THE CONTROL OF MULTIVARIABLE TIME-DELAYED PROCESSES AND A GENERALIZED SMITH PREDICTOR

A thesis presented in partial fulfilment of the requirements

for the degree of

Doctor of Philosophy
in Production Technology at
Massey University

Robert Andrew Crawford

1994

Abstract

In this thesis the description, analysis and control of time-delayed multivariable processes are investigated, particularly the descriptions of multivariable processes that facilitate a multivariable extension of the Smith predictor.

Two new pseudo-commutativity results for matrix multiplication are presented. These results are used to show that a general time-delayed transfer function can be decomposed into three components representing input-delays, output-delays and the delay-free dynamics of the process. It is also shown that any such time-delayed transfer function can also be written in a form in which all the delays appear as output-delays.

These time-delayed transfer functions are used in the development of a multivariable Smith predictor.

It is also shown that the pseudo-commutativity results can be applied to non-delayed processes. In particular a new method, based on these results, for reformulating a transfer function description of a process as a state-space description is developed.

A case study of a time-delayed process is investigated.

Acknowledgments

I would like to thank everybody that have helped in any way with this thesis. Special thanks must be extended to the staff of the Departments of Mathematics, Statistics and Production Technology, without whose very kind assistance, guidance and, not least of all, financial support this project would never have been started. I would also like to thank the various office-mates and fellow students in these departments especially Arron Parshotam, Marie Sleeman, Ming Tan and Robert Lynn, all of whom provided relief, insight and the occasional diversion from the topic at hand.

The patience and tolerance of Catherine Wallace, my partner, must also acknowledged, without Catherine life would have been so much simpler but nowhere near as much fun!

Lastly I would like to thank my long suffering supervisors, Paul Austin and Mike Carter.

TABLE OF CONTENTS

PART A

INTRODUCTION

1 INTRODUCTION

1.1 Introduction

This dissertation takes as its theme the control of time-delayed processes. In particular an investigation is made of the control of multivariable time-delayed processes in which the delays occur in both the input and output paths.

Time-delayed processes are common in industrial and technological systems and include, for example, distillation columns (Wood and Berry, 1973), multi-effect evaporators, (Song *et al,* 1986, Crawford and Austin, 1988*),* paper-making machines (Astrom 1967), ore-crushing processes (Borison and Syding, 1976), blending processes (Singh and McEwan, 1975), cold rolling steel mills (Smith 1957) and catalytic crackers (Smith 1957).

Continuous time-delayed processes are difficult to control since models of these processes have infinite dimension. Algorithms that predict or compensate for the time-delays are required. Few such algorithms have appeared in the control literature, especially for multivariable processes.

The Smith predictor (Smith, 1957) is a commonly used method of controlling single-input single-output time-delayed processes. The Smith predictor enables control engineers to design a controller for the equivalent delay-free process and apply that control law, in conjunction with the Smith predictor to control the time-delayed process.

In single-input single-output linear processes, the order in which the dynamics and delays occur is not important. That is, it is of little consequence to the mathematical model of the process if the overall time-delay is due to delays in the actuations or in the measurements. However, in multivariable processes it is evident that input-delays have different effects to output-delays and many of the single-input single-output methodologies for controlling time-delayed processes, including the Smith predictor, cannot be easily extended to the multivariable case.

In this thesis, methods that simplify the analysis of multivariable processes are developed. These methods allow the manipulation of any of the time-delays so that their effects can appear to be as input or output delays. These results greatly simplify the development of a multivariable Smith predictor.

1.2 Outline of Chapters

The thesis is divided into four sections:

Part A Introduction

Chapter 2 contains a literature review and defines the notation and terminology that will be used in the remainder of the dissertation. Throughout this dissertation both time-domain and frequency-domain representations will be used. In order to be as inclusive as possible, a unified approach, Middleton and Goodwin (1990), is used where ever possible. For completeness this unified approach, which includes continuous-time and discrete-time as special cases will be summarised. The literature review will cover some of the methods of controlling time-delayed processes, both single-input single-output and multivariable.

Part B Analysis Of Time-Delayed Processes

In Chapter 3 frequency domain representations of multivariable time-delayed processes are discussed. Two matrix decompositions are developed that allows a pseudo-commutativity of matrix products and allows an element by element matrix product to be written as a normal matrix product. These two decomposition theorems allow a general multivariable time-delayed transfer function to be written in a suitable form so that a generalised Smith predictor can be designed.

Chapter 4 presents a method for the construction of a state-space description from multivariable transfer functions. The decomposition results of Chapter 3 are used to develop a new transformation between a multivariable transfer function, describing a time-delayed process, and a state-space description of this time-delayed process and an equivalent delay-free process to be constructed. This transformation is useful since it gives a block diagonal state-transition matrix.

Chapter 5 investigates a discrete-time state-space description, the Non-Minimal State-Space (NMSS) description. This state-space description has some features that make it attractive for the design and implementation of adaptive controllers for time-delayed processes. A new form of NMSS description is developed using the decomposition results of Chapter 3.

Part C A Generalised Smith Predictor

In Chapter 6 the multivariable Smith predictor is developed using the results of Chapter 3. A simulation example using this Smith predictor is presented. An alternative derivation of the Smith predictor, using the Internal Model Control (IMC) structure is also considered.

Chapter 7 investigates state-space representations of this multivariable Smith predictor based on the time-delayed state-space descriptions developed in Chapter 4. A Smith predictor based on the time-delayed NMSS description, developed in Chapter 5, is also investigated.

Chapter 8 investigates some of the robustness and sensitivity properties of the Smith predictor. It is shown that the Smith predictor can only be applied to open-loop stable processes. Some of the robustness and stability properties of the multivariable Smith predictor are investigated. This analysis extends the results of Palmor and Halevi (1983) and Owens and Raya (1982) to the types of Smith predictor developed in Chapter 6.

Part D An Evaporator: A Case-Study Of A Time-Delayed Process

Chapters 9 and 10 examine a case study of a time-delayed process. This case study, an evaporator, will illustrate some of the problems of implementation of a multivariable controller on a real process plant: utilising a novel, and cheap, method of implementation.

2 Notation and Literature Review

2.1 Introduction

In this chapter the notation and terminology used throughout the remainder of the thesis is presented.

2.2 DELAY-FREE SYSTEM DESCRIPTIONS AND NOTATION

A *Process* or *System* is any dynamic structure in which there are inputs, states and outputs. A diagram of a general process is shown in Figure 2.2.1.

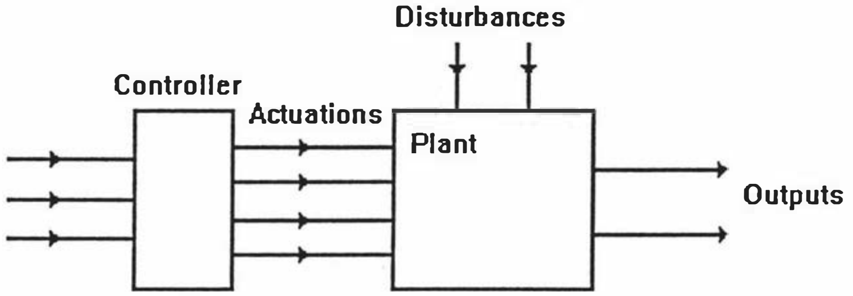

Figure 2.2.1

A *Plant* is a portion of a process, usually related to a physical plant such as an evaporator or a boiler.

Measurements, or *Outputs*, of a process are measurable quantities that are available from the process. Typically these might consist of temperatures, densities and flowrates.

Control inputs, or *actuations* are inputs to the process that can be manipulated in order that the outputs follow, as closely as possible, a selected pattern or trajectory.

Disturbances are inputs to the process that cannot be manipulated.

A *Controller* is a device that sets values for each actuation

Delay free processes can be described using *State-space descriptions*. A continuous-time linear process can be described as:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \qquad (2.2.1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{v}(t) \qquad (2.2.2)$$

and a discrete-time linear process can be described as:

$$\mathbf{x}(t+1) = \Phi\mathbf{x}(t) + \Gamma\mathbf{u}(t) + \mathbf{w}(t) \qquad (2.2.3)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{v}(t) \qquad (2.2.4)$$

where $\mathbf{x}(t)$ is the *State-vector* at time t. Unless stated otherwise $\mathbf{x}$ is $\boldsymbol{R}^{n}$.

$\mathbf{y}(t)$ is the vector of *Measured outputs*. Unless stated otherwise $\mathbf{y}$ is $\boldsymbol{R}^{r}$.

$\mathbf{u}(t)$ is the *Control Input vector*. Unless stated otherwise $\mathbf{u}$ is $\boldsymbol{R}^{q_1}$.

$\mathbf{w}(t)$ is the *Disturbance vector*. Unless stated otherwise $\mathbf{w}$ is $\boldsymbol{R}^{q_2}$.

$\mathbf{v}(t)$ is the *Measurement Noise vector*. Unless stated otherwise $\mathbf{v}$ is $\boldsymbol{R}^{r}$.

A and Φ [$\boldsymbol{R}^{n \times n}$] are the *State-Transition matrices.*

B and Γ [$\boldsymbol{R}^{n \times q_1}$] are the *Control Input matrices.*

and C [$\boldsymbol{R}^{r \times n}$] is the *Measurement matrix.*

The *Backward-shift* and *Forward-shift operators* are often used in discrete-time linear processes. The backward-shift operator, q^{-1}, has the property:

$$q^{-1}x(t) = x(t-1) \qquad (2.2.5)$$

The forward shift operator, q, has the property:

$$qx(t) = x(t+1) \qquad (2.2.6)$$

Discrete-time linear processes can also be described using *Difference Equations* or ARMAX, Auto Regressive Moving Average with auXiliary inputs, descriptions:

$$A(q^{-1})\mathbf{y}(t) = B(q^{-1})\mathbf{u}(t) + E(q^{-1})\mathbf{w}(t) + \mathbf{v}(t) \qquad (2.2.7)$$

where $\boldsymbol{A}(q^{-1})$ $[\in \mathcal{R}^{r \times r}]$, $\boldsymbol{B}(q^{-1})$ $[\in \mathcal{R}^{q_1 \times r}]$ and $\boldsymbol{E}(q^{-1})$ $[\in \mathcal{R}^{q_2 \times r}]$ are polynomial matrics in q^{-1}.

2.3 Frequency Domain Representations of Multivariable Processes

In the frequency domain process descriptions have commonly been written in terms of the Laplace transform, the z-transform or the delta-transform. These three transformations have historically been used in different ways. However, they do possess certain similarities, especially in the way in which time-delays are transformed. These transformations will be briefly reviewed to allow the use of a unified approach, proposed by Middleton and Goodwin (1990). This unified approach will be used in the remainder of the thesis.

2.3.1 The Laplace Transform

In continuous time, the Laplace transform is an often used and well understood method of studying differential equations. The underlying idea of the Laplace transform is the transformation of differential equations in the time-domain to algebraic equations in the frequency-domain.

The one-sided Laplace transform is defined as

$$F(s) = \mathcal{L}\{f(t)\} = \int_{0}^{\infty} e^{-st} f(t) dt \qquad (2.3.1)$$

The Laplace transformation has two important properties in relation to the subject matter of this work:

$$\mathcal{L}\left\{\frac{d^n f}{dt^n}\right\} = s^n F(s) - \sum_{i=1}^{n-1} s^i f^{(i)}(0) \qquad (2.3.2)$$

and

$$\mathcal{L}\{f(t-k)U(t-k)\} = e^{-ks}F(s), \qquad (2.3.3)$$

where

$$U(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (2.3.4)$$

These properties provide the means for transforming time-delayed systems of differential equations to the frequency domain.

Equation (2.3.2) allows the state-space description in Equations (2.2.1) and (2.2.3) to be written in the frequency domain as:

$$\mathbf{y}(s) = \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}(\mathbf{B}\mathbf{u}(s) + \mathbf{w}(s)) + \mathbf{v}(s) \qquad (2.3.5)$$

from which the s-domain transfer function between **u** and **y** can be seen to be

$$G(s) = \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\mathbf{u}(s) \qquad (2.3.6)$$

2.3.2 The Z-Transform

The z-transform (Astrom and Wittenmark, 1984, p15, and Power and Simpson, 1978, p106-110) is a discrete-time analogy to the Laplace transform.

The z-transform has a close association with the backward-shift operator, q^{-1}. This close relationship allows for the easy calculation of the z-transform from the difference equation describing the processes. If the difference equation describing the open-loop process is

$$A(q^{-1})\mathbf{y}(t) = B(q^{-1})\mathbf{u}(t) \qquad (2.3.7)$$

with the initial conditions that **y**(0)=**0** and **u**(0)=**0**, the z-transform transfer function is

$$G(z^{-1}) = A(z^{-1})^{-1}B(z^{-1}) \qquad (2.3.8)$$

In the z-domain a k time-unit time-delay is represented by z^{-k} instead of the $e^{-\tau s}$ that represents a time-delay in the Laplace-transform domain. This results in the discrete-time time-delayed open-loop transfer function having only a finite, although possibly large, number of zeros, whereas the continuous-time process will have effectively an infinite number of zeros due to the isolated essential singularities at infinity resulting from the terms $e^{-\alpha_j s}$.

2.3.3 The Delta Operator and the Delta Transform

Middleton and Goodwin (1990, p 43-47) present the *delta operator*

$$\delta = \frac{q-1}{\Delta} \qquad (2.3.9)$$

where q is the forward shift operator and Δ is the *sample-time*.

The delta operator appears very much like a Eulerian estimate of the derivative. This suggests that if a continuous-time system is modelled using the delta operator, it would appear very much like a differential equation. The parameters obtained for the delta operator model of a system as the sample-time, Δ, tends to zero correspond with the parameters of the differential equation that describes the underlying process. In addition, the numerical properties of the delta operator have been shown to be superior to those of the backward-shift operator. (Middleton and Goodwin1986).

Since the Laplace transform is so useful with continuous-time processes, it would seem to be useful to develop a corresponding discrete-time transform using the delta-operator. This transform, called the delta-transform, should have the property that as the sample-time decreases the discrete transform will tend to the Laplace transform.

The delta transform of f(t) is defined to be

$$\mathscr{D}(f(t)) = \Delta \sum_{k=0}^{\infty} f(t_k)(1+\Delta\gamma)^{-k} \tag{2.3.10}$$

it can be shown that as $\Delta \rightarrow 0$

$$\begin{aligned} \mathscr{D}(f(t) &\rightarrow \lim_{\Delta\rightarrow 0} \sum f(t_k)(1+\Delta\gamma)^{-k}\Delta \\ &= \int_0^{\infty} e^{-\gamma t} f(t)dt \\ &= \mathscr{L}(f(t)) \end{aligned} \tag{2.3.11}$$

which is the Laplace transform of f(t). A formal proof that the Laplace transform is the limiting case of the delta-transform is given in Middleton and Goodwin (1990, p69-70).

A table of delta-transforms can be easily constructed from a table of z-transforms using the identity

$$\mathscr{D}\{f(\gamma)\} = \Delta F(z^{-1})_{z=\Delta\gamma+1} \tag{2.3.12}$$

where $F(z^{-1})$ is the z-transform of f(t). Such tables can be found in Middleton and Goodwin (1990, p56 and p90).

In particular the pure delay $F(z^{-1}) = z^{-k}$ can be transformed as:

$$\mathcal{D}\{f(\gamma)\} = (1+\Delta\gamma)^{-k}$$
$$= E(\gamma,-k) \qquad (2.3.13)$$

which , as $\Delta \to 0$ can be seen to tend to $e^{-\gamma k}$.

2.3.4 Unified Transformation Method

Middleton and Goodwin (1990, p65-76) suggests a unified approach to analysis using a unified frequency domain variable, γ, representing z^{-1}, γ or s as required. Similarly the operators, $\frac{d}{dt}$, q^{-1} and δ are represented by a single operator, ρ.

Consider the process described by the state-space description

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{Ax}(t) + \mathbf{Bu}(t) \qquad (2.3.14)$$

$$\mathbf{y}(t) = \mathbf{Cx}(t) \qquad (2.3.15)$$

By assuming a zero-order hold on the inputs this can be written in the discrete form

$$\mathbf{x}(t+\Delta) = \mathbf{\Phi x}(t) + \mathbf{\Gamma u}(t) \qquad (2.3.16)$$

Equation (2.3.16) can be rewritten in the delta operator form using either of the two methods suggested by Middleton and Goodwin (1990, p46-47). The first method uses the property presented in Equation (2.3.12). The second method uses

$$\rho\mathbf{x}(t) = \mathbf{A}_\gamma\mathbf{x}(t) + \mathbf{B}_\gamma\mathbf{u}(t)$$
$$= \mathbf{\Omega Ax}(t) + \mathbf{\Omega Bu}(t) \qquad (2.3.17)$$

where $$\Omega = \frac{1}{\Delta}\int_0^\Delta e^{\mathbf{A}t}dt$$
$$= \mathbf{I} + \frac{\mathbf{A}\Delta}{2!} + \frac{\mathbf{A}^2\Delta^2}{3!} + \frac{\mathbf{A}^3\Delta^3}{4!} + ... \qquad (2.3.18)$$

The three forms of the state-space description, Equations (2.3.14), (2.3.17) and (2.3.18) can be written in terms of the unified operator as

$$\gamma\mathbf{x}(\gamma) = \mathbf{A}_\gamma\mathbf{x}(\gamma) + \mathbf{B}_\gamma\mathbf{u}(\gamma) \tag{2.3.19}$$

which can be written in terms of the unified transform as

$$\mathbf{y}(\gamma) = \mathbf{C}(\gamma\mathbf{I}-\mathbf{A}_\gamma)^{-1}\mathbf{B}_\gamma\mathbf{u}(\gamma) \tag{2.3.20}$$

The transfer function between **u** and **y** can be seen to be $\mathbf{C}(\gamma\mathbf{I}-\mathbf{A}_\gamma)^{-1}\mathbf{B}_\gamma$

2.4 TIME-DELAYED PROCESS DESCRIPTIONS

In the previous section time-domain and frequency-domain representations of delay-free processes were briefly described. In this section representations of time-delayed processes are introduced. Representations of time-delayed processes will be explored in depth in Chapter 3.

Time-delays occur both naturally and artificially in processes. They occur commonly with transportation processes where, for example, a product must either be physically moved from one part of the process to another, or cooled sufficiently to be handled. Delays can also occur due to the time taken for measurements to be made. Artificial delays are sometimes introduced to create low-order, time-delayed models of processes that are, in reality, of high order.

Consider a single-input single output process in which the input to the plant is delayed by τ units of time. The plant can be described as:

$$\frac{dx}{dt} = ax(t)+bv(t) \qquad (2.4.1)$$

where

$$v(t) = u(t-\tau) \qquad (2.4.2)$$

The Laplace transform of this process is

$$x(s) = \frac{b}{(s-a)}v(s) \qquad (2.4.3)$$

or

$$= \frac{b}{(s-a)}e^{-ts}u(s) \qquad (2.4.4)$$

In this process description it is clear that the delay free dynamics are represented by the rational portion of the transfer function, $\frac{b}{(s-a)}$, and the delay is represented by the exponential, $e^{-\tau s}$.

If the process were described using the unified approach the form of the transfer function would not change. The unified transfer function of the process would be:

$$g^*(\gamma) \quad = \frac{b}{(\gamma-a)}E((\gamma,-\tau)) \qquad (2.4.5)$$

The delay-free dynamics and the delay components of the transfer function, Equation (2.4.5), can still be clearly identified.

In this process the delay was assumed to be an *input-delay* since the delay in the transfer function occurs as a result of a delay in the actuation. This delay could occur, as suggested by Marshall, (1979, p2) if the plant is remote from the controller and the signals are transmitted acoustically. In this case a delay of 0.003 sm^{-1} would be incurred. This delay could be significant if the distance is large or the delay-free dynamics are fast.

Some measurements take an appreciable length of time to be performed and in some processes it is not always possible to place a sensor close to the plant due to environmental hazards. In these cases *measurement*, or *output*, *delays* would occur.

2.5 CONTROL OF TIME-DELAYED PROCESSES

In this section methods that have been used to control time-delayed processes will be reviewed.

2.5.1 Interactor Matrix Methods

An interactor matrix for the proper transfer function $\mathbf{G}(\gamma)=\mathbf{A}(\gamma)^{-1}\mathbf{B}(\gamma)$, is $\xi(\gamma)$ with the property

$$\lim_{\gamma \to \infty} \xi(\gamma).G(\gamma) = \mathbf{K} \qquad (2.5.1)$$

where $\mathbf{K}$ is a non-singular constant matrix.

Elliott and Wolovich (1984) recognised that the interactor matrix of Wolovich and Falb (1976) was the multivariable extension of the concept of time delay. This discovery gave rise to a number of algorithms using the interactor matrix. Using this device various control design algorithms have been studied. For example, De Souza (1983) and Dugard, Goodwin and Xianya (1984) use the interactor matrix as the basis for their controller designs.

These methods give rise primarily to model reference controllers. A major problem with the interactor matrix methods is that the control law obtained depends on the order in which the variables appear in the input and output vectors.

Goodwin and Sin (1984, p133-142) give a good account of this approach.

2.5.2 Input-Output Matrix Method

An alternative approach to the interactor matrix has been studied by Toivonen (1984). In this approach, the delays are identified as being either input or output delays. The transfer function can then be factorised into diagonal input and output delay matrices and a matrix that contains no delay elements. This approach is extended in Chapter 3 of this thesis to allow non-diagonal input and output delay matrices.

Tade, Bayoumi and Bacon (1987a, 1987b) presented a self-tuning controller based on this idea of factorisation of the transfer function into input and output delay matrices and showed that if there is an interactor matrix then there is an input-output delay structure.

Following Tade *et al* (1987a, 1987b) let the input delay matrix be $\mathbf{D_1}(\gamma)$, and the output delay matrix be $\mathbf{D_2}(\gamma)$.

Together these have the property

$$\lim_{\gamma \to \infty} D_1(\gamma)G(\gamma)D_2(\gamma) = \mathbf{K_{io}} \qquad (2.5.2)$$

In general $\mathbf{K_{io}}$ is not the same as the $\mathbf{K}$ used in the definition of the interactor matrix.

The results presented in Chapter 3 provide some techniques for the decomposition of a general time-delayed process into the components $\mathbf{D_1}(\gamma)$, $\mathbf{D_2}(\gamma)$ and $G(\gamma)$.

2.5.3 Internal Model Control

The basic idea of the Internal Model Control (IMC) structure is the explicit inclusion in the block diagram description of the closed-loop process of the process model used in the design of the controller. This facilitates an analysis of the system's closed-loop behaviour when there is mismatch between the model and the process. This representation allows a robustness analysis of the system to be made.

The Internal Model Control method has been investigated by Garcia and Morari (1982, 1984a, 1984b), Morari (1983), Holt and Morari (1984,1985a, 1985b), Morari and Zafiriou (1989) and Garcia, Prett and Morari (1989).

Jerome and Ray (1986) and Shanmugathasan and Johnston (1988a, 1988b) used the IMC structure to reduce the system's settling-time by increasing certain of the system's delays.

2.5.4 The Smith Predictor

The Smith predictor (Smith 1957) allows the user to design a controller as if there were no delay and then use this controller in conjunction with a compensator, the Smith predictor, to control the delayed process.

The single-input single-output Smith predictor design method relies on the freedom to consider the single-input single-output delays as output-delays in order that any non-output delays can be passed through the transfer function. Any delays in the system then appear as an output delay.

Jerome and Ray (1986) consider the Smith predictor to have three properties. Firstly, when using the Smith predictor the time-delay is eliminated from the closed-loop characteristic equation. Secondly, for set-point changes the Smith predictor provides the controller with an immediate prediction of the effects of the control action on the system outputs. Finally, the Smith predictor implicitly factors the plant into two parts. These parts are the contribution of the delay, which is invertible only with prediction, and what might be considered the dynamics of the process (that part of the transfer-function that can be inverted without introducing predictive elements).

In the multivariable case it is not, in general, possible for the delays to be moved through the transfer function in the same simple way because matrix multiplication does not have the commutatively property. Results which enable the delays to be passed through the transfer function will be developed in Chapter 3.

Furukana and Shumemura (1983), Watanabe and Ito (1981) and Gawthrop (1977) pointed out that the classical Smith predictor cannot stabilise an unstable time-delayed process. However, De Paor (1985) and De Paor and Egan

(1989) produced a modification to the Smith predictor that allows for the stabilisation of unstable time-delayed processes.

Various multivariable versions of the Smith predictor have been developed.

Alevisakis and Seborg (1973, 1974) produced a multivariable Smith predictor for two special cases: two distinct delays in the measurement and a single delay affecting all the inputs equally.

Marshall (1979) and Walton and Marshall (1984) considered the problem of mismatch between the real process and the model. They showed that in some cases mismatch actually improves the performance of the closed-loop system.

Marshall (1979) explored two-input, two-output time-delayed processes using block diagrams. However Marshall did not generalise his results beyond this simple case, or using these results as a motivation for decomposing the matrix transfer function into a product of a matrix containing the delays and a matrix containing the delay-free dynamics.

Ogunnaike and Ray (1979) presented a Smith predictor method that assumes that all the pure delays and the transfer function coefficients and orders are known exactly, but do not consider the delays and the dynamics as being sequential operations acting on the inputs.

Jerome and Ray (1986) improved on this extension of the Smith predictor by allowing a much more general predictor. Using the internal model control (IMC) [Garcia and Morari (1982, 1984a, 1984b), Morari (1983), Holt and Morari (1984,1985a, 1985b)] structure they showed that improvements to the control could be made if some of the delays are increased. Shanmugathasan and Johnston (1988a, 1988b) presented a systematic method of choosing those delays in the system that should be increased in order to improve the control.

Astrom and Wittenmark (1984, p238) show that a discrete time version of the Smith predictor, in the single-input single-output case, can be derived using a pole-placement design.

Chotai and Young (1988) made a connection between the Smith predictor and a Non-Minimal State Space (NMSS) description.

2.5.5 Other Methods of Control of Time-Delayed Processes

Fuller (1968) showed that for a time-delayed system with all the delays equal the optimal controller has a similar structure to the Smith predictor. The controller found using this approach is called a predictor controller.

The idea of predictor controllers with more general delay structures has been investigated by Mee (1973) where it was shown that even when the delays are of different time lengths, only non time-delayed problems need to be solved, as with the Smith predictor, but the solution becomes very complicated. The solution involves a sequence of dynamic optimisation sub-problems that must be solved for each time step. Due to the large amount of computation involved, this approach is not, at present, suitable for adaptive control.

2.5.6 Robustness of Time-Delayed Controllers

In addition to a process being stable, industrial controllers must be robust in the face of a number of possible failures: inaccurate modelling, time-variation in the process, noise and sensor failure.

There is a large body of literature associated with the robustness of controllers. The robustness of adaptive controllers has been studied, for example, by Cluett, Shah and Fisher (1987), Cook and Chen (1985), Gawthrop (1985a, 1985b) and Ioannou and Sun (1988) The robustness of adaptive controllers is closely related to the convergence properties of the control schemes. Ljung (1977a,b) made major contributions to the analysis of the convergence properties of adaptive control schemes.

The problem of controlling a process that has been inaccurately modelled has received a great deal of attention in recent years. The advent of H_∞

control has made a major contribution to the understanding of this type of robustness. The IMC structure techniques have been used by Morari and Zafiriou (1989) to investigate the closed-loop stability of inaccurately modelled processes

The robustness to modelling errors of the Smith predictor and of other methods for the control of time-delayed processes has also been investigated. The researchers include: Marshall (1979) and Garland and Marshall (1974) Palmor (1980) and Palmor and Halevi (1983), Owens and Raya (1982), Palmor and Shinnar (1981), Horowitz (1983), Owens and Chotai (1983), and Chotai, Owens, Raya and Wang (1984).

2.6 NOTATION USED

Throughout this thesis the following conventions as to type-face will be adhered to: bold upper-case text will denote matrices; bold lower case text will denote vector quantities; italic text will denote polynomial or rational functions and an asterisk will indicate that the quantity includes time-delay.

$[\mathbf{X}]_{ij}$	=	ijth element of matrix **X**
$\mathbf{X}^{T}$	=	transpose of matrix **X**
$\hat{\mathbf{X}}$	=	an estimate of **X**
$\mathbf{X}^{-1}$	=	Inverse of matrix **X** (if **X** is square and of full rank)
$\mathbf{X}^{-}$	=	a generalised right inverse of **X**
$\mathbf{X}^{*}$	=	**X** contains time-delays
$\mathbf{X}^{+}$	=	**X** has been manipulated using either Theorem 3.1 or Theorem 3.2
$\Delta\mathbf{X}$	=	Difference between **X** and $\hat{\mathbf{X}}$, ie $\Delta\mathbf{X} = \mathbf{X} - \hat{\mathbf{X}}$

2.7 NOMENCLATURE

A	n×n	state-transition matrix
B	n×q_1	Control Input matrices (continous time)
C	r×n	Measurement matrix
$\boldsymbol{x}$	n×1	state vector
$\boldsymbol{y}$	r×1	measurement vector
w	q_2×1	disturbance vector
v	r×1	Measurement noise vector
$\boldsymbol{u}$	q_1×1	control input vector

Operators commonly used in this thesis

q	forward shift operator	$qx(t) = x(t+1)$
q^{-1}	backward shift operator	$q^{-1}x(t) = x(t-1)$
$\mathcal{L}\{f(t)\}$	Laplace transform	$\int_0^\infty e^{-st} f(t)dt$
$\frac{d^i}{dt^i}$	ith derivative	
s	laplace domain operator	
γ_δ	delta-transform operator	
z^{-1}	z-domain operator	
δ	delta operator	$\delta = \frac{q-1}{\Delta}$
ρ	generalized operator	$\frac{d}{dt}, q^{-1}, \delta$
γ	generalised frequency domain variable	s,z^{-1}, γ_δ
E(γ,−τ)	Generalized exponential	

$\mathcal{D}\{f(t)\}$	Delta transform	$\Delta\sum_{k=0}^{\infty} f(t_k)(1+\Delta\gamma_\delta)^{-k}$,$t_k=t_{k-1}+\Delta$, k=1,2,...
$\mathcal{Z}\{f(t)\}$	Z-transform	$\sum_{k=0}^{\infty} z^{-k} f(t_k)$, $t_k=t_{k-1}+\Delta$, k=1,2,...
Δ	Sample time	
$g^*(\gamma)$	Single-input single-output time-delayed transfer function	
$g(\gamma)$	Single-input single-output transfer function	
$d(\gamma)$	pure-delay operator (single-input single-output)	

	description
$\xi(\gamma)$	interactor matrix
$D_1(\gamma)$	output delay matrix (r×m)
$D_2(\gamma)$	input delay matrix (q×m)
$G(\gamma)$	multivariable transfer function
$G^*(\gamma)$	time-delayed multivariable transfer function
$G_D(\gamma)$,$G_D{}^*(\gamma)$	multivariable transfer function in which the input and output delays have been removed.
n_a,n_b,n_c	number of different time-delays appearing in the state-transition,output and input delay matrices respectively
e_i	jth column of identity matrix
$\mathbf{D}_{\mathbf{Y}_i}$	diagonal matrix having $\mathbf{Y}_i$, the ith row of $\mathbf{Y}$ down its main diagonal

$\mathbf{D}_{\mathbf{Y}_i}$	diagonal matrix having $\mathbf{Y}_i$, the ith row of $\mathbf{Y}$ down its main diagonal
$\mathbf{G}_1^+(\gamma)$	delay-free transfer function matrix of dimension $mn^2q\times 2$
$\mathbf{G}_2^+(\gamma)$	delay-free transfer function matrix of dimension $p\times pm^2n$
$\mathbf{D}_{p1}(\gamma)$	delay matrix containing only delays of the plant transfer function ($m\times nm$)
$\mathbf{D}_{p2}(\gamma)$	delay matrix containing only delays of the plant transfer function ($nm\times m$)
$\mathscr{D}_i(\gamma)$	delay matrix containing only input delays of the plant ($nm\times n^2mq$)
$\mathscr{D}_o(\gamma)$	delay matrix containing only delays (pm^2 $\times nm$)
$g_{ij}^+(\gamma)$	scalar transfer function relating ith output of G^+ to jth input to G^+
$g_{ijk}^+(\gamma)$	kth quotient in the partial fractions expansion of $g_{ij}^+(\gamma)$
$B^*(q^{-1})$	Control Input matrix including time-delays
$\mathbf{C}^*(q^{-1})$	Measurement matrix including time-delays
B^+, C^+	the delay-free components of $B^*(q^{-1})$ and $\mathbf{C}^*(q^{-1})$ respectively
$\mathbf{D}_u(q^{-1}), D_v(q^{-1})$	the delay components of $B^*(q^{-1})$ and $\mathbf{C}^*(q^{-1})$ respectively
δ_{ij}	Kronicker delta $\delta_{ij} = \begin{cases} 1 & if\, i=j \\ 0 & otherwise \end{cases}$
Transfer function	discrete time $\mathbf{G}(z^{-1}) = \mathbf{A}(z^{-1})^{-1}\mathbf{B}(z^{-1})$

	continous time	$\mathbf{G}(s)=\mathbf{A}(s)^{-1}\mathbf{B}(s)$
	unified	$\mathbf{G}(\gamma)=\mathbf{A}(\gamma)^{-1}\mathbf{B}(\gamma)$
State space description	continous time	$\frac{d\mathbf{x}}{dt}=\mathbf{Ax}(t)+\mathbf{Bu}(t)$ $\mathbf{y}(t)=\mathbf{Cx}(t)$
	discrete time	$\mathbf{x}(t+1)=\mathbf{\Phi x}(t)+\mathbf{\Gamma u}(t)$ $\mathbf{y}(t)=\mathbf{Cx}(t)$
	unified time	$\rho\mathbf{x}=\mathbf{A}_{\gamma}\mathbf{x}(t)+\mathbf{B}_{\gamma}\mathbf{u}(t)$ $\mathbf{y}(t)=\mathbf{Cx}(t)$

Chapters 9 and 10 - units

Symbol	description	units	expected range
F	Concentrate flow-rate	proportion valve open	0-1
W	Cooling water flow-rate	proportion valve open	0-1
D	Output density	kg/kg	1.10 - 1.111
T	Final effect temperature	^{o}C	38-44^{o}C
P	Steam pressure	psi	7-15 psi
DF	Incoming whey density	kg/kg	1.00-1.05
TF	Incoming whey temperature	^{o}C	0-20^{o}C
PW	Water pressure	psi	
D_R	Density set point	kg/kg	1.10 - 1.111
T_R	Temperature set point	^{o}C	38-44^{o}C
T'(k)	T(k)-steady-state value of T	o	
D'(k)	D(k)-steady-state value of D	kg/kg	
F'	F(k)-steady-state value of F	proportion	
W'	W(k)-steady-state value of W	proportion	

PART B

ANALYSIS OF TIME-DELAYED PROCESSES

3 Multivariable Time-Delayed System Representations

3.1 Introduction

In this chapter representations of time-delays in multivariable processes are investigated. In Sections 3.2 and 3.3 it will become clear that time-delays in multivariable processes can be classified in three ways: input-delays, output-delays and plant-delays. These types of delay will be explored and the consequences of a process being comprised of them will be discussed.

In Sections 3.4 some of the problems associated with the representation of multivariable time-delayed processes will be addressed. Without loss of generality systems without time-delays can be considered as the sub-class of time-delayed systems in which all the delays are zero.

In Section 3.5 Theorem 3.1, a result that allows a pseudo-commutativity of matrix multiplication, is proved.

In Section 3.6 Theorem 3.2, a result that enables an element by element matrix product to be written as the product of two matrices is proved.

These results are motivated by time-delay considerations and are used in Section 3.7 where it is shown that any time-delayed process can be represented as though all the delays act on the inputs of the process or, alternatively, on the outputs of the process. However Theorems 3.1 and 3.2 can also be applied to non-delayed processes, as will be shown in Chapter 4.

3.2 Representations of Continuous-Time Multivariable Time-Delayed Processes

In this section representations of continuous time delayed process are considered. These provide motivation for the results in the subsequent sections of this chapter. In Section 3.2.1 existing representations of continuous-time multivariable delayed processes are considered. These provide a motivation for a general representation of time-delayed processes which is presented in Section 3.2.2.

3.2.1 Existing Representations of Continuous-Time Multivariable Time-Delayed Processes.

Ogunnaike and Ray (1979) considered a general form of a continuous-time transfer function matrix:

$$\frac{d\mathbf{x}(t)}{dt} = \sum_{i=1}^{n_A}\mathbf{A}_i\mathbf{x}(t-\alpha_i) + \sum_{j=1}^{n_B}\mathbf{B}_j\mathbf{u}(t-\beta_j) \tag{3.2.1}$$

$$\mathbf{y}(t) = \sum_{i=k}^{n_C}\mathbf{C}_k\mathbf{x}(t-\gamma_k) \tag{3.2.2}$$

where where $x(t) \in \mathbf{R}^{n\times 1}$,$\mathbf{y}(t) \in \mathbf{R}^{r\times 1}$,$\mathbf{u}(t) \in \mathbf{R}^{q\times 1}$, α_i, β_j and γ_k are time-delays, the $\mathbf{A}_i$ are matrices of dimension $n \times n$, the $\mathbf{B}_j$ are matrices of dimension $n \times q$ and the $\mathbf{C}_k$ are matrices of dimension $r \times n$.

Processes that can be described in this form, Equations (3.2.1) and (3.2.2), are extremely general. A transfer function representation for such a process can be obtained by taking the Laplace transform of Equations (3.2.1) and (3.2.2).

$$y(s) = \left(\sum_{k=1}^{n_C}\mathbf{C}_k e^{-\gamma_k s}\right)\left(s\mathbf{I} - \sum_{i=1}^{n_A}\mathbf{A}_i e^{-\alpha_i s}\right)\left(\sum_{j=1}^{n_B}\mathbf{B}_j e^{-\beta_j s}\right) \tag{3.2.3}$$

Consider now only the subclass of all time-delayed processes in which $n_A = 1$ and $\alpha_i = 0$.

The transfer function can be obtained as:

$$\mathbf{y}(s) = \boldsymbol{G}^*(s)\mathbf{u}(s) \qquad (3.2.4)$$

where

$$\mathbf{G}^*(s) = \boldsymbol{D}_1(s)\boldsymbol{G}_{\mathbf{p}}(s)\boldsymbol{D}_2(s) \qquad (3.2.5)$$

and

$$D_1(s) = \sum_{k=1}^{n_C} C_k e^{-\gamma_k s} \qquad (3.2.6)$$

$$D_2(s) = \sum_{j=1}^{n_B} \mathbf{B}_j e^{-\beta_j s} \qquad (3.2.7)$$

$$G_{\mathbf{p}}(s) = (s\mathbf{I} - \mathbf{A})^{-1} \qquad (3.2.8)$$

Note that, as it is defined, $G^*(s)$ has a structure that is a less general representation of pure delays than would normally arise from typical modelling procedures, using for example step, impulse or frequency response measurements of the actual process. Such modelling procedures typically give rise to a transfer function matrix of the form:

$$G^{*}(s)=\begin{pmatrix} g_{11}(s)\,g_{12}(s)\ldots g_{1n}(s)\\ g_{21}(s)g_{22}(s)\ldots g_{2q}(s)\\ \cdot \quad \cdot \quad \cdot\\ \cdot \quad \cdot \quad \cdot\\ \cdot \quad \cdot \quad \cdot\\ g_{r1}(s)\;g_{r1}(s)\ldots\;g_{rq}(s)\end{pmatrix} \quad \text{where } g_{ij}(s)=\frac{a_{ij}(s)}{b_{ij}(s)}e^{-\tau_{ij}s} \qquad (3.2.9)$$

and the $a_{ij}(s)$ and the $b_{ij}(s)$ are polynomials in s.

A transfer function of the form shown in Equation (3.2.9) can be represented in the form:

$$G^{*}=D_1G_p{}^{*}D_2 \qquad (3.2.10)$$

using a method adapted from Tade *et al* (1988a,b). By letting

$$d_{1i}=\underset{j}{\max}\,(\tau_{ij}) \qquad (3.2.11)$$

$$d_{2j}=\underset{i}{\max}\,(\tau_{ij}-d_{1i}) \qquad (3.2.12)$$

$$\tau_{ij}{}^{*}=\tau_{ij}-d_{1i}-d_{2j} \qquad (3.2.13)$$

it can be shown that $D_1(s)$,$D_2(s)$ and $G_p{}^{*}(s)$ can be written as follows:

$$D_1(s)=\begin{pmatrix} e^{-d_{11}s} & 0 & \ldots & 0\\ 0 & e^{-d_{12}s} & & \cdot\\ \cdot & & & \\ 0 & \ldots & & e^{-d_{1r}s}\end{pmatrix} \qquad (3.2.14)$$

$$D_2(s)=\begin{pmatrix} e^{-d_{21}s} & 0 & \ldots & 0\\ 0 & e^{-d_{22}s} & & \cdot\\ \cdot & & & \\ 0 & \ldots & & e^{-d_{2q}s}\end{pmatrix} \qquad (3.2.15)$$

and

$$G_p^*(s) = \begin{pmatrix} \frac{a_{11}(s)e^{-\tau_{11}^*s}}{b_{11}(s)} & \frac{a_{12}(s)e^{-\tau_{12}^*s}}{b_{12}(s)} & \cdots & \frac{a_{1q}(s)e^{-\tau_{1q}^*s}}{b_{1q}(s)} \\ \frac{a_{21}(s)e^{-\tau_{21}^*s}}{b_{21}(s)} & \frac{a_{22}(s)e^{-\tau_{22}^*s}}{b_{22}(s)} & \cdots & \frac{a_{2q}(s)e^{-\tau_{2q}^*s}}{b_{2q}(s)} \\ \vdots & \vdots & & \vdots \\ \frac{a_{r1}(s)e^{-\tau_{r1}^*s}}{b_{r1}(s)} & \frac{a_{r2}(s)e^{-\tau_{r2}^*s}}{b_{r2}(s)} & \cdots & \frac{a_{rq}(s)e^{-\tau_{rq}^*s}}{b_{rq}(s)} \end{pmatrix} \tag{3.2.16}$$

In this section it has been shown that the time-delays in a multivariable transfer function can appear in D_1, as output-delays, in D_2, as input-delays, or appear in G_p^*, as plant-delays. In Section 3.2.2 a more general representation of continuous-time multivariable time-delayed processes will be developed.

3.2.2 A General Representation of Continuous-Time Multivariable Time-Delayed Processes

Equation (3.2.3) is of a more general form than that considered in Section 3.2.1 since it includes the possibility of state-delays. However for the purpose of this work, the most general form considered will be:

$$G^*(s) = D_1(s)G_p^*(s)D_2(s) \tag{3.2.17}$$

where

$$G_p{}^*(s) = \begin{pmatrix} \dfrac{a_{11}(s)e^{-\tau_{11}{}^*s}}{b_{11}(s)} & \dfrac{a_{12}(s)e^{-\tau_{12}{}^*s}}{b_{12}(s)} & \cdots & \dfrac{a_{1n}(s)e^{-\tau_{1n}{}^*s}}{b_{1n}(s)} \\ \dfrac{a_{21}(s)e^{-\tau_{21}{}^*s}}{b_{21}(s)} & \dfrac{a_{22}(s)e^{-\tau_{22}{}^*s}}{b_{22}(s)} & \cdots & \dfrac{a_{2n}(s)e^{-\tau_{2n}{}^*s}}{b_{2n}(s)} \\ \vdots & \vdots & & \vdots \\ \dfrac{a_{m1}(s)e^{-\tau_{m1}{}^*s}}{b_{m1}(s)} & \dfrac{a_{m2}(s)e^{-\tau_{m2}{}^*s}}{b_{m2}(s)} & \cdots & \dfrac{a_{mn}(s)e^{-\tau_{mn}{}^*s}}{b_{mn}(s)} \end{pmatrix} \tag{3.2.18}$$

and

$$D_1(s) = \sum_{k=1}^{n_C} C_k e^{-\gamma_k s} \tag{3.2.19}$$

$$D_2(s) = \sum_{j=1}^{n_B} B_j e^{-\beta_j s} \tag{3.2.20}$$

$D_1(s)$ is $r{\times}m$, $D_2(s)$ is $n{\times}q$ and $G_p(s)$ is $m{\times}n$. The dimension of $G^*(s)$ remains as before $r{\times}q$. $\mathbf{C}_k$, γ_k and $\mathbf{B}_j$, β_j are not necessarily linked to the general system description shown in Equations (3.2.1) and (3.2.2). This is a more general form of time-delayed representation than that presented in Section 3.2.1 since it allows the delay operators, $D_1(s)$ and $D_2(s)$ to be non-square.

The transfer function, Equations (3.2.17) (3.2.18) (3.2.19) and (3.2.20), can be seen to be comprised of three components: $D_1(s)$, the output-delays; $D_2(s)$, the input-delays and $G_p{}^*(s)$, the plant itself, involving delayed elements.

3.3 Discrete-Time and Unified Transform Time-Delayed Multivariable Process Descriptions

In a completely analogous manner a general discrete time-delayed process transfer function description can be obtained, using the z-transform

$$G^*(z) = D_1(z)G_p^*(z)D_2(z) \qquad (3.3.1)$$

where

$$D_1(z) = \sum_{i=1}^{n_C} C_i z^{-i} \qquad (3.3.2)$$

$$D_2(z) = \sum_{i=1}^{n_B} B_i z^{-i} \qquad (3.3.3)$$

and

$$[G_p^*(z)]_{ij} = \frac{z^{-\tau_{ij}} b_{ij}(z)}{a_{ij}(z)} \qquad (3.3.4)$$

where $a_{ij}(z)$ and $b_{ij}(z)$ are polynomials in z^{-1}.

