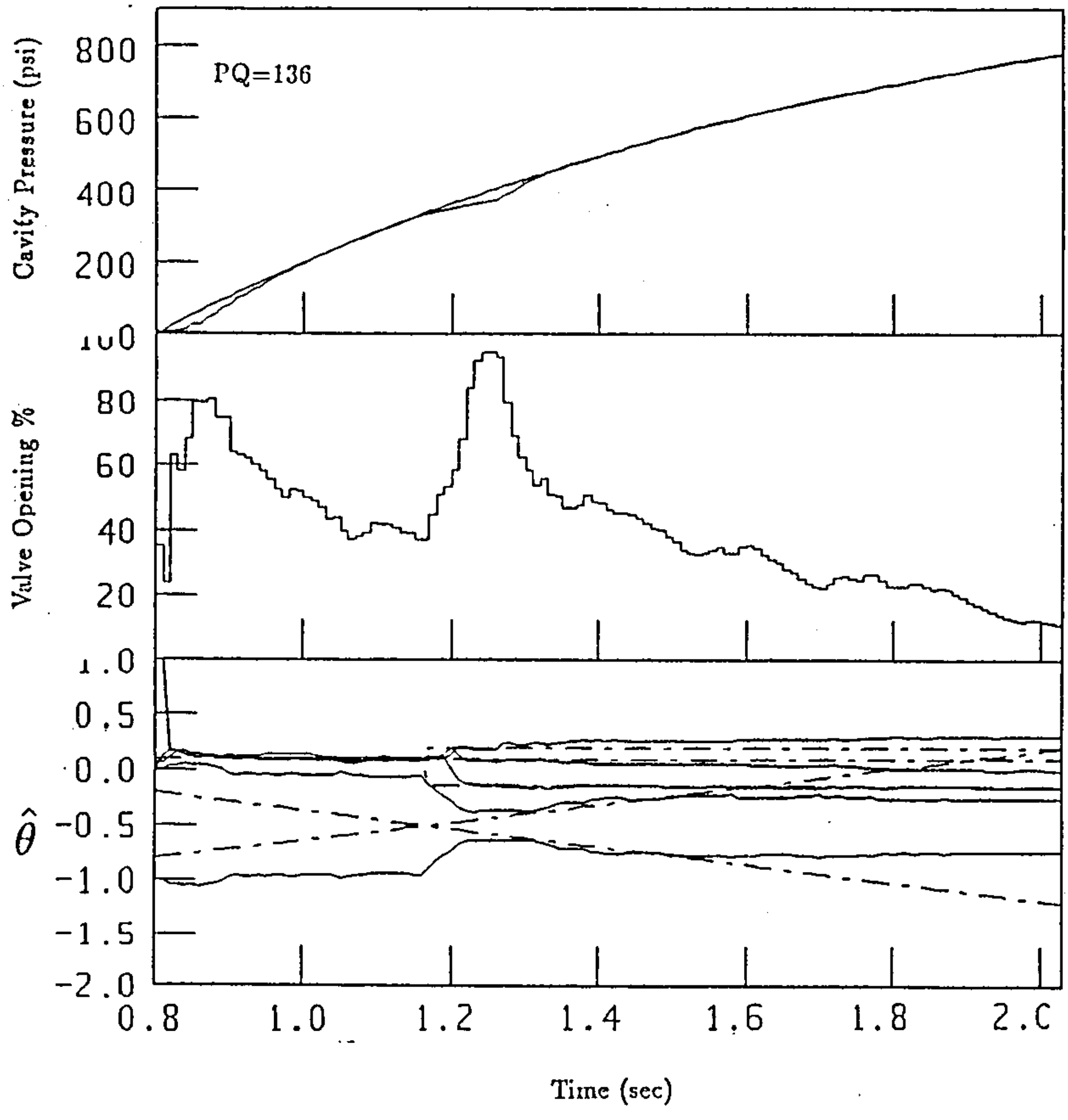

Figure 5.18: Effect of Noise on the Identified Model With the GPC Controller.