Using the generalised exponential function and the unified transform (Middleton and Goodwin, 1990) outlined in Section 2.3, the continuous-time and the discrete-time time-delay representations can be considered in a single framework. In this form, the general representation of a time-delayed multivariable process is of the same form as that presented in Section 3.2.2, except that the delay operators are replaced with generalised exponential functions, $E(\gamma,-\tau_{\iota\varphi})$:

$$\mathbf{y}(\gamma) = \boldsymbol{G}^*(\gamma)\mathbf{u}(\gamma) = D_1(\gamma)G_p^*(\gamma)D_2(\gamma)\mathbf{u}(\gamma) \qquad (3.3.5)$$

where

$$D_1(\gamma) = \sum_{k=1}^{n_C} \mathbf{C}_k E(\gamma,-\gamma_k) \qquad (3.3.6)$$

$$D_2(\gamma) = \sum_{j=1}^{n_B} \mathbf{B}_j E(\gamma,-\beta_j) \qquad (3.3.7)$$

and

$$G_p^*(\gamma) = \begin{pmatrix} \frac{a_{11}(\gamma)E(\gamma,-\tau_{11})}{b_{11}(\gamma)} & \frac{a_{12}(\gamma)E(\gamma,-\tau_{12})}{b_{12}(\gamma)} & \cdots & \frac{a_{1n}(\gamma)E(\gamma,-\tau_{1n})}{b_{1n}(\gamma)} \\ \frac{a_{21}(\gamma)E(\gamma,-\tau_{21})}{b_{21}(\gamma)} & \frac{a_{22}(\gamma)E(\gamma,-\tau_{22})}{b_{22}(\gamma)} & \cdots & \frac{a_{2n}(\gamma)E(\gamma,-\tau_{2n})}{b_{2n}(\gamma)} \\ \vdots & \vdots & & \vdots \\ \frac{a_{m1}(\gamma)E(\gamma,-\tau_{m1})}{b_{m1}(\gamma)} & \frac{a_{m2}(\gamma)E(\gamma,-\tau_{m2})}{b_{m2}(\gamma)} & \cdots & \frac{a_{mn}(\gamma)E(\gamma,-\tau_{mn})}{b_{mn}(\gamma)} \end{pmatrix} \qquad (3.3.8)$$

where D_1 is a $r \times m$ matrix, D_2 is an n×q matrix and G_p^* is an m×n matrix.

This model will be used in the remainder of this thesis whenever a general time-delayed multivariable process is discussed.

3.4 ALTERNATIVE REPRESENTATIONS OF A SYSTEM'S TIME-DELAY STRUCTURE

Equations (3.3.5)-(3.3.8) provide a general description of a system's time-delay structure in terms of input, internal and output delays operating on the process variables. Since, in general, matrix products are not commutative, it is not easily possible to make input delays appear as output delays and vice versa, as illustrated by Example 3.1. However, the results presented in the later sections of this chapter provide a means by which the input-delays can be represented as output-delays.

EXAMPLE 3.1 : Consider a simple 2 by 2 transfer function in which the inputs are delayed by 2 and 1 time units respectively.

$$\boldsymbol{G}^*(\mathrm{s}) = \begin{pmatrix} \dfrac{1}{(s+1)} & \dfrac{3}{(s+2)} \\ \dfrac{2}{(s+3)} & \dfrac{-1}{(s+4)} \end{pmatrix} \begin{pmatrix} e^{-2s} & 0 \\ 0 & e^{-s} \end{pmatrix}$$

$$= \begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} & \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-2s}}{(s+3)} & \dfrac{-e^{-s}}{(s+4)} \end{pmatrix} \qquad (3.4.1)$$

Suppose that this transfer function is to be written in the form $\boldsymbol{G}^*(\mathrm{s}) = \boldsymbol{D}(\mathrm{s})\boldsymbol{G}(\mathrm{s})$ in which $\boldsymbol{G}(\mathrm{s})$ contains no time-delay elements and $\boldsymbol{D}(\mathrm{s})$ represents a matrix of output delay elements. One way to form $\boldsymbol{G}(\mathrm{s})$ from $\boldsymbol{G}^*(\mathrm{s})$ is by simply dropping out the delay elements. Then $\boldsymbol{G}(\mathrm{s})$ can be written as

$$\boldsymbol{G}(\mathrm{s}) = \begin{pmatrix} \dfrac{1}{(s+1)} & \dfrac{3}{s+2} \\ \dfrac{2}{s+3} & \dfrac{-1}{s+4} \end{pmatrix} \qquad (3.4.2)$$

and then $\boldsymbol{G}^*(\mathrm{s})$ would have to be

$$\boldsymbol{G}^*(\mathrm{s}) = \begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} & \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-2s}}{(s+3)} & \dfrac{-e^{-s}}{(s+4)} \end{pmatrix} = \begin{pmatrix} d_{11}(s) d_{12}(s) \\ d_{21}(s) d_{22}(s) \end{pmatrix} \begin{pmatrix} \dfrac{1}{(s+1)} & \dfrac{3}{s+2} \\ \dfrac{2}{s+3} & \dfrac{-1}{s+4} \end{pmatrix} = \mathbf{D}(s) \begin{pmatrix} \dfrac{1}{(s+1)} & \dfrac{3}{s+2} \\ \dfrac{2}{s+3} & \dfrac{-1}{s+4} \end{pmatrix} \tag{3.4.3}$$

where $D(s)$ is a 2 × 2 matrix which is to be determined. By expanding the matrix product the following set of equations can be obtained.

$$\begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} \\ \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-2s}}{(s+3)} \\ \dfrac{-e^{-s}}{(s+4)} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{(s+1)} & \dfrac{2}{s+3} & 0 & 0 \\ \dfrac{3}{s+2} & \dfrac{-1}{s+4} & 0 & 0 \\ 0 & 0 & \dfrac{1}{(s+1)} & \dfrac{2}{s+3} \\ 0 & 0 & \dfrac{3}{s+2} & \dfrac{-1}{s+4} \end{pmatrix} \begin{pmatrix} d_{11}(s) \\ d_{12}(s) \\ d_{21}(s) \\ d_{22}(s) \end{pmatrix} \tag{3.4.4}$$

Clearly this can be separated into two subproblems:

$$\begin{pmatrix} \dfrac{e^{-2s}}{s+1} \\ \dfrac{3e^{-s}}{s+2} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{s+1} & \dfrac{2}{s+3} \\ \dfrac{3}{s+2} & \dfrac{-1}{s+4} \end{pmatrix} \begin{pmatrix} d_{11}(s) \\ d_{12}(s) \end{pmatrix} \tag{3.4.5}$$

and

$$\begin{pmatrix} \dfrac{2e^{-2s}}{s+3} \\ \dfrac{-e^{-s}}{s+4} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{s+1} & \dfrac{2}{s+3} \\ \dfrac{3}{s+2} & \dfrac{-1}{s+4} \end{pmatrix} \begin{pmatrix} d_{21}(s) \\ d_{22}(s) \end{pmatrix} \tag{3.4.6}$$

From the first of these two subproblems it is clear that

$$\begin{pmatrix} d_{11}(s) \\ d_{12}(s) \end{pmatrix} = \begin{pmatrix} \frac{1}{s+1} & \frac{2}{s+3} \\ \frac{3}{s+2} & \frac{-1}{s+4} \end{pmatrix}^{-1} \begin{pmatrix} \frac{e^{-2s}}{s+1} \\ \frac{3e^{-s}}{s+2} \end{pmatrix} \tag{3.4.7}$$

and similarly from the second

$$\begin{pmatrix} d_{21}(s) \\ d_{22}(s) \end{pmatrix} = \begin{pmatrix} \frac{1}{s+1} & \frac{2}{s+3} \\ \frac{3}{s+2} & \frac{-1}{s+4} \end{pmatrix}^{-1} \begin{pmatrix} \frac{2e^{-2s}}{s+3} \\ \frac{-e^{-s}}{s+4} \end{pmatrix} \tag{3.4.8}$$

Since the inverse matrix exists, except for a finite number of values of s, and can be shown to be

$$\begin{aligned} \begin{pmatrix} \frac{1}{s+1} & \frac{2}{s+3} \\ \frac{3}{s+2} & \frac{-1}{s+4} \end{pmatrix}^{-1} &= \frac{1}{\frac{-1}{(s+1)(s+4)} - \frac{6}{(s+3)(s+2)}} \begin{pmatrix} \frac{-1}{s+4} & \frac{-2}{s+3} \\ \frac{-3}{s+2} & \frac{1}{s+1} \end{pmatrix} \\ &= \frac{(s+1)(s+2)(s+3)(s+4)}{(s+3)(s+2) + 6(s+1)(s+4)} \begin{pmatrix} \frac{1}{s+4} & \frac{2}{s+3} \\ \frac{3}{s+2} & \frac{-1}{s+1} \end{pmatrix} \end{aligned} \tag{3.4.9}$$

it is possible to write

$$d_{11}(s) = \frac{e^{-2s}(s+2)(s+3)+6e^{-s}(s+1)(s+4)}{(s+3)(s+2)+6(s+1)(s+4)} \qquad d_{12}(s) = \frac{3(e^{-2s} - e^{-s})(s+3)(s+4)}{(s+3)(s+2)+6(s+3)(s+4)}$$

$$d_{21}(s) = \frac{2(e^{-2s}-e^{-1s})(s+1)(s+2)}{(s+3)(s+2)+6(s+1)(s+4)} \qquad d_{22}(s) = \frac{6e^{-2s}(s+1)(s+4)+e^{-s}(s+2)(s+3)}{(s+3)(s+2)+6(s+1)(s+4)}$$

(3.4.10)

These $d_{ij}(s)$ terms are complicated to calculate even for this simple example. Furthermore the elements of $\boldsymbol{D}(s)$ bear little similarity to the relatively simple form of the delays in the original form of $\boldsymbol{G}^*(s)$.

3.5 A Pseudo-Commutativity Result

As has been shown by the preceding example representing input delays as output delays is not trivial since in general a matrix product cannot be commuted. However, a commutativity result that enables input-delays to be represented as output-delays is presented in this section.

Theorem 3.1 *Given matrices* **A**, *of dimension* $n \times m$ *and* **B**, *of dimension* $m \times q$, *the matrix product* **AB** *can be written as* $\mathcal{B}\mathcal{A}$ *where* $\mathcal{B}$ *is of dimension* $n \times mnq$, and $\mathcal{A}$ *is of dimension* $mnq \times q$. *The non-zero elements of the matrices* $\mathcal{A}$ and $\mathcal{B}$ *are rearrangements of* **A** *and* **B** *respectively. These matrices* $\mathcal{A}$ *and* $\mathcal{B}$ *are defined in equation* (3.5.6) *below.*

Proof: The ijth element of the product **AB** can be written by expanding the matrix multiplication, as

$$[\mathbf{AB}]_{ij} = \sum_{k=1}^{m} [\mathbf{A}]_{ik} [\mathbf{B}]_{kj} \tag{3.5.1}$$

Extra zero terms may be added to this sum

$$\begin{aligned}[\mathbf{AB}]_{ij} &= \sum_{k=1}^{m} [\mathbf{B}]_{k1} 0 + [\mathbf{B}]_{k2} 0 + [\mathbf{B}]_{k3} 0 + \ldots + \\ &[\mathbf{B}]_{k(j-1)} 0 + [\mathbf{A}]_{ik} [\mathbf{B}]_{kj} + [\mathbf{B}]_{k(j+1)} 0 + \ldots + [\mathbf{B}]_{kq} 0 \\ &= \sum_{k=1}^{m} [[\mathrm{B}]_{k1} [\mathrm{B}]_{k2} [\mathrm{B}]_{k3} \ldots\ldots [\mathrm{B}]_{k(j-1)}\ [\mathrm{B}]_{kj}\ [\mathrm{B}]_{k(j+1)} \ldots\ldots [\mathrm{B}]_{kq}]\ [\mathrm{A}]_{ik}\, \mathrm{e}_j\end{aligned} \tag{3.5.2}$$

where $\mathbf{e}_j$ is the jth column of $\mathbf{I}_q$, the identity matrix of dimension q.

Thus

$$[\mathbf{AB}]_{ij} = \sum_{k=1}^{m} \mathbf{B}_k[\mathbf{A}]_{ik}\mathbf{e}_j \qquad (3.5.3)$$

where $\mathbf{B}_k$ is the k^{th} row of **B**. Concatenating the rows of **B** together :

$$\mathcal{B}_R = [\mathbf{B}_1\ \mathbf{B}_2 \ldots\ldots \mathbf{B}_m]$$

it can be seen that

$$[\mathbf{AB}]_{ij} = \mathcal{B}_R \begin{pmatrix} [\mathbf{A}]_{i1}\mathbf{e}_j \\ [\mathbf{A}]_{i2}\mathbf{e}_j \\ \\ [\mathbf{A}]_{im}\mathbf{e}_j \end{pmatrix} \qquad (3.5.4)$$

Hence the i^{th} row of the product **AB** can be written as:

$$[[\mathbf{AB}]_{i1}\ [\mathbf{AB}]_{i2}\ \ldots\ [\mathbf{AB}]_{iq}] = \mathcal{B}_R \begin{pmatrix} [\mathbf{A}]_{i1}\mathbf{I}_q \\ [\mathbf{A}]_{i2}\mathbf{I}_q \\ \\ [\mathbf{A}]_{im}\mathbf{I}_q \end{pmatrix} \qquad (3.5.5)$$

By augmenting the matrices to form the n rows of the product, it can be seen that the product **AB** can be written as

$$\mathbf{AB} = \begin{pmatrix} \mathcal{B}_R & 0 \ldots & 0 \\ 0 & \mathcal{B}_R & \cdot \\ \cdot & \cdot & \cdot \\ 0 & 0 \ldots & \mathcal{B}_R \end{pmatrix} \begin{pmatrix} [\mathbf{A}]_{11}\mathbf{I}_q \\ [\mathbf{A}]_{12}\mathbf{I}_q \\ \cdot\cdot \\ [\mathbf{A}]_{1m}\mathbf{I}_q \\ [\mathbf{A}]_{21}\mathbf{I}_q \\ [\mathbf{A}]_{22}\mathbf{I}_q \\ \cdot\cdot \\ [\mathbf{A}]_{2m}\mathbf{I}_q \\ \cdot\cdot \\ \cdot\cdot \\ [\mathbf{A}]_{n1}\mathbf{I}_q \\ [\mathbf{A}]_{n2}\mathbf{I}_q \\ \cdot\cdot \\ [\mathbf{A}]_{nm}\mathbf{I}_q \end{pmatrix}$$

$$= \mathcal{B}\mathcal{A} \qquad (3.5.6)$$

This theorem provides what might be called a pseudo-commutativity result. In particular it will later be used to provide a means of reversing the order of the input-delays, $D_2(\gamma)$, and the plant transfer function, $D_1(\gamma)G_p(\gamma)$, so that to observers outside the system, it would appear as though the time-delays were acting on the outputs of a transfer function instead of on the inputs.

Example 3.2: The time-delayed transfer function, $G^*(s)$, used in Example 3.1 can be easily rewritten using Theorem 3.1 as though the input-delays were output-delays.

$$G^*(s) = \begin{pmatrix} \dfrac{1}{(s+1)} & \dfrac{3}{(s+2)} \\ \dfrac{2}{(s+3)} & \dfrac{-1}{(s+4)} \end{pmatrix} \begin{pmatrix} e^{-2s} & 0 \\ 0 & e^{-s} \end{pmatrix} \qquad (3.4.1)$$

$$=\begin{pmatrix} e^{-2s} & 0 & 0 & e^{-s} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-2s} & 0 & 0 & e^{-s} \end{pmatrix} \begin{pmatrix} \frac{1}{(s+1)} & 0 \\ 0 & \frac{1}{(s+1)} \\ \frac{3}{(s+2)} & 0 \\ 0 & \frac{3}{(s+2)} \\ \frac{2}{(s+3)} & 0 \\ 0 & \frac{2}{(s+3)} \\ \frac{-1}{(s+4)} & 0 \\ 0 & \frac{-1}{(s+4)} \end{pmatrix} \tag{3.5.7}$$

$$= \mathcal{D}_i(s)\mathbf{G}^+(s) \tag{3.5.8}$$

Here $\mathcal{D}_i(s)$ and $\boldsymbol{G}^+(s)$ are defined by the right hand side of Equation (3.5.7). $\mathcal{D}_i(s)$ is a delay operator on the outputs of the process $\boldsymbol{G}^+(s)$. Note how simply constructed this representation is compared with that of Example 3.1. The delays in $\mathcal{D}_i(s)$ are simply those in the original transfer function rearranged.

3.6 Decomposition of an Element by Element Matrix Product

Theorem 3.2: *The matrix* **M**:

$$\mathbf{M} = \begin{pmatrix} x_{11}y_{11} & x_{12}y_{12} & \cdots & x_{1n}y_{1n} \\ x_{21}y_{21} & x_{22}y_{22} & \cdots & x_{2n}y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1}y_{m1} & x_{m2}y_{m2} & \cdots & x_{mn}y_{mn} \end{pmatrix} \qquad (3.6.1)$$

which is an element by element product of two matrices **X** *and* **Y** *can be written as a matrix product of two matrices,* $\mathscr{X}$ and $\mathscr{Y}$, *where the elements of* $\mathscr{X}$ *and* $\mathscr{Y}$ *are rearrangements of the elements of* **X** *and* **Y** *respectively, that is*

$$\begin{pmatrix} \mathbf{X}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{X}_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \mathbf{X}_m \end{pmatrix} \begin{pmatrix} \mathbf{D}_{\mathbf{Y}_1} \\ \mathbf{D}_{\mathbf{Y}_2} \\ \vdots \\ \\ \vdots \\ \mathbf{D}_{\mathbf{Ym}} \end{pmatrix} = \mathscr{X}\mathscr{Y}$$

where X_i is the i^{th} row of **X** and $\mathbf{D}_{\mathbf{Y}i}$ is a diagonal matrix having $\mathbf{Y}_i$, the i^{th} row of **Y** down its main diagonal.

Proof: The i^{th} row of M can be written as

$$(x_{i1}\ x_{i2}\ x_{i3} \quad x_{in}) \begin{pmatrix} y_{i1} & 0 & . & . & & 0 \\ 0 & y_{i2} & 0 & . & & \\ . & 0 & & & & \\ & & 0 & y_{ij} & 0 & \\ & & . & . & 0 & \\ 0 & . & . & . & . & y_{in} \end{pmatrix} = \mathbf{X}_i\mathbf{D}_{\mathbf{Y}i} \qquad (3.6.2)$$

Here $\mathbf{X_i}$ is the i^th row of $\mathbf{X}$ and $\mathbf{D_{Y_i}}$ is a diagonal matrix having $\mathbf{Y_i}$, the i^th row of $\mathbf{Y}$ down its main diagonal. Hence it is possible to write the matrix $\mathbf{M}$ as

$$\mathbf{M} = \begin{pmatrix} \mathbf{X_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X_2} & \cdots & \\ \cdot & \cdot & & \\ \mathbf{0} & & & \mathbf{X_m} \end{pmatrix} \begin{pmatrix} \mathbf{D_{Y1}} \\ \mathbf{D_{Y2}} \\ \cdot \\ \cdot \\ \\ \mathbf{D_{Ym}} \end{pmatrix} = \mathcal{X}\mathcal{Y} \qquad (3.6.3)$$

This theorem provides a method in which plant-delays can be written as though they are output-delays, or alternatively as input-delays, as is demonstrated in the following example.

Example 3.3: The time-delayed transfer function, $\mathbf{G^*}$(s) can be easily rewritten using Theorem 3.2 as though the delays were output delays.

$$G^*(s) = \begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} & \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-3s}}{(s+3)} & \dfrac{-e^{-1.5s}}{(s+4)} \end{pmatrix} = \begin{pmatrix} e^{-2s} & e^{-s} & 0 & 0 \\ 0 & 0 & e^{-3s} & e^{-1.5s} \end{pmatrix} \begin{pmatrix} \dfrac{1}{(s+1)} & 0 \\ 0 & \dfrac{3}{s+2} \\ \dfrac{2}{s+3} & 0 \\ 0 & \dfrac{-1}{s+4} \end{pmatrix} \qquad (3.6.4)$$

$$= D_{p1}(s)G^+(s) \qquad (3.6.5)$$

Here $D_{p1}(s)$ and $G^+(s)$ are defined by the right hand side of Equation (3.6.5). $D_{p1}(s)$ is a delay operator on the outputs of the process $G^+(s)$. The delays in $D_{pl}(s)$ are simply those in the original transfer function, rearranged.

Alternatively the transfer function, equation (3.6.4), can be rewritten using Theorem 3.2 as though the delays were input-delays:

$$G^{*}(s) = \begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} & \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-3s}}{(s+3)} & \dfrac{-e^{-1.5s}}{(s+4)} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{(s+1)} & \dfrac{3}{s+2} & 0 & 0 \\ 0 & 0 & \dfrac{2}{s+3} & \dfrac{-1}{s+4} \end{pmatrix} \begin{pmatrix} e^{-2s} & 0 \\ 0 & e^{-s} \\ e^{-3s} & 0 \\ 0 & e^{-1.5s} \end{pmatrix} \tag{3.6.6}$$

$$= G^{+}(s)D_{p2}(s) \tag{3.6.7}$$

That is, the transfer function, Equation (3.6.4) has been rewritten so that the time-delays, which in the original transfer function, had appeared as input-delays, now appear as output-delays.

3.7 Decomposition of Time-Delayed Transfer Functions

In this section a theorem is presented which shows that any multivariable transfer function can be written with all the delays appearing as output delays, rather than as a mixture of input, plant and output delays. Alternatively the theorem can be used to write any multivariable transfer function with all the delays appearing as input delays.

Theorem 3.3: *The time-delayed multivariable transfer function* $G^*(s)$ *of Equation (3.3.5) can be written as*

$$G^*(\gamma) = D_1(\gamma) D_{p1}(\gamma) \mathscr{D}_i(\gamma) G_1{}^+(\gamma) \tag{3.7.1}$$

or as

$$G^*(\gamma) = G_2{}^+(\gamma) \mathscr{D}_0(\gamma) D_{p2}(\gamma) D_2(\gamma) \tag{3.7.2}$$

where

$G_1{}^+(\gamma)$ *is a delay-free transfer-function matrix of dimension* $mn^2q \times q$,

$G_2{}^+(\gamma)$ *is a delay-free transfer-function matrix of dimension* $p \times pm^2n$,

$D_{p1}(\gamma)$ *is of dimension* $m \times nm$ *and contains only the delays of the plant transfer function*

$D_{p2}(\gamma)$ *is of dimension* $mn \times n$ *and contains only the delays of the plant transfer function*

D_1 *and* D_2 *are matrices of dimensions* $r \times m$ *and* $q \times m$ *respectively*

$\mathscr{D}_o(\gamma)$ *is of dimension* $pm^2n \times mn$ contains only the output-delays

and $\mathscr{D}_i(\gamma)$ *is of dimension* $mn \times mn^2q$ *contains only the input-delays.*

REMARK: This theorem provides a way in which the general time delayed transfer function matrix, $G^*(\gamma)$ can be represented as though the delay matrices, $\mathcal{D}_i(\gamma)$, $D_p(\gamma)$, $D_1(\gamma)$, containing the input, plant and output delays respectively act sequentially on the output of a delay-free transfer function matrix, $G_1{}^+(\gamma)$. Aternatively the theorem can be used to present the time-delayed transfer function as though the delays $D_2(\gamma)$, $D_{p2}(\gamma)$, $\mathcal{D}_o(\gamma)$ act sequentially on the inputs and the resulting, delayed inputs are then acted upon by a delay-free transfer function , $G_2{}^+(\gamma)$.

Proof:

<u>Output Delay Form:</u>

From Equation (3.3.5) $G^*(\gamma) = D_1(\gamma)G_p(\gamma)D_2(\gamma)$. It must be shown that $G_p(\gamma)D_2(\gamma)$ can be represented as $D_p(\gamma)\mathcal{D}_i(\gamma)\mathcal{G}_1{}^+(\gamma)$. By identifying:

$$d_{ij}(\gamma) = E(\gamma,-t_{ij}{}^*) \tag{3.7.3}$$

and
$$g_{ij}(\gamma) = \frac{a_{ij}(\gamma)}{b_{ij}(\gamma)} \tag{3.7.4}$$

from the elements of $G_p(\gamma)$, Theorem 3.2 provides a way of decomposing $G_p(\gamma)$ as

$$G_p(\gamma) \quad = D_{p1}(\gamma)G_1(\gamma). \tag{3.7.5}$$

Hence

$$G^*(\gamma) \quad = D_1(\gamma)G_p(\gamma)D_2(\gamma) \tag{3.7.6}$$
$$= D_1(\gamma)D_{p1}(\gamma)G_1(\gamma)D_2(\gamma) \tag{3.7.7}$$

Theorem 3.1 shows that Equation (3.7.7) can be written as

$$G^*(\gamma) \;= D_1(\gamma)D_{p1}(\gamma)\mathscr{D}_i(\gamma)\mathscr{D}_1{}^+(\gamma) \tag{3.7.1}$$

as stated in the theorem.

Input Delay Form:

From Equation (3.3.5) $G^*(\gamma) = D_1(\gamma)G_p(\gamma)D_2(\gamma)$ · It remains to be shown that

$$D_1(\gamma)G_p(\gamma) \;= \mathscr{D}_2{}^+(\gamma)\mathscr{D}_o(\gamma)D_{p2}(\gamma) \tag{3.7.9}$$

Theorem 3.2 can be used to show that

$$G_p(\gamma) \;= G_2(\gamma)D_{p2}(\gamma) \tag{3.7.10}$$

and hence

$$\begin{aligned} G^*(\gamma) &= D_1(\gamma)G_p(\gamma)D_2(\gamma) && (3.7.11)\\ &= \mathscr{D}_2{}^+(\gamma)\mathscr{D}_o(\gamma)D_{p2}(\gamma)D_2(\gamma) && (3.7.12) \end{aligned}$$

as stated in the theorem.

This theorem provides the necessary tool to decompose any time-delayed transfer function into either a product of matrices of output-delays and the delay-free process or as a product of the delay-free process and input-delays. It is clear that this decomposition enlarges the inner dimension of the description, which might be thought to limit the usefulness of the decomposition. However the combination of this decomposition with a generalised Smith predictor, as will be shown in Chapter 6, shows that there is in fact no increase in the dimensionality of the closed-loop system.

3.8 REVIEW

In this chapter, representations of multivariable time-delayed processes have been investigated. A general form of time-delayed multivariable transfer function was derived in which the delays were identified as input, output or state delays.

Two key results, Theorems 3.1 and 3.2, were presented. Theorem 3.1 shows that a pseudo-commutativity of matrix multiplication is possible. Theorem 3.2 presents a method in which an element by element matrix product can be written as the product of two matrices.

Theorems 3.1 and 3.2 were then used to prove Theorem 3.3 which shows that a general time-delayed multivariable transfer function can be written with all the delays appearing as input-delays, or alternatively with all the delays appearing as output delays.

These results provide the necessary tools for the development of a multivariable Smith predictor, which is presented in Chapter 6. However, as will be shown in Chapter 4, Theorems 3.1 and 3.2 are completely general and can be applied to non time-delayed descriptions.

4 Formulation of State-Space Descriptions from Transfer Function Descriptions

4.1 Introduction

Model conversions between transfer function and state-space descriptions of processes are common problems encountered by control engineers. However, this reformulation often requires an artificial state-space to be created, or numerical methods, as opposed to analytical methods, to be applied.

The pseudo-commutativity theorems of Chapter 3 have a perhaps unsuspected use in converting transfer function descriptions into state-space descriptions. A method presented in this chapter offers a simple procedure to generate a block diagonal state-space description from any rational multivariable transfer function.

In Section 4.2, a simple 2 by 2 delay-free transfer function is reformulated as a state-space description, using Theorem 3.2. The method used in this transformation is then formalised in Section 4.3.

In Section 4.4 the reformulation method is extended to time-delayed processes.

In Section 4.5 the stability, observability and controllability of time-delayed processes is explored using an extension of delay-free state-space methods made possible by the formulation of time-delayed state-space descriptions in Section 4.4.

4.2 Reformulation of Transfer Function Descriptions as State-Space Descriptions

Consider Example 4.1 in which a state-space description of the process is obtained from a transfer function description of the process using Theorem 3.2.

Example 4.1:

Consider the delay-free process transfer function:

$$G(s) = \begin{pmatrix} \frac{1}{(s+1)} & \frac{3}{(s+2)} \\ \frac{2}{(s+3)} & \frac{-1}{(s+4)} \end{pmatrix} \tag{4.2.1}$$

Using Theorem 3.2 it may be represented as follows

$$G(s) = \begin{pmatrix} \frac{1}{(s+1)} & \frac{3}{(s+2)} \\ \frac{2}{(s+3)} & \frac{-1}{(s+4)} \end{pmatrix} = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{pmatrix} \begin{pmatrix} \frac{1}{(s+1)} & 0 \\ 0 & \frac{1}{s+2} \\ \frac{1}{s+3} & 0 \\ 0 & \frac{1}{s+4} \end{pmatrix} \tag{4.2.2}$$

$$= CG^{+}(s) \tag{4.2.3}$$

where

$$\mathbf{C} = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{pmatrix} \tag{4.2.4}$$

and

$$G^{+}(s) = \begin{pmatrix} \frac{1}{(s+1)} & 0 \\ 0 & \frac{1}{s+2} \\ \frac{1}{s+3} & 0 \\ 0 & \frac{1}{s+4} \end{pmatrix} \tag{4.2.5}$$

By using Theorem 3.2 again $G^{+}(s)$ can be expanded so that:

$$G(s) = \mathbf{C}\begin{pmatrix} \frac{1}{s+1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{s+2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{s+3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{s+4} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad (4.2.6)$$

Since rows 2, 3, 6 and 7 of the far right matrix in Equation (4.2.6) are all zero, these rows and the corresponding columns of the central matrix in Equation (4.2.6) make no contribution to the transfer function and may be removed. Hence

$$G(s) = \mathbf{C}\begin{pmatrix} \frac{1}{s+1} & 0 & 0 & 0 \\ 0 & \frac{1}{s+2} & 0 & 0 \\ 0 & 0 & \frac{1}{s+3} & 0 \\ 0 & 0 & 0 & \frac{1}{s+4} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbf{C}(s\mathbf{I}\text{-}\mathbf{A})^{-1}\mathbf{B} \qquad (4.2.7)$$

where

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (4.2.8)$$

The second equality of equation (4.2.7) defines $(s\mathbf{I} - \mathbf{A})^{-1}$. The matrix $\mathbf{A}$ can easily be seen to be the diagonal matrix

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \tag{4.2.9}$$

Hence a state-space description of this delay-free process is

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax}(t) + \mathbf{Bu}(t) \tag{4.2.10}$$

$$\mathbf{y}(t) = \mathbf{Cx}(t) \tag{4.2.11}$$

where **A**, **B**, and **C** are given by Equations (4.2.9), (4.2.8) and (4.2.4) respectively.

In this example, a state-space description of the process has been obtained from the transfer function description of the process using the Theorems 3.1 and 3.2. This derivation of a state-space description was extremely simple compared with other methods of deriving state-space descriptions from transfer function descriptions. The method can be extended to any multivariable transfer function matrix as is formalised in the following section.

4.3 A Method for Transforming Transfer Function Representations of Processes to State-Space Descriptions

In this section the method, suggested in Section 4.2, for obtaining state-space descriptions from transfer function descriptions of processes is formalised.

COROLLARY 4.1: *Any transfer function description of a process can be transformed to a state-space description using Theorem 3.2.*

Proof: There are four cases that must be considered. They are:

1. All elements of the multivariable transfer function matrix are first order;
2. One element of the multivariable transfer function is of second or higher degree with no repeated roots in that element;
3. Several elements of the multivariable transfer function have degree greater than one but no element has repeated roots;
4. Repeated roots in an element of the multivariable transfer function.

4.3.1 Case 1: All Elements of the Transfer Function are First Order

In Case 1, the transfer function is of the form

$$G^{*}(\gamma) = \begin{pmatrix} c_{11}\frac{b_{11}}{\gamma-a_{11}} & c_{12}\frac{b_{12}}{\gamma-a_{12}} & \cdots & c_{1q}\frac{b_{1q}}{\gamma-a_{1q}} \\ c_{21}\frac{b_{21}}{\gamma-a_{21}} & c_{22}\frac{b_{22}}{\gamma-a_{22}} & \cdots & c_{2q}\frac{b_{2q}}{\gamma-a_{2q}} \\ . & . & & . \\ . & . & & . \\ c_{r1}\frac{b_{r1}}{\gamma-a_{r1}} & c_{r2}\frac{b_{r2}}{\gamma-a_{r2}} & \cdots & c_{rq}\frac{b_{rq}}{\gamma-a_{rq}} \end{pmatrix} \qquad (4.3.1)$$

Using Theorem 3.2 this can be written as:

$$G(\gamma) = \mathbf{C}G^{+}(\gamma) \qquad (4.3.2)$$

where

$$\mathbf{C} = \begin{pmatrix} c_{11}b_{11} & c_{12}b_{12} & \cdots & c_{1q}b_{1q} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & c_{21}b_{21} & \cdots & 0 & 0 \\ . & . & \cdots & . & . & \cdots & . & . \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & c_{rq}b_{rq} \end{pmatrix}$$

$$(4.3.3)$$

and

$$G^{+}(\gamma) = \begin{pmatrix} \frac{1}{\gamma+a_{11}}\mathbf{I}_q \\ \frac{1}{\gamma+a_{12}}\mathbf{I}_q \\ . \\ . \\ . \\ \frac{1}{\gamma+a_{rq}}\mathbf{I}_q \end{pmatrix} \qquad (4.3.4)$$

Defining $\tilde{\mathbf{B}}$ such that

$$[\tilde{\mathbf{B}}]_{ij} = \begin{cases} 1 & \text{if } [G^{+}]_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.3.5}$$

Theorem 3.2 can be used to show that

$$G^{+} = \begin{pmatrix} G_1^{+} & 0 & 0 & \ldots & 0 \\ 0 & G_2^{+} & 0 & \ldots & 0 \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ 0 & 0 & 0 & \ldots & G_q^{+} \end{pmatrix} \begin{pmatrix} \mathbf{D}_{\tilde{\mathbf{B}}_1} \\ \mathbf{D}_{\tilde{\mathbf{B}}_2} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{D}_{\tilde{\mathbf{B}}_q} \end{pmatrix} \tag{4.3.6}$$

$$= \tilde{\mathbf{A}}\,\tilde{\mathbf{B}}^{+} \tag{4.3.7}$$

where $\mathbf{D}_{\tilde{\mathbf{B}}_i}$ is the diagonal matrix with the i^{th} row of $\tilde{\mathbf{B}}$ along the main diagonal.

From the structure of G^{+} in Equation (4.3.4) it is clear that each G_i^{+} (and the corresponding $\tilde{\mathbf{B}}$) contains exactly one non-zero element. Hence the columns of $\tilde{\mathbf{A}}$ and the rows of $\tilde{\mathbf{B}}^{+}$ containing only zero entries make no contribution to the transfer function and can be removed leaving q non-zero columns of $\tilde{\mathbf{A}}$ and q rows of $\tilde{\mathbf{B}}^{+}$, which can be identified with $(\gamma\mathbf{I}\text{-}\mathbf{A})^{-1}$ and $\mathbf{B}$ respectively.

$$G(\gamma) = \mathbf{C}(\gamma\mathbf{I}-\mathbf{A})^{-1}\mathbf{B} \tag{4.3.8}$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & .. & 0 \\ \cdot & \cdot & \cdot & & \\ 0 & 0 & 0 & \ldots & 1 \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & & 0 \\ & & & & \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix} \tag{4.3.9}$$

and

$$(\gamma\mathbf{I}\text{-}\mathbf{A})^{-1} = \begin{pmatrix} \frac{1}{\gamma\text{-}a_{11}} & 0 & 0\ldots & 0 \\ 0 & \frac{1}{\gamma\text{-}a_{12}} & 0\ldots & 0 \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & \ldots & \frac{1}{\gamma\text{-}a_{rq}} \end{pmatrix} \qquad (4.3.10)$$

Clearly since $(\gamma\mathbf{I}\text{-}\mathbf{A})^{-1}$ is a square and diagonal matrix, it can be inverted to obtain **A**:

$$\mathbf{A} = \begin{pmatrix} a_{11} & 0 & 0\ldots & 0 \\ 0 & a_{12} & 0\ldots & 0 \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & \ldots & a_{rq} \end{pmatrix} \qquad (4.3.11)$$

It is clear that each individual root adds one row and column to the state-transition matrix, one row to the **B** matrix and one column to the **C** matrix.

Example 4.1 illustrates the method of obtaining a state-space description of a two-input two-output transfer function description in which each element is of first order.

4.3.2 Case 2: One Element of the Transfer Function is of Second or Higher Degree

In the case where one element of the transfer function is of order two or higher that element of the transfer function is of the form

$$g_{ij}(\gamma) = c_{ij}\frac{\sum_{k} b_k\gamma^k}{\sum_{l} a_l\gamma^l} \qquad 0 \le k \le l, \qquad (4.3.12)$$

Since the order of the inputs and outputs is arbitrary, they can be reordered so that this element is in the $(1,1)^{\text{th}}$ position.

Hence as before:

$$G(\gamma) = \begin{pmatrix} c_{11}g^{+}{}_{11}(\gamma) & c_{12}g^{+}{}_{12}(\gamma) & \cdots & c_{1q}g^{+}{}_{1q}(\gamma) \\ c_{21}g^{+}{}_{21}(\gamma) & c_{22}g^{+}{}_{22}(\gamma) & \cdots & c_{2q}g^{+}{}_{2q}(\gamma) \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ c_{r1}g^{+}{}_{r1}(\gamma) & c_{r2}g^{+}{}_{r2}(\gamma) & \cdots & c_{rq}g^{+}{}_{rq}(\gamma) \end{pmatrix} \qquad (4.3.13)$$

$$= \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1q} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & c_{21} & . & \cdots & 0 \\ 0 & 0 & \cdots & . & . & . & \cdots & c_{rq} \end{pmatrix} \begin{pmatrix} g^{+}{}_{11}(\gamma) & 0 & .. & 0 \\ 0 & g^{+}{}_{12}(\gamma) & \cdots & 0 \\ . & . & \cdots & \\ 0 & 0 & \cdots & g^{+}{}_{1q}(\gamma) \\ g^{+}{}_{21}(\gamma) & 0 & \cdots & 0 \\ 0 & g^{+}{}_{22}(\gamma) & \cdots & \\ & & & g^{+}{}_{2q}(\gamma) \\ . & . & & . \\ g^{+}{}_{r1}(\gamma) & & & \\ & g^{+}{}_{r2}(\gamma) & & \\ & & & g^{+}{}_{rq}(\gamma) \end{pmatrix}$$

(4.3.14)

By assuming that there are no repeated roots in the denominator a partial fractions expansion of the first element of this transfer function matrix can be written as

$$g^{+}{}_{11}(\gamma) = \sum \frac{\beta_{11k}}{\gamma - \alpha_{11k}} = \sum g^{+}{}_{11k}(\gamma) \qquad (4.3.15)$$

One way of representing this summation in $G(\gamma)$ would be to add a row and column for each of these addends:

$$G(\gamma) = \begin{pmatrix} c_{11} & \dots & c_{11} & c_{12} & \dots & c_{1q} & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & c_{21} & . & \dots & 0 \\ \\ 0 & \dots & 0 & 0 & \dots & . & . & . & \dots & c_{rq} \end{pmatrix} \begin{pmatrix} g^{+}_{111}(\gamma) & 0 & .. & 0 \\ g^{+}_{112}(\gamma) & 0 & .. & 0 \\ \\ g^{+}_{11n}(\gamma) & 0 & .. & 0 \\ 0 & g^{+}_{12}(\gamma) & \dots & 0 \\ . & . & \dots & \\ 0 & 0 & \dots & g^{+}_{1q}(\gamma) \\ g^{+}_{21}(\gamma) & 0 & & \\ & & & .. \\ \\ 0 & 0 & \dots & g^{+}_{rq}(\gamma) \end{pmatrix}$$

(4.3.16)

where
$$g^{+}_{11j} = \sum \frac{\beta_{11j}}{\gamma - \alpha_{11j}} \qquad (4.3.17)$$

and applying the result of Theorem 3.2 on the right hand matrix, cancelling out the rows and columns with only zero entries leaves:

$$\begin{pmatrix} g^{+}{}_{111}(\gamma) & 0 & \dots & 0 \\ g^{+}{}_{112}(\gamma) & 0 & \dots & 0 \\ & & & \\ g^{+}{}_{11n}(\gamma) & 0 & \dots & 0 \\ 0 & g^{+}{}_{12}(\gamma) & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & \dots & g^{+}{}_{1q}(\gamma) \\ g^{+}{}_{21}(\gamma) & 0 & \dots & . \\ & & & \\ 0 & 0 & \dots & g^{+}{}_{rq}(\gamma) \end{pmatrix} = \qquad .$$

$$\begin{pmatrix} \frac{1}{\gamma-\alpha_{111}} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\gamma-\alpha_{112}} & \dots & 0 & 0 & \dots & 0 \\ . & . & & . & . & & 0 \\ 0 & \dots & \frac{1}{\gamma-\alpha_{11n}} & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & \frac{1}{\gamma-\alpha_{12}} & \dots & . & . \\ . & . & & & & & \frac{1}{\gamma-\alpha_{rq}} \end{pmatrix} \begin{pmatrix} \beta_{111} & 0 & \dots & 0 \\ \beta_{112} & 0 & \dots & 0 \\ . & . & & . \\ \beta_{11n} & 0 & & 0 \\ 0 & \beta_{12} & \dots & 0 \\ . & . & . & . \\ 0 & 0 & & \beta_{rq} \end{pmatrix}$$

$$= (\gamma\mathbf{I}\text{-}\mathbf{A})^{-1}\mathbf{B} \qquad (4.3.18)$$

As before the **A** matrix can be easily obtained since the $(\gamma\mathbf{I}\text{-}\mathbf{A})^{-1}$ matrix is diagonal.

Remarks:

1: The transfer function $G(\gamma) = \mathbf{C}(\gamma\mathbf{I}\text{-}\mathbf{A})^{-1}\mathbf{B}$ can be written as

$$[\mathbf{C}_{11}{}^{+}\ \mathbf{C}_{12}{}^{+}\ \ \mathbf{C}_{qr}{}^{+}] \begin{pmatrix} (\gamma\mathbf{I}\text{-}\mathbf{A}_{11}{}^{+})^{-1} & 0 & 0 & \dots \\ 0 & (\gamma\mathbf{I}\text{-}\mathbf{A}_{22}{}^{+})^{-1} & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & \dots & (\gamma\mathbf{I}\text{-}\mathbf{A}_{qr}{}^{+})^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{B}_{11}{}^{+} \\ \mathbf{B}_{12}{}^{+} \\ \\ \\ \mathbf{B}_{qr}{}^{+} \end{pmatrix}$$

$$(4.3.19)$$

where $\mathbf{A}_{ij}{}^{+}$ is the diagonal matrix with the poles of the ijth element of the transfer function,

$\mathbf{B}_{ij}{}^{+}$ contains the rows of $\mathbf{B}^{+}$ that contain the coefficients b_{ij} (and in the case of $\mathbf{B}^{+}{}_{11}$ contains the coefficients b_{ijk})

and $\mathbf{C}_{ij}{}^{+}$ are the columns of $\mathbf{C}^{+}$ that contain the coefficients c_{ij}.

2. A number of possible transformations of this process description exist: the order in which the inputs and outputs are indexed could be modified or the order in which the submatrices, $(\gamma\mathbf{I}\text{-}\mathbf{A}_{ij}{}^{+})$ appear on the diagonal of the innermost matrix of Equation (4.3.19) could be changed.

3. If the order of the inputs is modified, then only the order of the columns of $\mathbf{B}^{+}$ is changed. Similarly if the order of the outputs is modified then only the order of the rows of $\mathbf{C}^{+}$ is changed. In both these cases, the ordering of the innermost matrix is not effected.

4. If two of the submatrices $(\gamma\mathbf{I}\text{-}\mathbf{A}^{+}{}_{ii})$ are interchanged, the order of the corresponding $\mathbf{B}_{ij}{}^{+}$ are interchanged and the corresponding $\mathbf{C}_{ij}{}^{+}$ matrices are interchanged, the overall process description is unaffected. However, since in the innermost matrix only diagonal elements on the main diagonal have been interchanged, the innermost matrix remains diagonal. Hence the corresponding state-space description will retain a diagonal state-transition matrix.

Example 4.2: The process transfer function

$$G(s) = \begin{pmatrix} \frac{1}{(s+2)(s+1)} & \frac{3}{(s+2)} \\ \frac{2}{(s+3)} & \frac{-1}{(s+4)} \end{pmatrix} \qquad (4.3.20)$$

can be expanded using Theorem 3.2 so that:

$$G(s) = \mathbf{C}\begin{pmatrix} \frac{1}{(s+2)(s+1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{s+2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{s+3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{s+4} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 3 \\ 2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} \tag{4.3.21}$$

where $\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ (4.3.22)

Since the first element of this transfer function can be written as:

$$\frac{1}{(s+2)(s+1)} = \frac{1}{(s+1)} - \frac{1}{(s+2)} \tag{4.3.23}$$

the transfer function can be expanded as:

$$G(s) = \mathbf{C}^{+}\begin{pmatrix} \frac{-1}{(s+2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{(s+1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{s+2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{s+3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{s+4} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 3 \\ 2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} \tag{4.3.24}$$

where

$$\mathbf{C}^{+} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{4.3.25}$$

Since rows 2, 3, 6 and 7 of the far right matrix in Equation (4.3.24) are all zero, these rows and the corresponding columns of the central matrix in Equation (4.3.24) make no contribution to the transfer function and may be removed. Hence

$$G(s) = \mathbf{C}^{+}\begin{pmatrix} \frac{-1}{(s+2)} & 0 & 0 & 0 \\ \frac{1}{s+1} & 0 & 0 & 0 \\ 0 & \frac{1}{s+2} & 0 & 0 \\ 0 & 0 & \frac{1}{s+3} & 0 \\ 0 & 0 & 0 & \frac{1}{s+4} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 0 \\ 0 & -1 \end{pmatrix} \qquad (4.3.26)$$

or

$$G(s) = \mathbf{C}^{+}\begin{pmatrix} \frac{1}{(s+2)} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{s+1} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{s+2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{s+3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{s+4} \end{pmatrix}\mathbf{B} \qquad (4.3.27)$$

where

$$\mathbf{B} = \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 3 \\ 2 & 0 \\ 0 & -1 \end{pmatrix} \qquad (4.3.28)$$

The matrix **A** can easily be seen to be the diagonal matrix

$$\mathbf{A} = \begin{pmatrix} -2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & -4 \end{pmatrix} \qquad (4.3.29)$$

Hence a state-space description of this delay-free process is

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{4.3.30}$$

$$\mathbf{y}(t) = \mathbf{C}^{+}\mathbf{x}(t) \tag{4.3.31}$$

where **A**, **B** and **C** are given by Equations (4.3.29), (4.3.28) and (4.3.25) respectively.