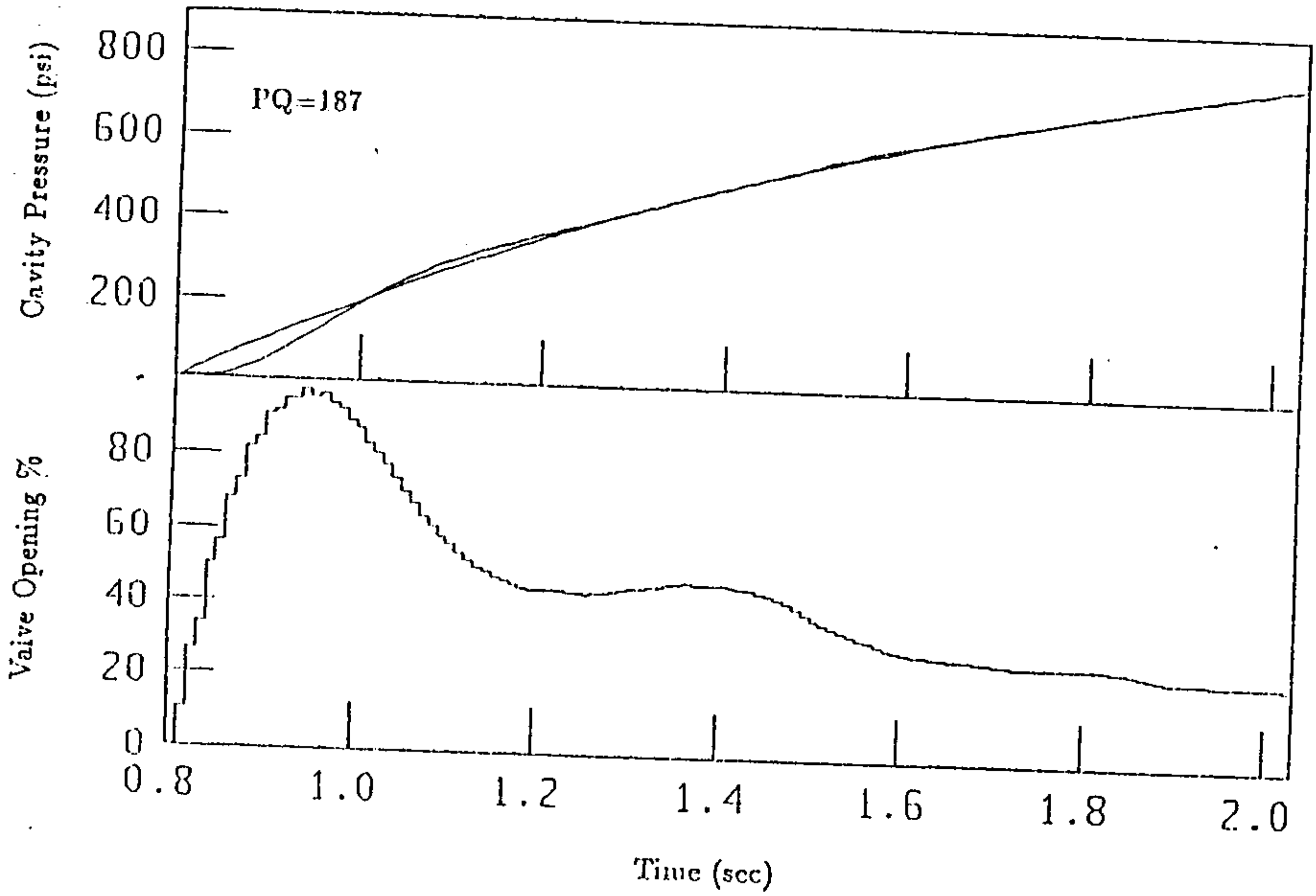

Figure 5.19: Application of the PID Controller to the Identified Model (Without Averaging to the Parameters).

is shown in Figure 5.20. The controller parameters [λ,N1,N2,NU] are chosen as [.5,3,10,1]. The GPC still gives better performance than the PID. Data filtering is used in this case and helped in improving the performance of the GPC.

5.4 Conclusions

The results of the simulation study presented in this chapter have shown the feasibility of using the GPC control algorithm to achieve control over cavity pressure in the injection molding process.

On the other hand, it was seen that the PID controller can give satisfactory performance provided that the right settings of the controllers are chosen, which is not an easy task in practical applications.

The GPC is a nice algorithm with a lot of tuning knobs which can be manipulated to give a certain behaviour. However, the main problem with the GPC is associated with the estimation quality, which will affect the performance of the controller.

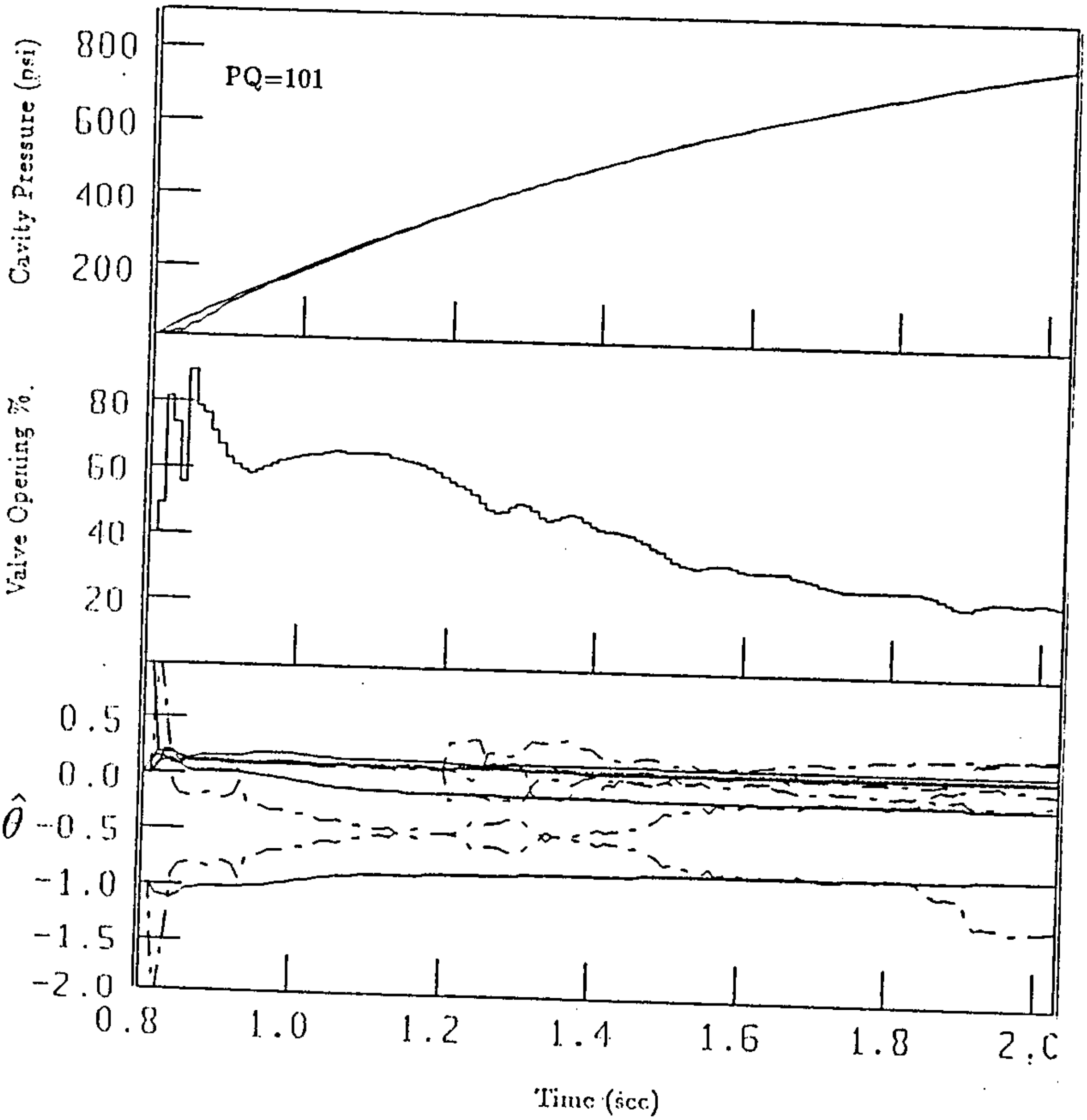

Figure 5.20: Application of the GPC Controller to the Identified Model (Without Averaging to the Parameters).

Chapter 6

CONCLUSIONS

The complex nature of the injection molding process makes it difficult to control the process adequately using conventional controllers such as the PID one, which is a good controller provided that it is well tuned. In addition, the PID can't be used for multivariable control purposes.

This study have introduced an advanced control algorithm, the GPC, to the injection molding process. The results show that the performance of this controller is better than that of the PID and does not need retuning of its parameters when the dynamics of the process changes. Moreover, the tuning knobs of the GPC were used to provide distinct closed-loop behaviour of the controller.

The identification algorithm plays an important role in the controller behaviour, since the GPC calculations depends on the identified model of the process. So, proper set of the estimator parameters should be chosen according to the process properties.

The already developed models are useful for understanding the behaviour of the process, and to test the control strategies,they do not reflect he nonlinearity and time-variation of the process. So, complete dynamic models are needed which represents the process more accurately. These models should incorporate the interactions between the process variables.

Future research which extends the application to control overall process (all stages) is needed, through both simulation and practical evaluation. Furthermore, a multivariable version of the GPC is recommended to be evaluated on the process, in order to compensate for the complex interaction between the process variables. It should be kept in mind that many problems may arise in practical application that were not seen in simulation due to assumptions and approximations.

REFERENCES

[1] Z. Tadmor and G. Costas, "Principles of Polymer Processing",1st Edition, John Wiley and Sons, Canada(1979).

[2] D. A. Fara, "Control of Nozzle and Cavity Pressure During Filling and Packing in Thermoplastic Injection Molding", Ph.D. Thesis, McGill University, Montreal (1988) .

[3] B. Sanchagrin, " Process Control of Injection Molding ", Poly. Eng. Sci.,23, (1983), pp. 431-442.

[4] A. Shankar and W. Paul , " A mathematical Model for the Evaluation of Injection Molding Machine Control", Transactions of the ASME, vol 104' march(1982), pp. 86-92.

[5] R. C. Ricketson and K. K. Wang , " Injection Molding Process Control Based on Empirical Models", ANTEC , vol 33(1987), pp. 231-234 .

[6] Carl Y. Ma, "A Design Approach to a Computer Controlled Injection Molding Machine", Poly. Eng. Sci.,14,11 (1974), pp. 768-772.

[7] M. R. Kamal, W. I. Patterson, D. Abu Fara and A. Haber, "A Study in Injection Molding Dynamics", Poly. Eng. Sci. 24,9(1984), pp. 686-691 .

[8] W. I. Patterson, M. R. Kamal and V. G. Gomes, "Dynamic Modelling and Control of Melt Temperature in Injection Molding ", ANTEC (1985), pp. 754- 758 .

[9] M. R. Kamal, W. I. Patterson, N.Conley and D. Abu Fara, "Dynamics and Control of Pressure in the Injection Molding of Thermoplastics", Poly. Eng. Sci., 27, 18 (1987), pp. 1403-1410 .

[10] D. W. Clarke, "Digital Control Techniques and Applications for Industrial Processes", Unpublished.

[11] D. W. Clarke and P. J. Gawthrop, "Self-Tuning Controllers", Proc. IEE, 122, 9(1975), pp.929-934.

[12] K. J. Astrom, "Theory and Applications of Adaptive Control Techniques- A Survey", Automatica, 19, 5(1983), pp. 471-486.

[13] C. H. Mohtadi, "Studies in Advanced Self-Tuning Algorithms", Ph.D. thesis, Oxford University (1987).

[14] D. W. Clarke, C. H. Mohtadi and P. S. Tuffs, "Generalized Predictive Control-Part I. The Basic Algorithm", Automatica, 23, 2(1987), pp. 137-148.

[15] D. W. Clarke, C. H. Mohtadi and P. S. Tuffs, "Generalized Predictive Control-Part II. Extensions and Interpretations", Automatica, 23, 2(1987), pp. 149-160.

[16] D. W. Clarke and C. H. Mohtadi, "Properties of Generalized Predictive Control", Automatica, 25,6(1989), pp. 859-875 .