4.3.3 Case 3: More than One Element is of Higher Order than Unity, but No Repeated Roots in any Element of the Transfer Function.

In the case where more than one element in the transfer function is of higher order a similar method can be applied. It is possible to move any element of the transfer function into the (1,1)th position by rearranging the rows and columns of the transfer function. Hence it is possible, by using Remark 4.1 to identify a diagonal submatrix $\mathbf{A}_{ij}$ as well as matrices $\mathbf{B}_{ij}^{+}$ and $\mathbf{C}_{ij}^{+}$ corresponding to each element of the transfer function. Hence it is a simple process to construct a state-space description for the system.

4.3.4 Case 4: Repeated Roots.

The remaining case to be considered is the case of repeated roots in any element of the transfer function.

Assume initially that the remainder of the transfer function is 'well-behaved' and that a single element, the (k,l)th element, of the transfer function that has repeated roots is of the form:

$$g_{kl}^{+}(\gamma) = \sum \frac{\beta_{klj}^{+}}{\gamma - \alpha_{klj}} + \sum_{i=1}^{n-1} \frac{\beta_{kli}\gamma^{j}}{(\gamma - \alpha_{kl})^{n}} \tag{4.3.32}$$

where j sums over the non-repeated roots.

The non-repeated roots, a_{klj}, can be dealt with as in Case 2 which leaves the subproblem of the repeated roots, a_{kl}. That portion of the input-output relationship that the repeated roots determine is given by:

$$y'(\gamma) = \sum_{i=1}^{n-1} \frac{\beta_{kli}\gamma^{i}}{(\gamma-\alpha_{kl})^{n}} u(\gamma) \qquad (4.3.33)$$

Expanding Equation (4.3.33) as:

$$(\gamma\text{-}a_{kl})^{n}y' = \sum_{i=1}^{n-1} b_{kli}\gamma^{i}u' \qquad (4.3.34)$$

It can be seen that Equation (4.3.34) can be written, using the binomial expansion, as:

$$\sum_{i=1}^{n} \binom{n}{i}(-a_{kl})^{(n-i)}\gamma^{i}y' = \sum_{i=1}^{n-1} b_{kli}\gamma^{i}u' \qquad (4.3.35)$$

A state-space description of this subproblem can then be constructed. One such state-space description would be:

$$\rho x(t) = \mathbf{A}_{kl}^{+}x(t) + \mathbf{B}_{kl}^{+}u'(t) \qquad (4.3.36)$$

$$y'(t) = \mathbf{C}_{kl}^{+}x(t) \qquad (4.3.37)$$

where

$$\mathbf{A}_{kl}^{+} = \begin{pmatrix} \mathbf{A}_{uu} & 0 \\ \mathbf{A}_{yu} & \mathbf{A}_{yy} \end{pmatrix} \qquad (4.3.38)$$

$$\mathbf{B}_{kl}^{+} = \begin{pmatrix} \mathbf{B}_{u} \\ \mathbf{B}_{y} \end{pmatrix} \qquad (4.3.39)$$

$$C^{+}_{kl} = \begin{bmatrix} C_u \\ C_y \end{bmatrix} \tag{4.3.40}$$

$$\mathbf{A}_{yu} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ . & . & . & \cdots & . \\ 0 & 0 & 0 & \cdots & 0 \\ b_{kl1} & b_{kl2} & b_{kl3} & \cdots & b_{kln-1} \end{pmatrix} \tag{4.3.41}$$

$$\mathbf{A}_{yy} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ . & . & . & \cdots & . \\ 0 & 0 & 0 & \cdots & 1 \\ \binom{n}{1}(-a_{kl})^{n-1} & \binom{n}{2}(-a_{kl})^{n-2} & \binom{n}{3}(-a_{kl})^{n-3} & \cdots & \binom{n}{n-1}(-a_{kl})^{1} \end{pmatrix} \tag{4.3.42}$$

$$\mathbf{A}_{uu} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ . & . & . & \cdots & . \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{4.3.43}$$

$$\mathbf{B}_{u} = \begin{pmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} \tag{4.3.44}$$

$$\mathbf{B}_{y} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ 1 \end{pmatrix} \tag{4.3.45}$$

$$\mathbf{C}_{y}^{\cdot} = (1 \ \ 0 \ \ 0 \ \ \ldots \ \ 0 \ \ 0) \tag{4.3.46}$$

This procedure can be repeated for each element in the transfer function that has repeated roots. A simple extension to Remark 4.1 ensures that it is possible to identify a block diagonal submatrix $\mathbf{A}_{ij}^{+}$ as well as matrices $\mathbf{B}_{ij}^{+}$ and $\mathbf{C}_{ij}^{+}$ corresponding to each element in the transfer function.

Example 4.3: Consider the process described by the transfer function:

$$G(s) = \begin{pmatrix} \dfrac{1}{(s+1)^2} & \dfrac{3}{(s+2)} \\ \dfrac{2}{(s+3)} & \dfrac{-1}{(s+4)} \end{pmatrix} \tag{4.3.47}$$

The $(1,1)^{th}$ element of this transfer function is the only element of this transfer function that has repeated roots. A state-space description for this element can be written as:

$$\frac{d\mathbf{x}_{11}(t)}{dt} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \mathbf{x}_{11}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}_1(t) \tag{4.3.48}$$

and

$$\mathbf{y}_1(t) = [1 \quad 0]\mathbf{x}_{11}(t) \tag{4.3.49}$$

The remaining elements in the transfer function are the same as those in the transfer function used in Example 4.1. Hence a state-space description of the process described by Equation (4.3.47) could be presented as:

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & -2 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & -4 \end{pmatrix} x(t) \; + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 3 \\ 2 & 0 \\ 0 & -1 \end{pmatrix} u(t) \tag{4.3.50}$$

and

$$\mathbf{y}(t) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \mathbf{x}(t) \qquad (4.3.51)$$

This method will always produce a state-space description which is block diagonal. However there are circumstances when it is possible to obtain a smaller state-space description by other analytical methods. Further work is required to extend the method described in this chapter to ensure that it produces minimal state-space descriptions.

4.4 TIME-DELAYED STATE-SPACE DESCRIPTIONS

In this section state-space descriptions of time-delayed processes are derived. It will be shown that these state-space descriptions are interesting since the state-transition matrices are the same in both descriptions and the time-delays appear only in the inputs or the outputs.

COROLLARY 4.2: *Any time-delayed process of the form presented in Equations* (3.3.5)-(3.3.8) *can be written in the state-space form*

$$\rho \mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + B^{*}(q^{-1})\mathbf{u}(t) \tag{4.4.1}$$

$$\mathbf{y}(t) = C^{*}(q^{-1})\mathbf{x}(t) \tag{4.4.2}$$

in which the delays appear either in the input matrix, $B^{*}(q^{-1})$ *or in the measurement matrix* $C^{*}(q^{-1})$, *where*

$$\mathbf{B}^{*}(q^{-1}) = \sum_{i=1}^{n_{b^{*}}} \mathbf{B}_i q^{-\tau_i} \tag{4.4.3}$$

and

$$\mathbf{C}^{*}(q^{-1}) = \sum_{i=1}^{n_{c^{*}}} \mathbf{C}_i q^{-\tau_i} \tag{4.4.4}$$

where $n_{b^{*}} \le n_b + n\ m$ and $n_{c^{*}} \le n_c + n\ m$

Proof:

The process can be represented as in Equations (3.3.5)-(3.3.8).

$$\mathbf{y}(\gamma) = G^{*}(\gamma)\mathbf{u}(\gamma) = D_1(\gamma)G_p(\gamma)D_2(\gamma)\mathbf{u}(\gamma) \tag{3.3.5}$$

$$D_1(\gamma) = \sum_{k=1}^{n_C} C_k E(\gamma,-\gamma_k) \tag{3.3.6}$$

$$D_2(\gamma) = \sum_{j=1}^{n_B} B_j E(\gamma,-\beta_j) \tag{3.3.7}$$

and

$$G_p(\gamma)=\begin{pmatrix} \frac{a_{11}(\gamma)E(\gamma,-\tau_{11})}{b_{11}(\gamma)} & \frac{a_{12}(\gamma)E(\gamma,-\tau_{12})}{b_{12}(\gamma)} & \cdots & \frac{a_{1n}(\gamma)E(\gamma,-\tau_{1n})}{b_{1n}(\gamma)} \\ \frac{a_{21}(\gamma)E(\gamma,-\tau_{21})}{b_{21}(\gamma)} & \frac{a_{22}(\gamma)E(\gamma,-\tau_{22})}{b_{22}(\gamma)} & \cdots & \frac{a_{2n}(\gamma)E(\gamma,-\tau_{2n})}{b_{2n}(\gamma)} \\ \vdots & \vdots & & \vdots \\ \frac{a_{m1}(\gamma)E(\gamma,-\tau_{m1})}{b_{m1}(\gamma)} & \frac{a_{m2}(\gamma)E(\gamma,-\tau_{m2})}{b_{m2}(\gamma)} & \cdots & \frac{a_{mn}(\gamma)E(\gamma,-\tau_{mn})}{b_{mn}(\gamma)} \end{pmatrix} \tag{3.3.8}$$

By expanding Equation (3.3.5) it can be seen that the elements of $G^*(\gamma)$ can be expressed as:

$$\left[G^*(\gamma)\right]_{ij} = \left[\sum_{k=1}^{ni,j} E(\gamma,\tau_{ijk})g_{ijk}(\gamma)\right]_{ij} \tag{4.4.5}$$

Now the $E(\gamma,\tau_{ij})$ terms can be grouped either with the b_{ij} or with the c_{ij} coefficients. If the $E(\gamma,\tau_{ijk})$ terms are grouped with the c_{ijk} coefficients the elements $b^*{}_{ijk}$ and $c^*{}_{ijk}$ can be defined as:

$$c^*{}_{ijk} = c_{ijk}E(\gamma,-\tau_{ijk}) \tag{4.4.6}$$

$$\begin{aligned} b^*{}_{ijk} &= b_{ijk} \\ &= b_{ijk}E(\gamma,0) \end{aligned} \tag{4.4.7}$$

Otherwise, grouping the $E(\gamma,\tau_{ijk})$ with the b_{ijk} coefficients allows the definition of b^*_{ijk} and c^*_{ijk} as

$$\begin{aligned} c^*_{ijk} &= c_{ijk} \\ &= c_{ijk}E(\gamma,0) \end{aligned} \qquad (4.4.8)$$

and

$$b^*_{ijk} = b_{ijk}E(g,\tau_{ijk}) \qquad (4.4.9)$$

In either case the transfer function can be written as

$$G(\gamma)=\begin{bmatrix} \sum_{k=1}^{n_{1.1}} c_{11k}{}^* b_{11k}{}^* g_{11k}(\gamma) & \sum_{k=1}^{n_{1.2}} c_{12k}{}^* b_{12k}{}^* g_{12k}(\gamma) & .. & \sum_{k=1}^{n_{1.q}} c_{1qk}{}^* b_{1qk}{}^* g_{1qk}(\gamma) \\ \sum_{k=1}^{n_{2.1}} c_{21k}{}^* b_{21k}{}^* g_{21k}(\gamma) & \sum_{k=1}^{n_{2.2}} c_{22k}{}^* b_{22k}{}^* g_{22k}(\gamma) & .. & \sum_{k=1}^{n_{2.q}} c_{2qk}{}^* b_{2qk}{}^* g_{2qk}(\gamma) \\ .. & .. & . & .. \\ \sum_{k=1}^{n_{r.1}} c_{r1k}{}^* b_{r1k}{}^* g_{r1k}(\gamma) & \sum_{k=1}^{n2} c_{r2k}{}^* b_{r2k}{}^* g_{r2k}(\gamma) & .. & \sum_{k=1}^{n_{r.q}} c_{rqk}{}^* b_{rqk}{}^* g_{rqk}(\gamma) \end{bmatrix}$$

(4.4.10)

The methodology used in Corollary 4.1 can be used to present the process description as:

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathrm{B}^*\mathbf{u}(t)) \qquad (4.4.11)$$

$$\mathbf{y}(t) = \mathbf{C}^*\mathbf{x}(t) \qquad (4.4.12)$$

The matrices $\mathbf{B}^*$ and $\mathbf{C}^*$ are of the form

$$M^*(\gamma)= \sum_k E(\gamma,-\tau_k)M_k{}^* \qquad (4.4.13)$$

It can be easily seen that $\mathbf{M}^*(\gamma)\mathbf{x}(\gamma)$ is the generalized transform of :

$$\sum_k M_k \mathbf{x}(t-\tau_k) = \left(\sum_k M_k q^{-\tau_k}\right)\mathbf{x}(t) \tag{4.4.14}$$

$$= M^*(q^{-1})\mathbf{x}(t) \tag{4.4.15}$$

where

$$M^*(q^{-1}) \quad = \sum_k \mathbf{M}_k{}^* \mathbf{q}^{-\tau_i} \tag{4.4.16}$$

where k sums over all the time-delays, and $M_i{}^*$ are a set of constant matrices that can be identified from the coefficients of M^* .

Hence Equations (4.4.11) (4.4.12), can also be written in the form:

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}^*(q^{-1})\mathbf{u}(t) \tag{4.4.17}$$

$$\mathbf{y}(t) \ = \mathbf{C}^*(q^{-1})\mathbf{x}(t) \tag{4.4.18}$$

One notable feature is that the state-transition matrix, **A**, is the same for both the input-delay form and the output-delay form. This is not entirely unexpected since this matrix determines the poles of the process. However it is now clear that even the plant-delays, that could not previously be separated from the dynamics of the process, can now be presented using either input or output delays.

In this form of state-space description the delays are intermingled with the delay-free dynamics of the process in the $B^*(q^{-1})$ and $C^*(q^{-1})$ matrices. The following corollary provides a means in which these can be separated.

COROLLARY 4.3: *Any time-delayed process of the form presented in Equations* (3.3.5)-(3.3.8) *can be written in either the input delay form of state-space description*

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}^{+}D_u(q^{-1})\mathbf{u}(t) \tag{4.4.19}$$

$$\mathbf{y}(t) = D_y(1)\mathbf{C}^{+}\mathbf{x}(\mathbf{t}) \tag{4.4.20}$$

or in the output delay form of state-space description:

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}^{+}D_u(1)\mathbf{u}(t) \tag{4.4.21}$$

$$\mathbf{y}(t) = D_y(q^{-1})\mathbf{C}^{+}\mathbf{x}(\mathbf{t}) \tag{4.4.22}$$

Proof: Corollary 4.2 provides a means in which the process can be described by Equations (4.4.1), (4.4.2). It remains to be shown that this state-space form can be written in either the form of Equations (4,4,19)-(4.4.20) or in the form of Equations (4.4.21)-(4.4.22).

In Corollary 4.2 the delays were grouped with either the b_{ijk} or with the c_{ijk}. if the $E(\gamma,-\tau)$ were associated with the b_{ijk}, then implicitly $E(0,-\tau)=1$ was associated with the c_{ijk} alternatively if the $E(\gamma,-\tau)$ were associated with the c_{ijk}, then implicitly $E(\gamma,0)=1$ was associated with the b_{ijk}. In either case Theorem 3.2 provides a means by which $B^*(q^{-1})$ and $C^*(q^{-1})$ can be written as

$$\mathbf{B}^*(q^{-1}) = \mathbf{B}^{+}\mathbf{D}_u(q^{-1}) \tag{4.4.23}$$

and

$$\boldsymbol{C}^*(q^{-1}) = \boldsymbol{D}_y(q^{-1})C^+ \qquad (4.4.24)$$

If the $E(\gamma,-\tau)$ were grouped with the b_{ijk} then the delays were considered to be input-delays, in which case $\boldsymbol{D}_y(q^{-1})=\boldsymbol{D}_y(1)$. Hence it can be seen that the process can be described in the input-delay form, Equations (4.4.19)-(4.4.20).

Alternatively if the delays were grouped with the c_{ijk} the they were considered to be output-delays and the process can be described in the output-delay form, Equations (4.4.21)-(4.4.22).

Example 4.4 : Consider the simple 2 by 2 transfer function

$$G^*(s) = \begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} & \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-3s}}{(s+3)} & \dfrac{-e^{-1.5s}}{(s+4)} \end{pmatrix} \qquad (4.4.25)$$

By grouping the delays in the c^*_{ij} elements the transfer function can be written using Corollary 4.1 in an output-delay state-space form:

$$\frac{dx}{dt} = \mathbf{A}\mathbf{x}(t) + \begin{pmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 0 \\ 0 & -1 \end{pmatrix}\mathbf{u}(t) \qquad (4.2.26)$$

$$y(t) = \begin{pmatrix} q^{-2} & q^{-1} & 0 & 0 \\ 0 & 0 & q^{-3} & q^{-1.5} \end{pmatrix}x(t) \qquad (4.4.27)$$

where

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \qquad (4.4.28)$$

or by grouping the delays in the b_{ij}^* the process can be rewritten using Corollary 4.1 in an input delay form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax}(t) + \begin{pmatrix} q^{-2} & 0 \\ 0 & 3q^{-1} \\ 2q^{-3} & 0 \\ 0 & -q^{-1.5} \end{pmatrix} \mathbf{u}(t) \qquad (4.4.29)$$

$$\mathbf{y}(t) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \mathbf{x}(t) \qquad (4.4.30)$$

Where **A** is defined in Equation (4.4.28).

By defining $\mathbf{B}^+$, $D_u(q^{-1})$ and $D_y(q^{-1})$

$$\mathbf{B}^+ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \qquad (4.4.31)$$

$$D_u(q^{-1}) = \begin{pmatrix} q^{-2} & 0 \\ 0 & q^{-1} \\ q^{-3} & 0 \\ 0 & q^{-1.5} \end{pmatrix} \qquad (4.4.32)$$

and

$$D_y(q^{-1}) = \begin{pmatrix} q^{-2} & q^{-1} & 0 & 0 \\ 0 & 0 & q^{-3} & q^{-1.5} \end{pmatrix} \qquad (4.4.33)$$

it can be seen, by inspection, that in both cases the process could be written as:

$$\frac{\mathbf{dx}}{\mathrm{dt}}(\mathrm{t}) = \mathbf{Ax}(\mathrm{t}) + \mathbf{B}^{+}\mathbf{v}(\mathrm{t}) \tag{4.4.34}$$

$$\mathbf{y}(\mathrm{t}) = \mathbf{C}^{*}\mathbf{x}(\mathrm{t}) \tag{4.4.35}$$

When the process is described using the input-delay form it can be seen that

$$\mathbf{v}(\mathrm{t}) = \begin{pmatrix} q^{-2} & 0 \\ 0 & q^{-3} \\ q^{-1} & 0 \\ 0 & q^{-1.5} \end{pmatrix} \mathbf{u}(t)$$

$$= D_{\mathrm{u}}(\mathrm{q}^{-1})\mathbf{u}(\mathrm{t}) \tag{4.4.36}$$

and

$$C^{*} = D_{y}(1) \tag{4.4.37}$$

and when the process is described in the output-delayed form

$$\mathbf{v}(\mathrm{t}) = \begin{pmatrix} \mathrm{q}^{-2} & 0 \\ 0 & \mathrm{q}^{-3} \\ \mathrm{q}^{-1} & 0 \\ 0 & \mathrm{q}^{-1.5} \end{pmatrix}_{\mathrm{q}=1} \mathbf{u}(\mathrm{t})$$

$$= \mathbf{D}_{u}(1)\mathbf{u}(\mathrm{t}) \tag{4.4.38}$$

and

$$C^{*} = D_{y}(q^{-1})$$
$$= \begin{pmatrix} q^{-2} & q^{-3} & 0 & 0 \\ 0 & 0 & q^{-1} & q^{-1.5} \end{pmatrix} \tag{4.4.39}$$

4.5 Analysis of Time-Delayed State-Space Descriptions

There are well established methods for assessing the stability, controllability and observability of state-space descriptions of delay-free processes such methods are described by numerous authors, for example, Kwakernaak and Sivan (1972), Ogata, (1989) and Middleton and Goodwin (1990). In this section it will be shown that the properties of time-delayed state-space descriptions can be found by considering the properties of an equivalent delay-free state-space description.

Definition 4.1: The time-delayed state-space description

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + B^*(q^{-1})\mathbf{u}(t) \tag{4.5.1}$$

$$\mathbf{y}(t) = C^*(q^{-1})\mathbf{x}(t) \tag{4.5.2}$$

has the *auxiliary* or equivalent delay-free state-space description

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + B^*(1)\mathbf{u}(t) \tag{4.5.3}$$

$$\mathbf{y}(t) = C^*(1)\mathbf{x}(t) \tag{4.5.4}$$

Remark: It can be easily seen that the state-space description, Equations (4.5.3)-(4.5.4) contains no time-delays. This state-space description is clearly very closely related to the time-delayed process, Equations (4.5.1)-(4.5.2).

Lemma 4.4: *The open-loop poles of this auxiliary state-space description, Equations (4.5.3)-(4.5.4) exactly coincide with the poles of the time-delayed process.*

Proof: The poles of the open-loop process can be found by finding the eigenvalues of the state-transition matrix **A**. Clearly the state-transition matrix for both the time-delayed and the auxiliary state-space descriptions are identical. Hence the eigenvalues must also be identical.

Lemma 4.5: *The time-delayed process, Equations (4.5.1)-(4.5.2) is completely observable if an auxiliary delay-free state-spaces description of the process is completely observable.*

Proof: The time-delayed process can be written in either an input-delayed or an output-delayed form. By choosing to write the process in an input-delay form it can be seen immediately that the observability matrix of the time-delayed process

$$Q = \begin{bmatrix} \mathbf{C}^*(1) \\ \mathbf{C}^*(1)\mathbf{A} \\ \vdots \\ \mathbf{C}^*(1)\mathbf{A}^{n-1} \end{bmatrix} \tag{4.5.5}$$

is exactly the same as the observability matrix of the auxiliary state-space description.

Lemma 4.6: *The time-delayed process, Equations (4.5.1)-(4.5.2) is completely controllable if an auxiliary delay-free state-spaces description of the process is completely controllable.*

Proof: The time-delayed process can be written in either a input-delayed or an output-delayed form. By choosing to write the process in an output-delay form it can be seen immediately that the controllability matrix of the time-delayed process

$$P = \begin{bmatrix} \mathbf{B}^*(1) & \mathbf{B}^*(1)\mathbf{A} & \cdots & \mathbf{B}^*(1)\mathbf{A}^{n-1} \end{bmatrix} \tag{4.5.6}$$

is exactly the same as the controllability matrix of the auxiliary state-space description.

4.6 Review

In this chapter a method of obtaining state-space descriptions from transfer functions has been presented. This method has three advantages:

The first advantage is that the method naturally generates state-space descriptions that are block diagonal. Each of the blocks in the state-space descriptions can easily be identified with the individual elements of the transfer function.

The second advantage is that the method does not rely on numerical methods apart from determining a partial fraction representation of each element. Hence if the poles and zeros of the process are known exactly, the state-space description will be exact.

The third advantage is that the method easily extends to time-delayed processes. When multivariable time-delayed transfer functions are transformed, the engineer has a number of choices as to where the time-delays will appear as input delays or as output delays or some combination of the two.

5 DISCRETE-TIME NON-MINIMAL STATE-SPACE DESCRIPTIONS

5.1 INTRODUCTION

This chapter introduces the Non-Minimal State-Space (NMSS) description. The NMSS description is a discrete-time state-space description that is extremely useful for studying time-delayed processes. Two forms of NMSS description are described in this chapter. The first form, which has previously been applied to control of time-delayed processes, includes all the delayed inputs and outputs in the state vector. The second form is developed using the results of Chapter 3 for the discrete-time version of the general time-delayed process description. This second form is a discrete-time analogy of the time-delayed state-space descriptions developed in Chapter 4.

5.2 Non-Minimal State Space (NMSS) Process Descriptions

In this section a Non-Minimal State-Space (NMSS) description will be constructed that describes a multivariable time-delayed process. The NMSS description is a discrete-time state-space description that, while intuitively simple, has not received very much attention in the literature. Hesketh (1981, 1982) developed a self-tuning controller based on an NMSS description. Chotai and Young (1988), showed that NMSS descriptions could be used to implement a discrete-time form of Smith predictor. Crawford and Austin (1988) used a controller based on an NMSS description to control an evaporator. Young Behzadi Wang and Chotai (1987) Wang and Young (1988a, 1988b), and Young Behzadi and Chotai (1988) consider an extension of NMSS descriptions called Proportional Integral Plus (PIP) in which by using the NMSS description, the control action implicitly includes integral action.

In this chapter it is assumed that the processes considered are discrete-time processes. Hence it will be assumed, in this chapter, that the backward-shift operator is defined for only discrete time-steps, that is:

$$q^{-k}x(x) = x(t-k) \quad k \in I^{+}, \text{ the positive integers,}$$

The NMSS description can be constructed by explicitly using the definition of the backward shift operator. Equation (2.2.3) is a general input-output model of a time-delayed multivariable process, such as would be obtained from modelling the process using a model identification technique such as recursive least-squares:

$$\boldsymbol{A}(q^{-1})\mathbf{y}(t) = \boldsymbol{B}(q^{-1})\mathbf{u}(t) + \boldsymbol{E}(q^{-1})\mathbf{w}(t) \qquad (2.2.3)$$

where $y \in \mathbf{R}^{r\times1} . u \in \mathbf{R}^{q_1\times1} . w \in \mathbf{R}^{q_2\times1}$

This equation consists of three parts: The autoregressive component,

$$A(q^{-1}) = \begin{bmatrix} a_{11}(q^{-1}) & a_{12}(q^{-1}) & \cdots & a_{1r}(q^{-1}) \\ a_{21}(q^{-1}) & a_{22}(q^{-1}) & \cdots & a_{2r}(q^{-1}) \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}(q^{-1}) & a_{r2}(q^{-1}) & \cdots & a_{rr}(q^{-1}) \end{bmatrix} \tag{5.2.1}$$

and two moving average components,

$$\boldsymbol{B}(q^{-1}) = \begin{pmatrix} q^{-d_{11}}b_{11}(q^{-1}) & q^{-d_{12}}b_{12}(q^{-1}) & \ldots & q^{-d_{1q1}}b_{1q1}(q^{-1}) \\ q^{-d_{21}}b_{21}(q^{-1}) & q^{-d_{22}}b_{22}(q^{-1}) & \ldots & q^{-d_{2q1}}b_{2q1}(q^{-1}) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ q^{-d_{r1}}b_{r1}(q^{-1}) & q^{-d_{r2}}b_{r2}(q^{-1}) & \ldots & q^{-d_{rq1}}b_{rq1}(q^{-1}) \end{pmatrix} \tag{5.2.2}$$

and

$$\boldsymbol{E}(q^{-1}) = \begin{pmatrix} q^{-k_{11}}e_{11}(q^{-1}) & q^{-k_{12}}e_{12}(q^{-1}) & \ldots & q^{-k_{1q1}}e_{1q2}(q^{-1}) \\ q^{-k_{21}}e_{21}(q^{-1}) & q^{-k_{22}}e_{22}(q^{-1}) & \ldots & q^{-k_{2q1}}e_{2q2}(q^{-1}) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ q^{-k_{r1}}e_{ri}(q^{-1}) & q^{-k_{r2}}e_{r2}(q^{-1}) & \ldots & q^{-k_{rq1}}e_{rq2}(q^{-1}) \end{pmatrix} \tag{5.2.3}$$

where

$$a_{ij}(q^{-1}) = \delta_{ij} + \sum_{k=1}^{n_{aij}} a_{ijk}q^{-k}, \tag{5.2.4}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if i=j} \\ 0 & \text{otherwise} \end{cases} \tag{5.2.5}$$

and the elements $b_{ij}(q^{-1})$,and $e_{ij}(q^{-1})$ are of the form

$$x_{ij}(q^{-1}) = \sum_{k=1}^{n_{x_{ij}}} x_{ijk}q^{-k}. \tag{5.2.6}$$

where $n_{x_{ij}}$ is the order of polynomial x_{ij}. The d_{ij} and the k_{ij} are the pure delay elements associated with each element of $B(q^{-1})$ and $E(q^{-1})$ respectively. That is, the elements of the matrices $A(q^{-1})$, $B^*(q^{-1})$ and $E^*(q^{-1})$ are polynomials in the backward shift operator, q^{-1}.

The $A(q^{-1})$ polynomial matrix can be written as

$$A(q^{-1}) = \mathbf{I} + \sum_{k=1}^{n_a} \mathbf{A}_k q^{-k} \tag{5.2.7}$$

where

$$n_a = \max_{i\,j} \; n_{a_{ij}} \tag{5.2.8}$$

the $B(q^{-1})$ polynomial matrix can be written as

$$B(q^{-1}) = \sum_{k=1}^{n_b} \mathbf{B}^*{}_k q^{-k} \tag{5.2.9}$$

and similarly the $E(q^{-1})$ polynomial matrix can be written as

$$E(q^{-1}) = \sum_{k=1}^{n_c} \mathbf{E}^*{}_k q^{-k} \tag{5.2.10}$$

where the $\mathbf{A}_i$, $\mathbf{B}_j$ and $\mathbf{E}_k$ are constant matrices (identified from the coefficients of the a_{ij}, b_{ij} and e_{ij} polynomials),

$$n_b = \max_{ij}(d_{ij} + n_{b_{ij}}) \qquad (5.2.11)$$

and

$$n_e = \max_{ij}(k_{ij} + n_{e_{ij}}) \qquad (5.2.12)$$

It can be seen by substituting (5.2.7) (5.2.9) and (5.2.10) into (2.2.3) that

$$\mathbf{y}(t) = -\left(\sum_{i=1}^{n_a} \mathbf{A}_i q^{-i}\right)\mathbf{y}(t) + \left(\sum_{j=1}^{n_b} \mathbf{B}_j q^{-j}\right)\mathbf{u}(t) + \left(\sum_{k=1}^{n_e} \mathbf{E}_k q^{-k}\right)\mathbf{w}(t) \qquad (5.2.13)$$

which can be easily rewritten using the definition of the backward shift operator as:

$$\mathbf{y}(t) = -\left(\sum_{i=1}^{n_a} \mathbf{A}_i \mathbf{y}(t-i)\right) + \left(\sum_{j=1}^{n_b} \mathbf{B}_j \mathbf{u}(t-j)\right) + \left(\sum_{k=1}^{n_e} \mathbf{E}_k \mathbf{w}(t-k)\right) \qquad (5.2.14)$$

The NMSS description can be obtained by choosing

$$\mathbf{x}(t)^T = [\mathbf{y}(t)^T, \mathbf{y}(t-1)^T, \ldots, \mathbf{y}(t-n+1)^T, \mathbf{u}(t-1)^T, \mathbf{u}(t-2)^T, \ldots, \mathbf{u}(t-n_b+1)^T,$$
$$\mathbf{w}(t-1)^T, \mathbf{w}(t-2)^T, \ldots, \mathbf{w}(t-n_e+1)^T] \qquad (5.2.15)$$

since the process description can then be written as

$$\mathbf{y}(t) = (-\mathbf{A}_1 \; -\mathbf{A}_2 \ldots -\mathbf{A}_n \;\; \mathbf{B}_2\, \mathbf{B}_3 \ldots \mathbf{B}_{n_b} \;\; \mathbf{E}_2\, \mathbf{E}_3 \ldots \mathbf{E}_{n_e}))\mathbf{x}(t-1)$$
$$+ \mathbf{B}_1 \mathbf{u}(t-1) + \mathbf{E}_1 \mathbf{d}(t-1) \qquad (5.2.16)$$

Hence the process can be written in NMSS form as:

$$\mathbf{x}^{*}(t) = \Phi^{*}\mathbf{x}^{*}(t\text{-}1) + \Gamma^{*}\mathbf{u}(t\text{-}1) + \Delta^{*}\mathbf{w}(t\text{-}1) \qquad (5.2.17)$$

and

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}^{*}(t) \qquad (5.2.18)$$

where

$$\Phi^{*} = \begin{bmatrix} \Phi_{11} & \Phi^{*}_{12} & \Phi^{*}_{13} \\ \Phi^{*}_{21} & \Phi^{*}_{22} & \Phi^{*}_{23} \\ \Phi^{*}_{31} & \Phi^{*}_{32} & \Phi^{*}_{33} \end{bmatrix} \qquad (5.2.19)$$

and

$$\Phi_{11} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \cdots & \mathbf{A}_n \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{0} \end{bmatrix} \qquad (5.2.20)$$

$$\Phi^{*}_{12} = \begin{pmatrix} \mathbf{B}_2 & \mathbf{B}_3 & \ldots & \mathbf{B}_{n_b} \\ 0 & 0 & \ldots & 0 \\ . & . & & . \\ 0 & 0 & \ldots & 0 \end{pmatrix} \qquad (5.2.21)$$

$$\Phi^{*}_{13} = \begin{pmatrix} \mathbf{E}_2 & \mathbf{E}_3 & \ldots & \mathbf{E}_{n_e} \\ 0 & 0 & \ldots & 0 \\ . & . & & . \\ 0 & 0 & \ldots & 0 \end{pmatrix} \qquad (5.2.22)$$

$$\Phi^{*}_{22} = \begin{pmatrix} 0\,0\,0 \ldots 0\,0 \\ \mathrm{I}\,0\,0 \ldots 0\,0 \\ 0\,\mathrm{I}\,0 \ldots 0\,0 \\ .\;.\;.\;\ldots\;.\;. \\ .\;.\;.\quad\;\;.\;. \\ 0\,0\,0 \ldots \mathrm{I}\,0 \end{pmatrix} \qquad (5.2.23)$$

$$\Phi^{*}_{33} = \begin{pmatrix} 0\,0\,0 \ldots 0\,0 \\ \mathrm{I}\,0\,0 \ldots 0\,0 \\ 0\,\mathrm{I}\,0 \ldots 0\,0 \\ .\;.\;.\;\ldots\;.\;. \\ .\;.\;.\quad\;\;.\;. \\ 0\,0\,0 \ldots \mathrm{I}\,0 \end{pmatrix} \qquad (5.2.24)$$

$\Phi^*_{21},\Phi^*_{23},\Phi^*_{31},\Phi^*_{32}$ are zero matrices of appropriate dimensions.

$$\Gamma^* \; = ((\mathbf{B}_1)^T \mathbf{0\ 0\ 0\ I\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}\,)^T \qquad (5.2.25)$$

$$\Delta^* \; = ((\mathbf{E}_1)^T \mathbf{0\ 0\ 0\ 0\ 0\ 0\ 0\ I\ 0\ 0\ 0\ 0\ 0})^T \qquad (5.2.26)$$

and

$$\mathbf{C} \; = (\mathbf{I\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 0\ 0}) \qquad (5.2.27)$$

This state-space description has dimension:

$$r\ n_a+q_1(n_b-1)+q_{2(}(n_e-1)\times r\ n_a+q_1 n_b+q_2 n_e \qquad (5.2.28)$$

The NMSS description, Equations (5.2.17) and (5.2.18), has an advantage over many other state-space descriptions: no observer or filter is required in order to obtain the state-vector since the state-vector of an NMSS description consists of the input-output record of the plant.

The NMSS description also has the advantage that when it is used to describe time-delayed processes it includes all the delayed inputs and the delayed outputs in the state-vector. Chotai and Young (1988) showed that it acts as an implicit Smith predictor since it makes the closed-loop appear, to the controller, to be a delay-free process. The implicit inclusion of the delays in the state-space description has also been used in the optimal control of an evaporator (Crawford and Austin, 1988) as will be seen in Chapter 9.

However this description does have some disadvantages. One disadvantage is that the transition matrix, Φ, for any NMSS description is singular. Hence computing optimal state-feedback controllers is more difficult than is the case with some other state-space descriptions since the solution of the discrete-time Riccati equation by the eigenvector method of Potter [Pappas, Laub and Sandell, 1980] cannot be used. However, it is still possible to solve the Riccati equation by a Schur vector method (Laub, 1979), a generalised eigenvector method, Pappas, Laub and Sandell (1980), or by approximating the solution by iterative means (Goodwin and Sin, 1984, p513-515).

Another disadvantage of the NMSS description is that the dimension of the state-transition matrix for an NMSS description is large. The large dimensions of the NMSS description detract from the attractiveness of applying the NMSS description to real problems due to the consequentially large memory and computational time requirements. For example, the model of the evaporator used in Chapter 9 is a 2 input, 2 output, third-order ARMAX model with a maximum delay of 5 time-units. Hence the NMSS description obtained for this process can be seen, using Equation (5.2.28) to have a state-transition matrix of dimension 20 × 20.

In the following section a method, based on the results of Chapter 3, will provide a way in which the dimensions of the NMSS description can be reduced.

5.3 Time-Delayed NMSS Descriptions

In the input-output representation presented in Equation (5.2.1) the $\boldsymbol{B}(\mathrm{q}^{-1})$ and $\boldsymbol{E}(\mathrm{q}^{-1})$ polynomial matrices include the time-delays of the process. This means that the state-space description obtained in Section 5.2 include all the delays implicitly in the model. In this section the results of Chapter 3 are used to obtain a NMSS description of a time-delayed process that does not include the pure time-delays in the state-space description but rather places all the delays in a matrix acting on the inputs and hence the delays are not involved in the dynamics.

The $\boldsymbol{B}(\mathrm{q}^{-1})$ matrix polynomial, written as

$$\boldsymbol{B}(\mathrm{q}^{-1}) = \begin{pmatrix} \mathrm{q}^{-\mathrm{d}_{11}}b_{11}(\mathrm{q}^{-1}) & \mathrm{q}^{-\mathrm{d}_{12}}b_{12}(\mathrm{q}^{-1}) & \ldots & \mathrm{q}^{-\mathrm{d}_{1q1}}b_{1q1}(\mathrm{q}^{-1}) \\ \mathrm{q}^{-\mathrm{d}_{21}}b_{21}(\mathrm{q}^{-1}) & \mathrm{q}^{-\mathrm{d}_{22}}b_{22}(\mathrm{q}^{-1}) & \ldots & \mathrm{q}^{-\mathrm{d}_{2q1}}b_{2q1}(\mathrm{q}^{-1}) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \mathrm{q}^{-\mathrm{d}_{r1}}b_{r1}(\mathrm{q}^{-1}) & \mathrm{q}^{-\mathrm{d}_{r2}}b_{r2}(\mathrm{q}^{-1}) & \ldots & \mathrm{q}^{-\mathrm{d}_{rq1}}b_{rq1}(\mathrm{q}^{-1}) \end{pmatrix} \tag{5.2.2}$$

can be written, using Theorem 3.2, as

$$\boldsymbol{B}(\mathrm{q}^{-1}) = \boldsymbol{B}^{+}(\mathrm{q}^{-1})\boldsymbol{D}_{u}(\mathrm{q}^{-1}) \tag{5.3.1}$$

where

$$\boldsymbol{B}^{+}(\mathrm{q}^{-1}) = \begin{pmatrix} b_{11}\ b_{12}\ ..\ b_{1q1}\ 0\ \ 0\ \ 0...... & \\ 0\ \ \ 0\ \ \ \ 0\ ..\ \ 0\ b_{21}\ b_{22} & \\ & b_{r1}\ ....b_{rq1} \end{pmatrix} \tag{5.3.2}$$

and

$$D_u(q^{-1}) = \begin{pmatrix} q^{-d_{11}} & 0 & 0 & \cdots & 0 \\ 0 & q^{-d_{12}} & 0 & \cdots & 0 \\ . & . & & & . \\ 0 & 0 & 0 & \cdots & q^{-d_{1q1}} \\ q^{-d_{21}} & 0 & 0 & \cdots & 0 \\ 0 & q^{-d_{22}} & 0 & \cdots & 0 \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ 0 & 0 & 0 & \cdots & q^{-d_{rq1}} \end{pmatrix} \tag{5.3.3}$$

and similarly

$$\boldsymbol{E}(q^{-1}) = \boldsymbol{E}^{+}(q^{-1})\boldsymbol{D}_o(q^{-1}) \tag{5.3.4}$$

where

$$\mathbf{E}^{+}(q^{-1}) = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1q_2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & e_{21} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & & & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & e_{r(q_2-1)} & e_{rq_2} \end{bmatrix} \tag{5.3.5}$$

and

$$D_o(q^{-1}) = \begin{pmatrix} q^{-d_{11}} & 0 & 0 & \cdots & 0 \\ 0 & q^{-d_{12}} & 0 & \cdots & 0 \\ . & . & & & . \\ 0 & 0 & 0 & \cdots & q^{-d_{1q1}} \\ q^{-d_{21}} & 0 & 0 & \cdots & 0 \\ 0 & q^{-d_{22}} & 0 & \cdots & 0 \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ 0 & 0 & 0 & \cdots & q^{-d_{rq2}} \end{pmatrix} \tag{5.3.6}$$

It is possible to write the system description as

$$\mathbf{x}^{+}(t) = \Phi^{+}\mathbf{x}^{+}(t-1) + \Gamma^{+}D_u(q^{-1})\mathbf{u}(t-1) + \Delta^{+}D_O(q^{-1})\mathbf{w}(t-1) \tag{5.3.7}$$

where

$$\Phi^* = \begin{bmatrix} \Phi_{11} & \Phi^+_{12} & \Phi^+_{13} \\ \Phi^+_{21} & \Phi^+_{22} & \Phi^+_{23} \\ \Phi^+_{31} & \Phi^+_{32} & \Phi^+_{33} \end{bmatrix} \tag{5.3.8}$$

$$\Phi_{11} = \begin{pmatrix} A_1 & A_2 & A_3. & \ldots & A_{n-1} & A_{na} \\ I & 0 & 0 & \ldots & 0 & 0 \\ 0 & I & 0 & \ldots & 0 & 0 \\ . & . & . & & . & . \\ 0 & 0 & 0 & \ldots & I & 0 \end{pmatrix} \tag{5.3.9}$$

$$\Phi^+_{12} = \begin{pmatrix} B_2{}^+ & B_3{}^+ & \ldots & B_{nb}{}^+ \\ 0 & 0 & \ldots & 0 \\ . & . & & . \\ 0 & 0 & \ldots & 0 \end{pmatrix} \tag{5.3.10}$$

$$\Phi^+_{13} = \begin{pmatrix} E_2{}^+ & E_3{}^+ & \ldots & E_{nc}{}^+ \\ 0 & 0 & \ldots & 0 \\ . & . & & . \\ 0 & 0 & \ldots & 0 \end{pmatrix} \tag{5.3.11}$$

$$\Phi^+_{22} = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ I & 0 & 0 & \ldots & 0 & 0 \\ 0 & I & 0 & \ldots & 0 & 0 \\ . & . & . & & . & . \\ 0 & 0 & 0 & \ldots & I & 0 \end{pmatrix} \tag{5.3.12}$$

$$\Phi^+_{33} = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ I & 0 & 0 & \ldots & 0 & 0 \\ 0 & I & 0 & \ldots & 0 & 0 \\ . & . & . & & . & . \\ 0 & 0 & 0 & \ldots & I & 0 \end{pmatrix} \tag{5.3.13}$$

$\Phi^+_{21.}\Phi^+_{23},\Phi^+_{31},\Phi^+_{32}$ are zero matrices of appropriate dimensions.

$$\Gamma^+ = ((B_1{}^+)^T 0\ 0\ 0\ I\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T \tag{5.3.14}$$

$$\Delta^{+} = ((\mathbf{E}_1{}^{+})^{\mathrm{T}}\mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{I}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0})^{\mathrm{T}} \tag{5.3.15}$$

and

$$\mathrm{C}^{+} = (\mathbf{I}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}) \tag{5.3.16}$$

The row dimension of Φ^{+} is:

$$(r\times n_a+r\times q_1(n_b-1)+r\times q_2(n_e-1))\times(r\times n_a+r\times q_1(n_b{}^{+}-1)+rq_2(n_e{}^{+}-1)) \tag{5.3.17}$$

Hence the difference in matrix dimensions, ΔN, when there are no disturbances between the two NMSS descriptions is

$$\begin{aligned}\Delta N &= r\ n_a+q_1(n_b{}^{+}-1)-(r\ n_a+r\times q_1(n_b-1))\\ &= q_1(n_b{}^{+}-1)-r\times q_1(n_b-1)\end{aligned} \tag{5.3.18}$$

Defining d as:

$$d = \max_{i \text{ and } j} d_{ij} \tag{5.3.19}$$

it can be seen that

$$n_b = n_b{}^{+}+d \tag{5.3.20}$$

and that

$$\Delta N = q_1(rd + (n_b{}^{+}-1)(1-r)) \tag{5.3.21}$$

From Equation (5.3.22) it can be seen that ΔN will be positive if

$$d > (n_b^+ - 1)(1-r)/r \qquad (5.3.22)$$

That is, when the maximum delay is sufficiently large to satisfy Inequality (5.3.22), the dimension of the state-transition matrix, Φ^+, Equation (5.3.8) is smaller than that of Φ^* , Equation (5.2.19), in which the delays are included in the state-transition matrix.

If the process contains disturbance inputs, as well as control inputs, the analysis of the comparative dimensions is more complicated since there are three possible options that can be considered:

- all delays are included in the state-transition matrix (the method presented in Section 5.2);
- only the control-input delays are in the state-transition matrix;
- only the disturbance-input delays are in the state-transition matrix and no delays in the state-transition matrix (the method presented in Section 5.3).