[17] D. C. Paulson , " Guide to Injection Machine Control", SPE Journal, 27(1971), pp. 37-41.

[18] J. W. Peter, "Prelude to Automated Process Control", SPE Technical papers, 28(1972), pp. 57-59.

[19] I. J. Catic, "Controlled Injection Molding Systems:One Way for Better Injection Molding of Thermoplastics ", SPE Technical papers, 22(1976), pp. 391-394.

[20] D. B. Hunkar and J.P. Dejean, "A Process And Production Controller for Injection Molding ", SPE Technical papers,24 (1978), pp. 130-132.

[21] D. B. Hunkar, "The Interdependence of Part Parameters on Process Control Adjustable Functions in Injection Molding of Thermoplastics" , SPE Technical Papers,21(1978), pp. 161-164 .

[22] J. Border and N.P. Suh, "Intelligent Injection Molding ", SPE Technical Papers, 28,40(1982), pp. 323-326.

[23] M. A. Davis, "Servocontrolled Injection Molding", SPE Technical Papers, 22(1976), pp. 618-623.

[24] M. A. Davis, "Process Control Applied to the Injection Molding of Thin Wall Parts", SPE Technical papers, 23(1977), pp. 66-70.

[25] W. J. Thayer and M. A. Davis, "Controls for Injection Molding of Thermoplastics ", Advances in Plastic Technology,28 (1981), pp. 28-44.

[26] Carl Y. Ma, "The Development of Process Control for the Molding Processing Industry ", SPE Technical Papers, 20 (1974), pp. 263-265.

[27] P. G. Chambers, "Control Systems for Injection Molding Machines" , Plastics and Rubber Processing ,June(1975), pp. 71-74.

[28] K. K. Wang *et al.* , " Computer - Aided Design and Fabrication of Molds and Computer Control of Injection Molding", Cornell University, Report no. 11 (1985).

[29] J. Parnaby and A. M. Eissa, " Modelling and Computer Control of the Injection Molding Process, Part I: Mathematical Modelling", Unpublished.

[30] J. Parnaby, A. M. Eissa and P. Battye, " Modelling and Computer Control of the Injection Molding Process, Part II: Model Reference Computer Control and Experimental Test of the Theoritical Model", Unpublished.

[31] K. K. Wang *et al.* , " Process Control :Estimation of Controlled System Parameters for Injection Velocity Control", Cornell University, Report no. 10 (1984).

[32] A. R. Agrawal and I. O. Pandelidis, "Non-Minimum Phase Self Tuner for Control of Ram Velocity in Injection Molding ", ANTEC (1987), pp. 235-237 .

[33] D. Abu Fara, M. R. Kamal and W. I. Patterson, "Evaluation of Simple Dynamic Models and Controllers for Hydraulic and Nozzle Pressure in Injection Molding ", Poly. Eng. Sci. 25,11(1985), pp. 714-723 .

[34] K. Srinivasan, T. Srinivasan and G. Maul, "Improvements in Closed Loop Control of Thermoplastic Injection Molding Process", ANTEC(1991), pp. 343-345 .

[35] K. Srinivasan and T. Srinivasan, "Learning Control of Melt Pressure in Injection Molding Process", ASME Winter Annual Meeting, (1991), pp. 1-9 .

[36] P. E. Wellstead and M. Zarrop, "Self-Tuning Systems", 1st Edition, John Wiley and Sons, England (1991), pp. 71-159 .

[37] K. G. Astrom and B. Wittenmark , "On Self-Tuning Regulators ", Automatica, 9, 2(1973), pp. 185-199.

[38] V. Peterka, "Adaptive Digital Regulation of Noisy Systems", IFAC Symposium on Identification and Process Parameter Estimation, Prague(1970) .

[39] D. W. Clarke, "Introduction to Self-Tuning Controllers:in Self-Tuning and Adaptive Control",editor Harris and Billings, Peter Periginus (1981) .

[40] B. Wittenmark, "Stochastic Adaptive Control Methods: A Survey ", Int. J. Cont. , 21, 5(1975), pp. 705-730 .

[41] D. W. Clarke, A. Hodgson and P. Tuffs, "The Offset Problem and K-Incremental Predictors in Self-Tuning Control", Proceedings IEE, vol 130, no 5, (1983), pp. 217-225 .

[42] D. W. Clarke and P.J. Gawthrop, "Self-Tuning Control", Proc. IEE, 126, 6(1979), pp. 633-640.

[43] P. E. Wellstead, D. Prager and P. Zanker, "Pole-Assignment self-tuning regulator", Proceedings IEE, 126,8(1979), pp. 781-787 .

[44] Y. Al-Assaf, "Self-Tuning Control:Theory and Applications", Ph.D. thesis, Oxford University (1988).

[45] E. P. Lambert, "The Industrial Application of Long - Range Predictions ", Ph.D. Thesis, Oxford University (1987).

[46] M. Lambert, "Adaptive Control for Flexible Systems", Ph.D. Thesis, Oxford University (1987).

[47] A. J. Hodjson , "Problems of Integrity in Application of Adaptive Controllers", Ph.D thesis, Oxford univ. (1983).

[48] T. R. Fortescue, "Work on Astrom's Self-Tuning Regulator: Handover Report', Report, Department of Chemical Engineering and Chemical Technology, Imperial College(1977) .

[49] P. S. Tuffs, "Self-Tuning Control:Algorithms and Applications ", Ph.D. thesis, Oxford University (1988).

Appendix A

The Standard Least Squares (SLS) method is derived here for estimation of the CARIMA model parameters. From chapter 3, this model can be written as:

$$A(z^{-1})y(t) = z^{-kd}B(z^{-1})u(t-1) + \frac{C(z^{-1})\zeta(t)}{\Delta} \tag{A.1}$$

The polynomials and parameters in the above model were defined in chapter 3. The model can be written in a compact form, using vector notation:

$$\Delta y(t) = \phi^T\theta + \zeta(t) \tag{A.2}$$

where ϕ is a vector containing the measured input/ouput data including the noise values. θ is the vector of the actual model parameters. This vector is usually unknown and should be estimated based on the available set of data. Now assume a model of the system of the correct structure:

$$\Delta y(t) = \phi^T\hat{\theta} + \hat{\epsilon}(t) \tag{A.3}$$

where $\hat{\theta}$ is the vector of the estimated model parameters and $\hat{\epsilon}(t)$ is the corresponding modelling (or fitting) error.

Our aim is to select $\hat{\theta}$ so that the overall modelling error is minimized. This modelling error can be written from the above equations as:

$$\hat{\epsilon}(t) = \zeta(t) + \phi^T(\theta - \hat{\theta}) \tag{A.4}$$

and for N runs (or for N set of data points), the model can be written as:

$$Y = \Phi\hat{\theta} + \hat{E} \tag{A.5}$$

where Y is a vector containing the output values, Φ is a matrix containing all the input/output data and $\hat{E}$ is a vector containing the values of modelling error. Rearranging the last equation:

$$\hat{\epsilon}(t) = Y - \Phi\hat{\theta} \tag{A.6}$$

The SLS method selects an estimate $\hat{\theta}$ which minimizes the following cost function which represents the sum of sqaures of errors:

$$J = \sum_{t=1}^{N} \hat{\epsilon}^2(t) = \hat{E}^T\hat{E} \tag{A.7}$$

rewriting the above equation :

$$J = (Y - \Phi\hat{\theta})^T(Y - \Phi\hat{\theta}) \tag{A.8}$$

rearrangin another time:

$$J = Y^TY - \hat{\theta}^T\Phi^TY - Y^T\Phi\hat{\theta} + \hat{\theta}^T\Phi^T\Phi\hat{\theta} \tag{A.9}$$

minimizing J with respect to $\hat{\theta}$:

$$\frac{\partial J}{\partial \hat{\theta}} = -2\Phi^TY + 2\Phi^T\Phi\hat{\theta} = 0 \tag{A.10}$$

simplifying the above equation:

$$\Phi^T\Phi\hat{\theta} = \Phi^TY \tag{A.11}$$

and this solves for a unique minimum if the second derivative is positive:

$$\frac{\partial^2 J}{\partial \hat{\theta}^2} = 2\Phi^T\Phi \tag{A.12}$$

which is positive definite. This means that the solution $\hat{\theta}$ is a unique minimum, therfore the least squares estimation is given by:

$$\hat{\theta} = [\Phi^T \Phi]^{-1} [\Phi^T Y] \qquad (A.13)$$

Appendix B

```
C***************************************************************
C       RECURSIVE LEAST SQUARES   SUBROUTINE                  **
C       This program calculates the parameters estimates of   **
C       an assumed model using the RLS identification algorithm **
C***************************************************************
        SUBROUTINE ESTI(M,NY,NA,DY,DU,THETA,P,NPAR,FORGET,SEGMA,E3,FMIN,IFT)
        REAL PHI(50),K(50),THETA(50),P(50,50),PHP(50),PHP2(50),PHP3(50,50)
     1,DU(50),DY(50),PH3(50),R(50,50),PP1(50,50)
        NA1=NA+1
C-----------------------------------------------
C       The vector PHI Contains input-output data
        Y=DY(NY))
        PHI(1)=DY(NY-2)-DY(NY-1)
        DO 330 I=NA1,NPAR
330     PHI(I)=DU(NY-I+NA1)
C-----------------------------------------------
C       Calculation of the prediction error
        ER=Y
        DO 225 I=1,NPAR
225     ER=ER-THETA(I)*PHI(I)
C-----------------------------------------------
C       Multiplication of the Covariance matrix
C       by the data vector PHI
C       PHP=PHI*P ( VECTOR)
        DO 220 I=1,NPAR
220     PHP(I)=0
        DO 221 I=1,NPAR
        DO 221 J=1,NPAR
221     PHP(I)=PHP(I)+PHI(J)*P(J,I)
C-----------------------------------------------
C       Multiplication of the data vector PHI by the
C       PHP(PHI*P) vector, the result is a scalar
C       PH=PHP*PHI (SCALAR VALUE)
        PH=0
        DO 222 I=1,NPAR
222     PH=PH+PHP(I)*PHI(I)
        PH=PH+1
C-----------------------------------------------
C       Calculation of the correction gain vector
        DO 214 I=1,NPAR
214     PHP2(I)=0.
        DO 5 I=1,NPAR
        DO 226 J=1,NPAR
226     PHP2(I)=PHP2(I)+P(I,J)*PHI(J)
5       K(I)=PHP2(I)/PH
C-----------------------------------------------
C       Variable forgetting factor
        FORGET=1-(ER**2/ABS(PH)/SEGMA)
C-----------------------------------------------
C       Updating the covariance matrix [P(t)]
C       PHP3=K*(P*PHI)   (MATRIX)
328     DO 9 I=1,NPAR
        DO 9 J=1,NPAR
        PHP3(I,J)=K(I)*PHP2(J)
9       P(I,J)=(P(I,J)-K(I)*K(J)*PH)/FORGET
C-----------------------------------------------
C       Model parameters updating, Theta
        DO 11 I=1,NPAR
11      THETA(I)=THETA(I)*ER*K(I)
        RETURN
```