Each of these options could be considered in a particular application. Although for large delays it is clear that the method of Section 5.3 will result in smaller state-space descriptions, it is possible that one of the other options, listed above, could result in a smaller state-transition matrix if the delays are of the same magnitude as the model order.

In the example of the evaporator used in Chapter 9, if the process was described using this NMSS description, the state-transition matrix, Φ^+, would have been 14 by 14.

However, the NMSS description Equations 5.3.8 has the disadvantage that there are time-delays operators, $D_u(q^{-1})$ and $D_d(q^{-1})$ in the description and to use this description a compensator for the time-delays must be used.

5.4 REVIEW

In this chapter Non Minimal State-Space (NMSS) descriptions have been investigated. NMSS descriptions have some distinct advantages for the description and control of time-delayed processes. However the potentially large size of the state transition matrices would appear to make NMSS descriptions less attractive. A modification of the NMSS description, using the results of Chapter 3 suggests one way in which a limitation of the NMSS description can to some extent be overcome.

PART C

A GENERALIZED SMITH PREDICTOR

6 MULTIVARIABLE SMITH PREDICTORS

6.1 INTRODUCTION

The Smith predictor is perhaps the best known method of controlling a time-delayed process. It has commonly been applied to continuous-time single-input single-output systems of the form $g^*(s) = e^{-\tau s}g^+(s)$. However, the application of Smith predictor methods to multivariable systems has received much less attention.

In Section 6.2 a derivation of the single-input single-output Smith predictor is presented which will be used as a motivation for several multivariable extensions of the Smith predictor that depend on the structure of the delay matrices in the transfer function description of the process.

In Section 6.3, the main result of this chapter, a multivariable extension of the Smith predictor to quite general time-delayed multivariable processes is presented.

In Section 6.4 the Internal Model Control (IMC) structure is introduced. The IMC methodology is applied to time-delayed processes and the relationships between the controllers designed this way and those designed using the generalised Smith predictor are explored.

Throughout this chapter it will be assumed that the process is modelled exactly and is noise-free, as was the case with the original Smith predictor.

6.2 SINGLE-INPUT SINGLE-OUTPUT SMITH PREDICTORS

In this section the single-input single-output Smith predictor is developed using Smith's Method. In subsequent sections the same methodology will be used for multivariable processes.

Smith equated the transfer functions obtained from two closed-loop processes: one being the time-delayed process with a controller of unknown design, as shown in Figure 6.2.1 and the other being the desired closed-loop, Figure 6.2.2, in which the delays lie outside the feedback loop and hence act only on the outputs of the system.

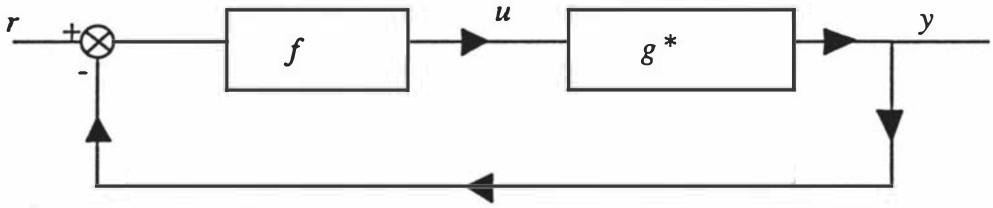

Figure 6.2.1

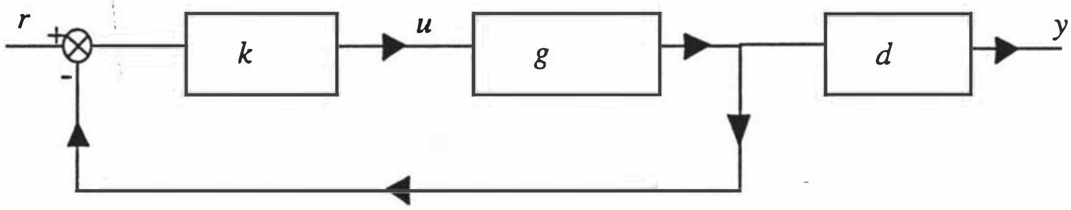

Figure 6.2.2

The closed-loop transfer function of the system portrayed in Figure 6.2.1 is

$$\frac{g^*f}{(1+g^*f)} \tag{6.2.1}$$

The transfer function of the desired closed-loop system, Figure 6.2.2, is

$$\frac{dgk}{(1+gk)} \tag{6.2.2}$$

Equating the two transfer functions:

$$\frac{g^*f}{(1+g^*f)} = \mathrm{d}\frac{gk}{(1+gk)} \tag{6.2.3}$$

collecting the terms containing g^* and f and putting

$$h \;= g^*\text{-}dg \tag{6.2.4}$$

it can be seen that:

$$g^*f \;= dg\frac{k}{(1+hk)} \tag{6.2.5}$$

This suggests that if the plant transfer function, g^*, and the model, dg are equal, $g^*(\gamma)= d(\gamma)g(\gamma)$ then

$$f(\gamma) \;= \frac{k}{(1+kh)} \tag{6.2.6}$$

This is the traditional form of the Smith predictor. The block-diagram representation of the controller transfer function, $f(\gamma)$, is given in Figure 6.2.3.

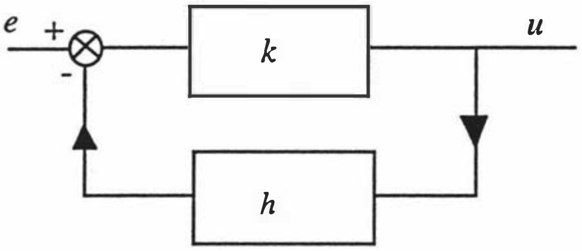

Figure 6.2.3

Clearly in the single-input single-output case it does not matter whether the delay is associated with the input, the output or is a plant delay since scalar multiplication is commutative. However, in multivariable processes the orders in which the delays appear in the process description become important since matrix multiplication is not commutative. It is the lack of this commutatively property that has so far limited the extensions of Smith's method to multivariable processes.

6.3 MULTIVARIABLE SMITH PREDICTORS

One way of extending the Smith predictor to multivariable systems would be to consider the multivariable extensions of Smith's method, as is shown in Section 6.3.1

6.3.1 A MULTIVARIABLE EXTENSION OF SMITH'S METHOD

Smith's method, which was used in Section 6.2 to derive a Smith predictor for a time-delayed single-input single-output process can be extended to multivariable processes by considering the multivariable extensions of Figures 6.2.1 and 6.2.2. These are shown in Figures 6.3.1 and 6.3.2. In Figure 6.3.2 it has been assumed that the transfer function matrix, $G^*(\gamma)$ can be written as the product of the matrices $D(\gamma)$ and $G(\gamma)$, where $D(\gamma)$ is a square matrix. In Section 4.3.1 it was shown that such a decomposition does exist for time-delayed processes that can be described in the form of Equations (3.3.5)-(3.3.8) .

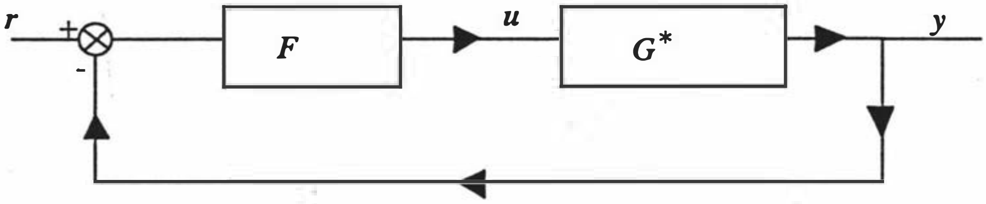

Figure 6.3.1

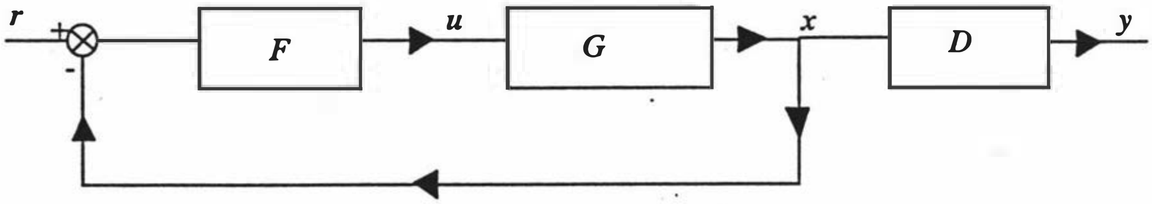

Figure 6.3.2

In these Figures, $G(\gamma)$ is the delay-free plant, $G^*(\gamma)$ is the delayed process, $D(\gamma)$ is a multivariable delay-operator and $K(\gamma)$ is the controller designed to close the delay-free loop. By equating the transfer functions obtained from these block diagrams it is readily seen that:

$$D(\mathrm{I}+GK)^{-1}GK \quad = (\mathrm{I}+G^*F)^{-1}G^*F \tag{6.3.1}$$

$$DGK(\mathrm{I}+GK)^{-1} \quad = (\mathrm{I}+G^*F)^{-1}G^*F \tag{6.3.2}$$

Pre-multiplying by $(\mathbf{I}+\mathbf{G}^*\mathbf{F})$ and post-multiplying by $(\mathbf{I}+\mathbf{GK})$:

$$\begin{aligned}(\mathbf{I}+\mathbf{G}^*\mathbf{F})\mathbf{DGK} &= \mathbf{G}^*\mathbf{F}(\mathbf{I}+\mathbf{GK})\\ \mathbf{DGK}+\mathbf{G}^*\mathbf{FDGK} &= \mathbf{G}^*\mathbf{F}(\mathbf{I}+\mathbf{GK})\\ \mathbf{DGK} &= \mathbf{G}^*\mathbf{F}(\mathbf{I}+\mathbf{GK}-\mathbf{DGK})\end{aligned} \tag{6.3.3}$$

or

$$G^*F \quad = DGK(\mathrm{I}+HK)^{-1} \tag{6.3.4}$$

which can be rewritten using the matrix identity $(\mathbf{I}+\mathbf{AB})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{I}+\mathbf{BA})^{-1}$ as

$$G^*F \quad = DG(\mathrm{I}+HK)^{-1}K \tag{6.3.5}$$

where

$$H(\gamma) \quad = G^+(\gamma) - G^*(\gamma). \tag{6.3.6}$$

Hence the multivariable Smith predictor obtained is of the same form as the single-input single-output case if $G^*(\gamma) = D(\gamma)G^+(\gamma)$ and $D(\gamma)$ is a square matrix.

However, in general, $G^*(\gamma)$ is of the form given in Equations (3.3.5)-(3.3.8):

$$\mathbf{y}(\gamma) \;= \boldsymbol{G}^{*}(\gamma)\mathbf{u}(\gamma) = \boldsymbol{D}_1(\gamma)\boldsymbol{G}_{\mathrm{p}}(\gamma)\boldsymbol{D}_2(\gamma)\mathbf{u}(\gamma) \qquad (3.3.5)$$

where $$\boldsymbol{D}_1(\gamma) = \sum_{k=1}^{n_C} C_k E(\gamma,-\gamma_k) \qquad (3.3.6)$$

$$\boldsymbol{D}_2(\gamma) = \sum_{j=1}^{n_B} \mathbf{B}_j E(\gamma,-\alpha_j) \qquad (3.3.7)$$

and

$$\boldsymbol{G}_p(\gamma)=\begin{pmatrix} \dfrac{a_{11}(g)E(\gamma,-\tau_{11})}{b_{11}(\gamma)} & \dfrac{a_{12}(\gamma)E(\gamma,-\tau_{12})}{b_{12}(\gamma)} & \cdots & \dfrac{a_{1n}(\gamma)E(\gamma,-\tau_{1n})}{b_{1n}(\gamma)} \\ \dfrac{a_{21}(\gamma)E(\gamma,-\tau_{21})}{b_{21}(\gamma)} & \dfrac{a_{22}(\gamma)E(\gamma,-\tau_{22})}{b_{22}(\gamma)} & \cdots & \dfrac{a_{2n}(\gamma)E(\gamma,-\tau_{2n})}{b_{2n}(\gamma)} \\ \vdots & \vdots & & \vdots \\ \dfrac{a_{m1}(\gamma)E(\gamma,-\tau_{m1})}{b_{m1}(\gamma)} & \dfrac{a_{m2}(\gamma)E(\gamma,-\tau_{m2})}{b_{m2}(\gamma)} & \cdots & \dfrac{a_{mn}(\gamma)E(\gamma,-\tau_{mn})}{b_{mn}(\gamma)} \end{pmatrix} \qquad (3.3.8)$$

There are several special cases of delay-structures that will be considered. These possibilities arise from the different ways that the delays arise within the process. The possibilities include:

a) all the delays are identical: $\boldsymbol{G}^{*}(\gamma) = E(\gamma,-\tau)\boldsymbol{G}(\gamma)$;

b) the delays arise only from the outputs of the plant: $\alpha_{ij}= 0$ and $\tau_{ij}=0 \;\forall i,j$;

c) the delays arise from either the inputs or the outputs in a simple manner: the delay matrices $\boldsymbol{D}_1(\gamma)$ and $\boldsymbol{D}_2(\gamma)$ are both square;

and d) the input and output delay matrices are non-square and internal plant delays exist.

Each of these cases will be explored in turn.

6.3.2 Multivariable Smith Predictors when all the Delays are Identical

The transfer function matrix describing a process with all the overall delays identical can be written as

$$\mathbf{G}^*(\gamma) = \mathrm{E}(\gamma,-\tau)\mathbf{G}(\gamma) \tag{6.3.8}$$

where $\mathrm{E}(\gamma,-\tau)$ is a scalar operator and τ is the length of the pure delay.

The multivariable version of the derivation of the Smith predictor is, in this case, a simple extension of the single-input single-output case.

6.3.3 Multivariable Smith Predictors when the Delays Act only on the Plant's Outputs

If in Equation (3.3.5) all the internal plant delays were zero and the input delays were also zero so that $D_2(\gamma) = \mathbf{I}$ then $D(\gamma)= D_1(\gamma)$ and the multivariable Smith predictor would be in the same form as that obtained for the single-input single-output case because the process transfer function can be written as:

$$G^*(\gamma) = D(\gamma)G(\gamma) \tag{6.3.9}$$

This allows the straightforward multivariable generalisation of the Smith predictor as noted in Section 6.3.1 and F can be easily identified from Equation (6.3.5) to be

$$F = (\mathbf{I} + HK)^{-1}K \tag{6.3.10}$$

A block diagram representation of $F(\gamma)$ is shown in Figure 6.3.3.

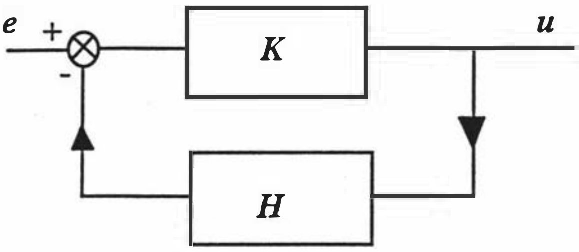

Figure 6.3.3

The Smith predictor, ***F*** is of the same form as the single-input single-output Smith predictor.

It is easy to see that the case presented in Section 6.3.2 is a special case of this result when all the output-delays are identical.

6.3.4 MULTIVARIABLE SMITH PREDICTORS WHEN THE DELAYS ACT EITHER ON THE INPUTS OR THE OUTPUTS IN A SIMPLE MANNER

In this case it is assumed that the input and output delay matrices are both square and there are no internal, or plant, delays. Smith's method, as was shown in Section 6.3.1, requires that the process description be represented as the product of a delay operator, ***D***, and the delay-free dynamics, ***G***. The output delays, represented by a matrix of delay operators, premultiply the delay-free dynamics of the process. However, the input-delays, which post-multiply the delay-free dynamics do not appear in a form in which Smith's method can be applied.

A multivariable Smith predictor could be constructed if the input-delays were of the form:

$$D_2(\gamma) = \mathrm{E}(\gamma,-\tau)\mathbf{I} \qquad (6.3.11)$$

since if the input-delays were of this form the scalar time-delay term, E(γ,−τ) could be passed through the matrix product to appear as though it were an output-delay.

· Alternatively, if D_2 is not of the form of Equation (6.3.11), additional input delays, D_3, could be added so that:

$$D_2D_3 = \tau(\gamma).I \qquad (6.3.12)$$

where $\tau(\gamma)$ is a scalar function containing delay-operators. The result of adding these delays is that the effective controller, F^*, becomes:

$$\begin{aligned} F^* &= D_3(I+KH)^{-1}K \qquad (6.3.13) \\ &=D_3F \end{aligned}$$

as shown in Figure 6.3.4.

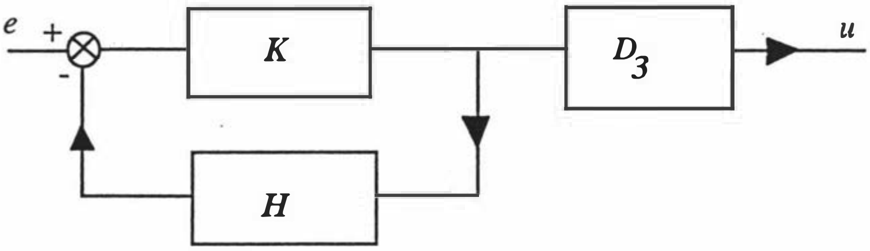

Figure 6.3.4

The closed-loop transfer function can be written as:

$$D(\gamma)G^*(\gamma)(I+EG^*)^{-1} \qquad (6.3.14)$$

where $D(\gamma)= \tau(\gamma)D_1(\gamma)$ where $D_2(\gamma)D_3(\gamma) = \tau(\gamma)$,

Time-delays have been added to multivariable Smith predictor by Jerome and Ray (1986) and Shanmaguthasan and Johnston (1988a, 1988b) in order to optimise response times.

6.3.5 Non-Square Delay Operator : a Generalisation of the Smith Predictor

In the previous sections it was assumed that the delay matrices were square and that there were no internal plant delays. It was found that a Smith predictor could be constructed that maintained a delay-structure only if the delays acted on the outputs or if all the input delays were the same, each represented as $E(\gamma,-\tau)$.

Now consider the more general delay structures allowed by Equations (3.3.5)-(3.3.8). Theorem 3.3 can be used to rewrite Equation (3.3.5) as:

$$\mathbf{y}(\gamma) \quad = D_1(\gamma)D_{p1}(\gamma)D_i(\gamma)G_1{}^+(\gamma)\mathbf{u}(\gamma) \qquad (6.3.15)$$

Smith's method, Section 6.3.1 can then be applied to this process description to obtain a multivariable Smith predictor. However this would, at first appearance, seem to be an impractical solution since the dimensions of the vector quantities, **r** and **x**, are not necessarily the same, as is evident from the block diagram of the 'ideal', or desired, closed loop system drawn in Figure 6.3.2.

This problem, concerning the difference in the dimensions of the vector quantity $\mathbf{x}(\gamma)=G^+\mathbf{u}(\gamma)$ and $\mathbf{y}(\gamma)=D_1(\gamma)D_{p1}(\gamma)D_i(\gamma)G^+(\gamma)\mathbf{u}(\gamma)$, can be solved by adding an extra block diagram element, **E**, to the actual process as shown in Figure 6.3.5, where **E** is of the same dimensions as the delay operator, $D(\gamma)=D_1(\gamma)D_{p1}(\gamma)D_i(\gamma)$.

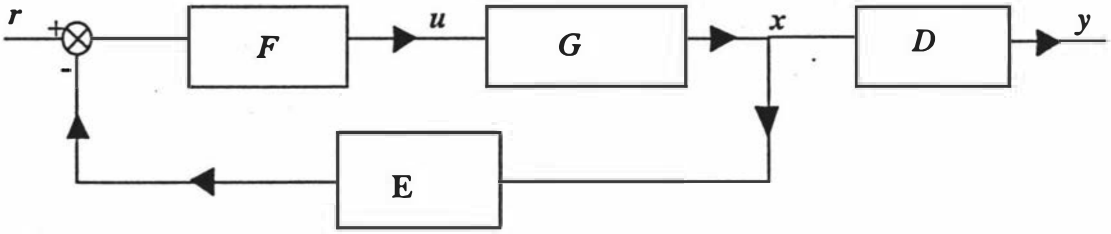

Figure 6.3.5

By the same matrix manipulation process as was performed in Section 6.3.1, the controller/predictor block, $\boldsymbol{F}$, can be determined. Equating the transfer functions from Figures 6.3.5 and 6.3.1:

$$D_1 D_{pl} D_i (\mathrm{I} + G^+ K \mathrm{E})^{-1} G^+ K = (\mathrm{I} + G^* F)^{-1} G^* F \qquad (6.3.16)$$

which can be rewritten using the matrix identity $(\mathbf{I} + \mathbf{AB})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{I} + \mathbf{BA})^{-1}$ as

$$D_1 D_{pl} D_i G^+ K (\mathrm{I} + \mathrm{E} G^+ K)^{-1} = (\mathrm{I} + G^* F)^{-1} G^* F \qquad (6.3.17)$$

Premultiplying by $(\mathrm{I} + G^* F)$ and post-multiplying by $(\mathrm{I} + \mathrm{E} G^+ K)$ this equality can be rewritten as:

$$(\mathrm{I} + G^* F) D_1 D_{pl} D_i G^+ K = G^* F (\mathrm{I} + \mathrm{E} G^+ K) \qquad (6.3.18)$$

Defining

$$H = (\mathrm{E} - D_1 D_p D_i) G^+(\gamma) \qquad (6.3.19)$$

and gathering together the terms containing $G^* F$ we have:

$$G^* F = D_1 D_{pl} D_i G^+ K (\mathrm{I} + H K)^{-1} \qquad (6.3.20)$$

Since by Theorem 3.3 $G^* = D_1 D_{pl} D_i G^+$, Equation (6.3.20) reveals that:

$$\begin{aligned} F &= K (\mathrm{I} + H K)^{-1} \\ &= (\mathrm{I} + K H)^{-1} K \end{aligned} \qquad (6.3.21)$$

The Smith predictor described by Equation (6.3.20) is of the same form as the traditional Smith predictor except that the identity matrix. has been replaced by **E.** This **E** matrix can be free to the designer to choose so long as the delay-free process with the transfer function matrix $G(\gamma) = \mathbf{E}G^{+}(\gamma)$ is controllable and observable. There are a number of possible choices for **E**. These include:

1. $\mathbf{E} = D_1(1)D_p(1)D_u(1)$
2. $\mathbf{E} = D_p(1)D_u(1)$ (in the case where $D_1(\gamma)$ is square)
3. $[\mathbf{E}]_{ij} = \begin{cases} 1 & \text{if } [D_1(1)D_p(1)D_u(1)]_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}$

It is clear that when the delays are only output-delays or all the input-delays are the same, the second and third choices for **E** give the same Smith predictor as was found in Section 6.3.1. The first choice gives the Smith predictor found by Alevisakis and Seborg (1973, 1974) if the input delays are equal. Further research is needed to explore other possible choices of **E**.

EXAMPLE 6.1: The process used in Examples 3.1, 3.2 and 4.1 was used for numerical simulations of the process. The process description obtained in Example 3.2 was

$$G^{*}(s) = \begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} & \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-3s}}{(s+3)} & \dfrac{-e^{-1.5s}}{(s+4)} \end{pmatrix} = \begin{pmatrix} e^{-2s} & e^{-s} & 0 & 0 \\ 0 & 0 & e^{-3s} & e^{-1.5s} \end{pmatrix} \begin{pmatrix} \dfrac{1}{(s+1)} & 0 \\ 0 & \dfrac{3}{s+2} \\ \dfrac{2}{s+3} & 0 \\ 0 & \dfrac{-1}{s+4} \end{pmatrix} \qquad (6.3.22)$$

$$= D_p(s)G^{+}(s) \qquad (6.3.23)$$

In order to use the Smith predictor, a controller for the delay-free transfer function

$$G(s) = \mathbf{E}G^{+}(s) \tag{6.3.24}$$

must be found.

A state-space description of this delay-free process was obtained in Example 4.1:

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax}(t) + \mathbf{Bu}(t) \tag{6.3.25}$$

$$\mathbf{y}^{+}(t) = \mathbf{Cx}(t) \tag{6.3.26}$$

where

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \tag{4.2.9}$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4.2.8}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{pmatrix} \tag{4.2.4}$$

Using this state-space description it is a straightforward process to design an output feedback control law. The closed-loop system was simulated using SIMULINK using a proportional-integral output feedback controller. The first output, y_1 and the reference signals r_1 are plotted in Figure 6.3.6 and the second output $y_2$and r_2 are shown in Figure 6.3.7.

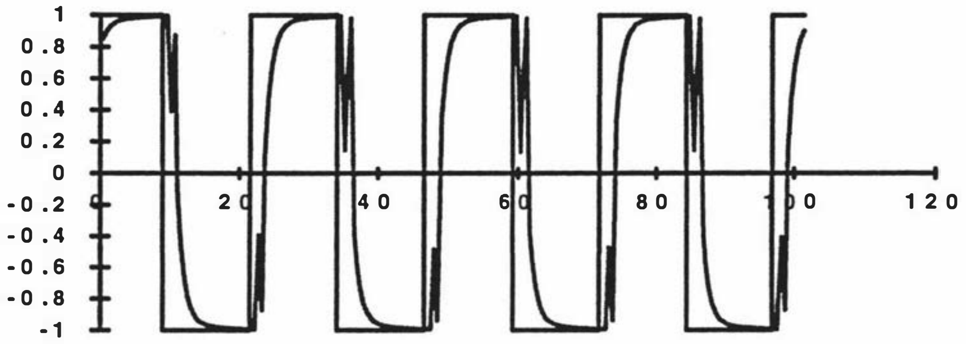

Figure 6.3.6

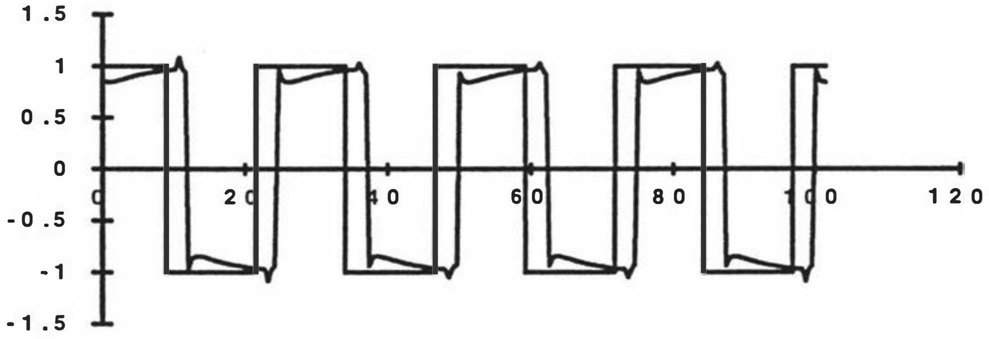

Figure 6.3.7

6.4 Internal Model Control (IMC)

The Internal Model Control (IMC) philosophy is based on the explicit inclusion of a model of the process in the controller. In this way it is possible to consider the effects, on the robustness and stability properties of the closed-loop system, of model mismatch, both in the structural parameters of the process description, such as the time-delays and model orders, and in the coefficients of the process. IMC uses the concept of perfect control, exactly matching and countering the dynamics of the open-loop processes. Clearly this is not always possible without including predictions or adding problematic unstable poles or non-minimum phase zeros to the closed-loop process, but the IMC methodology suggests some methods for the analysis of processes that contain problematic delays, poles or zeros .

In Sections 6.4.1-6.4.3, it will be assumed that the process has the same number of inputs as outputs. In Sections 6.4.4 and 6.4.5 it will be shown that, by using the matrix decompositions found in Section 3.3, it is possible to extend some of the ideas of the IMC structure to non-square processes, and to hence provide an alternative method of obtaining Smith predictors and to provide an extension of IMC to processes in which the number of inputs differs from the number of outputs.

6.4.1 Derivation of the IMC Closed-Loop Process Structure

The Internal Model Control (IMC) closed-loop process structure can be derived by considering Figures 6.4.1-6.4.3. Figure 6.4.1 is a typical traditional block diagram representation of a multivariable process:

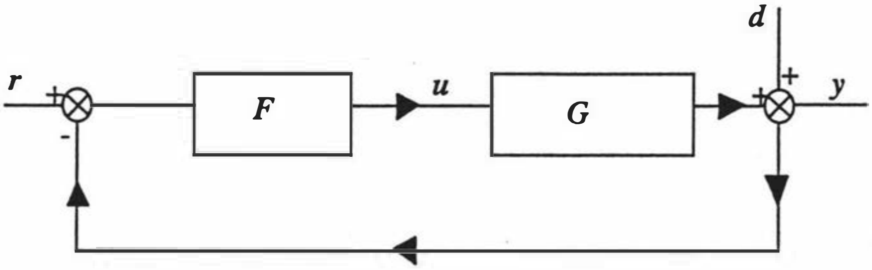

Figure 6.4.1

Figure 6.4.2 is obtained by adding and subtracting the process-model, $\hat{G}$, to the block diagram. This obviously does not change the process transfer function.

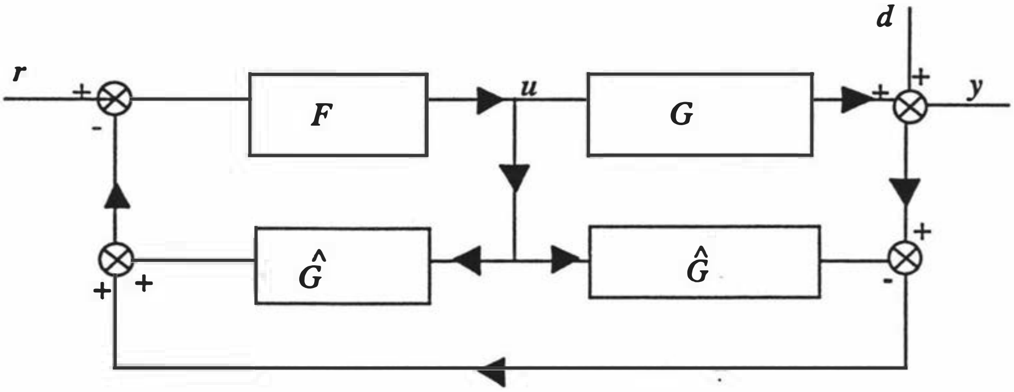

Figure 6.4.2

Figure 6.4.3 is obtained by grouping the inner-loop as one element, *C*, in the block diagram, that is

$$C = (\mathbf{I} + \hat{G} F)^{-1} F \tag{6.4.1}$$

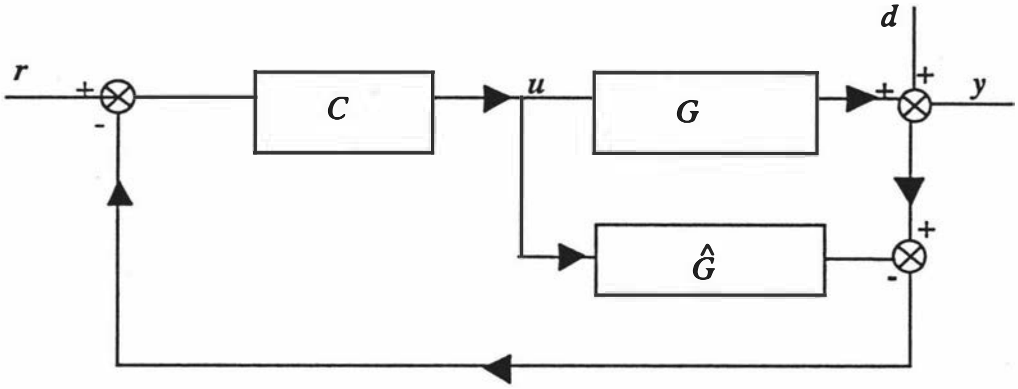

Figure 6.4.3

Figure 6.4.3 represents the IMC structure. The controller C explicitly includes the model, $\hat{G}$,of the process.

Clearly there is a relationship between the IMC controller, C, and the traditional controller, F, shown in Figure 6.4.2. By rearranging Equation (6.4.1) F, the traditional controller, can be found to be:

$$F = (\mathbf{I} - C\hat{G})^{-1}C \qquad (6.4.2)$$

From Figure 6.4.3 it can be seen that the control inputs, **u**, are determined by the control law:

$$\mathbf{u}(\gamma) = (\mathbf{I} + C(G - \hat{G}))^{-1}C(\mathbf{r} - \mathbf{d}) \qquad (6.4.3)$$

Hence the process output, **y**, can be described by:

$$\mathbf{y}(\gamma) = G(\mathbf{I} + C(G - \hat{G}))^{-1}C(\mathbf{r} - \mathbf{d}) + \mathbf{d} \qquad (6.4.4)$$

Using the IMC structure it is possible to design the element C so that the closed-loop process has the desired properties of stability and robustness. Two properties of the IMC structure that have been found to be of use are presented in the next section.

6.4.2 Properties of the IMC Structure

There are two properties of the IMC structure that are commonly used (Morari, 1983, Morari and Zafiriou, 1989) as evidence of the usefulness of the structure:

Property 1: Dual stability criteria

The closed-loop process is stable if

a) the process is modelled exactly

b) the open-loop process is stable and

c) the IMC controller, *C* is stable. That is, both the controller, *C*, and the open-loop plant, *G*, have poles only inside the stability region.

Proof: (Morari and Zafiriou, 1989)

The closed-loop process can be described by:

$$\begin{aligned} \mathrm{y}(\gamma) &= G(\mathbf{I} + C\,(\,G - \hat{G})\,)^{-1} C\,(\,\mathbf{r} - \mathbf{d}\,)\; + \mathbf{d} \\ &= GC\,(\mathbf{r} - \mathbf{d})\; + \mathbf{d} \end{aligned} \qquad (6.4.5)$$

since the process is exactly modelled. Clearly this is stable if *G* is stable and *C* is stable.

Property 2: Perfect control

The closed-loop process follows the reference signal exactly, and hence offers perfect control if

a) the process is modelled exactly and

b) $C = \mathbf{G}^{-1}$

Proof: (Morari and Zafiriou, 1989)

If $C = \mathbf{G}^{-1}$ and $G = \hat{G}$ then the closed-loop transfer function becomes:

$$\begin{aligned}\mathbf{y}(\gamma) &= G(\mathbf{I} + C(G - \hat{G})^{-1}C(\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= GC(\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= (\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= \mathbf{r}(\gamma) \qquad (6.4.6)\end{aligned}$$

6.4.3 Time-Delayed Processes and IMC

Holt and Morari (1985) pointed out that the transfer function of a time-delayed process cannot, in general, be inverted without the introduction of predictive elements. That is, the inverse of the process transfer function may be non-causal and that the idea of perfect control cannot be applied to processes with time-delays. However, if the process description can be decomposed into the delays and the delay-free dynamics:

$$G^{*}(\gamma) = D(\gamma)G(\gamma) \qquad (6.4.7)$$

where $D(\gamma)$ and $G(\gamma)$ are square matrices and the delay-free dynamics $G(\gamma)$ can be inverted to produce $C(\gamma) = G(\gamma)^{-1}$ with $C(\gamma)$ stable, then if $C(\gamma)$ is used as the controller applied to the process, it follows from the proof of Property 2, Section 6.4.2, that the closed-loop system becomes:

$$\mathbf{y}(\gamma) = D(\gamma)(\mathbf{r} - \mathbf{d}) + \mathbf{d} \qquad (6.4.8)$$

assuming that the process is exactly modelled.

By choosing C as the product

$$C = G^{-1}C_{cl} \qquad (6.4.9)$$

where $$C_{cl} = (\mathrm{I} + GK)^{-1}GK \tag{6.4.10}$$

it can be seen that the closed-loop process has the transfer function

$$\begin{aligned}\mathbf{y}(\gamma) &= DGC(\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= DG G^{-1} C_{cl}(\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= DC_{cl}(\mathbf{r} - \mathbf{d}) + \mathbf{d}\end{aligned} \tag{6.4.11}$$

From Equations (6.4.10) and (6.4.11) it can be seen that the IMC controller is:

$$\begin{aligned}C &= G^{-1}(\mathrm{I} + GF)^{-1}GF \\ &= (\mathrm{I} + KG)^{-1}K\end{aligned} \tag{6.4.12}$$

Equation (6.4.2) provides a means of obtaining the traditional controller of this process:

$$\begin{aligned}F &= (\mathrm{I} - CG^{*})^{-1}C \\ &= (\mathrm{I} - (\mathrm{I} + KG)^{-1}KDG)^{-1}(\mathrm{I} + KG)^{-1}K \\ &= ((\mathrm{I} + KG) - KDG)^{-1}K \\ &= (\mathrm{I} + KH)^{-1}K\end{aligned} \tag{6.4.13}$$

This can be seen to be the traditional Smith predictor, as was obtained in Section 6.3.1. Hence the Smith predictor for a process in which the delays and the dynamics can be decomposed into a product of square matrices, ***D*** and ***G***, can be obtained using IMC techniques.

In general, as was shown in Chapter 3, it is not always possible to decompose a time-delayed process so that the delay operator and the dynamics are both square multivariable transfer functions. Some authors, for example Holt and Morari (1985), Shanmugathasan and Johnston (1988a, 1988b), square down the process so that there are an equal number of inputs and outputs to the controlled process. In the next section, it will be shown that, by using the decompositions of Chapter 3, it is possible to extend the ideas of IMC to non-square processes without squaring down the system.

6.4.4 IMC FOR NON-SQUARE DELAY-FREE SYSTEMS

Consider now a process in which the multivariable transfer function has a different number of inputs from the number of outputs.

$$\mathbf{y}(\gamma) \;= G(\gamma)\mathbf{u}(\gamma) \qquad (6.4.14)$$

where $G(\gamma)$ is a p×q matrix.

The results of Sections 6.4.1 and 6.4.2 can be extended to non-square processes as is shown in the following results.

Result 6.4.1 *The closed-loop process, Equation* (6.4.14) *will, in IMC form, have the transfer function*

$$\mathbf{y}(\gamma) \;= G(\,\mathbf{I} + C(\,G - \hat{G}\,)^{-1}C\,(\mathbf{r} - \mathbf{d}) \;+ \mathbf{d} \qquad (6.4.15)$$

If in addition the process is modelled exactly then the closed-loop process will have the transfer function,

$$\mathbf{y}(\gamma) \;= G(\gamma)C(\gamma)(\mathbf{r} - \mathbf{d}) + \mathbf{d} \qquad (6.4.16)$$

which will be stable if both ***G*** *and* ***C*** *are stable.*

Proof: The first part of the result can be obtained trivially by following exactly the same steps for the non-square case as were taken in Section 6.4.1 for the case when G and K were square.

Equation (6.4.16) follows immediately since, when the process is modelled exactly the difference between the transfer functions representing the actual process and the model of the process $G - \hat{G}$ will always be zero.

Clearly, as in the Dual stability criteria, Property 1, if both G and C are stable then their product will be stable, hence the closed-loop process will also be stable.

Result 6.4.2 *If*

a) the process, presented in Equation (6.4.14), is exactly modelled

b) both G and C are stable

c) *G is of rank q*

d) $q \leq p$

e) C is chosen to be the right generalised inverse of G, $[GG^T]^{-1}$.

then the process is perfectly controlled.

Proof: Result 6.4.1 shows that if conditions a) and b) are satisfied then the closed loop process will be stable with transfer function

$$\mathbf{y}(\gamma) = G(\gamma)C(\gamma)(\mathbf{r} - \mathbf{d}) + \mathbf{d} \qquad (6.4.17)$$

Since G is of rank q and $q \leq p$ it can be shown that $(G)^- = G^T[GG^T]^{-1}$exists. Hence choosing C as

$$C = (G)^- = G^T[GG^T]^{-1} \qquad (6.4.18)$$

ensures that

$$GC \ = GG^T[GG^T]^{-1} = \mathbf{I} \tag{6.4.19}$$

Clearly in order for C to be stable G^T and $[G^TG]^{-1}$ must both be stable. This requires that the poles and zeros of G lie inside the stability region. That is, the process must be stable and non-minimum phase in order that it can be perfectly controlled, in the IMC sense.

6.4.5 IMC for Non-Square Time-Delayed Systems

Considered now the general multivariable time-delayed system, Equations (3.3.5)-(3.3.8). By the decompositions of Chapter 3 it is possible to describe the open-loop process:

$$G^*(\gamma) = \begin{pmatrix} d_{11}g_{11} & d_{12}g_{12} & \cdots & d_{2q}g_{2q} \\ d_{21}g_{21} & d_{22}g_{22} & \cdots & d_{1q}g_{2q} \\ \cdot & \cdot & & \cdot \\ d_{p1}g_{p1} & d_{p2}g_{p2} & \ldots. & d_{pq}g_{pq} \end{pmatrix} \tag{6.4.20}$$

as

$$G^*(\gamma) \ = D(\gamma)G^+(\gamma) \tag{6.4.21}$$

where

$$D(\gamma) = \begin{pmatrix} d_{12}(\gamma) & d_{12}(\gamma) & ..d_{1q}(\gamma) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{21}(\gamma) & \ldots & & . \\ 0 & 0 & 0 & 0 & & . & d_{pq}(\gamma) \end{pmatrix} \tag{6.4.22}$$

and

$$G^+ = \begin{pmatrix} g_{11} & 0 & \ldots & 0 \\ 0 & g_{12} & \ldots & 0 \\ & & & \\ g_{21} & 0 & \ldots & 0 \\ 0 & . & . & \\ . & & g_{22} & \ldots \\ & & & \\ 0 & & & g_{pq} \end{pmatrix} \tag{6.4.23}$$

Since it can be seen that C must be of dimension q×p and that, from inspection of Equation (6.4.23) it can be seen that G^+ is of dimension pq×q, it is not possible to find C such that $G^+C = \mathbf{I}$ unless q=1.

By comparison with the method used to solve the multivariable Smith predictor in which the identity matrix was replaced by **E,** a similar technique is considered.

Result 6.4.3 *If*

a) *the process, presented in Equation (6.4.14), is exactly modelled*

b) *both* **G** *and* **C** *are stable*

c) *G is of rank q*

d) $q \leq p$

e) *C is chosen so that* $G^+C = C_{cl}$

where $C_{cl} = (\mathbf{I} + G^+\mathbf{K}\mathbf{E})^{-1}G^+\mathbf{K}$ (6.4.24)

and **E** *is a constant p×pq matrix.*

then the IMC method will produce a control of the same form as the Smith predictor.

Remark: C_{cl} is the transfer function of the closed-loop process shown in Figure 6.4.4.

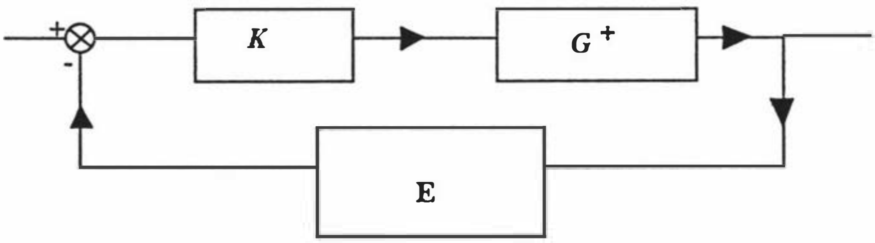

Figure 6.4.4

Proof: Result 6.4.1 shows that if conditions a) and b) are satisfied then the closed-loop process will be stable with transfer function

$$\begin{aligned} \mathbf{y}(\gamma) &= G^*(\gamma)C(\gamma)(\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= D(\gamma)G(\gamma)C(\gamma)(\mathbf{r} - \mathbf{d}) + \mathbf{d} \qquad (6.4.25) \end{aligned}$$

Since G is of rank q and $q \leq p$ it can be shown that $(G^+)^- = (G^+)^T[(G^+)(G^+)^T]^{-1}$ exists. Hence it can be seen from Assumption (e) that:

$$\begin{aligned} G^+C &= C_{cl} \\ \Rightarrow \qquad C &= (G^+)^- C_{cl} \qquad (6.4.26) \end{aligned}$$

If the process is exactly modelled then the closed-loop process becomes:

$$\begin{aligned} \mathbf{y}(\gamma) &= G^*(\mathbf{I} + C(G^* - \hat{G^*}))^{-1}C(\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= DG^+C(\mathbf{r} - \mathbf{d}) + \mathbf{d} \\ &= DC_{cl}(\mathbf{r} - \mathbf{d}) + \mathbf{d} \qquad (6.4.27) \end{aligned}$$

The closed-loop transfer function between **r** and **y** can be easily identified from Equation (6.4.27) as:

$$D(\gamma)C_{cl}(\gamma) \qquad (6.4.28)$$

This can be seen to be the same as that found in Section 6.3.1 using Smith's method.

The traditional controller can be found to be

$$\begin{aligned} F &= (\mathbf{I} - CG^*)^{-1}C \\ &= C(\mathbf{I} - DG^+C)^{-1} \\ &= C(\mathbf{I} - DC_{cl})^{-1} \\ &= (G^+)^- C_{cl}(\mathbf{I} - DC_{cl})^{-1} \qquad (6.4.29) \end{aligned}$$

By expanding C_{cl} it can be seen that

$$F = (G^{+})^{-}(\mathrm{I}+ G^{+}\mathrm{KE})^{-1}G^{+}\mathrm{K}(\mathrm{I} - D(\mathrm{I}+ G^{+}\mathrm{KE})^{-1}G^{+}\mathrm{K})^{-1} \tag{6.4.30}$$

By using the matrix identity $(\mathbf{I} + \mathbf{AB})^{-1}\mathbf{A} = \mathrm{A}(\mathbf{I} + \mathbf{BA})^{-1}$ it can be seen that

$$\begin{aligned} F &= (G^{+})^{-}G^{+}\mathrm{K}(\mathrm{I}+ \mathrm{E}G^{+}\mathrm{K})^{-1}(\mathrm{I} - DG^{+}\mathrm{K}(\mathrm{I}+ \mathrm{E}G^{+}\mathrm{K})^{-1})^{-1} \\ &= \mathrm{K}(\,(\mathrm{I}+ \mathrm{E}G^{+}\mathrm{K}) - DG^{+}\mathrm{K})^{-1} \end{aligned} \tag{6.4.31}$$

By defining

$$H = (\mathrm{E}- D)G^{+} \tag{6.4.32}$$

it can be seen that F can be written as

$$F = \mathrm{K}(\mathrm{I} + H\mathrm{K})^{-1} \tag{6.4.33}$$

which can be seen to be the transfer function form of the generalised multivariable Smith predictor.