Appendix C

```
C*************************************************************
C       MAIN PROGRAM                                        **
C       This main program calls the GPC or PID subroutines  **
C       and plots the response and control activity versus  **
C       time                                                **
C*************************************************************
        DIMENSION THETA(50),TH(10,2000),P(50,50),E(2000),E2(2000),T(2000),
     2BETA(2000),YG(50),W(2000),U(50),W1(50),YP(2000),UP(2000),T3(2000),
     3X1(2000),X2(2000),X3(2000),XA1(2000),XA2(2000),XA3(2000),
     5XB1(2000),XB2(2000),XB3(2000),X4(2000),X5(2000),X6(2000),
     4UP2(2000),T2(2000),TP(10),YF(50),UF(50),UF2(50),YF1(50),
     6E4(2000),A(10),B(10),UPID(2000)
        REAL LAMBDA,KC
C*********************************************************************
C       Read no. of time steps and no. of elements in the data vectors
        READ(1,*)NS,NY
C       Read no. of polynomial parameters and their values
        READ(1,*)NA,NB
        READ(1,*)(A(I),I=1,NA)
        READ(1,*)(B(I),I=1,NB)
C       No. of parameters NPAR=NA+NB
        NPAR=NA+NB
C*********************************************************************
C       Read the time delay in samples
        READ(1,*)KD,IFT
C       Input the T-polynomial coefficients
        READ(1,*)(TP(I),I=1,2)
C       Read the GPC parameters values
        READ(1,*)LAMBDA,N1,N2,NU
C       Read the RLS parameters
        Read(1,*)SEGMA,COV
C*******************************
C       Call the random number generation subroutine
        CALL RN(E,E2)
C*******************************
C       Initial Covariance matrix size
        DO 16 I=1,NPAR
        DO 16 J=1,NPAR
        P(I,J)=0
16      P(I,I)=COV
C*******************************
C       Read initial parameters values
        READ(1,*)(THETA(I),I=1,NPAR)
C*******************************
C       Calculation of the process output
        M=1
C       YG=A(I)*Y+B(I)*U
24      YA=0.
        YB=0.
        DO 37 I=1,NA
37      YA=YA-A(I)*YG(NY-I)
        DO 38 I=1,NB
38      YB=YB+B(I)*U(NY-KD-I)
        YG(NY)=YA+YB
C*********************************************************
C       Choice of either the GPC or the PID
        READ(1,*)JC
        GOTO(320,321)JC
C*********************************************************
C       Data filtering through the T-filter
```

```
320 `   UF(NY)=(U(NY-1)-U(NY-2)-TP(2)*UF(NY-1))/TP(1)
        YF1(NY)=(YG(NY)-TP(2)*YF1(NY-1))/TP(1)
        YF(NY)=(YG(NY)-YG(NY-1)-TP(2)*YF(NY-1))/TP(1)
        UF2(NY)=(U(NY-KD-1)-U(NY-KD-2)-TP(2)*UF2(NY-1))/TP(1)
C***********************************************************
C       Call identification subroutine
        CALL ESTI(M,NY,NA2,YF,UF2,THETA,P,NPAR,FORGET,SEGMA,E3,FMIN,IFT)
        DO 2 I=1,NPAR
2       TH(I,M)=THETA(I)
C************************************************************
C       Call GPC  subroutine
        CALL GPC(M,W1,THETA,NA,NB,YG,YF1,UF,TP,D1U,NY,LAMBDA,
     2N1,N2,NU,KD)
C       Get the current control signal
        U(NY)=D1U+U(NY-1)
        GOTO 200
C************************************************************
C       CALL PID SUBROUTINE
C       Read the PID parameters
321     READ(1,*)KC,TS,TOI,TOD
        CALL PID(M,NY,YG,U,W,U1,TS,KC,TOI,TOD)
        U(NY)=U1
C*************************************************************
200     IF(M.GE.NS) GOTO 23
C       Jacketing of the control signals
        IF(U(NY).GT.100)U(NY)=100
        IF(U(NY).LT.0)U(NY)=0
        CALL SH(NY,YG,U)
55      M=M+1
        GO TO 24
C*****************************************
C       PLOTTING SUBROUTINE
        DO 76 I=1,NS
        X1(I)=TH(1,I)
        X2(I)=-1-TH(1,I)
        X3(I)=TH(2,I)
        X4(I)=TH(3,I)
76      X5(I)=TH(4,I)
        READ(1,*)XMIN,XMAX,YMIN,YMAX,XS,YS
        READ(1,*)CMIN,CMAX,PMIN,PMAX,YS2,YS3
        call iniplt(I,.FALSE.,1.0)
C       Plot the process output along with the set point
        call viewport(0,5000,3500,7000)
        call graphboundary(900,4900,100,3000)
        call scale(XMIN,XMAX,YMIN,YMAX)
        call axis(XS, '10.1','Time (sec) ',0,YS,'10.0','Pc (psi) ',2)