6.5 REVIEW

In this chapter Smith predictors have been investigated. In particular a multivariable Smith predictors for time-delayed processes of the form given in Equations (3.3.5)-(3.3.8) have been developed using the results of Chapter 3. These generalised Smith predictors closely resemble the traditional Smith predictor that has been used for single-input single output processes.

The multivariable Smith predictor, which was the subject of this chapter, was developed in two ways: using Smith's method and via the IMC methodology. This multivariable Smith predictor clearly separates the delays from the time-delay dynamics of the process, as is necessary by using Smith's method. Smith's method also explicitly removes the time-delays from the characteristic equation of the closed loop process. The multivariable Smith predictor also contains a model of a delay free process and this model provides a prediction of the effect of the control action on this delay-free processes outputs. Hence this generalised multivariable Smith predictor has the three properties that Jerome and Ray (1986) consider a Smith predictor to possess.

7 TIME-DOMAIN REPRESENTATIONS OF SMITH PREDICTORS

7.1 INTRODUCTION

In this chapter generalised multivariable Smith predictors are developed in the time-domain using state-space descriptions. Traditionally, Smith predictors have been presented in the frequency-domain, where processes are represented as transfer functions. The implementation of a Smith predictor in this way has had to include approximations to time-delays, using mathematical devices such as Padé approximations. Allowing the multivariable Smith predictor to be represented in the time-domain has two advantages. The first advantage is that a control law can be designed using state-space techniques. The second advantage arises if the control law is implemented digitally since the delays can be represented as a queue of measurements rather than an approximation to an exponential function.

In Section 7.2 the generalised multivariable Smith predictor developed in Section 6.3 will be used to identify the requirements for a time-domain implementation of a Smith predictor. It will be shown that a key component of a time-domain representation of a Smith predictor is a time-domain model of an equivalent delay-free process.

As in Chapter 6, throughout this chapter it will be assumed that the processes are modelled exactly, as was the case with the original Smith predictor.

7.2 TIME-DOMAIN REPRESENTATIONS OF SMITH PREDICTORS

In this section the derivation of a Smith predictor based around the time-delayed state-space descriptions developed in Section 4.4 is discussed.

It was shown in Section 6.3 that a Smith predictor for a process in which the transfer function can be written in the output-delay form. A block diagram of such a process with a Smith predictor is shown in Figure 7.2.1.

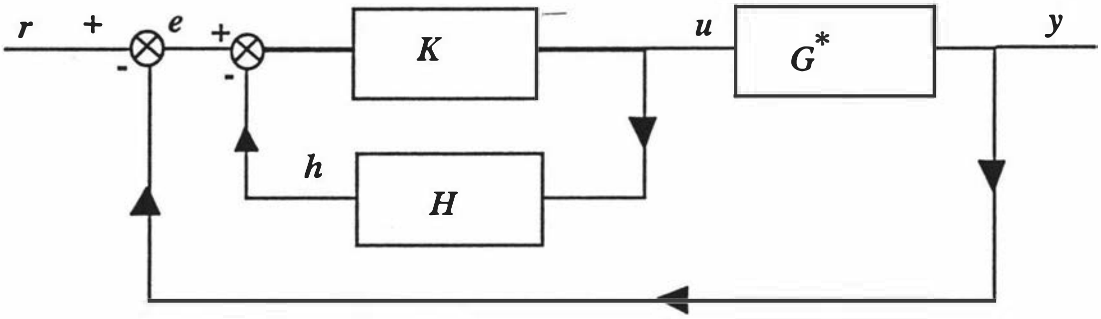

Figure 7.2.1

In this process description the time-delayed transfer function, G^* can be expressed as the product:

$$G^*(\gamma) = D(\gamma)G^+(\gamma) \tag{7.2.1}$$

and the control-inputs can be calculated as

$$\mathbf{u}(\gamma) = (\mathbf{I} + \mathbf{K}H)^{-1}\mathbf{K}\varepsilon(\gamma) \tag{7.2.2}$$

where

$$H(\gamma) = (\mathbf{E}\text{-}D(\gamma))G(\gamma) \tag{7.2.3}$$

and

$$\varepsilon(t) = \mathbf{r}(t)-\mathbf{y}(t) \qquad (7.2.4)$$

K is the controller designed for an equivalent delay-free process and **E** is a constant matrix with the same dimensions as $D(\gamma)$.

Figure 7.2.1 makes it clear, that if output-delays were used, the transfer function matrix for $\mathbf{h}(\gamma)$ is

$$\begin{aligned} \mathbf{h}(\gamma) &= H(\gamma)\mathbf{u}(\gamma) \\ &= (\mathbf{E} - D(\gamma))G^{+}(\gamma)\mathbf{u}(\gamma) \qquad (7.2.5) \end{aligned}$$

Premultiplying Equation (7.2.2) by $(\mathbf{I} + \mathbf{K}H)$ it can be seen that an expression for the control inputs in terms of $\varepsilon(\gamma)$ and $\mathbf{h}(\gamma)$ can be obtained:

$$\begin{aligned} (\mathbf{I} + \mathbf{K}H)\mathbf{u}(\gamma) &= \mathbf{K}\varepsilon(\gamma) \qquad (7.2.6) \\ \mathbf{u}(\gamma) &= \mathbf{K}\varepsilon(\gamma) - \mathbf{K}H(\gamma)\mathbf{u}(\gamma) \\ \mathbf{u}(\gamma) &= \mathbf{K}\varepsilon(\gamma)-\mathbf{K}\mathbf{h}(\gamma) \\ \mathbf{u}(\gamma) &= \mathbf{K}(\varepsilon(\gamma)-\mathbf{h}(\gamma)) \qquad (7.2.7) \end{aligned}$$

Inverse transforming Equation (7.2.7) gives:

$$\mathbf{u}(t) = \mathbf{K}(\varepsilon(t)-\mathbf{h}(t)) \qquad (7.2.8)$$

From Equation (7.2.8) it is clear that in order to implement a Smith predictor in the time-domain it is necessary to have a control law that will improve the control of the equivalent delay-free process, **E***G* which is represented by **K** and **h**(t) the output from the Smith predictor return matrix.

Time-delayed processes can be represented using Corollary 4.3 as either input-delayed or an output-delayed state-space descriptions. In either of these forms, as was shown in Section 4.5, an equivalent delay-free state-space description can be readily obtained, the Auxiliary state-space:

$$\rho\tilde{\mathbf{x}}(t) = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{B}^{+}{}_{u}(1)\mathbf{u}(t) \qquad (7.2.9)$$

$$\tilde{\mathbf{y}}(t) = \mathbf{D}_{y}(1)\mathbf{C}^{+}\tilde{\mathbf{x}}(t) \qquad (7.2.10)$$

It can be seen that this state-space description can be transformed to

$$\tilde{\mathbf{y}}(\gamma) = \mathbf{E}\boldsymbol{G}^{+}(\gamma) \qquad (7.2.11)$$

where $\mathbf{E} = \boldsymbol{D}(1)$.

This state-space description can be used to design a controller that will then be used in the Smith predictor. The controller can be designed using any state-space controller design method.

From Equation (7.2.5) it can be seen that for the output-delay form the Smith predictor return matrix, $\mathbf{h}(\gamma)$, can be written as

$$\mathbf{h}(\gamma) = \tilde{\mathbf{y}}(\gamma) - \mathbf{y}(\gamma) \qquad (7.2.12)$$

which by inverse transforming can be expressed in the time-domain as:

$$\mathbf{h}(t) = \tilde{\mathbf{y}}(t) - \mathbf{y}(t) \qquad (7.2.13)$$

where $\tilde{\mathbf{y}}(t)$ is the time-domain output-vector from the Auxiliary delay-free state-space description, Equation (7.2.9)-(7.2.10).

If the process were written using Corollary 4.3 in input-delay form:

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}^{+}D_{u}(q^{-1})\mathbf{u}(t) \tag{7.2.14}$$

$$\mathbf{y}(t) = D_{y}(1)\mathbf{C}^{+}\mathbf{x}(t) \tag{7.2.15}$$

it can be seen that $\mathbf{h}(t)$ can be expressed as

$$\begin{aligned}\mathbf{h}(t) &= \tilde{\mathbf{y}}(t) - \mathbf{y}(t)\\ &= D_{y}(1)C^{+}\tilde{x}(t) - D_{y}(1)C^{+}x(t)\\ &= D_{y}(1)C^{+}(\tilde{x}(t) - x(t))\end{aligned} \tag{7.2.16}$$

By augmenting the state-spaces descriptions, Equations (7.2.9)-(7.2.10) and (7.2.14)-(7.2.15) the following state-space description can be obtained:

$$\rho\mathbf{X} = \begin{bmatrix}\mathbf{A} & 0\\ 0 & \mathbf{A}\end{bmatrix}\mathbf{X}(t) + \mathbf{B}^{+}\begin{bmatrix}D_{u}(1)\\ D_{u}(q^{-1})\end{bmatrix}u(t) \tag{7.2.17}$$

$$\mathbf{y}(t) = D_{y}(1)\mathbf{C}^{+}\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\mathbf{X}(t) \tag{7.2.18}$$

it can be seen that $\mathbf{h}(t)$ can be obtained from $\mathbf{X}(t)$ as:

$$\mathbf{h}(t) = D_{y}(1)\mathbf{C}^{+}\begin{bmatrix}\mathbf{I} & -\mathbf{I}\end{bmatrix}\mathbf{X}(t) \tag{7.2.19}$$

Lemma 4.5 shows that the observability of the time-delayed state-space description, Equations (7.2.17)-(7.2.19), is the same as the controllability of its auxiliary state-space:

$$\rho\tilde{\mathbf{X}} = \begin{bmatrix}\mathbf{A} & 0\\ 0 & \mathbf{A}\end{bmatrix}\tilde{\mathbf{X}}(t) + \mathbf{B}^{+}\begin{bmatrix}D_{u}(1)\\ D_{u}(1)\end{bmatrix}u(t) \tag{7.2.20}$$

$$\tilde{\mathbf{y}}(t) = D_{y}(1)\mathbf{C}^{+}\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\tilde{\mathbf{X}}(t) \tag{7.2.21}$$

The controllability matrix of the state-space description, Equations (7.2.20)-(7.2.21) can be found to be:

$$\mathbf{Q}=D_y(1)\begin{bmatrix} \mathbf{C}^{+}\begin{bmatrix}0 & \mathbf{I}\end{bmatrix} \\ \mathbf{C}^{+}\begin{bmatrix}0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{A}^{T} & 0 \\ 0 & \mathbf{A}^{T}\end{bmatrix} \\ \mathbf{C}^{+}\begin{bmatrix}0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{A}^{T} & 0 \\ 0 & \mathbf{A}^{T}\end{bmatrix}^{2} \\ \vdots \end{bmatrix} \qquad (7.2.22)$$

which is not of full rank, implying that the auxiliary state-space description is not completely observable and hence that the time-delayed state-space, Equations (7.2.17)-(7.2.18) is not completely observable.

Since the augmented state-space description is not completely observable an observer, or filter for **X**(t) cannot be constructed. Hence **h**(t) cannot be obtained from process measurements. The only way to obtain **h**(t) then is to model it using, for example, Equations (7.2.17) and (7.2.19) inside the controller.

It can also be seen that **h**(t) could be obtained from the state-space description:

$$\rho\mathbf{x_h}(t) = \mathbf{A}\mathbf{x_h}(t) + \mathbf{B}^{+}(D_u(1)-D_u(q^{-1}))\mathbf{u}(t) \qquad (7.2.23)$$

$$\mathbf{h}(t) = D_y(1)\mathbf{C}^{+}\mathbf{x_h}(t) \qquad (7.2.24)$$

Assuming that a good estimate of the initial value of $\tilde{\mathbf{x}}(t)$ can be obtained and that the model of the process is good, then a reasonable estimate of $\tilde{\mathbf{x}}(t)$, when t > 0, can be obtained from input data as:

$$\tilde{x}= E(\mathbf{A},t)\tilde{x}(0)+\int_{0}^{t}(E(\mathbf{A},\tau-t)\mathbf{B}^{+}D_u(1)\mathbf{u}(\tau))\,d\,\tau \qquad (7.2.25)$$

If the process is open-loop stable, then as $t \rightarrow \infty$ the effect of the initial conditions, $E(\mathbf{A},t)\tilde{\mathbf{x}}(0)$, will diminish. However, if the process is not open-loop stable then the effect of the initial conditions will dominate and cause the closed-loop process to also be unstable.

If Corollary 4.3 were used to write the process in an output-delay form:

$$\rho\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}^{+}{}_{u}(1)\mathbf{u}(t) \tag{7.2.26}$$

$$\mathbf{y}(t) = D_y(q^{-1})\mathbf{C}^{+}\mathbf{x}(t) \tag{7.2.27}$$

it can be seen that $\mathbf{h}(t)$ can be obtained from $\mathbf{x}(t)$ as:

$$h(t) = (D_y(1) - D_y(q^{-1}))\mathbf{C}^{+}\mathbf{x}(t) \tag{7.2.28}$$

In a simular way to the analysis performed for the input-delay form, it can be seen that a model must be maintained to calculate $\mathbf{h}(t)$ and that the process must be open-loop stable.

The two methods, developed above, for calculating $\mathbf{h}(t)$ could easily be implemented digitally. An algorithm to find the control, $\mathbf{u}(t)$, could be written as:

```
Estimate $\mathbf{h}(t)$ using the model and old inputs.
Measure $\mathbf{y}(t)$ and the setpoint $\mathbf{r}(t)$
 Calculate      $\varepsilon(t) = \mathbf{y}(t) - \mathbf{r}(t)$
                $\varepsilon_h(t) = \varepsilon(t) - \mathbf{h}(t)$
 Calculate $\mathbf{u}(t) = \mathbf{K}\varepsilon_h(t)$
```

Example 7.1: Consider the process used in Examples 3.3, 4.1 and 6.1

$$\boldsymbol{G}^*(s) = \begin{pmatrix} \dfrac{e^{-2s}}{(s+1)} & \dfrac{3e^{-s}}{(s+2)} \\ \dfrac{2e^{-3s}}{(s+3)} & \dfrac{-e^{-1.5s}}{(s+4)} \end{pmatrix} = \begin{pmatrix} e^{-2s} & e^{-s} & 0 & 0 \\ 0 & 0 & e^{-3s} & e^{-1.5s} \end{pmatrix} \begin{pmatrix} \dfrac{1}{(s+1)} & 0 \\ 0 & \dfrac{3}{s+2} \\ \dfrac{2}{s+3} & 0 \\ 0 & \dfrac{-1}{s+4} \end{pmatrix} \qquad (7.2.29)$$

In Example 4.4 input-delay and output-delay forms of state-space descriptions for this process were obtained. The output-delay form was

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}(t) + \begin{pmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 0 \\ 0 & -1 \end{pmatrix} \mathbf{u}(t) \qquad (4.2.26)$$

$$\mathbf{y}(t) = \begin{pmatrix} q^{-2} & q^{-1} & 0 & 0 \\ 0 & 0 & q^{-3} & q^{-1.5} \end{pmatrix} \mathbf{x}(t) \qquad (4.4.27)$$

where

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \qquad (4.4.28)$$

and the input-delay form was:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}(t) + \begin{pmatrix} q^{-2} & 0 \\ 0 & 3q^{-1} \\ 2q^{-3} & 0 \\ 0 & -q^{-1.5} \end{pmatrix} \mathbf{u}(t) \qquad (4.4.29)$$

$$\mathbf{y}(t) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \mathbf{x}(t) \qquad (4.4.30)$$

The auxiliary delay-free state-space description of this process was obtained in Example 4.1 as:

$$\frac{d\tilde{\mathbf{x}}}{dt} = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) \tag{7.2.30}$$

$$\tilde{\mathbf{y}}(t) = \mathbf{C}\tilde{\mathbf{x}}(t) \tag{7.2.31}$$

where

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \tag{4.2.9}$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4.2.8}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{pmatrix} \tag{4.2.4}$$

$\mathbf{h}(t)$ can be found using Equation (7.2.16) as:

$$\mathbf{h}(t) = D_y(1)\mathbf{C}^{+}\tilde{\mathbf{x}}(t) - D_y\mathbf{C}^{+}\mathbf{x}(t)$$

$$\mathbf{h}(t) = \tilde{\mathbf{y}}(t) - \mathbf{y}(t)$$

$$= D_y(1)\mathbf{C}^{+}(\tilde{\mathbf{x}}(t) - \mathbf{x}(t)) \tag{7.2.16}$$

$$= \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{pmatrix}(\tilde{\mathbf{x}}(t) - \mathbf{x}(t)) \tag{7.2.32}$$

If the process were written in an output-delay form, then Equation (7.2.28) would give:

$$h(t) = (D_y(1) - D_y(q^{-1}))\mathbf{C}^{+}\mathbf{x}(t) \tag{7.2.28}$$

$$=\left(\begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{pmatrix}-\begin{pmatrix} q^{-2} & q^{-1} & 0 & 0 \\ 0 & 0 & q^{-3} & q^{-1.5} \end{pmatrix}\right)\mathbf{x}(t) \qquad (7.2.33)$$

7.3 REVIEW

In this chapter a method for using state-space descriptions of multivariable time-delayed processes in an implementation of multivariable Smith predictors have been presented. These state-space Smith predictors have an advantage over traditional Smith predictors since any time-domain state-space method for designing delay-free process controllers can be used in conjunction with this form of Smith predictor.

8 Robust Stability And Sensitivity To Modelling Errors Of The Smith Predictor

8.1 Introduction

The preceding chapters have shown that the multivariable Smith predictor can control multivariable time-delayed processes. However it is important to recognise, as with any device, the limitations of the multivariable Smith predictor. In this chapter the robustness, stability and the sensitivity to modelling errors of the multivariable Smith predictor are considered.

In Section 8.2 it will be shown that the Smith predictor cannot be applied to open-loop unstable processes.

In Section 8.3 additive and multiplicative error models are discussed. These models are used to describe the differences between the actual process and the model of the process that is used for designing the Smith predictor.

Section 8.4 considers a perturbation analysis of the multivariable Smith predictor.

Section 8.5 contains a discussion of the practical stability properties of a multivariable Smith predictor.

In Section 8.6 a robust stability result due, to Owens and Raya (1982), is extended to the multivariable Smith predictor.

8.2 The Smith Predictor and Open-Loop Unstable Processes

In this section it will be shown that, in general, a multivariable Smith predictor cannot be applied to open-loop unstable processes. This result is derived using the time-delayed state-space description of Section 4.4 and the Smith predictor controller that was derived in Section 6.3.

Theorem 8.1: *A time-delayed closed-loop system controlled with a Smith predictor has 2n modes, n of these are those of the open-loop process and the other n are the modes of the equivalent delay-free closed-loop system, where n is the order of the state-space description.*

Proof: (Adapted from Palmor and Halevi, 1983)

Corollary 4.3 shows that a time-delayed process can be described by the state-space description:

$$\rho\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}^{+}D_u(q^{-1})\mathbf{u}(t) \qquad (8.2.1)$$

$$\mathbf{y} = D_y(1)\mathbf{C}^{+}\mathbf{x} \qquad (8.2.2)$$

The Smith predictor scheme based on a state-feedback control law, calculates the input, **u**(t) as:

$$\mathbf{u}(t) = \mathbf{K}(\mathbf{x}(t) - \mathbf{h}(t)) \qquad (8.2.3)$$

where

$$\rho\mathbf{h}(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}^{+}(D_u(1) - D_u(q^{-1}))\mathbf{u}(t) \qquad (8.2.4)$$

Writing $\mathbf{X}(t)^T = [\mathbf{x}(t)^T\ \mathbf{h}(t)^T]$, it follows from Equations (8.2.1), (8.2.3) and (8.2.4) that the closed-loop system can be described by

$$\rho\mathbf{X}(t) = \begin{pmatrix} \mathbf{A}+\mathbf{B}^{+}D_u(q^{-1})\mathbf{K} & -\mathbf{B}^{+}D(q^{-1})\mathbf{K} \\ \mathbf{B}^{+}(D_u(1)-D_u(q^{-1}))\mathbf{K} & \mathbf{A}-\mathbf{B}^{+}(\mathbf{E}-D_u(q^{-1}))\mathbf{K} \end{pmatrix}\mathbf{X}(t)$$

(8.2.5)

The modes of the overall system are given by the roots of

$$\det\begin{pmatrix} \mathbf{I}\rho-\mathbf{A}-\mathbf{B}^{+}D_u(q^{-1})\mathbf{K} & \mathbf{B}^{+}D_u(q^{-1})\mathbf{K} \\ -\mathbf{B}^{+}(D_u(1)-D_u(q^{-1}))\mathbf{K} & \mathbf{I}\rho-\mathbf{A}+\mathbf{B}^{+}(D_u(1)-D_u(q^{-1}))\mathbf{K} \end{pmatrix} = 0$$

(8.2.6)

Since $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$ and

$$\det\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{pmatrix} = \det\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix} = 1$$

it can be seen that

$$\begin{aligned} &\det\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}\det\begin{pmatrix} \mathbf{I}\rho-\mathbf{A}+\mathbf{B}^{+}\boldsymbol{D}(q^{-1})\mathbf{K} & -\mathbf{B}^{+}\boldsymbol{D}(q^{-1})\mathbf{K} \\ -\mathbf{B}^{+}(\mathbf{E}_u-\boldsymbol{D}(q^{-1}))\mathbf{K} & \mathbf{I}\rho-\mathbf{A}\mathbf{B}^{+}(\mathbf{E}_u-\boldsymbol{D}(q^{-1}))\mathbf{K} \end{pmatrix}\det\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{pmatrix} \\ &\quad = \det\begin{pmatrix} \mathbf{I}\rho-\mathbf{A} & -\mathbf{B}^{+}\boldsymbol{D}(q^{-1})\mathbf{K} \\ \mathbf{0} & \mathbf{I}\rho-\mathbf{A}-\mathbf{B}^{+}\mathbf{E}_u\mathbf{K} \end{pmatrix} \\ &\quad = \det(\mathbf{I}\rho-\mathbf{A})\det(\mathbf{I}\rho-\mathbf{A}+\mathbf{B}^{+}\mathbf{E}_u\mathbf{K}) \end{aligned}$$

(8.2.7)

Hence it can be seen that the modes of the overall closed-loop system consist of two parts: the modes of the open-loop system and the modes of the equivalent delay-free closed-loop plant.

Corollary: *The multivariable Smith predictor cannot be used to stabilise a time-delayed open-loop unstable process.*

Proof: The result follows immediately since the overall closed-loop process will contain among its modes, the (unstable) open-loop modes of the process, which make the closed-loop unstable.

This result has previously been commented on by various authors, including Gawthrop (1977) Watanbe and Ito (1981) and Furukawa and Shimemura (1983) and has often been cited as a major disadvantage of the Smith predictor. Despite this, the Smith predictor remains as a viable controller design tool for the vast majority of time-delayed processes since most industrial processes are in fact stable.

De Paor (1985) and De Paor and Egan (1989) produced modified Smith predictors that allowed certain open-loop unstable processes to be controlled via a Smith predictor-like control scheme. An extension of the ideas of the generalised Smith predictor, that was presented in Chapters 4 and 5, to encompass their work is yet to be attempted.

8.3 Additive and Multiplicative Error Models of a Time-Delayed Process

In most practical applications, the dynamics and time-delays are not known exactly and there is likely to be a certain amount of mismatch between the model of the process and the actual plant. The modelling errors, in time-delayed processes, can be divided into two distinct groups: the errors in the estimates of the time-delays and the errors in the model of the delay-free dynamics. There are, in the literature, a number of results that give tolerance bounds on the modelling errors for the time-delays and the dynamics that ensure that the controlled process remains stable. In this section two error models of time-delayed processes are presented that will be investigated in the following sections of this chapter.

8.3.1 Additive Error Models of a Time-Delayed Process

Consider the general time-delayed process of Equations (3.3.5)-(3.3.8):

$$\mathbf{y}(\gamma) = G^*(\gamma)\mathbf{u}(\gamma) = D_1(\gamma)G_p(\gamma)D_2(\gamma)\mathbf{u}(\gamma) \qquad (3.3.5)$$

where $$D_1(\gamma) = \sum_{k=1}^{n_C} C_k E(\gamma,-\gamma_k) \qquad (3.3.6)$$

$$D_2(\gamma) = \sum_{j=1}^{n_B} B_j E(\gamma,-\beta_j) \qquad (3.3.7)$$

and

$$G_p(\gamma)=\begin{pmatrix} \frac{a_{11}(\gamma)E(\gamma,-\tau_{11})}{b_{11}(\gamma)} & \frac{a_{12}(\gamma)E(\gamma,-\tau_{12})}{b_{12}(\gamma)} & \cdots & \frac{a_{1n}(\gamma)E(\gamma,-\tau_{1n})}{b_{1n}(\gamma)} \\ \frac{a_{21}(\gamma)E(\gamma,-\tau_{21})}{b_{21}(\gamma)} & \frac{a_{22}(\gamma)E(\gamma,-\tau_{22})}{b_{22}(\gamma)} & \cdots & \frac{a_{2n}(\gamma)E(\gamma,-\tau_{2n})}{b_{2n}(\gamma)} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{a_{m1}(\gamma)E(\gamma,-\tau_{m1})}{b_{m1}(\gamma)} & \frac{a_{m2}(\gamma)E(\gamma,-\tau_{m2})}{b_{m2}(\gamma)} & \cdots & \frac{a_{mn}(\gamma)E(\gamma,-\tau_{mn})}{b_{mn}(\gamma)} \end{pmatrix} \tag{3.3.8}$$

It was shown in Chapter 3 that this general time-delayed process can be expressed as:

$$\begin{aligned} \mathbf{y}(\gamma) &= G^{*}\mathbf{u}(\gamma) \\ &= D_1 G_p D_2 \mathbf{u}(\gamma) \\ &= D(\gamma)G^{+}(\gamma) \end{aligned} \tag{8.3.1}$$

where $D(\gamma)$ is an $r\times qr$ matrix and $G^{+}(\gamma)$ is a $qr\times q$ matrix.

It will be assumed that the time-delayed process is modelled by:

$$\begin{aligned} \hat{\mathbf{y}}(\gamma) &= \hat{G}^{*}\mathbf{u}(\gamma) \\ &= \hat{D}\hat{G}^{+}\mathbf{u}(\gamma) \end{aligned} \tag{8.3.2}$$

where $\hat{G}^{+}(\gamma)$ and $\hat{D}(\gamma)$ are the sums

$$\hat{G}^{+}(\gamma) = G^{+} + \Delta G \tag{8.3.3}$$

and

$$\hat{D}(\gamma) = D + \Delta D \tag{8.3.4}$$

An alternative method of considering the inaccuracies in delay operators is a multiplicative error model. This model of errors in the delays will be considered in the next section.

8.3.2 A Multiplicative Modelling Error Model

Consider the delay element:

$$\hat{d}_{ij}(\gamma) = E(\gamma,-\hat{\tau}_{ij})=E(\gamma,-(\tau_{ij}+\Delta\tau_{ij})) \qquad (8.3.5)$$

in which

$$\Delta\tau_{ij} = \hat{\tau}_{ij} - \tau_{ij} \qquad (8.3.6)$$

From the properties of generalised exponentials this can be written as:

$$\begin{aligned}\hat{d}_{ij}(\gamma) &= E(\gamma,-\Delta\tau_{ij})E(\gamma,-\tau_{ij}) \\ &= \Delta d_{ij}(\gamma)d_{ij}(\gamma)\end{aligned} \qquad (8.3.7)$$

where

$$\Delta d_{ij}(\gamma) = E(\gamma,-\Delta\tau_{ij}) \qquad (8.3.8)$$

and

$$d_{ij}(\gamma) = E(\gamma,-\tau_{ij}) \qquad (8.3.9)$$

Applying this model to each element of the multivariable delay model $D(\gamma)$:

$$\hat{D} = \begin{bmatrix} \hat{d}_{11} & \hat{d}_{12} & \cdots & \hat{d}_{1q} \\ \hat{d}_{21} & \hat{d}_{22} & \cdots & \hat{d}_{2q} \\ \vdots & \vdots & & \vdots \\ \hat{d}_{p1} & \hat{d}_{p2} & \cdots & \hat{d}_{pq} \end{bmatrix} = \begin{bmatrix} \Delta d_{11} d_{11} & \Delta d_{12} d_{12} & \cdots & \Delta dq d_{1q} \\ \Delta d_{21} d_{21} & \Delta d_{22} d_{22} & \cdots & \Delta d_{2q} d_{2q} \\ \vdots & \vdots & & \vdots \\ \Delta d_{p1} d_{p1} & \Delta d_{p2} d_{p2} & \cdots & \Delta d_{pq} d_{pq} \end{bmatrix}$$

(8.3.10)

$\hat{D}(\gamma)$ can be written, using Theorem 3.2, as:

$$\hat{D} = \begin{pmatrix} \Delta d_{11} & \Delta d_{12} & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \Delta d_{r(q-1)} & \Delta d_{rq} \end{pmatrix} \begin{pmatrix} d_{11} & 0 & .. & \\ 0 & d_{12} & \cdot & \\ & \cdot & \cdot & \\ \cdot & & & \\ & & d_{rq} & \end{pmatrix}$$

$$= \Delta D^{+} \tilde{D}$$

(8.3.11)

Similarly, using Theorem 3.2, it is possible to expand **D** as

$$D = E^{+} \tilde{D}$$

(8.3.12)

where

$$E^{+} \quad = \Delta D^{+}(0)$$

(8.3.13)

The additive error model of the time-delays, Equation (8.3.4), can be rewritten as:

$$\begin{aligned} \Delta D &= \hat{D} - D \\ &= (E^{+} - \Delta D^{+})\tilde{D} \end{aligned}$$

(8.3.14)

where ΔD^{+} is as defined in Equation (8.3.11).

If only one of the delays has a mismatch between the true delay and the modelled delay, say the kl$^{\text{th}}$ element, then it can be seen that

$$\Delta \mathcal{D} = \begin{pmatrix} 0 & 0 & .. & 0 \\ 0 & 1-\Delta d_{kl} & .. & . \\ . & . & & . \\ 0 & 0 & .. & 0 \end{pmatrix} \tilde{\mathcal{D}} \tag{8.3.15}$$

8.4 Perturbation Analysis of Mismatch in Smith Predictors

Consider the block diagram representation of the multivariable Smith predictor, Figure 8.4.1, in which the modelled process is used to construct the Smith predictor and controller.

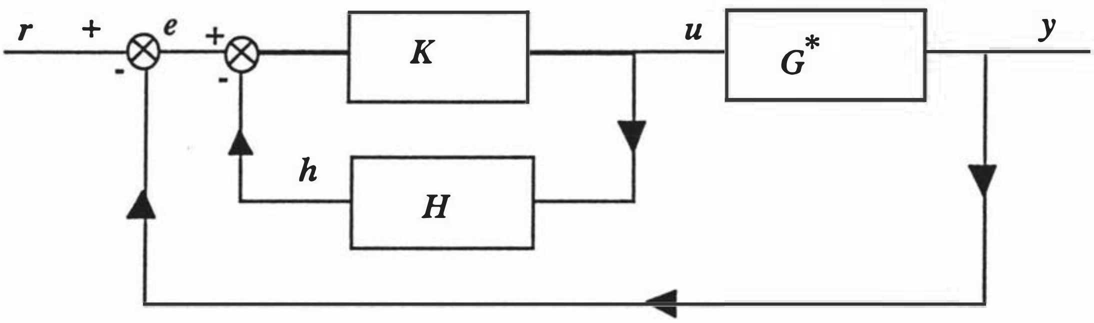

Figure 8.4.1

This block diagram can be rearranged as Figure 8.4.2

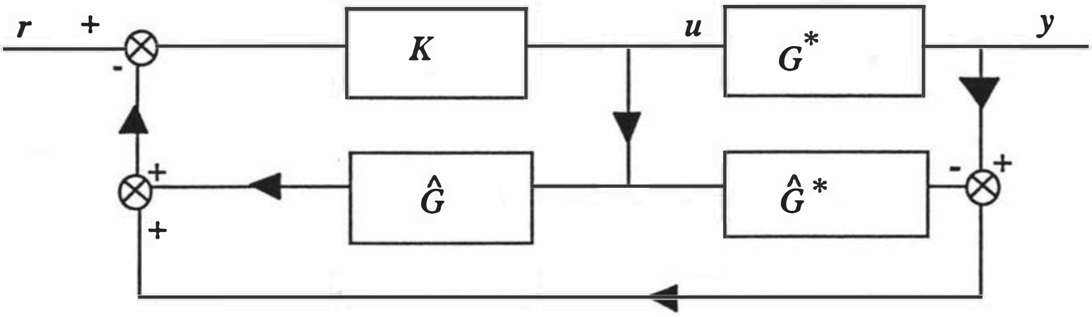

Figure 8.4.2

By expanding

$$\begin{aligned}\hat{G}^*(\gamma) &= \hat{D}(\gamma)\hat{G}^+(\gamma)\\ &= \hat{D}(\gamma)(G^+(\gamma) + \Delta G(\gamma)\end{aligned}$$

Figure 8.4.2 can be rearranged as shown in Figure 8.4.3.

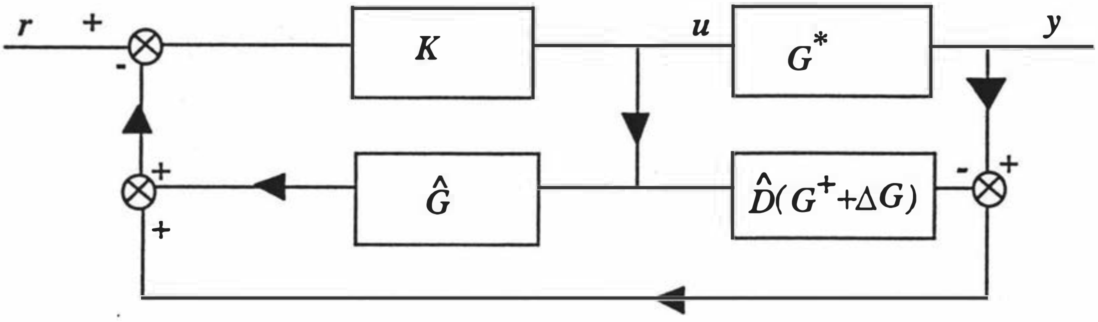

Figure 8.4.3

By rearranging the block diagram Figure 8.4.3 Figures 8.4.4 and 8.4.5 can be easily obtained.

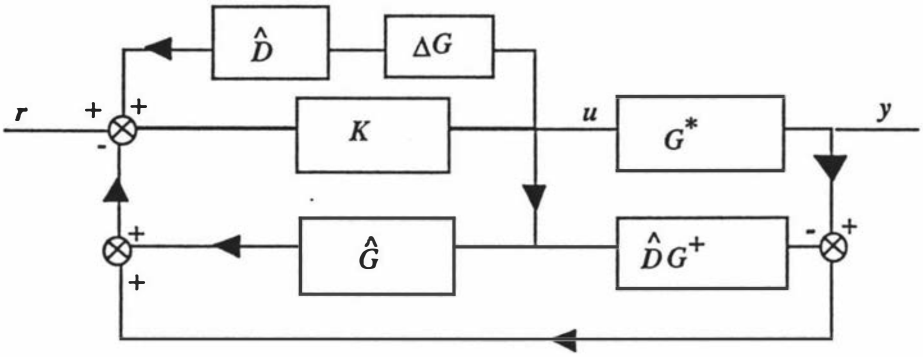

Figure 8.4.4

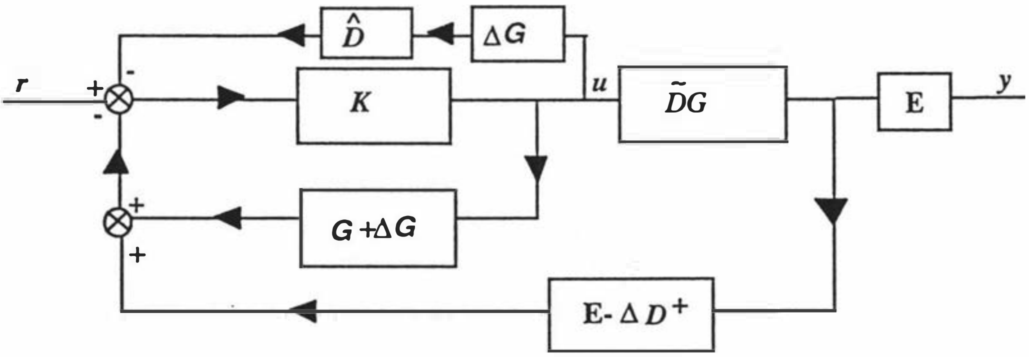

Figure 8.4.5

This block diagram, Figure 8.4.5, illustrates the interaction between the two types of modelling error: the errors in the estimates of the time-delays and the errors in the estimates of the coefficients.

It can be seen that when $\boldsymbol{D} = \hat{\boldsymbol{D}}$ that the time-delays affect the controls through the term in $\Delta\mathbf{G}$ in the upper loop around $\boldsymbol{K}$. This means that if the delay is large, offsets in the model of $\boldsymbol{G}$ can cause instability.

Conversely when $\Delta\boldsymbol{G}= \mathbf{0}$, but the delays are not exactly modelled, there is an outer loop with the term $\tilde{\boldsymbol{D}}(\mathbf{E}^{+} - \Delta D^{+})$. The affect of this outer loop can be likened to adding a derivative effect since the second term can be approximated, to first order, as:

$$(\Delta \boldsymbol{D}^{+} - \mathbf{E}^{+}) \quad \equiv \begin{pmatrix} \delta_{11}\gamma & \delta_{12}\gamma & .. & 0 \\ . & . & & . \\ 0 & 0 & ... & \delta_{qr}\gamma \end{pmatrix} \qquad (8.4.1)$$

where

$$\delta_{ij} \equiv -\Delta\tau_{ij} \qquad (8.4.2)$$

This derivative-like feedback loop will respond rapidly if the delays are not well matched. The single-input single-output case of this perturbation analysis has been investigated by Garland and Marshall (1974, 1975), Marshall (1979, p103-104) and Marshall and Salehi (1982). Marshall (1979) exploits deliberate mismatch to improve the performance of controlled processes.

8.5 Practical Stability of Multivariable Smith Predictors

Practical stability, a weak form of robust stability, has been investigated in the single-input single-output case and in limited forms of the multivariable case. Palmor (1980) investigated practical stability of single-input single-output processes. Palmor and Halevi (1983) investigated practical stability of processes in which the control law was diagonal.

In this section some practical stability results are obtained for the multivariable Smith predictor developed in Chapter 6. The results developed include those of Palmor and Halevi (1983) as special cases.

By letting $G(\gamma,\mu)$ be the transfer function in the γ-domain that has the parameter γ and coefficient values μ, it is possible to make the following definition:

Definition 8.1: (Palmor and Halevi, 1983) A control system is practically stable (ps) if:

(a) it is asymptotically stable ie all poles of $G(\gamma,\mu)$ are in the stability region.

and (b) there exists a bound, $\delta>0$, such that all the poles of $G(\gamma,\mu+\Delta\mu)$ are also in the stability region for all parameter differences $\|\Delta\mu\| < \delta$.

where $\|\cdot\|$ is a vector norm defined on the space of possible parameter values, μ.

The characteristic equation of the overall system can be found from the block diagram, Figure 8.4.1 to be

$$\det\,[\mathbf{I} + \mathbf{K}\mathbf{E}\hat{G}^{+} + \mathbf{K}(G^{*} - \hat{G}^{*}] = 0 \qquad (8.5.2)$$

By writing

$$S = (\mathbf{I} + \mathbf{KE}\hat{G})^{-1} \tag{8.5.3}$$

the characteristic equation, Equation (8.5.2), can be rewritten as

$$\det(S^{-1})\det(\,\mathbf{I} + SK(G^{*} - \hat{G}^{*}) = 0 \tag{8.5.4}$$

Since $\det(S^{-1}) = 0$ is the characteristic equation of the ideal (modelled) process, which can be assumed to have been designed to be stable, it is clear that there are no roots of $\det(S^{-1})=0$ in the unstable region. Thus, the closed-loop process will be unstable if the roots of

$$\det(\,\mathbf{I} + SK(G^{*} - \hat{G}^{*}) = 0 \tag{8.5.5}$$

lie outside the stability region.

Assume temporarily that the system is exactly modelled except for some mismatch in the elements of the transfer function relating the outputs to one of the inputs of the time-delayed process model. That is, it is assumed that

$$\hat{g}_{ij} = g_{ij} \quad \forall\, i\, j \quad j \neq k \tag{8.5.6}$$

$$\tau_{ij} = \hat{\tau}_{ij} \quad \forall\, i\, j \quad j \neq k \tag{8.5.7}$$

and

$$\hat{g}_{ik} + \Delta g_{ik} = g_{ik} \tag{8.5.8}$$

$$\hat{\tau}_{ik} + \Delta\tau_{ik} = \tau_{ik} \tag{8.5.9}$$

Furthermore $\Delta g_{kl}{}^{*}$ can be written as

$$\Delta g_{kl}{}^{*} = \hat{g}_{kl}[E(\gamma,-\tau_{kl})-E(\gamma,-\hat{\tau}_{kl})] + \Delta g_{kl}E(\gamma,-\hat{\tau}_{kl}) \qquad (8.5.10)$$

Since Equations (8.5.6) and (8.5.8) imply that

$$\Delta G^{*} = \begin{pmatrix} 0 & 0 & .. & \Delta g_{1k} & 0 & ... & 0 \\ 0 & 0 & .. & \Delta g_{2k} & 0 & ... & 0 \\ . & . & .. & . & . & ... & . \\ 0 & 0 & .. & \Delta g_{rk} & 0 & ... & 0 \end{pmatrix} \qquad (8.5.11)$$

it can be seen that the characteristic equation, Equation (8.5.5) can be written as:

$$\det(\mathbf{I} + P\Delta G^{*}) = \det(\mathbf{I} + S\mathbf{K}(G^{*} - \hat{G}^{*}) = 0 \qquad (8.5.12)$$

where P is the matrix

$$P = S\mathbf{K} \quad = (\mathbf{I} + \mathbf{KE\hat{G}})^{-1}\mathbf{K} \qquad (8.5.13)$$

The product $\mathbf{P}\Delta G^{*}$ consists of zeros except for the kth column.

Expanding the matrix and solving the determinant along the kth row it can be seen that the determinant is

$$\det(\mathbf{I} + S\mathbf{K}(G^{*} - \hat{G}^{*})) = 1 + \sum_{i=1}^{r} p_{ki}\,\Delta g_{ik}{}^{*} \qquad (8.5.14)$$

The process is unstable if the determinant is zero for γ values outside the stability region. (In the Laplace transform domain this is Re(s)>0.)

Equation (8.5.14) implies that

$$-1 \;= \sum_{i=1}^{r} p_{\mathrm{ki}}\, \Delta g_{\mathrm{ik}}^{*} \qquad (8.5.15)$$

Taking the absolute values of both sides of Equation (8.5.15) gives:

$$\begin{aligned} 1 \;&= \Big| \sum_{i=1}^{r} p_{\mathrm{ki}}\, \Delta g_{\mathrm{ik}}^{*} \Big| \\ &< |\Delta g_{max}^{*}| \sum |p_{\mathrm{ki}}| \end{aligned}$$

$$< |\hat{g}_{max}[E(\gamma,-\tau_{\mathrm{kl}})\text{-}E(\gamma,-\hat{\tau}_{\mathrm{kl}})] + \Delta g_{max} E(\gamma,-\hat{\tau}_{\mathrm{kl}})|| \sum p_{\mathrm{ki}}|$$

$$(8.5.16)$$

where $\Delta g_{max}^{*} \geq \Delta g_{ik}^{*} \quad \forall\, \mathrm{i}$

For γ outside the stability region the absolute value of the generalised exponential, $E(\gamma,\tau)$ can be bounded:

$$0 \;\leq E(\gamma,-\tau) \leq 1 \qquad \forall \gamma \text{ not in the stability region and for all } \tau > 0$$

Since $0 \leq E(\gamma,-\tau) \leq 1$ it can be seen from the triangle inequality that

$$\left|E(\gamma,-\tau_{kl}) - E(\gamma,-\hat{\tau}_{kl})\right| \leq \left|E(\gamma,-\tau_{kl})\right| + \left|E(\gamma,-\hat{\tau}_{kl})\right| \leq 2$$

By applying the triangle inequality to Equation (8.5.16) it can now be seen that

$$1 \;< [2|\hat{g}_{max}|\Sigma|p_{\mathrm{il}}| + |\Delta g_{\mathrm{max}}|\Sigma|p_{\mathrm{il}}|] \qquad (8.5.17)$$

and the system will be unstable if

$$1 < 2|\hat{g}_{max}|\Sigma|p_{i1}| + |\Delta g_{max}|\Sigma|p_{i1}|$$

for any value of γ in the unstable region.

Therefore, if the dynamics are modelled exactly ($\Delta g_{kl} = 0$), it can be seen that a necessary condition for the non-ideal process to be stable is that:

$$|\bar{g}_{max}|\Sigma|p_{i1}| \quad < \frac{1}{2} \qquad \forall \text{ k and l} \quad \text{and all } \gamma \text{ in the instability region.}$$

In the slightly more complicated case of the dynamics also containing mismatch then a bound for the amount of mismatch allowable can be constructed:

$$|\Delta g_{max}| \quad < 1/|\Sigma p_{i1}| - 2|g_{max}| \quad \forall \text{ k and l} \quad \text{and } \forall \ \gamma \text{ in the instability region.}$$

It is interesting to note that this bound does not include the difference in the time-delays.

This result shows that it is possible for a process in which some of the elements have modelling errors to remain stable and that bounds exist for the extent to which the elements can differ from the elements of the actual process.

8.6 A Robust Stability Result for the Multivariable Smith Predictor with Additive Modelling Errors

The Smith predictor determines that the control law should be of the form:

$$\mathbf{u}(\gamma) \;= (\mathbf{I}+\mathbf{K}H)^{-1}\mathbf{K} \tag{8.6.1}$$

where

$$\mathbf{H}(\gamma) = (\hat{\mathbf{D}}(\gamma) - \mathbf{E})\mathbf{G}(\gamma) \tag{8.6.2}$$

When this control law is applied, the closed-loop can be described by Figure 8.6.1

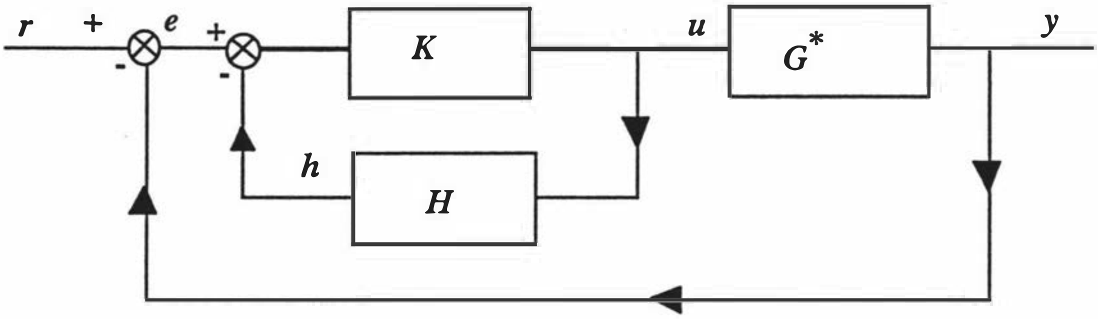

Figure 8.6.1

This block diagram can be rearranged as Figure 8.6.2

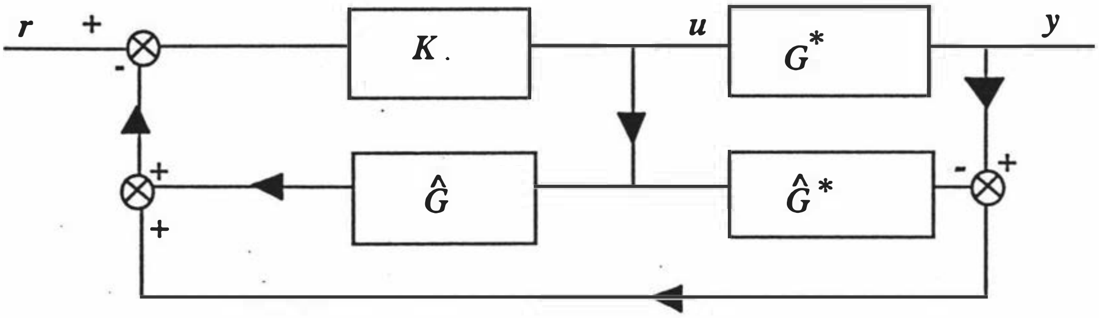

Figure 8.6.2

In this representation it is clear that if the model is exact, the outer-loop, which is the difference between the true output and the expected (modelled) output, $\Delta\mathbf{y}$, is zero and the controller is effectively feed-forward. If there is any noise in the process and the process is open-loop unstable then the Smith predictor scheme will fail to control the process as was shown in Section 8.2.

However, in general the model will not always exactly equal the real process and mismatch will occur. It is of interest to know how much mismatch can exist before the controller fails to stabilise the process. If a controller will continue to stabilise a process even when the process is not modelled exactly, then the system is said to have a degree of *Robust Stability*.

A wide range of literature exists that investigates the robustness of delay-free processes. However, much less is known about the robustness of time-delayed processes. It is known, however, that the traditional Smith predictor can be sensitive to modelling errors. In this section, the robust stability result of Owens and Raya (1982) is extended to include the multivariable Smith predictor. Owens and Raya (1982) investigated the case when the errors were assumed to be additive and the matrix delay-operator, $D(\gamma)$, acted only on the outputs and was a square matrix with non-zero elements only on the main diagonal. The extension to the non-square case is straightforward and relies only on the inclusion of the $\mathbf{E}$ matrix that is used in the construction of the multivariable Smith predictor.

Definition 8.2: *A vector norm is a measure of the size of a vector. It has the following properties:*

$\|\mathbf{x}\| \geq 0$

$\|\mathbf{a}\mathbf{x}\| = |\mathbf{a}|\|\mathbf{x}\|$

$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

There are several commonly used vector norms. These include the Euclidean norm and the absolute norm.

Definition 8.3: The induced matrix operator norm has the properties

$$\|X\| \geq 0$$

$$\|aX\| = |a|\|X\|$$

$$\|X + Y\| \leq \|X\| + \|Y\|$$

For example $\|\cdot\| = \max_{i} \sum_{j} |(.)ij|$ is the matrix norm induced by the vector norm $\|\cdot\| = \max_{i} |(.)i|$

Definition 8.4: A process is Bounded-Input Bounded-Output (BIBO) stable if for any finite initial conditions and for any $\mathbf{u}(t)$ *such that* $\|u(t)\|$ *exists and is finite, the outputs* $\mathbf{y}(t)$ *exist and have finite norm.*

Theorem 8.3: Contraction mapping theorem.

If the mapping T: U $\rightarrow$ U and $\|T(\mathbf{u}) - T(\mathbf{v})\| \leq \|\mathbf{u} - \mathbf{v}\|$ then the mapping has a convergence point.

Proof: See Ortega and Rheinboldt (1970)

By defining U to be the range of possible actuations, **u**, let Y be the space of possible outputs, **y**, and Z to be the space of possible vectors, **z**. Suppose that U_0 is a linear subspace of U,Y_0 is a linear subspace of Y and Z_0 is a linear subspace of Z; these are regarded as spaces of stable inputs, outputs and process-outputs, **z**, respectively. Assuming that these vector spaces have a norm topology with respect to which they are Banach Spaces, it is possible to state the following theorem.

Theorem 8.4: *A multivariable Smith predictor is Bounded-Input Bounded-Output (BIBO) stable if*

(a) *The delay-free plant* $G(\gamma)$ *and its model* $\hat{G}(\gamma)$ *maps* U_0 *into* Z_0

(b) *The delay component* **D**, *its model* $\hat{\mathbf{D}}$ *and* **E** *map* $\mathbf{Z_0}$ *into* $\mathbf{Z_0}$.

(c) *The restriction to* Y_0 *of the delay free mapping*

$$r \to u_a = (\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{Kr}$$

has range in U and has finite induced norm.

(d) $\lambda_1 = \| (\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{K}\Delta D\hat{G} \| < 1$

(e) $\lambda_2 = \frac{1}{1-\lambda_1} \| (\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{K}D\Delta G \| < 1$

Comment: Conditions (a) and (b) are a requirement that the plant and the model are both open-loop stable. Condition (c) requires that the closed-loop model is stable and bounded and finally conditions (d) and (e) provide upper bounds on the additive mismatches of the delays and the delay-free dynamics.

These requirements of stability echo those of the Internal Model Control structure discussed in Chapter 6.

Proof: (Adapted from Owens and Raya, 1982).

By assumption $G(\gamma)$ and $D(\gamma)$ are stable and bounded so it is sufficient to prove that $u \in U_0$ whenever $\mathbf{r} \in Y_0$.

From Figure 8.6.2 it can be seen that

$$\mathbf{u} = \mathbf{K}(\mathbf{r} - \mathbf{E}\hat{G}\mathbf{u} - (DG - \hat{D}\hat{G})\,\mathbf{u}) \qquad (8.6.3)$$

which can be written as

$$\mathbf{u} = (\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{K}[\mathrm{r}\text{-}(DG - \hat{D}\hat{G})\,\mathbf{u}] \qquad (8.6.4)$$

This is an equation in U of the form

$$\mathbf{u} = \mathbf{W_r}(\mathbf{u}) \qquad (8.6.5)$$

Conditions (a) to (c) ensure that $\mathbf{W_r}$ maps U_0 into U_0 for all $\mathbf{r}$ in Y_0.

Now

$$\begin{aligned}\lambda_0 &= \|(\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{K}(DG - \hat{D}\hat{G})\| \\ &= \|(\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{K}(D\Delta G + \Delta D\hat{G})\| \\ &< \|(\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{K}D\Delta G\| + \|(\mathbf{I} + \mathbf{KE}\hat{G})^{-1}\mathbf{K}\Delta D\hat{G}\|\end{aligned}$$

The two absolute values on the right hand side of this inequality can be replaced using assumptions (d) and (e) giving:

$$\begin{aligned}\lambda_0 &< \lambda_2(1\text{-}\lambda_1) + \lambda_1 \\ &< (1\text{-}\lambda_1) + \lambda_1 \;= 1\end{aligned} \qquad (8.6.6)$$

Thus it can be seen that

$$\|W_r(\mathbf{u})\| \leq \|\mathbf{u}\| \qquad (8.6.7)$$

Hence $W_r(\mathbf{u})$ is a contraction mapping of U_0 into itself. By the contractive mapping theorem $W_r(\mathbf{u})$ has only one fixed point. Since the process output $\boldsymbol{G}\mathbf{u}$ is finite (and since $\boldsymbol{G}\mathbf{u} \in Z_0$ from Theorem 8.4, assumption (a)), the process is BIBO as required.

8.7 REVIEW

In this chapter it has been proved that the poles of a Smith predictor controlled time-delayed process can be divided into two sets: the poles of the equivalent delay-free closed-loop process and the poles of the open-loop process. The proof utilises the results of Chapter 3 in order to decompose the transfer function in an appropriate manner. Hence it can be seen that a time-delayed open-loop unstable process cannot be stabilised using a Smith predictor.

Two ways in which modelling errors can be accounted for in process descriptions were examined. It was seen that some robustness results relating to Smith predictors could be derived for inexactly modelled process descriptions when either additive or multiplicative model errors were assumed.

PART D

AN EVAPORATOR: A CASE STUDY OF A TIME-DELAYED PROCESS

9 An Evaporator: A Case Study of a Controlled Multivariable Time-Delayed Process

9.1 Introduction

This chapter describes an evaporator which is an example of a multivariable time-delayed process. This particular evaporator is used to increase the concentration of solids in a stream of whey as part of the process to extract lactose.

The control group at Massey University was invited to explore the possibility of installing a modern control system onto the existing plant to improve the control and hence the efficiency relative to what is currently obtained.

The evaporator studied has a widely varying range of time-delays. These time-delays range up to about 14 minutes, although some of the inputs have an immediate effect.

This chapter reports the first stage of this case study, a simulation analysis of the current controller implementation and a multivariable controller based on a state-space description of the evaporator process. An analysis of the economic advantages of improved control of the evaporator is also provided.

9.2 Plant Description

The evaporator is a five-effect evaporator used for the extraction of water from whey. During the dairy season it processes about 38,000 litres of dilute whey each hour. The density of the whey is increased from about 6% solids to about 41% solids. The whey stream will solidify if the temperature within any part of the evaporator exceeds a threshold. It is known from operator experience that if the final effect temperature is maintained with a set-point of less than 47 degrees, the temperature of the whey stream throught the evaporator does not exceed the solidification threshold.

A schematic representation of the evaporator is presented in Figure 9.2.1. Steam is used as the heat-source to raise steam from the concentrate-flow. The steam raised by evaporation in the first-effect is used to raise steam in the second-effect, which is at a partial vacuum relative to the first-effect. This steam is used to heat the third-effect, which is at a partial vacuum relative to the second-effect. This pattern is repeated in five effects. In addition, to increase the thermal efficiency, some of the concentrate from the second and fifth effects is recycled. The amount of vacuum in the final effect is adjusted by varying the rate at which steam is condensed in the fifth effect. This is done by manipulating the cooling water flow-valve that controls the cooling water flow rate.

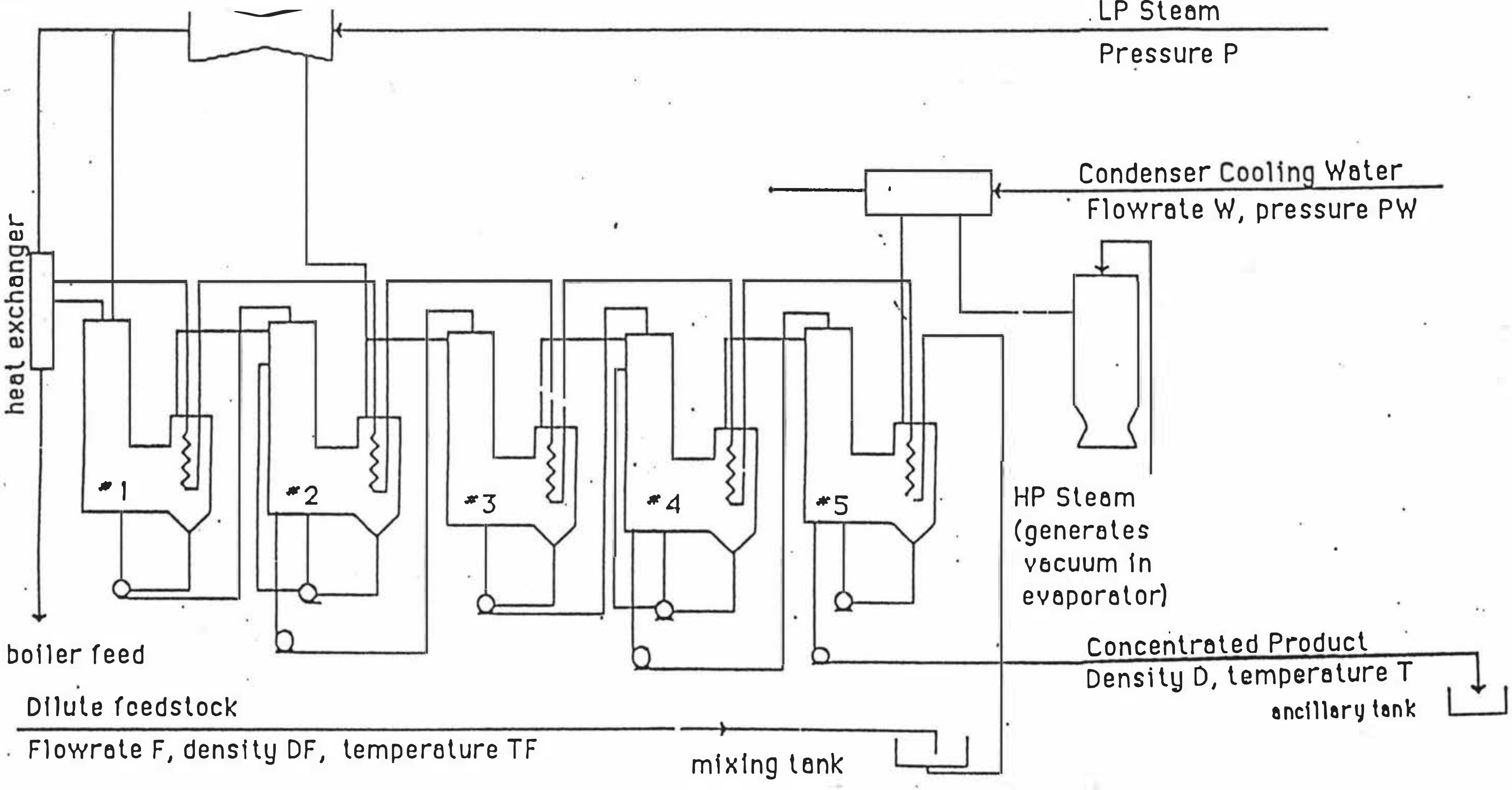
LP Steam
Pressure P
Condenser Cooling Water
Flowrate W, pressure PW
heat exchanger
#1
#2
#3
#4
#5
HP Steam
(generates
vacuum in
evaporator)
boiler feed
Concentrated Product
Density D, temperature T
ancillary tank
Dilute feedstock
Flowrate F, density DF, temperature TF
mixing tank

Figure 9.2.1

The evaporator is controlled by varying the concentrate flow-rate and cooling water flow rate. By increasing the concentrate flow-rate (F) the output density (D) can be decreased and by increasing the cooling water flow rate (W) the final effect temperature (T) can be controlled. However, increasing the concentrate flow-rate will also decrease the final effect temperature and increasing the cooling-water flow-rate will result in less evaporation and hence a lower output-density. Hence the evaporator dynamics have significant cross couplings between the effects of the actuation inputs.

The steam, used as the heat source, is provided on-site from a boiler. However, there are other users of the steam, in particular a steam-turbine driven generator. The steam pressure (P) must therefore be considered a disturbance with a non-zero mean. The incoming density of the concentrate is considered to be a disturbance since the whey arrives at the evaporator from several sources. Since the economic payoff between transportation costs and the cost of concentration differ between the various sources, the whey arrives at different densities. Hence the incoming whey density (DF) is considered a disturbance.

The temperature of the whey (TF) entering the evaporator is also a disturbance since newly arrived whey is significantly warmer that whey that has been stored for any significant length of time in chilled vats.

The cooling water is pumped from a nearby stream. It is assumed that the temperature of this water is reasonably constant. However, the water-pressure (PW) fluctuates with other demands for water being made within the factory.

This plant is a little different to most plants described in the literature, for example Nisenfeld (1985), since most evaporator operators use the steam-pressure as an actuation-input and consider the concentrate flow-rate as the disturbance.

A block diagram form of the evaporator is given in Figure 9.2.2.

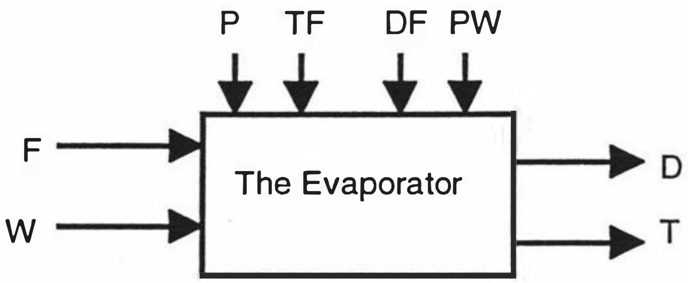

Figure 9.2.2

9.3 THE PRESENT CONTROL MECHANISM INSTALLED ON THE EVAPORATOR

The control scheme currently installed on the evaporator consists of two proportional-loops. One of these loops maintains the density of the out-flowing whey by adjusting the flow rate of the incoming-concentrate at a set-point determined by the operators. The other loop adjusts the cooling-water flowrate to keep the final effect temperature at a set-point. The two setpoints are periodically adjusted up or down by operation staff. These adjustments are based on the operators experience and have not been fully quantified. This controller configuration is portrayed in Figure 9.3.1.

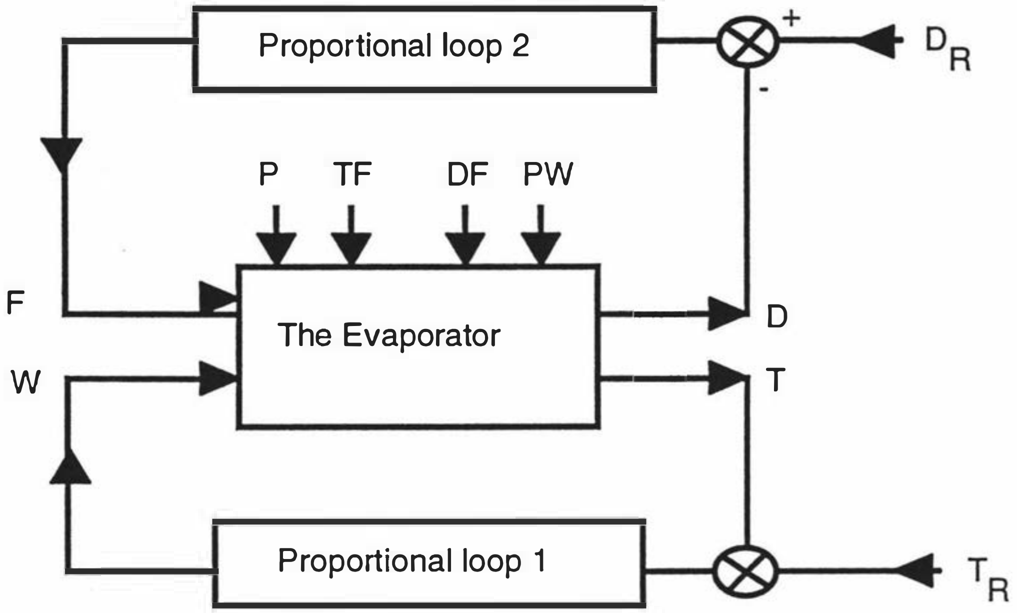

Figure 9.3.1

9.4 Model Identification of the Evaporator

The first stage of the preliminary study was obtaining a model of the process in order to assess the economic viability of implementing a multivariable controller.

It was decided to obtain a time-series model of the evaporator rather than an analytical model. This choice was made for two reasons: firstly, very little was actually known about the heat-transfer coefficients, the evaporation surface dimensions and the pressures inside the evaporator except that the heat-transfer coefficients varied during any evaporation run and, secondly, the slowness in variation in the dynamic response of the evaporator suggested that an adaptive linear multivariable control scheme would be sufficient to control the process.

In order to obtain a time-series model of the process, data was collected from the evaporator. This data-logging was performed using an inter-sample interval of 20 seconds. This sampling rate was chosen to be about 1/10th of the fastest time-constant (thought to be about 3 minutes). As the objective of this stage of the project was to obtain a model of the process running near the current operating set-points, under the existing control scheme, pseudo-random binary perturbations were made around the existing controller's concentrate flow-rate and cooling-water flow-rate actuation signals. Measurements were recorded of the fifth effect temperature (T), the concentrate flow-rate (F), the density of the outflowing concentrate (D), the steam pressure (P), and whether the concentrate flow-rate and cooling-water flow-rate were increased or decreased over the set-point levels decided by the currently implemented controller .

The other disturbances, the cooling-water pressure (PW), the incoming-density (DF), and the incoming-temperature (TF) were not initially considered to have significant effects on the plant so no provision for the on-line measurement of these variables was made. In fact it was during the course of data collection that the importance of these variables was realised and arrangements were made with the operations staff to ensure that these variables remained constant for the data collection time-periods.

Three runs of data were collected from the plant. The measurements were collected with an 8-bit A/D converter. These runs were made when the operators considered the plant to be close to steady state. Each run was approximately 30 minutes long, with data being collected every 20 seconds. In the first run the concentrate flow rate valve position was perturbed +/- 5% while the cooling water flow rate was held constant. In the second run the cooling water flow rate was perturbed +/- 5% and the conentrate flow rate was held constant. In the final run both the concentrate flow rate and the cooling water flow rate were perturbed +/- 5% around their setpoints. In addition during this final run the steam flow to the evaporator was manually perturbed.

During the course of the first run the operators found it necessary to change feed tanks of incoming whey. This whey was at a different temperature and density to that which had been used in the earlier part of the data collection run. The resulting shift of operating conditions for the evaporator meant that steady state was not again reached until well through the second data collection run. Unfortunately this was not realised until the data was analysed. There were no difficulties encountered during the third data collection run. As a result of this major disturbance, data from the first two runs were not used in the modelling process.

9.5 Data Analysis

The raw data from the evaporator was in the range 0 to 255. Appendix A contains the raw data from Run 3. The data collected was initially plotted and graphically analysed. This showed that the major cause of disturbances on the evaporator was variations in the steam pressure and gave some indication of the time-delays. A plot of the run three data is shown in Figure 9.5.1

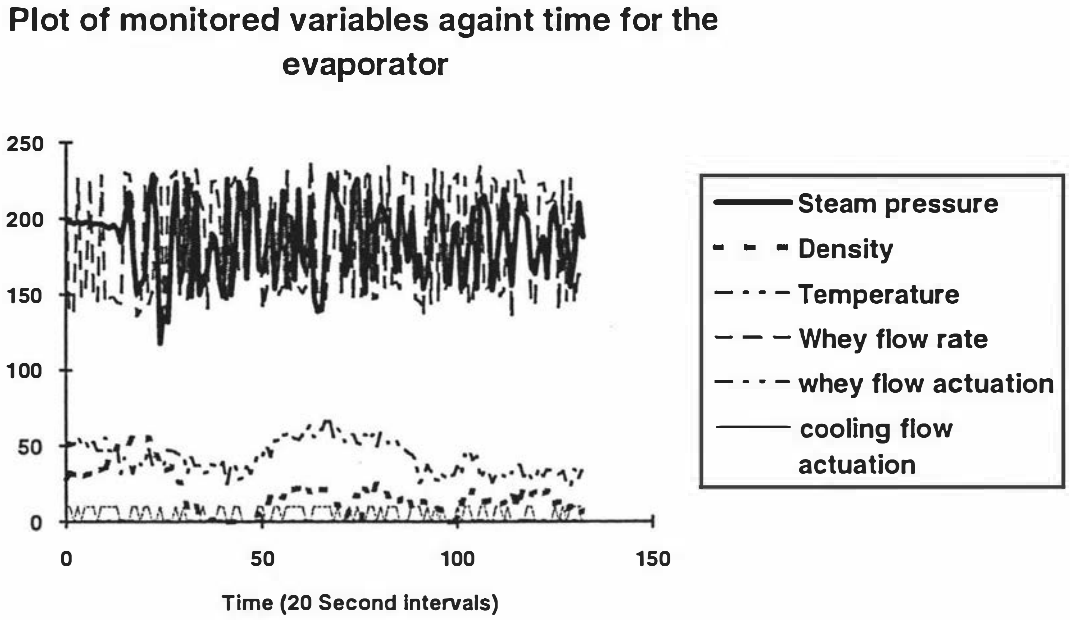

Figure 9.5.1

The power spectrum functions of the steam pressure measurements and the actuation signals were plotted, Figures 9.5.2-9.5.4.

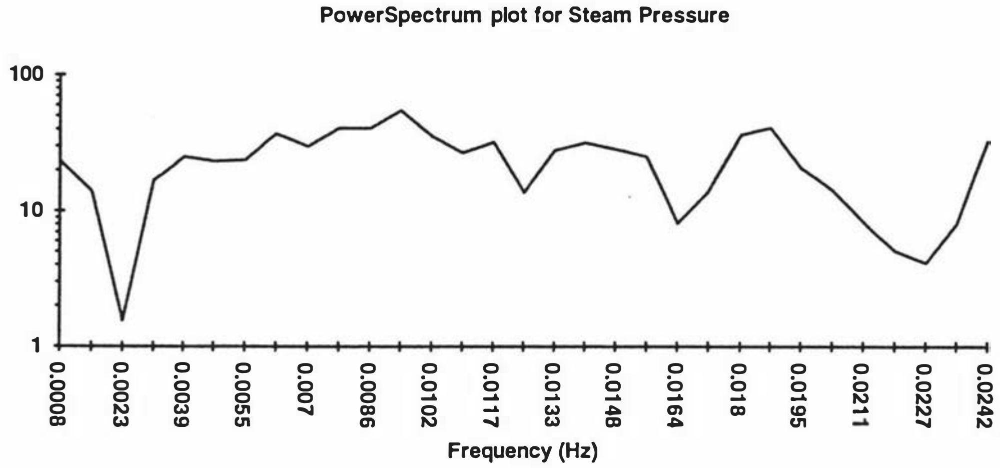

Figure 9.5.2

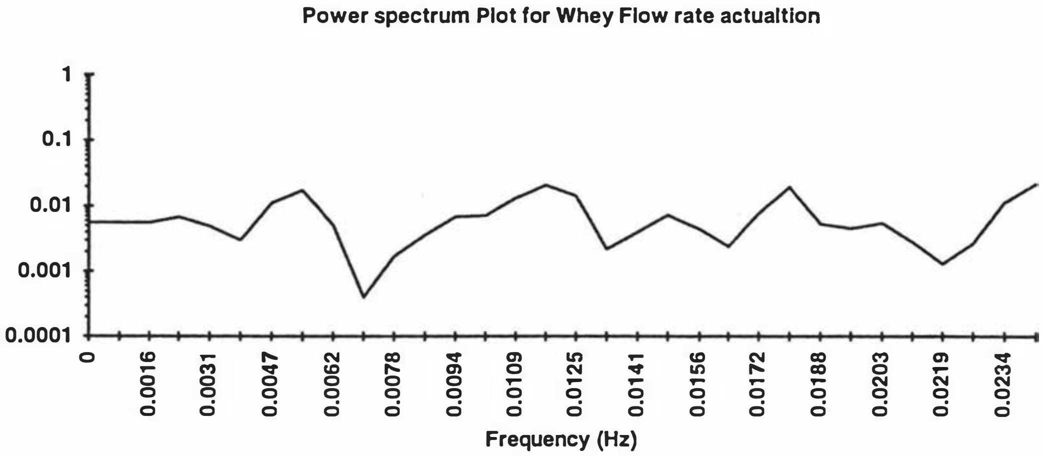

Figure 9.5.3

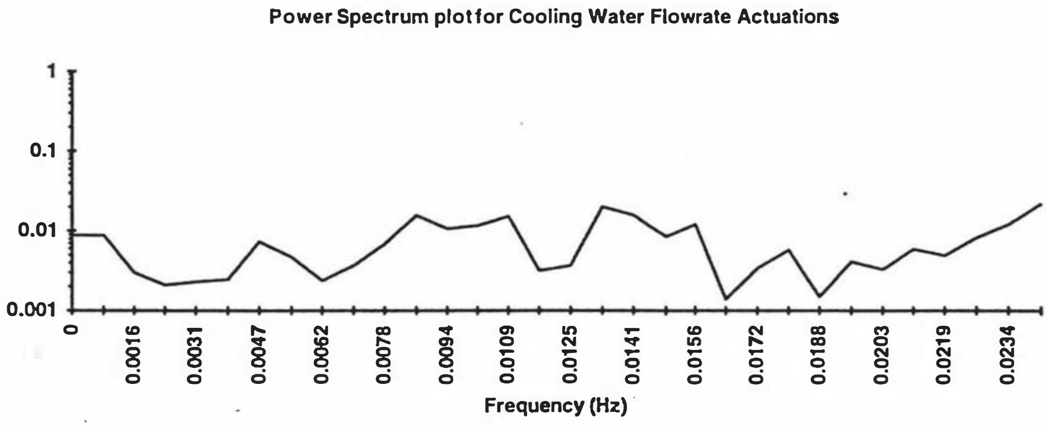

Figure 9.5.4

These plots show that there was sufficient excitation being provided to the process at all frequencies up to 1/2 the sample frequency of 1/20 Hz. This lends confidence to the claim that the evaporator was excited persistently by the input signals over the measurement period.

9.6 PARAMETER ESTIMATION

In order to minimise numerical problems the measurements were scaled and translated so that the range of values in any measurement channel was -100 and 100. The setpoint for each variable was taken to be zero in this transformed space. Table 9.6.1 shows how these ranges relate to the actual process variables.

	Value corresponding to -100	Value corresponding to 100
Temperature	38^{o}C	44^{o}C
Density	1.10kg/kg	1.111kg/kg
Steam pressure	7psi	15psi
Concentrate Flow-rate	5% decrease in valve position	5% increase in valve position
Cooling Water Flow-rate	5% decrease in valve position	5% increase in valve position

Table 9.6.1

9.6.1 ESTIMATION OF TIME DELAYS

Initial estimates of the delays were made using the cross-correlation method, Hannon and Robinson (1973).

The cross-correlation functions for each combination of input and output were calculated and the lag at which the cross-correlations were largest was identified and taken to be the time-delay. These estimates corresponded well with the indications suggested from visual analysis of the time-series. The delays were identified as given in Table 9.6.2

	Density	Temperature
Concentrate Flowrate	5 time-steps (100 sec)	2 time-steps (40 sec)
Cooling water flow rate	2 time-steps (40 sec)	3 time-steps (60 sec)
Steam pressure	1 time-steps (20 sec)	4 time-steps (80 sec)

Time-Delays between Inputs and Outputs in Evaporator

Table 9.6.2

9.6.2 ESTIMATION OF MODEL ORDER

Using the time-delays estimated in Section 9.6.1 using the cross correlation method, estimates were then made of the model order of the process. the model orders were determined using the prediction error method.

Akaike's information criteria (AIC) method, Akaike (1974, 1981) maximizes the *AIC:*

$$AIC = -2\ln(L(\theta, y)) + 2n_p \qquad (9.6.1)$$

where L is the maximum likihood function, θ are the model parameters and n_p is the number of parameters.

When the probability distributions of the observations are gaussian the AIC method is equivalent to minimizing

$$SSE_\theta + 2n_p \qquad (6.6.2)$$

where SSE_θ is the sum of squared errors for the model with parameters θ.

The model order, for each sub-model in the process, was then determined as that which minimizes Equation (9.6.2).

The model was identified as:

$$\begin{bmatrix} A_1(q^{-1}) & 0 \\ 0 & A_2(q^{-1}) \end{bmatrix}\begin{bmatrix} D(t+1) \\ T(t+1) \end{bmatrix} = \begin{pmatrix} H_{11}(q^{-1}) & H_{12}(q^{-1}) \\ H_{21}(q^{-1}) & H_{22}(q^{-1}) \end{pmatrix}\begin{pmatrix} F(t) \\ W(t) \end{pmatrix} + \begin{pmatrix} H_{13}(q^{-1}) \\ H_{23}(q^{-1}) \end{pmatrix} P(t) \tag{9.6.3}$$

where $A_1(q^{-1}) = 1 + 1.232q^{-1} - 0.092q^{-2} - 0.186q^{-3}$ (9.6.4)

$A_2(q^{-1}) = 1 + 0.408q^{-1} -0.248q^{-2} - 0.292q^{-3}$ (9.6.5)

and $H_{11}(q^{-1}) = q^{-5}(-0.028q^{-1} - 0.013q^{-3})$ (9.6.6)

$H_{12}(q^{-1}) = q^{-2}(-0.033q^{-1} + 0.012q^{-2} - 0.004q^{-3})$ (9.6.7)

$H_{13}(q^{-1}) = q^{-2}(-0.080q^{-1} + 0.094q^{-2} + 0.132q^{-3})$ (9.6.8)

$H_{21}(q^{-1}) = q^{-3}(-0.080q^{-1} + 0.094q^{-2} + 0.033q^{-3})$ (9.6.9)

$H_{22}(q^{-1}) = q^{-1}(-0.067q^{-1} - 0.032q^{-2} + 0.021q^{-3})$ (9.6.10)

$H_{23}(q^{-1}) =q^{-4}(0.062q^{-1} + 0.030q^{-2} + 0.091q^{-3})$ (9.6.11)

The R^2 values of the model was 66% for Density and 58% for final effect temperature.

The one-step ahead predictions that this model produced are shown in Figures 9.6.1 and 9.6.2.

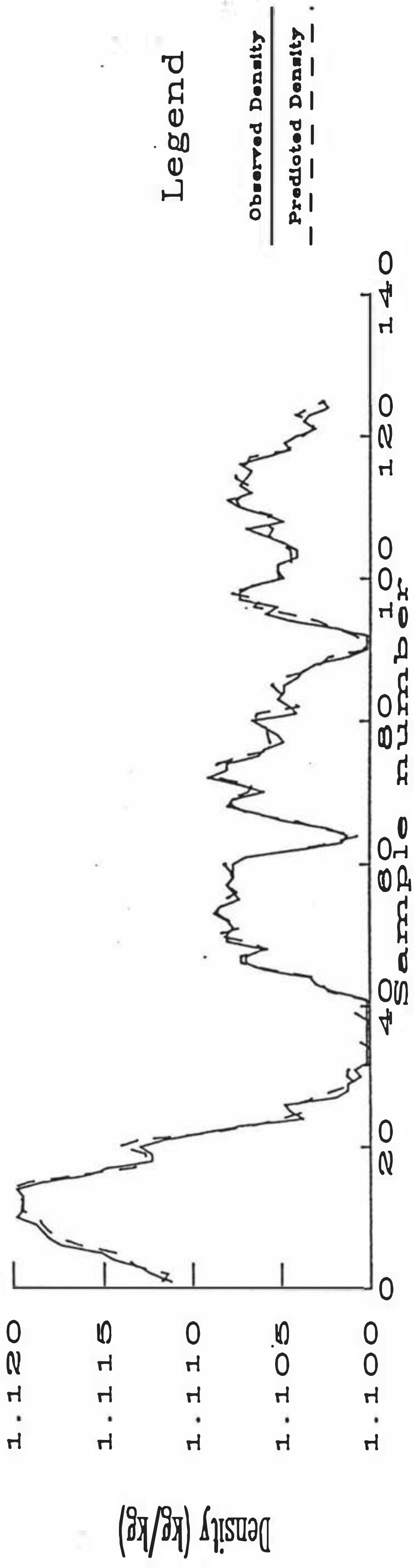
Legend
Observed Density
Predicted Density
0
20
40
60
80
100
120
140
Sample number
1.100
1.105
1.110
1.115
1.120
Density (kg/kg)

Figure 9.6.1

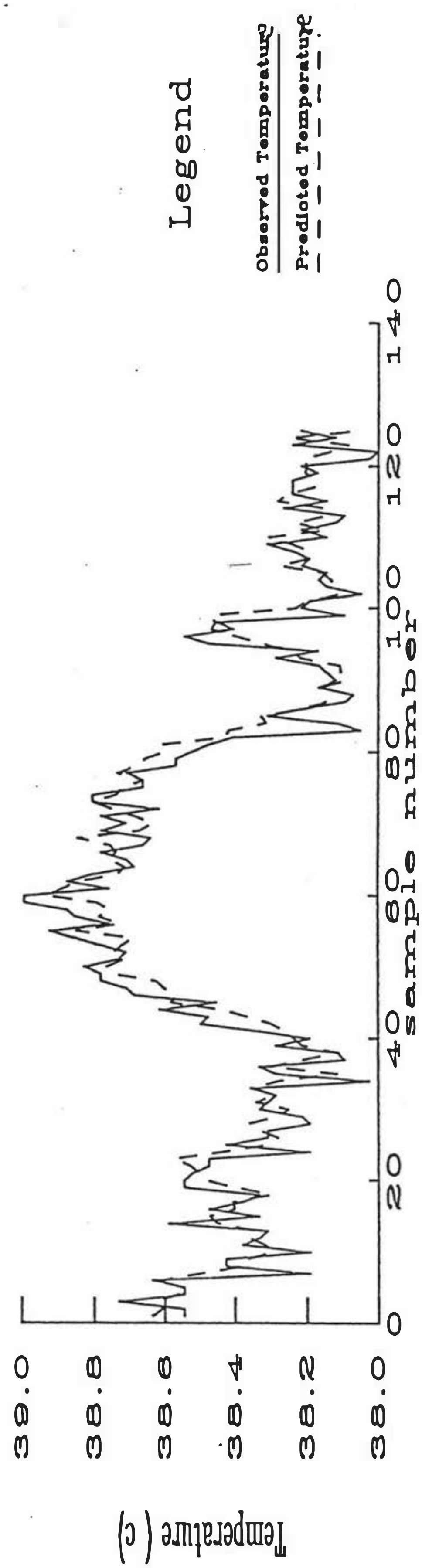
Legend
Observed Temperature
Predicted Temperature
Temperature (c)
39.0
38.8
38.6
38.4
38.2
38.0
0
20
40
60
80
100
120
140
sample number

Figure 9.6.2

This model provided an impression of what the evaporator was like at on the day on which data was collected. This model was then used as a basis for experimentation with different model descriptions and different controllers.

9.7 CONTROL AND SIMULATION

The model, Equations (9.6.3)-(9.6.11), was rewritten in a Non-Minimal State-Space (NMSS) form and used to design several control schemes. The first group of controllers considered were optimal state-feedback controllers using an NMSS description to design and implement the controller. These control schemes were designed to minimise the following quadratic performance index:

$$J = \sum_{k=1}^{\infty} (\, D'(k)\; T'(k)) \begin{pmatrix} q & 0 \\ 0 & q \end{pmatrix} \begin{pmatrix} D'(k) \\ T'(k) \end{pmatrix} + (\, F'(k)\; W'(k)) \begin{pmatrix} F'(k) \\ W'(k) \end{pmatrix} \tag{9.7.1}$$

where s'(k) is the value at the k^{th} interval of the deviation variable of the signal s(k), that is:

$$s'(k) \;= s(k) - \text{steady-state value of } s(k) \tag{9.7.2}$$

Using MATLAB and the model of the evaporator described in Section 9.6, Equations (9.6.3)-(9.6.11), optimal state-feedback controller gain matrices were found for the evaporator for each of three values of q: q=1; q=10 and q=100:

$$\mathbf{u}(t) \;= -\mathbf{K}_q\mathbf{x}(t) \tag{9.7.3}$$

These resulted in closed-loop systems that reacted progressively more quickly to restore the process to steady-state after a disturbance.

In an attempt to simulate a control system similar to that currently used on the evaporator, a control system comprising of two negative-feedback proportional control loops was also designed:

$$F'(k) \quad = -k_1 D'(k) \qquad (9.7.4)$$

$$W'(k)= -k_2 T'(k) \qquad (9.7.5)$$

In order to compare the performance of this control system with that of the optimal system, an attempt was made to determine the values of k_1 and k_2 as near as possible to the same order of magnitude as each other and such that the resulting closed-loop system was stable and had closed-loop poles that were in the same range of magnitude as the closed-loop poles of the optimal designs. The values selected were k_1=0.3 and k_2=0.1.

The positions of the closed-loop poles for each of the optimal controllers and for the proportional control system are presented in Table 9.7.1.

q=1	q=10	q=100	2-P loops	open-loop
0.89	0.65	0.52	0.97, 0.10	0.90
0.76	0.65	0.52	0.95,0.48	0.65
0.71	0.71	0.42	0.66,0.49	0.32
0.32	0.31	0.44	0.45,0.55	0.97
0.55	0.52	0.44	0.45,0.08	0.55
0.55	0.52	0.24	0.45,0.43	0.55
			0.55	

Table 9.7.1

For a particular steam pressure input sequence, the controlled performance, in terms of the standard deviations and maximum and minimum values of each of these signals in the simulated evaporator is shown in Table 9.7.2. In this table the -100 to 100 scale is used.

Control System		Standard Deviation	Minimum	Maximum
q=1	D	5.021	-7.88	10
	T	3.462	-6.55	10.66
	F	2.402	-6.10	5.74
	W	3.387	-3.61	12.70
q=10	D	4.258	-7.06	10
	T	2.639	-5.82	10.66
	F	5.040	-15.74	13.54
	W	8.553	-11.84	47.31
q=100	D	3.774	-6.80	10
	T	2.422	-5.40	10.66
	F	10.700	-28.43	31.75
	W	15.580	-21.01	107.01
2 P Loops	D	5.944	-9.11	10.81
	T	3.994	-4.03	11.26
	F	1.773	-3.24	2.73
	W	0.395	-1.12	0.40
Open-Loop	D	5.871	-9.11	10.98
	T	4.902	-6.50	11.33

Table 9.7.2

Two controllers, an optimal controller with q=10 and the 2 proportional loop controller, were simulated using the same disturbance sequences and the same initial conditions. The results from the simulations are presented in Figure 9.7.3.

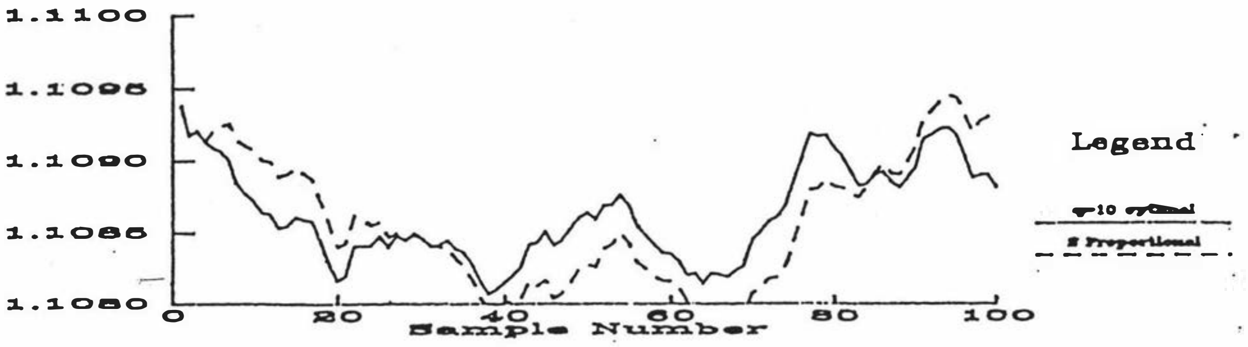

Figure 9.7.3

This graph shows that the density is much less variable using the optimal state feedback control law than when the 2-P loop control law is used. However, this reduction in the variability of the outputs is at the expense of an increase in the magnitude of the signals to the valves controlling the feed flow-rate and cooling-water flow-rate. The increased variance in these were however within operating constraints of the process.

9.8 ADVANTAGES OF IMPROVED CONTROL

The existing controller does not allow large enough gains in actuation signals whereas the multivariable controller could make use of appropriate amount of control action. By reducing the overall amount of steam used in the process, by increasing the amount of concentration that takes place in the evaporator, instead of in down stream stages of the process savings can be made to the process cost. Improved control could include, amongst its advantages a reduction in the amount of cleaning required due to fouling, or baking of the concentrate onto the evaporator walls.

The savings in steam usage can be measured since the thermal efficiencies of the various components of the extraction process are known. The evaporator has a thermal efficiency of about 0.14, that is 0.14kg of steam is used to raise 1kg of steam from the concentrate. This compares well with the thermal efficiencies of about 0.53 for the rest of the plant. Hence if more evaporation takes place in this part of the process than at down stream stages, savings can be made to the total amount of steam used in the extraction of concentrate.

The simulation suggested that the set-point for the out-flowing density could be raised by 2%. This would result in more evaporation taking place in the evaporator instead in down-stream components of the extraction process with a resulting saving in the amount of steam used for heating.