        CALL POLYLINE(T,W,NS,0,0,0,0,0)
        CALL POLYLINE(T,YP,NS,0,0,0,0,0)
C       Plot the control action
        call viewport(0,5000,0,3500)
        call graphboundary(900,4900,700,3400)
        call scale(XMIN,XMAX,CMIN,CMAX)
        call axis(XS,'10.1','Time (sec)',2,YS2,'10.0','Valve Op. (%) ',2)
        CALL POLYLINE(T2,UP2,NS2-2,0,0,0,0,0)
C       Plot the identified parameters
        call viewport(5000,10000,0,7000)
        call graphboundary(1000,4800,700,4500)
        call scale(XMIN,XMAX,PMIN,PMAX)
        call axis(XS,'10.1','Time (sec)',2,YS3,'10.1','Theta',2)
        CALL POLYLINE(T,XA1,NS,0,0,0,0,7)
```

```
      CALL POLYLINE(T,XA2,NS,0,0,0,0,7)
      CALL POLYLINE(T,XB1,NS,0,0,0,0,7)
      CALL POLYLINE(T,XB2,NS,0,0,0,0,7)
      CALL POLYLINE(T,XB3,NS,0,0,0,0,7)
      CALL POLYLINE(T,X1,NS,0,0,0,0,0)
      CALL POLYLINE(T,X2,NS,0,0,0,0,0)
      CALL POLYLINE(T,X3,NS,0,0,0,0,0)
C*********************************************
C     Shifting subroutine
C     This subroutine shifts the data vectors
C     by moving each element a one step Backward
C     (Two vectors are shifted each time)
      SUBROUTINE SH(NY,YG1,UD)
      DIMENSION YG1(50),UD(50)
      DO 32 K=1,NY-1
      YG1(K)=YG1(K+1)
32    UD(K)=UD(K+1)
      RETURN
      END
C*********************************************
C     RANDOM NUMBER GENERATION SUBROUTINE
      SUBROUTINE RN(E,E2)
      DIMENSION E(2000),E2(2000)
      N=2000
      E(1)=.5
      P=11
      K=35
      DO 1 I=2,N
      E(I)=E(I-1)*K/P
      D1=INT(E(I))
      E(I)=E(I)-D1
      E2(I)=(-1)**I*E(I)
1     CONTINUE
      RETURN
      END
C*********************************************
C     PID SUBROUTINE
      SUBROUTINE PID(M,NY,YG,U,W,U1,T,KC,TOI,TOD)
      DIMENSION W(2000),YG(50),U(50)
      REAL KC
C     The parameters of the digital PID
      CA0=KC*(1+(T/TOI)+(TOD/T))
      CA1=KC*(1+2*(TOD/T))
      CA2=KC*(TOD/T)
      E=W(M)-YG(NY)
      U1=U(NY-1)+CA0*E-CA1*E1+CA2*E2
      E2=E1
      E1=E
      RETURN
      END
C***********************************************************
C     GPC SUBROUTINE                                       **
C     This subroutine calculates the control output using  **
C     the Generalized Predictive Control method            **
C***********************************************************
      SUBROUTINE GPC(M,W,THETA,NA,NB,Y,YF,UF,TP,D1U,NY,LAMBDA
     5,N1,N2,NU,KD)
      REAL E(60),A(10),B(10),AD(50),DEL(2),G(50),G1(50),Y(50),G2(50),
     1PF(50),F1(50),DU(50),WF(50),DU1(50),W(50),THETA(50),GB(10,50),
     3F(10,50),FFM(50,50),GM(50,50),I1(50,50),GG(50,50),TP(10),
```

```
     2I2(50,50),GT(50,50),GTG(50,50),GG1(50,50),GG2(50,50),YF(50),
     4UF(50),GUD(50),FYD(50),FF(50,50)
      REAL LAMBDA
      NPAR=NA+NB
      NA1=NA+1
      NB=NB+KD
      NB1=NB-1
C     The identified model parameters
      A(1)=1
      A(2)=THETA(1)
      A(3)=-1-A(2)
      DO 86 I=1,KD
86    B(I)=0.
      DO 89 I=NA+1,NPAR
89    B(I-NA+KD)=THETA(I-1)
C     The delta operator
      DATA (DEL(I),I=1,2)/1,-1/
C     Multiplication the delta operator
C     by the A polynomial
      CALL PM(NA1,2,A,DEL,AD,NAD)
C     Start of Diophantine equation recursion(E,F are obtained)
      E(1)=1
C     Start of G polynomial partition (G,GB are obtained)
      G(1)=B(1)
      DO 90 I=1,NA1
90    F(I,1)=TP(I+1)-AD(I+1)*E(1)
      DO 14 I=1,NB1
14    GB(I,1)=B(I+1)-TP(I+1)*G(1)
      DO 1 J=2,N2
      E(J)=F(1,J-1)
      G(J)=(E(J)*B(1)+GB(1,J-1))/TP(1)
      DO 22 I=1,NA
22    F(I,J)=F(I+1,J-1)-AD(I+1)*E(J)
      F(NA1,J)=0.-AD(NA1+1)*E(J)
      DO 13 I=1,NB1
13    GB(I,J)=E(J)*B(I+1)-G(J)*TP(I+1)+GB(I+1,J-1)
1     CONTINUE
      DO 12 J=1,N2
      DO 3 I=1,J
      G1(I)=G(J-I+1)
C     The G-matrix in GPC
3     GM(J,I)=G1(I)
      DO 7 K=1,NB1
7     G2(K)=GB(K,J)
      DO 11 I=1,NA1
11    F1(I)=F(I,J)
C     Calculation of the parameters of f-vector in GPC(GU,FY)
      CALL DP(NB1,NY,G2,UF,GU)
      GUD(J)=GU
      CALL DP(NA1,NY,F1,YF,FY)
      FYD(J)=FY
      PF(J)=-(GU+FY)
12    CONTINUE
      DO 39 I=N1,N2
      DO 39 J=1,NU
39    GM(I-N1+1,J)=GM(I,J)