Assuming that the incoming whey flowrate is not changed, then 38000 ℓ of concentrate enter the evaporator at 6% solids. This means that 2280 ℓ of solids enters the evaporator each hour. When the solids exit the evaporator, they make up 41% of the volume. Hence the volume of concentrated whey (and water) exiting the evaporator is about 5561 ℓ/hr. If the solids content is increased to 43%, the outflowing volume decreases to 5302 ℓ/hr. The difference in total volume exiting the evaporator each hour of operation is the increased amount of steam raised in the evaporator, 260 ℓ/hr.

The 2% increase in solids content would therefore save about

260×(0.53-0.14) = 101.4kg/hr of steam used for heating

The dairy company estimates that steam cost about $20 per tonne. Since the evaporator operates 22 hours per day for 250 days of the year. This means that about $11,154 per annum could be saved.

The possible economic gains from improved control are marginal so the cost of implementing any form of improved controller had to be kept to a minimum in order to keep the improvements economically viable. A constraint imposed was to reduce to a minimum the software and hardware supplied by Massey University in order to reduce the development time. In order to further reduce the risks commercially available equipment and software was to be used whereever possible since these commercially available products have been well tested and are known to be reliable. Since the dairy company operating the evaporator had a PC and a suitable interface card already installed on the evaporator that was in use .only when the evaporator was in a cleaning cycle, the remaining 2 hours of each day, it was decided to implement a controller using the same PC.

In addition using a PC with good software would allow the easy exploration of some possible control strategies and controller designs.

9.9 REVIEW

In this chapter a model of an evaporator has been identified. The process which can be modelled as a discrete-time multivariable time-delayed system is currently controlled using two proportional control loops.

Using the model of the evaporator, a simulation study suggested that a multivariable controller could be applied to the evaporator and that financial savings could be made from improved control of the evaporator.

The quantifiable savings from improved usage of steam are small. However additional benefits such as improved knowledge of the process dynamics and reduced fouling of the evaporation surfaces make the project attractive to the dairy company.

10 Implementation of a Self-Tuning Controller for an Evaporator

10.1 INTRODUCTION

The model obtained in the first part of this case-study, Chapter 9, provides an impression of what the evaporator was like on a particular day. However, the evaporator dynamics are slowly changing as concentrate builds up on the evaporation surfaces. This fouling causes the heat-transfer coefficients to change as well as the flow-rate of concentrate through the tubes of the evaporator. Hence it was suggested that an adaptive control law should be implemented on this process. This chapter investigates several aspects of a proposed implementation of a self-tuning controller on the evaporator. This implementation was never tested on the process for two reasons; the dairy season finished before the controller could be implemented and evaluated, hence there was no whey to be evaporated, and the dairy company operating the evaporator then reviewed its development policy and decided to improve the steam pressure control, resulting in much better performance from the evaporator without the need for multivariable control of the evaporator. The controller was therefore only tested in simulation (Section 10.5)

Section 10.2 contains a brief review of some of the aspects of self-tuning control is presented.

Section 10.3 contains a description of the hardware and software that was selected for an implementation and the constraints and limitations that are imposed on the implementation by this choice of hardware and software.

In Sections 10.4 a framework for the control of the evaporator is described.

In Section 10.5 the model obtained in the first part of the project is used to simulate the plant operating under a self-tuning controller based on the non-minimal state-space (NMSS) description and optimal control. The results of this simulation show that a self-tuning controller is viable, at least on the simulated plant, on the grounds that: the time taken for the estimation and controller calculations was less than the inter-sample time-step; the coefficients converged; the self-tuning controller converges and the process variability is reduced.

10.2 A LITERATURE REVIEW OF SELF-TUNING CONTROL

Self-tuning control was first suggested by Kalman (1958) who developed an algorithm for controlling a plant where only the inputs, outputs, model-orders and time-delays were known. He constructed a hybrid computer, part analog and part digital, that used a recursive least-squares procedure to estimate the parameters of the process and implemented the control using a dead-beat control design law. The lack of powerful computers at the time and the lack of any convergence proofs meant that this method was severely limited.

The invention of the microprocessor in the 1970's provided the required technological impetus that resulted in practical self-tuning controllers.

Good surveys of in this field are found in Astrom, Borison, Ljung and Wittenmark (1977), Harris and Billings (1981), Astrom (1983), Astrom and Wittenmark (1984), Goodwin and Sin (1984) and Toivonnen (1984).

The property that the controller will converge to the controller that would have been obtained if the true process description had been known *a priori* was called the self-tuning property by Astrom and Wittenmark (1973). This is the property that differentiates self-tuning control from other forms of adaptive control.

The analysis of self-tuning controller systems is usually restricted to time-invariant systems. It is common to assume that an upper-bound on the order of the process is known and that something is known about the disturbances that affect the process.

A number of multivariable adaptive controllers have been presented in the literature.

Borison (1979) produced a multivariable minimum-variance self-tuning controller. This controller presumed the delays in each loop were equal.

Koivo (1980) extended this method to the Clarke-Gawthrop controller. Again all the delays were assumed to be equal. Keviczky and Kumar (1981) produced a generalisation of the Clarke-Gawthrop controller that allowed for a general $\boldsymbol{B}(q^{-1})$ polynomial matrix subject to $\boldsymbol{B}(0)$ being non-singular.

Prager and Wellstead (1980) gave a multivariable pole-assignment self-tuning regulator. This controller also assumes that the delays are all equal.

The self-tuning methods mentioned above are based around a polynomial matrix description of the process. Another common method of writing process descriptions is the state-space technique. There are many state-space descriptions. One type of state-space description is the Non-Minimal State-Space (NMSS) description discussed in Chapter 5. This description is of interest since, unlike many state-space descriptions, it does not require an observer or filter to be implemented in order that the state of the process be obtained from input-output data.

Peterka and Astrom (1973) produced a state-space adaptive controller that made use of a form of non-minimal state space (NMSS) description and used an optimal state feedback controller design law.

Hesketh (1981, 1982) used a NMSS description but used pole-placement as the controller design technique. His control law is found by a transformation of the coordinate system used to describe the state-vector. The inverse transform returns the control law to the required coordinate space. This method produces controllers that are the same as those obtained in Prager and Wellstead's multivariable controller, Prager and Wellstead, (1980).

A particular NMSS form has been used by Young, Behzadi and Chotai (1987), Wang and Young (1988a, 1988b), Chotai and Young (1988), Young Behzadi and Chotai (1988) and Cox, Boucher and Young (1988). These explorations have added various extensions to their NMSS description, in particular Proportional Integral Plus (PIP) control that involves adding an integral term to the NMSS description.

Identifying the parameters of a process while it is in closed loop is a problem associated with adaptive and self tuning controllers. The problem arises when a feedback control law is applied to a process and the coefficients of the process are estimated since the inputs and outputs are correlated. Soderstrom, Gustavsson and Ljung (1975), Ljung, Gustavsson and Soderstrom (1974) and Gustavsson, Ljung and Soderstrom (1977) give a survey of the solutions to this problem.

In Section 10.2.1 methods which have been used to determine model orders and time-delays are reviewed. In Section 10.2.2 methods for model coefficient estimation are reviewed. Some of the methods used to calculate the controller gain matrix are considered in Section 10.2.3.

10.2.1 ESTIMATES OF THE MODEL ORDERS AND TIME-DELAYS

Good estimates of the system parameters are needed to obtain good models of the process. If the estimates of the orders are too small then not all the dynamics of the process are found. If the estimates are too large then there are two problems: the amount of calculation is vastly increased and the algorithm may not converge (Ljung, 1977a, 1977b).

There are many tools for estimating the model orders. The most common way of selecting the model orders is based on the method of Akaike (Akaike 1981). The cost function that is minimised in this technique is a modification of that of maximum likelihood that includes a cost of increased model order. The technique used is to increase the order of a proposed model, calculating the cost function for each proposed model order. The true model order has been found when no significant decrease in the cost function is obtained by increasing the order any further.

There are fewer results to assist in the estimation of the delays of a process. One method proposed is to use Akaike's criterion in a similar manner to that used to find the order of a model, Mancher and Hensel (1985).

Another method mentioned in the literature, for example Hannon and Robinson (1973) and Hamon and Hannon (1974), is to use the cross-correlation function between the input and output. The pure delay is claimed to be the lag at which the maximum absolute value of the cross-correlation function is attained.

10.2.2 Model Estimation

The most common technique for model identification is the method of least squares. Other methods for identification include maximum likelihood, method of instrumental variables and the method of recursive maximum likelihood stochastic approximation and recursive least-squares method, the differences between these methods are the criteria used to select the best estimates of the set of coefficients. These methods can be considered in a common frame-work with much of the analysis being the same.

The least squares method has been used for estimating model coefficients since it was developed by Gauss in 1809 [Ljung and Soderstrom 1983, p12-63]. The least squares estimator minimises the cost function

$$J = \sum_{t=0}^{t_{fin}} w_t (y(t) - \hat{y}(t))^2 \qquad (10.2.1)$$

where w_t is a sequence of weightings and $\hat{y}(t)$ is the estimate of the output, y(t), made at time t. In the simplest implementation, all the weightings are the same,$w_t = 1 \; \forall \; t$.

The traditional least-squares method is best suited for offline analysis. A recursive form of the least squares method was developed by Plackett [Ljung and Soderstrom, 1983, p12-63]. If a linear model, $\hat{y}(t) = \Theta(t)^T\mathbf{X}(t)$, is used to describe the process, the recursive least-squares algorithm can be stated in the form,

$$\Theta(t) = \Theta(t-1) + \mathbf{K}(t)(\mathbf{y}(t)-\Theta'(t-1)\mathbf{X}(t)) \tag{10.2.2}$$

$$\mathbf{K}(t) = \frac{\mathbf{P}(t-1)\Theta(t)}{\alpha(t)^2+\Theta(t)^T\mathbf{P}(t-1)\Theta(t)} \tag{10.2.3}$$

and

$$\mathbf{P}(t) = \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\Theta(t)\Theta(t)\mathbf{P}(t-1)}{\alpha(t)^2+\Theta(t)^T\mathbf{P}(t-1)\Theta(t)} \tag{10.2.4}$$

where $\alpha(t)$ is a forgetting factor.

The forgetting factor, $\alpha(t)$, is related to the weightings w_t. If all the weightings are the same then $\alpha(t) = 1$. By changing the weightings, w_t, on the sum of squares cost function, it is possible to increase the relative weighting on newer data and correspondingly reduce the weighting on older data. There are several ways in which this can be performed. Exponential forgetting occurs if the weightings decrease geometrically with age. In this case $\alpha(t) = \alpha$ for $0<\alpha<1$. Alternatively if the weightings do not change with age except for a very low weighting on early data a forgetting factor in which $\alpha(t)$ increases to unity could be employed. Soderstrom, Gustavsson and Ljung (1975) suggest that a good choice of forgetting factor is $\alpha(t) = \alpha_0.\alpha(t-1) +(1- \alpha_0)$ with typical values of $\alpha(0)=0.95$ and $\alpha(0)=0.99$.

Another method of improving the performance of the recursive least squares (RLS) algorithm is to insert a covariance resetting or modification step into the algorithm (Goodwin and Sin, 1984 p65-68). This is done because while the RLS algorithm has large initial gain, **K**, the gain reduces dramatically in magnitude when the error covariance matrix becomes small. The covariance matrix is then reset to force the gain matrix to become larger. This commonly happens after 10-20 iteration steps.

Good summaries and derivations of the different methods of on-line, or recursive, model identification methods are given in Ljung and Soderstrom (1983, p16-61) and Goodwin and Sin (1984, p47-105 and p301-359).

Comparisons of different variations of the recursive least squares estimator and other on-line identification methods are given in Isermann, Baur, Bamberger, Kneppo and Siebert (1974), Saridis (1974) and Sen and Sinha (1975).

10.2.3 Persistent Excitation

In order for the recursive least squares algorithm, Equations (10.2.2)-(10.2.4) to converge it is necessary that the process receives sufficient excitation in the input sequence that the matrix $\Theta^T\Theta$ is nonsingular. Care must therefore be taken to ensure that the inputs to the process are persistently exciting or have sufficient richness to ensure that the covariance matrix, $\Theta^T\Theta$ is nonsingular. If the inputs are not persistently exciting the process, perturbations could be added to the process to ensure that the inputs are sufficiently rich.

The literature includes various results that ensure persistent excitation, For example Middleton and Goodwin (1990, p364) Goodwin and Sin (1984, p72).

10.2.4 Updating The Controller Gain Matrix

In discrete-time the time-invariant regulator problem can be formulated as

$$\min_{\mathbf{u}(t)} \quad J = \sum_{0}^{\infty} \mathbf{x}(t)^t\mathbf{Q}\mathbf{x}(t) + \mathbf{u}(t)^t\mathbf{R}\mathbf{u}(t) \qquad (10.2.5)$$

such that

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \qquad (10.2.6)$$

The dynamic programming approach of Kalman and Koepeke is reviewed in Kwakernaak and Sivan (1972 p 492-494). This version of the solution to the discrete time optimal regulator problem gives rise to the set of equations.

$$\mathbf{F}(t) = (\mathbf{R} + \mathbf{B}'(\mathbf{Q}+\mathbf{P}(t))\mathbf{B})^{-1}\mathbf{B}'(\mathbf{Q}+\mathbf{P}(t))\mathbf{A} \qquad (10.2.7)$$

$$\mathbf{P}(t\text{-}1) = \mathbf{A}[\mathbf{Q}+\mathbf{P}(t)][\mathbf{A}\text{-}\mathbf{B}\mathbf{F}] \qquad (10.2.8)$$

which are solved recursively from the finishing time. from which the optimal **u**(t) can be calculated. It can be shown (see Goodwin and Sin, 1984, p 513-515) that the steady-state solution can be obtained using the recursive approach and that this method will converge exponentially to the solution of the steady-state Riccati equation.

Another approach is to use the techniques of constrained optimisation. This can give rise to reformulation of the problem of solving the Riccati equation, as a generalised eigenvalue problem. Potter [Pappas, Laub and Sandell (1980)] showed that the continuous time optimal regulator can also be formulated in this way. If the transition matrix is non-singular then the set of equations that arise from applying the method of Lagrangian multipliers to the constrained optimisation problem can be rearranged as an eigenvalue and eigenvector problem, as shown by Vaughan (1970).

Laub (1979) shows that the Riccati equation can be solved in terms of Schur vectors.

Pappas, Laub and Sandell (1980) present the more general case in which it is not assumed that the transition matrix is invertible. They give two methods, one using generalised eigenvectors, the other using generalised Schur vectors.

Shieh Tsay and Yates (1983) show that the matrix sign function can be used to solve the discrete time Riccati equation using a method similar to that of Vaughan (1970). The matrix sign function can be viewed as a mapping of the eigenvalues to +/-1 but preserving the eigenvectors. There are several recursive algorithms for the estimation of the matrix sign function.

10.3 Implementation Considerations

In this section the hardware and software components of the existing and the proposed controllers are discussed.

10.3.1 The Modicon and the Ero Controllers

The evaporator is currently controlled using two controllers, a Modicon and an Ero. The Ero is used to regulate the cooling water flowrate. The Modicon is used to control the whey flowrate. Each of these controllers accepts 4-20mA signals from the various sensors. Both these controllers are typical industrial controllers and neither has extensive numerical capability. The lack of extensive floating-point arithmetic means that any multivariable controller could not be implemented using these controllers to perform the calculations. However, there is some spare input-output capacity that would enable the Modicon to act as an intermediary between a more advanced controller and the process.

10.3.2 Measurements Available on the Process

The evaporator, as described in Section 9.2 has nine variables that can be meaured. In addition to the measurements of the two inputs, W and F, and the two outputs, D and T, there are on the evaporator a number of other measurements available. These include: Water pressure (PW); water-temperature (WT), incoming whey-temperature (TF); incoming whey-density, (DF) and steam pressure (P). Except for the incoming whey-temperature, which is a 0-1v signal, all the signals are available from 4-20mA loops.

10.3.3 The Hardware available for Multivariable Process Control

The dairy company operating the evaporator have installed in the control-room of the evaporator an IBM-AT personal computer (PC). This computer was installed to assist in the management of the cleaning cycle of the evaporator and hence is available when the evaporator is not in a cleaning cycle. Attached to the PC is a Keithley interface device. This interface device consists of:

up to 16 Analog-digital conversion channels with 12 bit accuracy;
a 2 channel 12 bit digital-analog converter;
32 bit digital input
and 32 bit digital output.

The analog-digital (A/D) conversion channels can be selected to be 16 channels in the ranges 0-5v, 0-10v, or 8 channels in the range ±5v or ±10v. By tapping into the 4-20mA signals using a 250Ω resister, the signals can be interpreted as 1-5v signals by the A/D conversion hardware in the Keithley.

10.3.4 Software for the Multivariable Controller

The PC uses MS-DOS version 3.3, a standard operating system. This allows a number of possibilities for the software used to implement a multivariable controller on the plant. However the cost of purchase and implementation of many of these packages effectively eliminated them from consideration.

The selection of software was further restricted by two factors. MS-DOS is not a multitasking operating system. Hence any task-scheduling, such as making measurements and dispatching actuations at evenly spaced time-intervals, must be performed by software provided by the user, rather than either by the operating system itself or by an autonomous process with its own reliable timing method. The second factor was the desire to minimise the amount of programming that was involved in the project. The tasks that the software had to perform could be separated into two groups: the algorithm for calculating the actuations and the interface between this algorithm and the rest of the plant. It was decided to use MATLAB for the algorithm portion of the software.

MATLAB is a matrix manipulation package available for a wide range of computer architectures, including the IBM personal computer. It is a well tested, fast and robust toolbox of matrix and vector routines that can be used either interactively or, by writing either in MATLAB script files or by linking either FORTRAN or C code. Using MATLAB it is possible to very quickly code an algorithm for the calculation of the actuations and since MATLAB is interactive, it is possible to modify this code relatively simply. Since MATLAB was not restricted to only being run on the PC attached to the evaporator but could be run on any compatible computer, it was possible to run simulation tests of the evaporator control software without any possibility of upsetting the operation of the evaporator.

However, MATLAB is intended to be used as a design tool rather than for real-time implementation. Hence the only standard methods of data-transfer to and from MATLAB are via the keyboard, the screen and the hard-disk. The other major drawback of using MATLAB is that it does not contain any form of time-management to ensure that the time-steps are maintained to be of equal-length.

A simple, but ingenious, solution to these two drawbacks of using MATLAB was devised. This solution was based around a Terminate and Stay Resident (TSR) program. This program takes over control of the computer once every 18 milliseconds, scanning each input measurement and then returning control to the program that was running. Every 20 seconds as well as scanning the input measurements, the TSR program averages the measurements that have been made and places the resulting mean values in the keyboard buffer. It then collects the actuations that the foreground program, MATLAB, has placed in the screen buffer and reads a status-flag (one of the digital inputs). If the status-flag informs the TSR program that the multivariable controller is currently in charge of the evaporator, it then sends the actuations to the digital-analog converters and then returns control of the PC to the foreground program, MATLAB.

This TSR program successfully overcomes the limitations of MATLAB since the combination of the TSR program and a MATLAB program that repeatedly reads input from the keyboard, calculates the actuations and displays the actuations on the screen. The TSR program ensures that the intersample interval is a constant 20 seconds provided that MATLAB completes its calculations within that time-period. Using this combination it is therefore possibly, relatively cheaply, to implement a multivariable controller on a PC. Simulation studies showed that MATLAB required 12-16 seconds to calculate the actuations, well within the limit of 20 seconds.

A listing of the Pascal code for the TSR is presented in Appendix B. A listing of the MATLAB code is presented in Appendix C.

10.3.5 Proposed Mode of Operation

It was decided that any multivariable controller implemented on the evaporator would be positioned in parallel with any existing control scheme. Paramount among the reasons were:

- Personal computers compared to ruggardised industrial controllers such as the Ero and Modicon are relatively fragile and prone to hardware failures such as Hard-disk crashes, power failures, overheating, and contamination of electronics due to dust particles.
- The multivariable controller proposed was designed to operate near the stable steady state. Hence another controller had to be available when the evaporator was started up.

- For financial reasons the existing controllers were to act as the interface between the multivariable controller, on the PC, and the evaporator.
- Since the multivariable controller proposed was outside the operating experience of the operations or technical staff of the dairy company, it was felt that the operators should have a fall-back position of the existing control scheme.

In order to satisfy the requirement that the multivariable controller coexisted with the existing plant it was decided that the process hardware would be arranged so that all the actuations were transmitted to the evaporator via the Modicon that, at all times, continued to calculate the control actions that it would transmit to the evaporator if it were controlling the process. These actuations could then be used if for some reason the multivariable controller ceased to function. Figure 10.3.1 displays the data-flow paths between the controllers and the evaporator when the multivariable controller is controlling the evaporator.

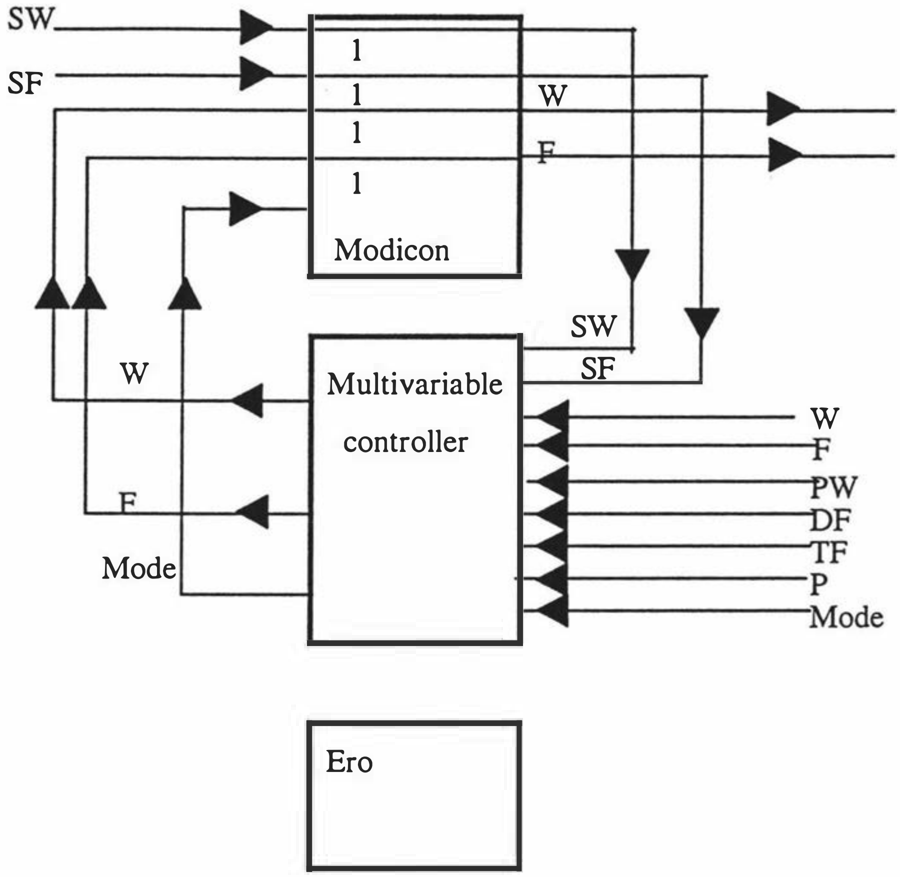

Figure 10.3.1

Figure 10.3.2 shows the data-flow paths when the Modicon and Ero calculates the actuations to be used to control the evaporator.

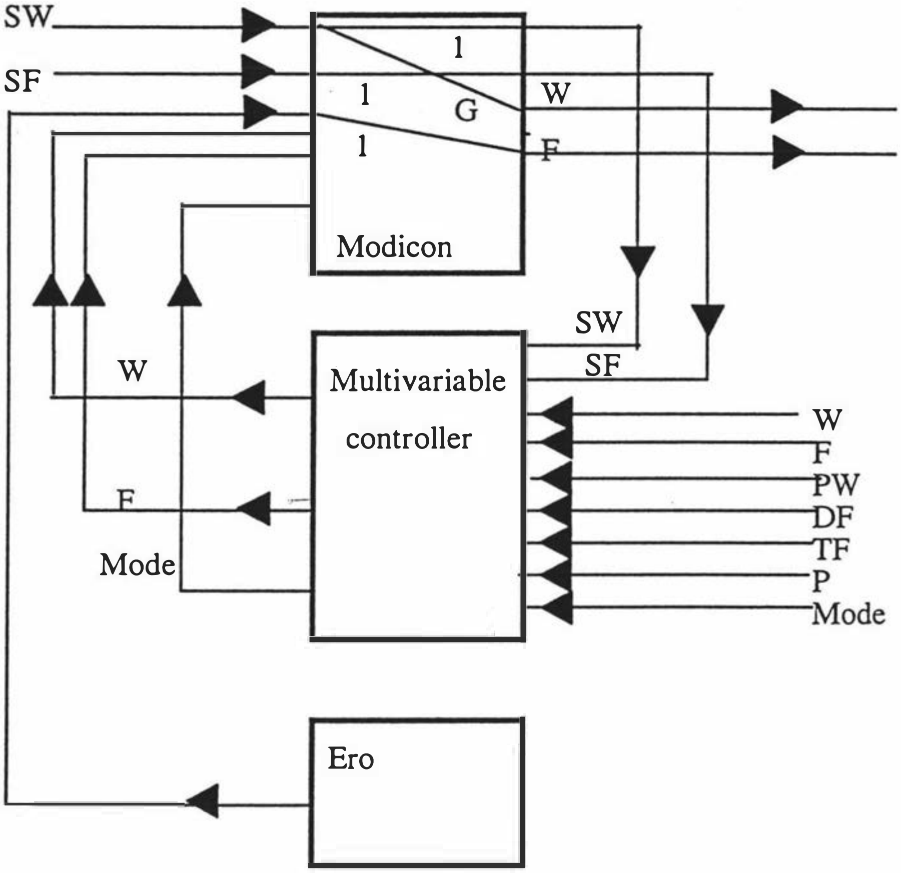

Figure 10.3.2

When the TSR program and MATLAB were running, there was no possible way of informing the software whether the existing controllers or the multivariable controllers was in fact controlling the evaporator since normally all communication to MATLAB, in which the multivariable software was written, passed through the keyboard that was being used to receive measurements from the Keithley interface device. The solution implemented involved the addition of a status-flag that was read by the TSR program and relayed to the MATLAB program and to the existing controller. The status-flag had four possible states. These states are listed in Table 10.3.1.

State	Multivariable Controller	Existing Controller
0	Watch	Control
1	Learn	Control
2	Learn and Control	Supervise
3	Control only	Supervise
4	Save	Supervise

Table 10.3.1

The Modicon received a binary flag: 0 if the Modicon was to control the evaporator or a 1 if the multivariable controller was in control, in which case the Modicon was in a supervising mode. If for some reason the PC and the Keithley failed to function, this flag would default to the 0 state and the Modicon and Ero controllers would resume control of the evaporator. The multivariable controller also received a 3-bit flag that informed it of the operational mode of the process. These modes were included so that the model estimation routine of the multivariable controller could be disabled, or the current model saved.

No modifications to the program of the Ero were required. The Modicon program alterations were trivial.

10.4 Algorithm Used to Implement a Self-Tuning Controller on the Evaporator

It was decided to implement the multivariable controller using a NMSS description. This choice was made since the state-vector of a NMSS description consists of the previous input output record of the process. However, in this implementation the actual dynamics of the process are not known and hence the model and the actual process are possibly mismatched. Moreover, only approximations to the time-delays are known and some of these time-delays change with the flowrates of the cooling-water and the whey. The results of Section 8.3 suggested that if a Smith predictor was implemented, large derivative-like action could result which, on the largely unknown process was undesirable.

The uncertainty in the model of the evaporator, especially the uncertainty in the estimates of the time-delays suggested that a Smith predictor should not be explicitly implemented, hence it was proposed that a controller based on a NMSS description, such as that described in Section 5.2 would be implemented.

It was also decided to implement a self-tuning controller. This choice was made since the process dynamics, while largely unknown and slowly time-varying, seemed to be sufficiently slow in changing that a self-tuning controller implemented using the PC could control the process successfully.

The control algorithm could thus be summarised as:

1) Given the model orders and the time-delays
2) Collect new measurements
3) Update process model
4) Recalculate controller gain matrix
5) Calculate new actuations
6) Dispatch actuations
7) Go to step 2.

A variety of different methods are available that could be utilised in stages 1, 3 and 4 of the overall algorithm proposed above. These various options were discussed in the Sections 10.2.1, 10.2.2 and 10.2.3.

10.5 Simulation of a Self-Tuning Controller on the Evaporator

When the evaporator was simulated using MATLAB the graphs shown in Figures 10.5.1-10.5.6 were obtained. In this simulation the self-tuning controller was placed in a learning mode for the first 100 time-steps during which PRB signals were added to a predetermined actuation signal to the evaporator. After 100 time-steps the actuations produced by the multivariable self-tuning controller were used to control the process.

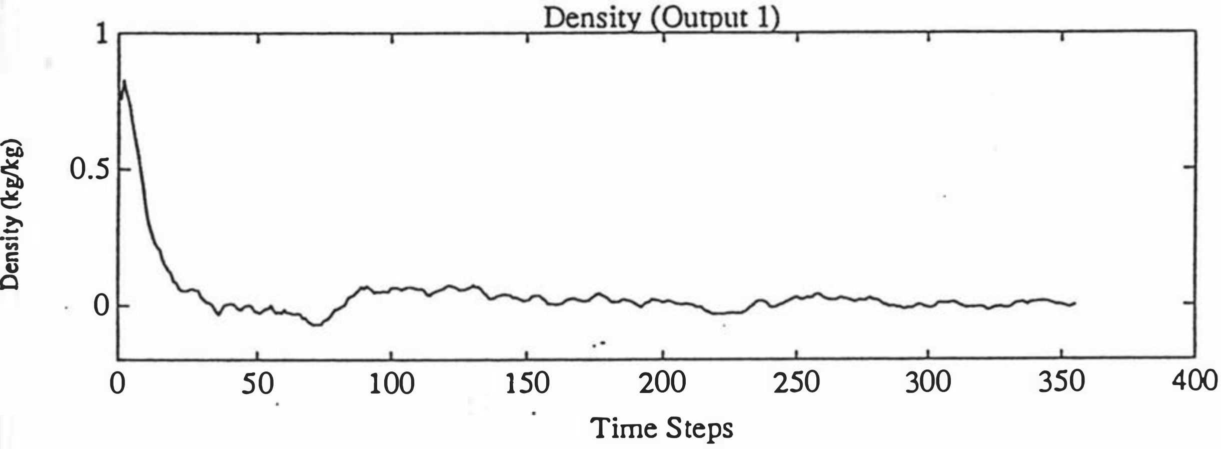

Figure 10.5.1

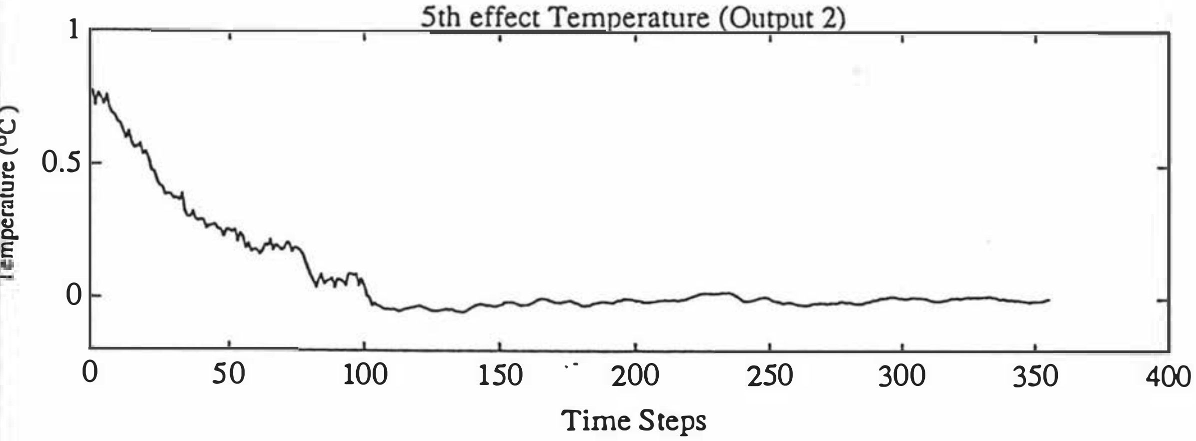

Figure 10.5.2

The plots of the two outputs, D and T against time are shown in Figures 10.5.1 and 10.5.2. It can be seen from these two graphs that the deviation from the setpoint of the temperature and the density decreased. In this particular simulation it can be seen that the temperature is especially well controlled by the self-tuning controller.

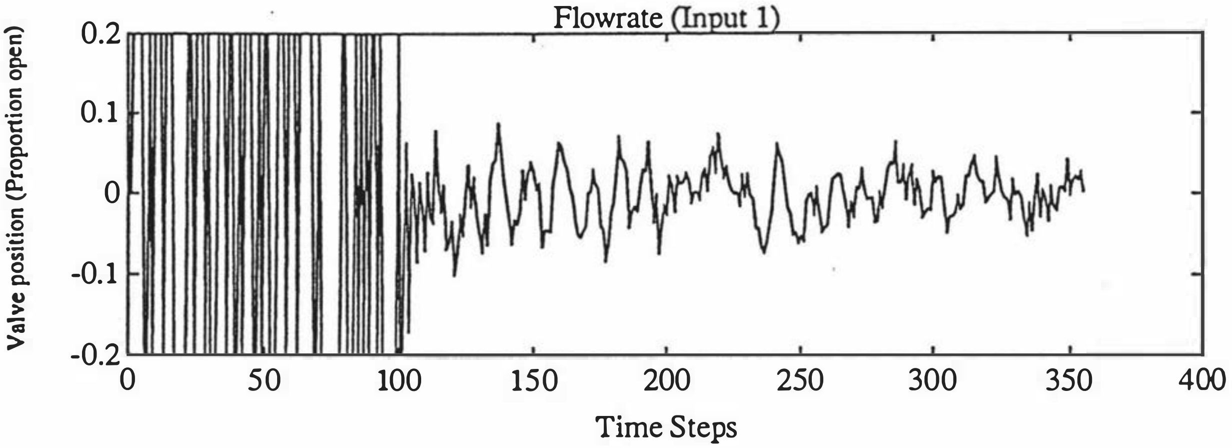

Figure 10.5.3

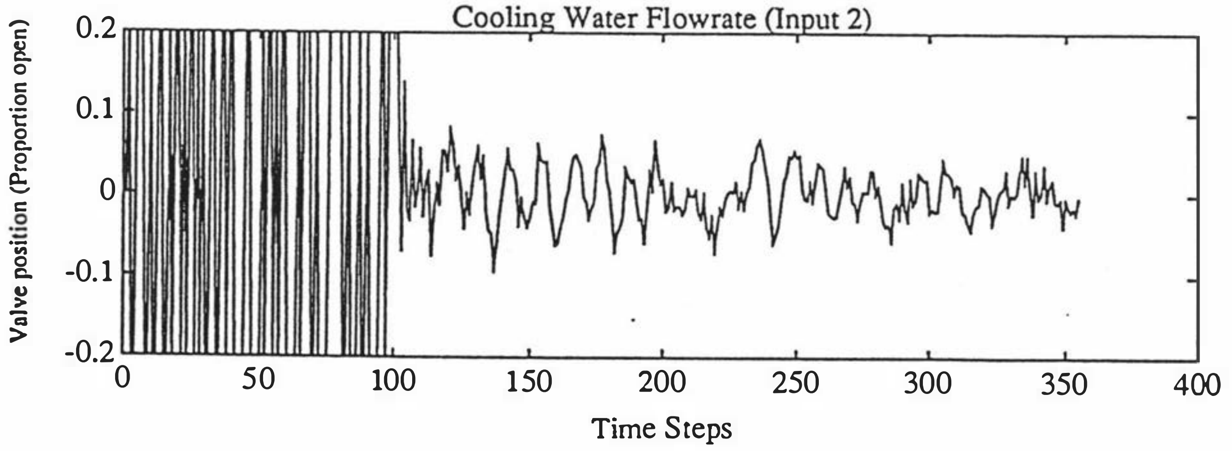

Figure 10.5.4

The plots of the two actuation inputs, W and F, against time are shown in Figures 10.5.3 and 10.5.4. In these the PRB signals can be clearly seen for the first 100 time-steps. After the 100 time-step it can be seen that the actuation inputs quickly settled to within a band either side of the steady-state levels.

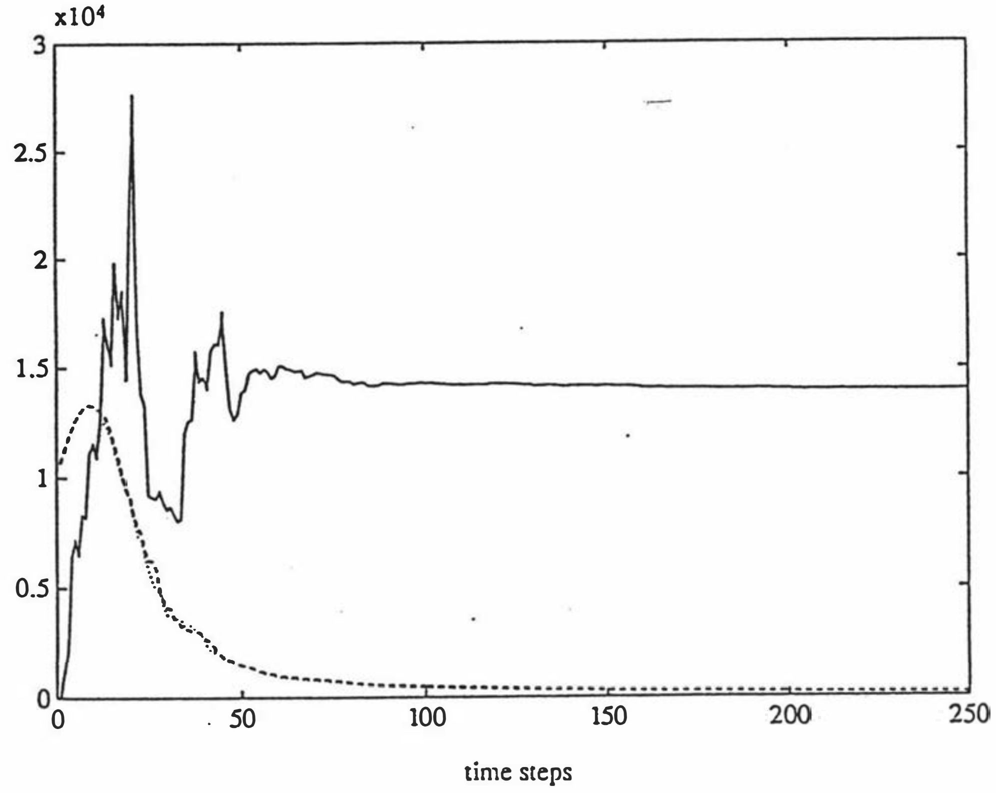

Figure 10.5.5

The plots of the traces of the solution of the Riccati equation and the covariance matrix against time, Figure 10.5.5, shows that the two traces converge to constant values. This suggests that the controller gain matrix has converged to a constant matrix and the recursive least squares algorithm estimates of the coefficients of the evaporator have converged.

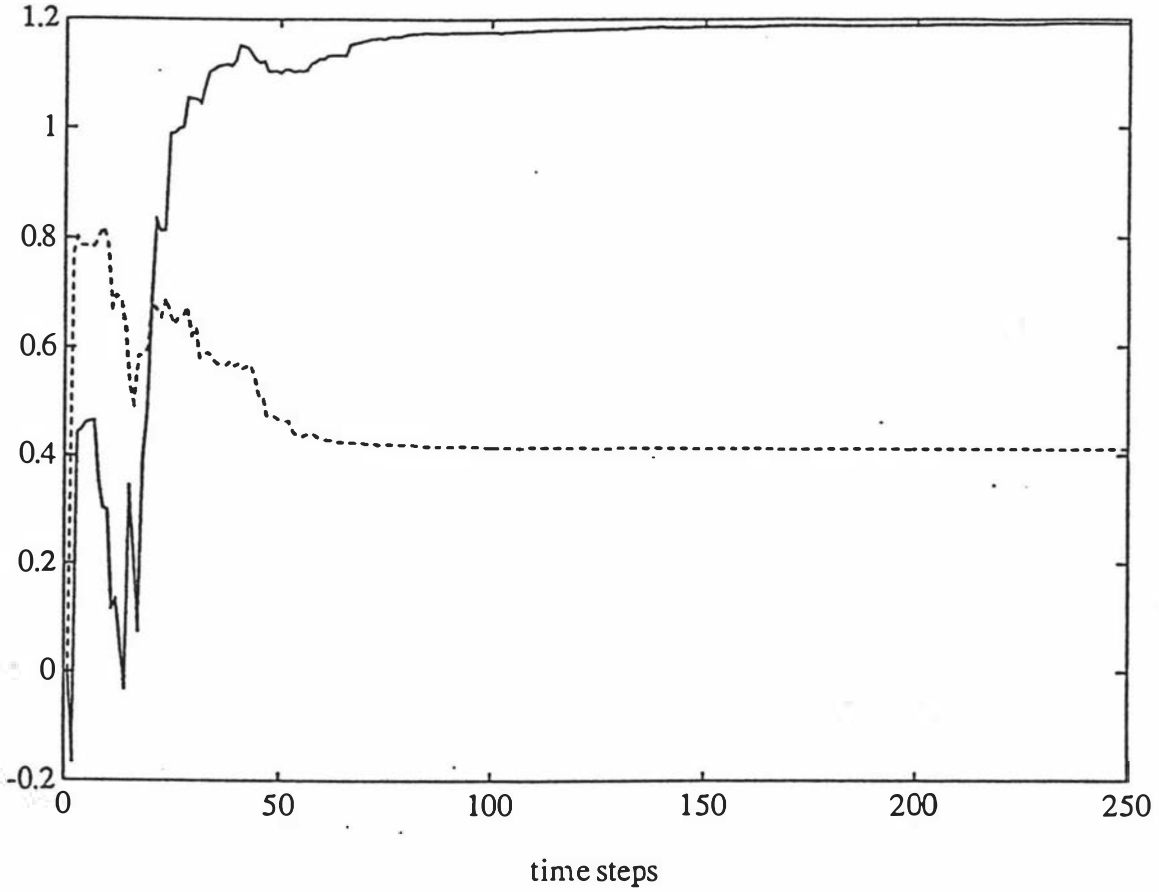

Figure 10.5.6

The plot of two of the coefficients against time, Figure 10.5.6, confirms that the coefficient estimates have converged.

10.6 REVIEW

In this chapter self tuning controllers have been reviewed. A self tuning controller for the evaporator, described in Chapter 9, was developed although this controller was never tested due to an end to the dairy season and a change in company research and development policy.

The controller was designed to meet the requirements of the dairy company and utilised as much existing equipment as possible, with the minimum of interference to their existing mode of operation. The design was such that, if any of the components of the multivariable controller failed, the existing controller would immediately resume control of the process. The multivariable controller was implemented in commercially available software, MATLAB, which has been proved to be numerically robust.

The simulation study showed that it was feasible to control the process using a self-tuning controller: the controller converged and the calculations required in each time-interval could be completed within the available time-slice.

11 CONCLUSION

11.1 Summary

In this thesis multivariable time-delayed processes have been investigated. In particular the method of controlling time-delayed processes using a device known as a Smith predictor has been explored.

The Smith predictor, which has been used on single-input single-output systems since the late 1950's relies on the commutativity of scalar multiplication to allow the translation of the delay operator through the transfer function. However, this commutativity property does not, in general, extend to the multivariable case. Several special cases of Smith predictors have appeared in the literature. These Smith predictors have relied on the structure of the delay operator, or a close approximation to the delay operator, to allow the operator to be commuted through the transfer function. The representation of multivariable processes using either state-space or transfer functions is common place. However, the representation of multivariable time-delays in process descriptions has yet to be fully explored. In Chapter 3 several of the methods of representation of multivariable time-delays were considered and a general form of representation developed. This formulation is sufficiently general to include majority of process descriptions.

In Chapter 3, two new results that allow a type of pseudo commutativity for matrices, and hence multivariable transfer functions, have been presented.

These results allowed the construction of a Generalised multivariable Smith predictor in Chapter 6.

Many techniques exist for the reformulation of a state-space description of a process as a transfer function description. However the converse task, reformulating a transfer function description as a state-space description often relies on numerical methods. The results of Chapter 3 were used in Chapter 4 to present a method of reformulating a multivariable transfer function description of a process as a state-space description. The state-space description obtained is, by default, in a block-diagonal form, which makes it extremely easy to work with in the time-domain.

In chapter 5 a discrete-time state-space description, the Non-Minimal State-Space (NMSS) description was investigated and a time-delayed variant of this description developed.

The time-delayed state-space descriptions, developed in chapter 4, allow for the implementation of a new form of Smith Predictor: a State-space Smith predictor was developed in chapter 7. This form of Smith predictor could prove to be extremely useful for digital implementation of Smith predictors.

It is well known that Smith predictors cannot control an open loop unstable process. In chapter 8 a proof of this limitation, using the results of Chapter 3 was presented. An investigation of the robustness of the generalised multivariable Smith predictor of Chapter 6 to modelling errors was also undertaken in Chapter 8.

In Chapters 9 and 10 a case study of a multivariable time-delayed process was presented. This process, with relatively long time-delays and numerous inputs and outputs is truly multivariable in nature. A NMSS description for this process was developed and used to investigate the feasibility of implementing a multivariable self-tuning controller.