C     Calculation of the Transpose of the G-matrix
      CALL MT(N2,NU,GM,GT)
C     Multiplication of the matrix by its transpose
      CALL MM(NY,NU,N2,NU,GT,GM,GTG)
```

```
      DO 111 I=1,NU
      DO 111 J=1,NU
      I1(I,J)=0
111   I1(I,I)=1.
C     The identity matrix
      CALL MC(NU,LAMBDA,I1,I2)
      CALL MA(NU,I2,GTG,GG)
C     If Nu>1 then call the matrix inversion subroutine
      IF(NU.NE.1)THEN
      CALL MI(NU,GG,GG1)
      ELSE
      GG1(1,1)=1/GG(1,1)
      ENDIF
      CALL MM(NY,NU,NU,N2,GG1,GT,GG2)
      CALL PA(N2,W,PF,WF,NWF)
C     Calculation of the optimum control increments vector
      CALL MV(NY,1,N2,GG2,WF,DU1)
      DIU=DU1(1)
C*********************************************
C     Used subroutines for the GPC
C*********************************************
C     Polynomial multiplication subroutine
C     To multiply two polynomials
      SUBROUTINE PM(NI,NJ,P1,P2,PP,K)
      DIMENSION P1(50),P2(50),PP(50)
      DO 100 K=1,NI+NJ
100   PP(K)=0
      DO 99 I=1,NI
      DO 99 J=1,NJ
      K=I+J-1
99    PP(K)=PP(K)+P1(I)*P2(J)
      RETURN
      END
C*********************************************
C     Dot product subroutine
C     To find the dot product of two vectors
C     or polynomials
      SUBROUTINE DP(ND,NY,H1,H2,P)
      DIMENSION H1(50),H2(50)
      P=0
      DO 97 I=1,ND
      J=NY-I+1
97    P=P+H1(I)*H2(J)
      RETURN
      END
C*********************************************
C     Polynomial addition subroutine
      SUBROUTINE PA(NI,P3,P4,PP2,K)
      DIMENSION P3(50),P4(50),PP2(50)
      DO 98 I=1,NI
98    PP2(I)=P3(I)+P4(I)
      RETURN
      END
C*********************************************
C     Matrix transpose subroutine
C     To find the transpose of a matrix
      SUBROUTINE MT(NA,NU,A,AT)
      DIMENSION A(50,50),AT(50,50)
      DO 5 I=1,NA
      DO 5 J=1,NU
```

```
5       AT(J,I)=A(I,J)
        RETURN
        END
C**********************************************
C       Matrix addition subroutine
        SUBROUTINE MA(NA,A1,A2,A3)
        DIMENSION A1(50,50),A2(50,50),A3(50,50)
        DO 85 I=1,NA
        DO 85 J=1,NA
85      A3(I,J)=A1(I,J)+A2(I,J)
        RETURN
        END
C**********************************************
C       Matrix multiplication subroutine
C       TO multiply two matrices
        SUBROUTINE MM(NY,N1,N2,N3,A4,A5,A6)
        DIMENSION A4(50,50),A5(50,50),A6(50,50)
        DO 110 I=1,NY
        DO 110 J=1,NY
110     A6(I,J)=0
        DO 6 I=1,N1
        DO 6 J=1,N3
        DO 6 K=1,N2
        RETURN
        END
C*******************************************
C       Matrix-vector  multiplication subroutine
C       To multiply vector by a matrix
        SUBROUTINE MV(NY,N1,N2,A7,A8,A9)
        DIMENSION A7(50,50),A8(50),A9(50)
        DO 120 I=1,NY
120     A9(I)=0
        DO 46 J=1,N1
        DO 46 I=1,N2
46      A9(J)=A9(J)+A7(J,I)*A8(I)
C46     PRINT*,'DU,GG2,WF',A9(J),A7(J,I),A8(J)
        RETURN
        END
C*********************************************
C       Matrix - constant  multiplication subroutine
C       To multiply a matrix by a constant value
        SUBROUTINE MC(NU,C,A7,A8)
        DIMENSION A7(50,50),A8(50,50)
        DO 87 I=1,NU
        DO 87 J=1,NU
87      A8(I,J)=C*A7(I,J)
        RETURN
        END
C*********************************************
C       Matrix inversion subroutine
C       To find the inverse of a square matrix
        SUBROUTINE MI(N,AA,AI)
        DIMENSION AA(50,50),AI(50,50)
        DO 101 I=1,N
        DO 101 J=1,N
        AI(I,J)=0
101     AI(I,I)=1
        DO 102 I=2,N
        DO 102 K=1,I-1
        IF(AA(I,K).EQ.0.) GO TO 102
```

```
      Z1=AA(K,K)/AA(I,K)
      DO 102 J=1,N
      AA(I,J)=AA(K,J)-Z1*AA(I,J)
      AI(I,J)=AI(K,J)-Z1*AI(I,J)
102   CONTINUE
      DO 103 I=N,2,-1
      DO 103 K=N,I,-1
      IF(AA(I-1,K).EQ.0.) GO TO 103
      Z2=AA(K,K)/AA(I-1,K)
      DO 103 J=N,1,-1
      AA(I-1,J)=AA(K,J)-Z2*AA(I-1,J)
      AI(I-1,J)=AI(K,J)-Z2*AI(I-1,J)
103   CONTINUE
      DO 104 I=1,N
      Z3=AA(I,I)
      DO 104 J=1,N
      AA(I,J)=AA(I,J)/Z3
      AI(I,J)=AI(I,J)/Z3
104   CONTINUE
      RETURN
      END
```