11.2 Recommendations

There are several developments made in this thesis that need further work. These include:

- The development of a method, or methods, for finding minimal state-space descriptions from transfer function form. It may be possible to develop an algorithm, based on the method presented in Chapter 4, that allows reformulating a multivariable transfer function as a minimal state-stace description.
- The investigation of the performance of the generalised Smith predictor on processes with disturbance inputs.
- Exploration of the Relationship between generalised Smith predictors and LQC and the effect of model mismatch on generalised Smith predictors.
- Explore the possibilities of combining the methods proposed by De Paor (1985) and De Paor and Egan (1989), for the control of open-loop unstable processes, and the Generalised Smith predictor presented in this thesis .
- In the evaporator case study, a multivariable self tuning controller was investigated as a possible control mechanism for the evaporator. It would be of interest to know how well this process could be controlled using a generalised multivariable Smith predictor.

APPENDIX A:RAW DATA COLLECTED FROM EVAPORATOR

This dataset is the data collected from the evaporator on Run 3.

The records are the measurments of the Steam pressure, Density, 5th effect Temperature and Whey flow rate. These measurements are in the range 0-255. The other two columns of data contain the valve positions. These are 0 and 1, corresponding to the valve being pertibated -5% or 5% respectively.

Time between samples is - 20.00 seconds

1 STEAM PRESS.

2 DENSITY

3 5TH EFFECT TEMP.

4 WHEY FLOW RATE

Output channel 1 is perturbed

Output channel 2 is perturbed

During this run the steam pressure was varied by hand

DATA POINT	STEAM PRESSURE	DENSITY	5TH EFF TEMP	WHEY FLOW RATE	WHEY FLOW ACTUATION	COOLING FLOW ACTUATION
0	200	29	49	210	0	1
1	198	32	52	138	0	1
2	197	31	51	138	0	0
3	198	32	56	226	0	1
4	197	32	52	149	0	1
5	198	30	53	154	0	0
6	197	32	48	226	0	1
7	197	31	52	148	0	1
8	197	31	47	146	0	0

9	197	34	47	228	1	1
10	195	36	55	146	0	1
11	194	40	47	148	1	1
12	196	42	47	148	1	1
13	193	48	51	145	1	1
14	185	50	32	144	1	0
15	201	51	42	230	1	0
16	217	52	42	229	0	0
17	178	55	32	211	1	1
18	151	54	40	136	0	1
19	156	54	38	139	0	0
20	162	54	37	222	0	1
21	212	55	49	144	0	1
22	229	51	38	151	1	0
23	197	44	44	227	0	0
24	118	41	40	225	0	1
25	141	34	38	138	0	0
26	132	34	47	223	1	0
27	198	36	47	228	0	0
28	224	32	46	231	0	1
29	158	26	44	147	0	0
30	169	20	44	230	1	1
31	226	10	32	147	1	0
32	149	12	42	221	1	0
33	217	13	37	232	0	0
34	150	5	37	213	1	0
35	160	3	32	203	0	1
36	167	3	33	141	1	0
37	190	1	38	224	0	0

38	187	2	37	225	0	0
39	160	0	36	208	1	1
40	149	0	39	146	1	1
41	226	0	25	147	0	0
42	150	0	36	221	1	0
43	174	0	38	227	1	1
44	224	0	28	165	0	1
45	217	0	29	165	0	0
46	161	0	36	226	0	0
47	226	0	32	233	0	0
48	225	0	38	208	1	0
49	167	5	45	209	1	1
50	165	8	44	141	1	1
51	183	9	50	153	0	0
52	205	16	43	214	1	0
53	154	19	53	210	0	1
54	179	19	54	150	0	1
55	218	16	57	155	0	0
56	155	22	57	231	1	1
57	201	22	59	153	0	1
58	160	21	55	153	1	1
59	219	22	54	154	0	1
60	172	24	57	151	0	1
61	165	23	60	155	1	0
62	214	20	63	237	0	0
63	148	21	56	209	1	1
64	139	22	60	152	1	1
65	140	21	61	151	0	1
66	193	21	66	147	1	1

67	229	21	66	175	0	1
68	223	19	56	175	1	0
69	210	11	61	233	0	1
70	206	4	58	152	0	0
71	163	3	53	231	1	0
72	172	6	54	229	0	1
73	219	13	57	148	0	0
74	226	17	52	221	1	1
75	177	2	51	149	1	0
76	151	21	57	227	1	0
77	218	16	54	232	0	1
78	160	20	57	147	0	1
79	199	25	50	148	1	0
80	209	22	58	227	0	1
81	203	22	58	155	1	0
82	195	17	52	234	0	1
83	185	16	52	154	0	1
84	165	13	54	158	1	1
85	213	14	48	154	0	0
86	189	16	48	230	0	0
87	172	18	46	228	1	0
88	204	11	44	229	1	1
89	161	12	41	147	0	1
90	176	14	26.	145	0	1
91	154	13	29	137	0	0
92	164	13	37	222	1	1
93	187	11	33	144	0	0
94	214	10	28	232	0	1
95	210	8	27	146	0	0

96	206	3	31	230	1	1
97	159	0	29	154	0	0
98	157	0	30	221	1	0
99	194	0	31	226	0	0
100	197	5	36	229	0	1
101	154	12	31	151	0	0
102	157	16	44	227	0	1
103	187	14	47	144	0	1
104	207	20	41	147	1	0
105	154	20	43	224	0	0
106	212	18	28	234	1	1
107	215	13	33	153	1	1
108	202	14	32	155	1	0
109	153	14	26	227	1	0
110	167	11	30	224	0	1
111	190	11	31	149	0	1
112	219	13	30	152	1	0
113	183	16	33	224	0	1
114	156	19	32	135	0	0
115	191	13	34	226	0	0
116	215	16	37	231	1	0
117	203	20	30	224	1	0
118	199	22	33	205	1	1
119	168	18	29	146	1	1
120	164	20	28	144	0	0
121	182	19	35	222	0	0
122	188	18	30	223	1	0
123	161	20	34	210	1	0
124	203	16	34	212	1	0

125	208	12	34	217	1	1
126	183	13	31	140	1	0
127	167	11	33	224	1	1
128	192	8	25	146	0	0
129	155	10	24	226	0	1
130	176	9	34	145	1	1
131	210	6	29	163	0	1
132	188	7	33	163	0	1
131	210	6	29	163	0	1
132	188	7	33	163	0	0

APPENDIX B: TSR PROGRAM DEVELOPED FOR CONTROLLING REAL PROCESS

```
program ScndTSR;

uses
CRT,DOS,Unit_Key,Unit_Scn,Unit_Str,Unit_Sup,Unit_Isr,Unit_Int,Unit_Pgm;
{with the execption of CRT and DOS, these units are part of the Turbo
professonal library. CRT and DOS are standard turbo pascal units }

const
  Keithley_add = $cff0;
  Max_Channel  = 12;
  Max_Point = 35;
  Max_Order   = 9;
  StartCount  = 5; {((Max_Point+1+StartCount)*Int_Ticks)*18.2=Sample period}
  Int_Ticks = 3;
  TickCount = 4;
  BlnkStr = '                    ';

var
  KeySignal : _SignalKey;
  TimerSignal : _SignalTime;
  ErrorCode : word;
  c_val,bad_channels: word;
  index,k,j,i,Count,Point,Channel,Ticks : byte;
  FiltVal : array[0..Max_Channel] of longint;
  Last_Val : array[0..Max_Channel] of integer;
  Order : array[0..Max_Order] of byte;
  Chan_Buff : array[0..Max_Channel,0..4] of Integer;
  Sort : array[0..4] of integer;
  Actuator,Tmp, Chan_Val : integer;
  ActStr : string[40];
  MeasStr : string[80];
  CRLF : string[2];
  FLAG : boolean;


{$F+}
procedure Keithley(var Signal : _Signal);
{$F-}
begin
  with Signal do
  begin

{ continue with emptying buffer if neccessary}

    if length(MeasStr)>0 then MeasStr:=__StuffKey(MeasStr);

    if _KeyAction = _Service then
    begin
      index := 0;
      Count := 0;
      Point := 0;
    MeasStr:='';
    Ticks:=0;
    FLAG := TRUE;
    for i:=0 to Max_Channel do FiltVal[i]:=0;
    ActStr:= BlnkStr
  end;
  if Ticks < TickCount then inc(Ticks)
  else if (_TimeAction = _Interval) and (FLAG) then
  begin
    inc(Count);
    Ticks:=0;
    if Count >= StartCount then
    begin

{mains zero cross detector -ve to +ve}

      k:=0;      {state unknown}
      repeat
        for i:=0 to 2 do
          Count:=Count;
        Mem[Keithley_Add : $8A] := 15;  {channel number}
        Mem[Keithley_Add : $81] := 6;      {motherboard slot}
        Mem[Keithley_Add : $98] := 0;      {start conversion}

{wait for A_D}
        i:=0;
        while (Mem[Keithley_Add : $98] = 255) and (i<200) do inc(i);
        if i>=200 then       {skip this detector if no response}
        begin
```

```
            k:=1 ; Chan_Val:=1
          end
          else

{else set k=1 when mains reading is negative}

          begin
            Chan_Val:=Mem[Keithley_Add : $82]
                    +((Mem[Keithley_Add : $83] and $0F) shl 8)-2047;
            if (k=0) and (Chan_Val<0) then k:=1
          end;
        until (k=1) and (Chan_Val>=0);

{now read channels}
        index:=(index+1) mod 5;
        for Channel:=0 to Max_Channel-1 do
        begin
          for i:=0 to 2 do Count:=Count;
          Mem[Keithley_Add : $8A] := Channel;  {channel number}
          Mem[Keithley_Add : $81] := 6;        {motherboard slot}
          Mem[Keithley_Add : $98] := 0;        {start conversion}

          {wait for A_D}
          i:=0;
          while (Mem[Keithley_Add : $98] = 255) and (i<200) do inc(i);

{use last value for this channel}

          if i > 199 then Chan_Val:=Last_Val[Channel];
```

```
{or get new value}

          j := Mem[Keithley_Add : $82];
          k := Mem[Keithley_Add : $83] and $0F;
          Chan_Val:=(k shl 8)+j;
{                 for j:=0 to 3 do chan_buff[channel,j]:= chan_buff[channel,j+1];}
{update chan_buff}
          if Chan_val >=2047  then chan_buff[channel,index]:= chan_val
           else bad_channels:= bad_channels or (1 shl channel);
          if Point > 4 then
           begin

{   move channel data to sort array for median filter}

           for j:=0 to 4 do sort[j]:=chan_buff[channel,j];
            For j:=0 to 3 do
              for i:=0 to 3-j do
                if Sort[i]>Sort[i+1] then
                  begin
                   Tmp:=Sort[i];
                   Sort[l]:=Sort[i+1]; Sort[i+1]:=Tmp
                  end;
           end;

{operate filter on this channel at current point}

        FiltVal[Channel]:=FiltVal[Channel]+longint(Sort[2]);
```

```
{      FiltVal[Channel]:=FiltVal[Channel]+longint(Chan_Val);}  {uncomment
this line and comment above line if you wish to deactivate the median filter}
    end;
    inc(Point);
    if Point > Max_Point then
    begin
     Point := 0;

{convert all filtered values to string and put into buffer}

     if __QueueKey >= 15 then
     begin
      MeasStr := '[';
      for i:=0 to Max_Order do
      begin
       C_Val := FiltVal[Order[i]] div (max_point-4);
       Last_Val[Order[i]]:=C_Val;
       MeasStr:=MeasStr+__RightStr(__ToDecStr(C_Val,2),2)+' ';
      end;

{ read digital port [mode information]}

      C_Val:=255-mem[keithley_add:$86];
      MeasStr:=MeasStr+__RightStr(__ToDecStr(C_Val,2),2)+' ';
      MeasStr:=MeasStr+__RightStr(__ToDecStr(bad_channels,2),2)+' ';
      MeasStr := MeasStr+']'+char($0D);
      { if controller is in control and learn, controland save or control
       and we have good signals on all channels then we are in
       control otherwise hand back control to modicon}
```

```
          if ((C_Val>=2) and (C_Val<=9)) then  mem[keithley_Add:$89]:=1
            else mem[keithley_Add:$89]:=0
        end;
        for i:=0 to Max_Channel do FiltVal[i]:=0;
        MeasStr := __StuffKey(MeasStr);
        bad_channels:=0;
{get actuator values from screen, if there are none skip actuator update}


         __ReadScn(0,0,@ActStr[1],25,0);
         __QuikScn(0,0,1,15,BlnkStr);


{else convert them to 12 bit integer and write to d-a channels}
        i:=1;
        while (ActStr[i] = ' ') and (i<25) do inc(i);
        j:=i;
        while (ActStr[j] <> ' ') and (j<=25) do inc(j);
        j:=j-i;
        if ActStr <> BlnkStr then
        begin
          Val(__SubStr(ActStr,i,j),Actuator,ErrorCode);
          mem[Keithley_Add:$84]:=0;
          mem[Keithley_Add:$85]:=lo(Actuator);
          mem[Keithley_Add:$84]:=1;
          mem[Keithley_Add:$85]:=Hi(Actuator) and $F;
          i:=i+j;
          while (ActStr[i] = ' ') and (i<25) do inc(i);
          j:=i;
          while (ActStr[j] <> ' ') and (j<=25) do inc(j);
          j:=j-i;
```

```
                Val(__SubStr(ActStr,i,j),Actuator,ErrorCode);
                mem[Keithley_Add:$84]:=2;
                mem[Keithley_Add:$85]:=lo(Actuator);
                mem[Keithley_Add:$84]:=3;
                mem[Keithley_Add:$85]:=Hi(Actuator) and $F;
                mem[Keithley_Add:$9D]:=0;
                Count := 0
              end
            end
          end
        end
    end;
end;



begin
{initialize the keithley}

  Mem[Keithley_Add : $9A] := 0;        {global gain = 1}
  Mem[Keithley_Add : $9D] := $40;      {enable D-A strobe}
  for i:=0 to Max_Channel do
    begin
      Last_Val[i]:=0;
      for j:=0 to 4 do
        Chan_buff[i,j]:=0;
    end;
  bad_channels:=0;
{    Disturbances}
```

```
 Order[0]:=2 ; Order[1]:= 3; Order[2]:=4 ; Order[3]:=9;

{    Y,s}

 Order[4]:=8 ; Order[5]:= 7;

{    U,s}

 Order[6]:=5 ; Order[7]:= 0;

{   Set Points}

 Order[8]:=12; Order[9]:=11;
 FLAG := False;
 CRLF := char($0D); {+char($0A);}
 Ticks:=0;
 MeasStr:='';
 with KeySignal do
 begin
   _HotKeySeq := _KeyTable[_Key_a_h];
   _Action := _Service
 end;
 with TimerSignal do
 begin
   _Ticks := Int_Ticks;
   _Action := _interval
 end;
 MeasStr:=__StuffKey('MATLAB'+CRLF+'kthly'+CRLF);
 __InsInt(@Keithley,4096,False,False,@KeySignal,1,@TimerSignal,1
  ,'Keithley',ErrorCode);
  __ExitRPgm(0,4096);
end.
```

APPENDIX C: MATLAB CODE

Main routine

```
% This m-file contains the main program loop for the evaporator.


% variables used.
%
% A, B, E, K                state space description of model formed
% data                      used to test algorithm agaist historical data
% dist                      history of previous disturbances
% engY, engYsetoutputs and output setpoints in engineering units.
% eta                       history of modelling errors.
% eti                       elapsed time in last iteration (computation time)
% evap              file in which the current description is saved.
% F                         the optimal state feedback gain matrix based on S
% itnum                     number of iterations of the riccati equation  (recursive
form) used %                                at each time step
% mode              current mode of operation learn, learn and control, control only,
%                                   control and save
% m                         Measurements made (array)
% order             Order of the polynomials
% N                         N is the size of the A matrix.
% ok                        always 1 unless something has gone wrong
% q1, q2                  number of inputs and disturbances respectively
% r                       number of outputs
% Q, R                    weighting matices for the controller design
% S                       the current estiate of the solution of the Riccati equation
% setpoint                history of previous setpoints. Range 0-4096
% t                       number of time steps since simulation begun
% time                    used to calculate eti
% tol                     tolerance used to see if need to continue model
estimation
% u                       history of last scrnlen inputs u(t), ... u(t-scrnlen). Range
0-4096
% U                       inputs with the setpoint removed. Range +/-4096
% uwant                   inputs that would have been sent if totally in control
% usent           last actuation sent . Range 0-4096
% usetpoint               history of set-point for the actuations. Range 0-4096
% uuset           input setpoint   (yet to determine how to set this variable)
% y                       history of last scrnlen outputs y(t), y(t-1),... y(t-scrnlen).
range 0-
%                                 4096
% ZA,ZB,ZE,ZK used in the construction of the state vector.
% uindex,yindex,}         index of which measurements are inputs, outputs and
disturbances.
% dindex          }
%

% This m-file consists of several parts
% 1)  Data collection
% 2)  Data display
% 3)  Estimation of transition matrices  (done in another m-file called estimate.m)
% 4)  Update of control law
% 5)  Update of state vector
% 6)  Calculation of new actuations


% It is assumed that either this program or EVAP.m has already be run. Evap.m
sets up all the required
% matrices and vectors for this program.
%
% The following files are called by this program:
%
% RLS.M, and engunit.m

% THESE MUST BE CHECKED BEFORE USE

  % the measurements made are ordered in the tsr program as the 4 disturbances
  % then the outputs (2) and the inputs (2) followed by the output setpoints

uindex = [7,8];
yindex = [5,6];
dindex = [1,2,3,4];
sindex = [9,10];

%load evap                         %Retrieve saved values
mode = 2;                          % assume mode is learn until is set otherwise
ok = 1;
while ok >0,                       % main loop
  %----------------------------------------------------------------------
  % PART 1   DATA COLLECTION AND ACTUATION
  % do some housekeeping tasks first
        t=t+1;
        eti=etime(clock,time);
        time=clock;
        % this block of the program sends the actuations to the Keithley
        % and reads in the new data from the Keithley.
        clc;
        disp(usent);                    % send the actuation to the keithley
        % in the real plant the measurements are made by a tsr. these are then
        % pushed into the keyboard buffer. From where they are read into
MATLAB
        % by the following lines.
        m = input(' ');
        %  if the process is being simulated, an m-file called data can be called
here to
        % provide measurements from simulated plant
        % m=data(t,:)
        i= m(12);
        if i==0
                good = 1; setstr(7*ones(1,30))
      else
                good=0;
        end;
        if mode~=m(11);  % not the same mode  as before
                mode = m(11);  % change mode
        if mode == 1   % newly entered learn mode so need to get the actuation
set-points
                uuset = m(uindex)';         %  for the actuations  are the values in
use. store
```

```
                                        % these for latter use
                    ysetp = m(yindex)';        % get the setpoint from the input-data,
m
          end;
          % display mode of operaion
          if mode ==0 disp('just watch')
            elseif mode== 1 disp('leam');
            elseif mode== 2 disp('learn and control')
            elseif mode== 3 disp('control')
            else disp('mode save')
          end;

   %---------------------------------------------------------
   % PART 2 DATA DISPLAY
   % This consists of two sub-sections
   % 1) update the stacks of stored data
   % 2) display the stacks
   %---------------------------------------------------------


          %-----------------------------------------------------------------------------------------
          % PART 2.1. UPDATING THE STACKS
          % Before the data can be displayed the stacks of previous values must be
updated.
          % The stacks are shifted down one place and the most recent data takes
the top
          % position in the stack. The oldest piece of data is thrown out.
          %
          % A complication is that data goes into keihley from a 4-20mA loop on
some of
```

```
        % the measurements. This when translated into voltages is 1-5v. giving
range
        % 4095/5 to 4095 when digitised.
        % On the disturbancces this offset is subtracted. In the case of inputs and
outputs,
        % we are working with deviation variables from the setpoint and
usetpoint, so this
        % offset is not important to the calculations, However, for the displayed
and
        % stored data, these values are given in engineering units.
        %---------------------------------------------------------------------------------
        % stack data into y,u, and dist stacks
        disp('display data');
        uwant=[usent', uwant(:,1:scrlen-1)];
        u=[m(uindex)',u(:,1:scrlen-1)] ;               % the true inputs in integers
0-4095  u=[m(uindex)',u(:,1:scrlen-1)];                % the true inputs in integers
0-4095
        setpoint = [ysetp,setpoint(:,1:scrlen-1)];     % get the setpoint from the
input-
                                                                % data, m
        usetpoint=[uuset,usetpoint(:,1:scrlen-1)];     % the setpoint for the inputs
is the
                                                                % running average of
the last scrlen
                                                                %     of     inputed
setpoints values
        U=u-usetpoint;                                 % remove the setpoint from
the
                                                                % inputs sent
```

```
dist=[m(dindex)',dist(:,1:scrlen-1)];                    % disturbances
y=[m(yindex)', y(:,1:scrlen-1)];
Y = y-setpoint;
yhat=[C*x,yhat(:,1:scrlen-1)];          % predicted outputs
tr=[trace(S),tr(1:scrlen-1)];           % tr is the stack of old trace of the
                                        % estimates of the riccati equation
                                        % solution
out1=[theta1(1:2*nr),out1(:,1:scrlen-1)];   % some of the coefficents, the
two
                                        % sets of autoregressive terms
out2=[theta2(1:2*nr),out2(:,1:scrlen-1)];
fprintf(outfname,'%g ',yhat(1,1));
fprintf(outfname,'%g ',yhat(2,1));
for i=1:24
        fprintf(outfname,'%g ',theta1(i));
end;
for i=1:24
        fprintf(outfname,'%g ',theta2(i));
end;
fprintf(outfname,'%g ',tr(1));
fprintf(outfname,'%g ',eti);
fprintf(outfname,'\n');
%-----------------------------------------------------------------------------------
% PART 2.2. DISPLAYING THE STACKS
%       In this opart of the program  the historical data is displayed.
%-----------------------------------------------------------------------------------
disp([A(:,1:2),B]);%pause(5);
clc,clg                                 % ensure that the screens are clear.
engY=y';
```

```
        engU=u';
        engUwant=uwant';
        engYset = setpoint';
        engYhat = (yhat+setpoint)';
        engY(:,1)=engunit(engY(:,1),1.1,1.3);
        engYhat(:,1)=engunit(engYhat(:,1),1.1,1.3);
        engYset(:,1)=engunit(engYset(:,1),1.1,1.3);
        engY(:,2)=engunit(engY(:,2),0,100);
        engYhat(:,2)=engunit(engYhat(:,2),0,100);
        engYset(:,2)=engunit(engYset(:,2),0,100);
        engU=(engU-820*ones(engU))*100/(4096-820);
        engUwant=(engUwant-820*ones(engUwant))*100/(4096-820);
        subplot(221),
        plot(t:-1:t-scrlen+1,[engY(:,1),engYset(:,1),engYhat(:,1)]),
        title('Density output')
        subplot(222),
        plot(t:-1:t-scrlen+1,[engY(:,2),engYset(:,2),engYhat(:,2)]),
        title('Temperature output')
        subplot(223),
        plot(t:-1:t-scrlen+1,[engU,engUwant]'),
        title('Inputs')
        subplot(224),
        plot(t:-1:t-scrlen+1,[eta',200*out1', 200*out2']),
        title(['Errors & coeff.  ', num2str(tr(1))])
%----------------------------------------------------------
% PART 3 ESTIMATION
%----------------------------------------------------------
 % mode is a variable assumed to have one of four values
 % 0 stop and hand over control
% 1 learn only, no control action sent
% 2 learn and control
% 3 control only
% 4 save and control
% the coefficents of the new A, B, E and K matrices are estimated. These
estimates are
% then arranged into their respective matrices.
    if ((mode==1)|(mode==2))    % then we must improve our estimates of
                                % the coefficients. To do this we must first
                        % form the vectors of predictors and then
                        % submit this, with the current coefficients
                        % to the recursive least squares routine,
                        % rls.m
    clear phi1;
    for j=1:nr
    phi1=[phi1;Y(:,j+1)];       % get y(t-j) (y(t) is in y(:,1), the first
                                % column of y)
    end;
    phi2=phi1;
    for j=1:q1                  % and the actuations
    phi1=[phi1;U(j,delay(1,j)+2:delay(1,j)+order+1)'];
    phi2=[phi2;U(j,delay(2,j)+2:delay(2,j)+order+1)'];
    end;
    for j=1:q2                  % and the disturbances
    phi1=[phi1;dist(j,delay(1,j+q1)+2:delay(1,j+q1)+order+1)'];
    phi2=[phi2;dist(j,delay(2,j+q1)+2:delay(2,j+q1)+order+1)'];
    end;
    % Retrive theta for the ith output and the covariance matric for it
    error=[Y(1,1)-theta1'*phi1;
```

```
        Y(2,1)-theta2'*phi2];
% run the recursive estimation routine
[theta1,p1,k1]=rls(lamda,phi1,error(1),theta1,p1);
[theta2,p2,k2]=rls(lamda,phi2,error(2),theta2,p2);
% save some workings to disk for offline analysis
% the trace of the covariance matrix
% the max value of the predictor gain (k) matrix
%--------------------------------------------------------------
%  UPDATE THE A,B,E,and K MATRICES
%--------------------------------------------------------------
for j=1:nr,
A((j-1)*r+1,1:r)=theta1((j-1)*r+1:j*r)';
A((j-1)*r+2,1:r)=theta2((j-1)*r+1:j*r)';
end;
count=r*nr;     % if K included then 2*r*nr
for j=1:q1
for k=1:order
                count=count+1;
                B((delay(1,j)+k-1)*r+1,j)=theta1(count);
                B((delay(2,j)+k-1)*r+2,j)=theta2(count);
end
end;
%          update the  non zeros components of E
for j=1:q2
for k=1:order
                count=count+1;
                E((delay(1,j+q1)+k-1)*r+1,j)=theta1(count);
                E((delay(2,j+q1)+k-1)*r+2,j)=theta2(count);
end;
```

```
    end;
    eta=[error,eta(:,1:scrlen-1)];
    mse=mse+error'*error;
    if (mode == 4),
        save  evap;
    end;
    %--------------------------------------------------------
    % PART 4  CONTROLLER DESIGN
    %--------------------------------------------------------
    % Update the feedback law using either Matrix sign function or iterative
method
    if (mode > 0)
        disp('control design');
        if max(max(isnan(S)))==1 % if the matrix has become
                                 %illconditioned then
                                 % nan's will appear in the matrix and
                                 %need to reset it.
            S=eye(Q);
        end;
        for i=1:itnum            % perform itnum iterations of of
                                 % recursive form of the solution of
                                 % the Riccarti equation
            F=(B'*S*B+R)\B'*S*A;
            S=A'*S*A-A'*S*B*F+Q;
        end;
    end;
    if mode >1
    %--------------------------------------------------------
    % PART 5 STATE RECONSTRUCTION
% This is a way of reconstructing the state vector using the
% input/output data, assuming that the oberver state space
% description has been used.
%--------------------------------------------------------
ZA = A(:,1:r)*Y(:,1:N);
ZB = B*U(:,1:N);
ZE = E*dist(:,1:N);
x= ZA(:,1) + ZB(:,1) + ZE(:,1)
for i=1:N-1
        x=x+[ZA(r*i+1:r*N,i+1);zeros(i*r,1)]+
[ZB(r*i+1:r*N,i+1);zeros(i*r,1)]                    +
[ZE(r*i+1:N*r,i+1);zeros(i*r,1)];
end;
%--------------------------------------------------------
% PART 6  ACTUATION CALCULATION
% State feedback controller based on the control law calculated earlier
%--------------------------------------------------------
usent=usetpoint(:,1)-F*x;
% usent must lie between 820 and 4095 and be an integer
for i=1:q1;
        usent(i)=max([820,usent(i)]);      % greater of 820 and usent(i)
        usent(i)=min([4095,usent(i)]);     % lesser of 4095 and usent(i)
end;
usent=round(usent');                   % Intergerise usent and make it a row
vector
end;
end;    % main loop
```

RLS.M

```
function [theta,p,k]=rls(lamda,phi,eta,theta,p);
% This m-file uses the rls method to update the estimates of
% the paramaters stored in vector theta using the new data stored in vector
% phi.
% Lamda is a forgetting factor specified by the user
% p is the covariance matrix
% eta contains the error (y-yhat)
k=p*phi/(lamda+phi'*p*phi);          % get the gain
theta=theta+k*eta;                  % calculate the changed theta
p=(p-k*phi'*p)/lamda;                % and update the covariance matrix estimate
```

References and Bibliography

Abrishamkar, F. and Bekey, G.A.

Estimation of time varying delays in linear stochastic systems. in *Proceedings of IFAC Identification and System Parameter Estimation*, York, UK, 1985.

Ahmed, M.S. and Sait, N.

State-space adaptive control through a modified bootstrap algorithm for parameter and state estimation. *IEE proc ptD* Vol 136 (5) 215-224 1989.

Akaike, H.

A new look at the statistical model identification. *IEEE Trans AC*-19 (6) 716-723, 1974.

Akaike, H.

Modern development of statistical methods. p169-184. In *Trends in Process and System Identification.* Edited Eykhoff, P. Pergamon Press New York, 1981.

Alevisakis, G., and Seborg, D.E.

An extension of the Smith Predictor method to multivariable linear systems containing time-delays. *Int. J. Control* , 3 (17) p541-551, 1973.

Alevisakis, G., and Seborg, D.E.

Control of multivariable systems containing time delays using a multivariable Smith predictor. *Chemical Engineering Science*, 29 p373-380, 1974.

Anderson, B.D.O., Bitmead, R.R., Johnson, C.R., Kokotovic, P.V., Kosut, R.L., Marcel, I.M.Y., Praly, L. and Riedle, B.D.

Stability of adaptive systems: passivity and averaging analysis. The MIT Press, Massachusetts, 1986, 326 p.

Astrom, K.J.

Computer control of a paper mashine - an application of linear stochastic control theory. *IBM journal research and development* 11 389-405, 1967.

Astrom, K.J.

Frequency domain properties of Otto Smith regulators. *Int. J. Control* 26 (2) 307-314, 1977.

Astrom, K.J.

Theory and applications of adaptive control -a survey. *Automatica* 19(5) 471-486, 1983.

Astrom, K.J., Borison, U., Ljung, L. and Wittenmark, B.

Theory and applications of self tuning regulators. *Automatica* 13 457-476, 1977.

Astrom, K.J. and Wittenmark, B.

On self-tuning regulators. *Automatica* 9 185-199, 1973.

Astrom, K.J and Wittenmark, B.

Computer controlled systems: theory and design. Prentice-Hall International, Englewood Cliffs, NJ 1984, 430 p.

Astrom, K.J. and Wittenmark, B.

Self tuning controllers based on pole-zero placement. *IEE Proc. ptD* 127 (3) 120-130, 1980.

Borison, U.

Self-tuning regulators for a class of multivariable systems. *Automatica* 15, p209-215, 1979.

Borison, U. and Syding, R.

Self-tuning control of an ore crusher *Automatica* 12,1-7, 1976.

Chotai, A. and Young, P.C.

Pole-placement design for time delay systems using a generalised discrete-time Smith predictor. *Proceedings of control 88 IEE* York, UK, 1988.

Chotai, A., Owens, D.H., Raya, A. and Wang, H.M.

Design of Smith control schemes for time-delay systems based on plant step data. *Int. J. Control*, 40 (2), p297-316, 1984.

Cluett, W.R., Shah, S.L. and Fisher, D.G.

Stable robust adaptive controller. *Int J Control* 45 (4) 1265-1273, 1987.

Cox, C., Boucher, R. Young, P.C.

Direct digital control using a personal computer: a new algorithm.p115-123. In *Personal Computers in Industrial Control.* Edited by Giorgi, B. Butterworths, 1988.

Cordero, O.A. and Mayne, D.Q.

Deterministic convergence of a self-tuning regulator with a variable forgetting factor. *Proc IEE prD* 128 (1) 19-23, 1981.

Crawford, R.A., and Austin, P.C.

Control of an evaporator. *Proceedings of the third international symposium on process engineering*, Sydney Australia 1988.

De Paor, A.M.

A modified Smith predictor and controller for unstable processes with time-delays. *Int. J. Control* 41 (4) p1025-1036, 1985.

De Paor, A.M. and Egan, R.P.K.

Extension and partial optimization of a modified Smith predictor and controller for unstable processes with time-delays. *Int. J. Control* 50 (4) 1315-1326, 1989.

De Souza, C.E., Goodwin, G.C., Mayne, D.O. and Palaniswami, M.

An adaptive control algorithm for linear systems having unknown time delay. *Automatica* 24 (3) 327-341, 1988.

Dugard, L., Goodwin, G.C. and Xianya, X.

The role of the interactor matrix in multivariable stochastic adaptive control. *Automatica*,20 (5) 701-709, 1984.

El-Sherief, H. and Sinha, N.K.

Bootstrap estimation of parameters and states of linear multivariable systems. *IEEE Trans* AC-24 (2) 340-343, 1979.

El-Sherief, H. and Sinha, N.K.

Suboptimal control of linear stochastic multivariable systems with unknown parameters. *Automatica* 18 (1) 101-105, 1982.

Elliott, H. and Wolovich, W.A.

Parameterization issues in multivariable adaptive control. *Automatica* 20 (5) 533-545, 1984.

Fuller, A.T.

Optimal non-linear control of systems with pure delay. *Int J Control* 8 (2) 148-168, 1968.

Furukawa, T. and Shimemura, E.

Predictive control for systems with delay. *Int. J. Control*, 37 (2) 399-412, 1983.

Garcia, C.E and Morari, M.

Internal model control 1 - A unifying review and some new results. *Ind. Engng Chem Proc Des. Dev.* 21 308, 1982.

Garcia, C.E and Morari, M.

Internal model control 2 - The multivariable case. *Ind Engng Chem Proc. Des Dev.* 24 (2) 472-484, 1984.

Garcia, C.E and Morari, M.

Internal model control 3 - Multivariable control law computation and tuning guidelines. *Ind Eng Chem Proc. Des Dev.* 24 (2) 484-494, 1984.

Garcia, C.E., Prett, D.M. and Morari, M.

Model predictive control: theory and practice -a survey. *Automatica* 25 (3) 335-348, 1989.

Garland, B. and Marshall, J.E.

Sensitivity considerations of Smith's method for time-delay systems. *Electronic letters* 10 (15) p308-309, 1974.

Garland, B. and Marshall, J.E.

Application of the sensitivity points method to a linear predictor control system. *Int. J. Control* 21 (4) 681-688, 1975.

Gawthrop, P.J.

Some interpretations of the self-tuning controller. *Proceedings IEE Control and Science* 124 (10) 889-894, 1977.

Gawthrop, P.J.

Comparative robustness of non-adaptive and adaptive control. *Proceedings of Control 85, IEE,* Cambridge, 1985.

Goodwin, G.C. and Sin, K.S.

Adaptive filtering prediction and control. Prentice-Hall, Englewood-Cliffs, New Jersey, 1984 540p

Gustavsson, I., Ljung, L. and Soderstrom, T.

Identification of processes in closed loop - identification and accuracy aspects. *Automatica* 13 (1) 59-75, 1977.

Hamon, B.V. and Hannon, E.J.

Spectral estimation of time delay for dispersive and non-dispersive systems. *Applied Statistician* 23 (2) 134-142, 1974.

Hannon, E.J.

Time series analysis. *IEEE Trans AC*-19 (6) 706-715, 1974.

Hannon, E.J. and Robinson, P.M.

Lagged regression with unknown lags. *J Royal Statistical Society, series B* 35 (2) 252-266, 1973.

Harris, C.J. and Billings, S.A. (Editor)

Self-tuning and adaptive control: theory and applications. Peregrinus Press Stevenage, 1981, 333p.

Hesketh, T.

Strategies for multivariable self-tuning control. PhD thesis Massey University, 1981, 186p.

Hesketh, T.

State-space pole-placing self-tuning regulator using input-output values. *IEE Proc pt D* 129 (4) 123-128, 1982.

Holt, B.R. and Morari, M.

Design of resilent processing plants -VI The effect of RHP zeros on dynamic resilience. *Chemical Engineering Science* 40 (1) 59-74, 1984.

Holt, B.R. and Morari, M.

Design of resilent processing plants -V The effect of deadtime on dynamic resilience. *Chemical Engineering Science* 40 (7) 1229-1237, 1985.

Horowitz, I.

Some properties of delayed controls (Smith predictor). *Int. J. control* 38 (5) 977-990, 1983.

Ioannou, P. and Sun, J.

Theory and design of robust direct and indirect adaptive-control schemes. *Int J. control* 47 (3) 775-813, 1988.

Isermann, R., Baur, U., Bamberger, W., Kneppo, P. and Siebert, H.

Comparison of six on-line identification and parameter estimation methods. *Automatica* 10 81-103, 1974.

Jerome, N.F. and Ray, W.H.

High performance multivariable control strategies for systems having time delays. *AIChE Journal* 32 (6) 914-931, 1986.

Kalman, R.E.

Design of a self-optimising control system. *Transactions ASME*, 80, 468-478, 1958

Keviczky, L. and Kumar, K.S.P.

Multivariable self-tuning regulator with generalized cost function. *Int J Control*, 1981.

Koivo, H.N.

A multivariable self tuning controller. *Automatica* 16 351-366, 1980.

Kwakernaak, H.

Optimal filtering in linear systems with time delays. *IEEE Trans* AC-12 (2) 169-173, 1967.

Kwakernaak, H. and Sivan, R.

Linear optimal control. Wiley, 1972, 575 p.

Laub, A.J.

A schur method for solving algebraic Riccati equations. *IEEE Trans* AC-24 (6) p913-921, 1979.

Ljung, L.

Analysis of recursive stochastic algorithms. *IEEE Trans AC*-22 (4) 551-575, 1977.

Ljung, L.

On positive real transfer functions and the convergence of some recursive schemes. *IEEE Trans AC*-22 (4) 539-551, 1977.

Ljung, L., Gustavsson, I. and Soderstrom, T.

Identification of linear multivariable systems operating under linear feedback control. *IEEE Trans AC*-19 (6) 836-840, 1974.

Ljung, L and Soderstrom, T.

Theory and practice of recursive identification. MIT press, Massachusetts, 1983, 529p.

Mancher, H. and Hensel, H.

Determination of order and deadtime for multivariable discrete-time parameter estimation methods. *Proceedings IFAC identification and system parameter estimation*, York, UK, 1985.

Marshall, J.E.

Extensions of O.J. Smith's method to digital and other systems. *Int. J. Control* 19 (5) 933-939, 1974.

Marshall, J.E.

Control of time-delay systems. Peregrinus Press on behalf of IEE, 1979, 225p.

Marshall, J.E., Chotai, A. and Garland, B.

A survey of time-delay system control methods. *Proceedings of control and its application, IEE*, Warwick, UK, 1981.

Marshall, J.E., Ireland, B. and Garland, B.

Comments on 'An extension of predictor control for systems with control-delays'. *Int. J. Control* 26 (6) p981-982, 1977.

Marshall, J.E. and Salehi, S.V.

Improvement of system performance by the use of time-delay elements. *IEE Proc. PtD* 129 (5) 177-181, 1982.

Marshall, J.E. and Walton, K.

Analytical design for time-delay system controllers. *Proceedings of the 4th IMA Int. Conference on Control theory*, ed Cook, P.A., Robinson College, Cambridge, 1984.

Mee, D.H.

An extension of predictor control for systems with control time delays. *Int J Control* 18 (6) 1151-1168, 1973.

Middleton, R.H. and Goodwin, G.C.

Improved finite word length characteristics in digital control using delta operators. *IEEE trans AC*-31 (11) 1015-1021, 1986.

Middleton, R.H. and Goodwin, G.C.

Adaptive control of time-varying linear systems. IEEE trans AC-33 (2) 150-155, 1988.

Middleton, R.H. and Goodwin, G.C.

Digital control and estimation: a unified approach. Prentice-Hall International, Englewood-Cliffs, New Jersey, 1990, 538 p.

Morari, M.

Design of resilient processing plants III. *Chem Engng Sci.* 38 (11) 1881-1891, 1983.

Morari, M. and Zafiriou, E.

Robust process control. Prentice-Hall Englewood-Cliffs, New Jersey, 1989, 488p.

Nisenfield, A.E.

Industrial evaporators: principles of operation and control. Monograph 9 Instrument Society of America, 1985.

Ogata, K.

Discrete-time control systems. Prentice-Hall International, Englewood Cliffs, N.J. 1987, 994 pp.

Ogunnaike, B.A.

A new approach to observer design for time-delay systems. *Int. J. Control* 33 (3) 519-542, 1981.

Ogunnaike, B.A. and Ray, W.H.

Multivariable controller design for linear sysytems having multiple time delays. *AIChE Journal* 25 (6) 1043-1057, 1979.

Ogunnaike, B.A. and Ray, W.H.

Incomplete state feedback for time-delay systems: Observer applications in multidelay compensation. *AIChE J.* 30 (5) 717-724, 1984.

Owens, D.H. and Chotai, A.

Robust controller design for linear dynamic systems using approximate models. *IEE Proc. PtD* 130 (2) 45-56, 1983.

Owens, D.H.

Multivariable and optimal systems. Academic press, London, 1981, 300pp.

Owens, D.H and Raya, A.

Robust stability of Smith predictor controllers for time-delay systems. *IEE Proc PtD* 129 (6) 298-304, 1982.

Palmor, Z.J.

Stability properties of Smith dead-time compensator controlers. *Int. J. Control* 32 (6) 937-949, 1980.

Palmor, Z.J. and Halevi, Y.

On the design and properties of multivariable dead-time compensators. *Automatica*, 19 (3) 255-264, 1983.

Palmor, Z.J. and Shinnar, R.

Design of advanced process controllers. *AIChE J.* 27 (5) p793-805, 1981.

Pappas, T., Laub, A.J. and Sandell, N.R.

On the numerical solution of the discrete-time algebraic Riccati equation. *IEEE Trans AC*-25 (4) 631-641, 1980.

Peterka, V. and Astrom, K.J.

Control of multivariable systems with unknown but constant parameters. *Proceedings of the 3rd IFAC Symposium on Identification and System Parameter Estimation*, 535-544 The Hague/Delft, 1973.

Power ,H.M. and Simpson, R.J.

Introduction to dynamics and control. McGraw-Hilll, London, 1978, 393pp.

Prager, D.L. and Wellstead, P.E.

Multivariable pole-assignment self tuning regulators. *IEE Proc ptD* 128 (1) 9-18, 1980.

Saridis, G.N.

Comparison of six on-line identification algorithms. *Automatica* 10 69-79, 1974.

Sen, A. and Sinha, N.K.

On-line system identifaction combining stochastic approximation and pseudoinverse. *Automatica* 11 425-429, 1975.

Shanmugathasan, N. and Johnston, R.D.

Exploitation of time delays for improved process control. *Int J control* 48 (3) 1137-1152, 1988.

Shanmugathasan, N. and Johnston, R.D.

Enhanced multivariable time delay compensation, *Proceedings of the third international symposium on process systems* Engineering, Sydney 1988.

Shieh, L.S., Tsay, Y.T. and Yates, R.E.

Some properties of matrix sign functions derived from continuous fractions. *IEE Proc prD* 130 (3) 111-118, 1983.

Singh, A. and McEwan, D.H.

The control of a process having appreciable transport lag-a laboratory case study. *IEEE Trans AC*, AC-22 (3) 396-401, 1975.

Smith, O.J.M.

Closer control of loops with dead time. *Chemical Engineering Progress* 53 (5) 217-219, 1957.

Soderstrom, T. Gustavsson, I. and Ljung, L.

Identifiably conditions for linear systems operating in closed loop. *Int. J. Control* 21 (2) 243-255, 1975.

Soliman, M.A. and Ray, W.H.

The optimal control of a process containing pure time-delays -II Transfer function models. *Chemical Engineering Science* 27, 2183-2188, 1972.

Song, H.K., Shah, S.L. and Fisher, D.G.

A self-tuning robust controller. *Automatica* 32 (5) 521-531, 1986.

Tade, M.O., Bayoumi, M.M. and Bacon, D.W.

Self-tuning controller design for systems with arbitrary time delays:Part I theoretical development. *Int J Sys Sci* June 1987.

Tade, M.O., Bayoumi, M.M. and Bacon, D.W.

Self-tuning controller design for systems with arbitrary Time delays:Part II Algorithms and simulation examples. *Int J Sys Sci* June, 1987.

Toivonen, H.T.

Multivariable adaptive control. *Modelling Identification and Control* 5 (1) 19-45, 1984

Vaughan, D.R.

A non-recursive solution for the matrix Riccati equation. *IEEE Trans AC-*15 597-599, 1970.

Walton, K. and Marshall, J.E.

Mismatch in a predictor control scheme: some closed-form solutions. *Int. J. Control* 40 (2) p403-410, 1984.

Wang, C.L. and Young, P.C.

Direct digital control by input-output, state variable feedback: theoretical background. *Int J. Control* 47 (1) 97-109, 1988.

Wang, C.L. and Young, P.C.

Direct digital pole-assignment control systems for multivariable systems based on input output state variable feedback. Proceedings *Control 88 IEE* York , UK, 1988.

Watanabe, K. and Ito, M.

A process model control for linear systems with delay. *IEEE Trans* AC-26 (6) 1261-1269, 1981.

Wolovich, W.A.

Linear multivariable systems. Springer-Verlag, New York 1974, 358pp.

Wolovich, W.A. and Falb, P.L.

Invariants and canonical forms under dynamic compensation. *SIAM J Control and Optimization* 14 (6) 996-1008, 1976.

Wood, R.K and Berry, M.W.

Terminal composition control of a binary distillation column. *Chemical Engineering Science* 28, 1707-1717, 1973.

Ydstie, B.E.

Adaptive control and estimation with forgetting factors. *Proceedings of IFAC Identification and system parameter estimation*, York, UK, 1985.

Ydstie, B.E. and Sargent, R.W.H.

Convergence and stability properties of an adaptive regulator with a variable forgetting factor. *Automatica* 22 (6) 749-751, 1986.

Young, P.C., Behzadi, M.A., Wang, C.L. and Chotai, A.

Direct digital and adaptive control by input-output state variable feedback pole assignment. *Int. J. Control* 46 (6) p1867-1881, 1987.

Young, P.C., Behzadi, M.A. and Chotai, A.

Self-tuning and self-adaptive PIP control systems. In *Implementation of self-tuning*. Edited Warwick, K., Peregrinus Press and IEE, 1988.