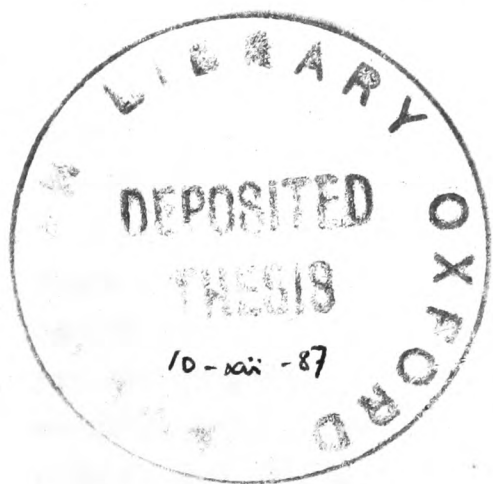


Process Control Applications of Long-Range Prediction



E. P. Lambert

St. John's College, Oxford

When you are a Bear of Very Little Brain, and you Think of Things, you find sometimes that a Thing which seemed very Thingish inside you, is quite different when it gets out into the open and has other people looking at it. A.A. Milne

Submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy at the University of Oxford. Trinity 1987

Submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy at the University of Oxford. Trinity 1987

ABSTRACT

The recent Generalised Predictive Control algorithm (Clarke *et al*, 1984,87) is a self-tuning/adaptive control algorithm that is based upon *long-range prediction*, and is thus claimed to be particularly suitable for process control application.

The complicated nature of GPC prevents the application of standard analytical techniques. Therefore an alternative technique is developed where an *equivalent* closed loop expression is repeatedly calculated for various control *scenarios*. The properties of GPC are investigated and, in particular, it is shown that 'default' values for GPC's design parameters give a mean-level type of control law that can reasonably be expected to provide robust control for a wide variety of processes.

Two successful industrial applications of GPC are then reported. The first series of trials involve the SISO control of soap moisture for a full-scale drying process. After a brief period of PRBS assisted self-tuning default GPC control performance is shown to be significantly better than the existing manual control, despite the presence of a large time-delay, poor measurements and severe production restrictions.

The second application concerns the MIMO inner loop control of a spray drying tower using two types of GPC controller: full multivariable MGPC, and multi-loop DGPC. Again after only a brief period of PRBS assisted self-tuning both provide dramatically superior control compared to the existing multi-loop gain-scheduled PID control scheme. In particular the use of MGPC successfully avoids any requirement for *a priori* knowledge of the process time-delay structure or input-output pairing. The decoupling performance of MGPC is improved by scaling and that of DGPC by the use of feed-forward. The practical effectiveness of GPC's design parameters (e.g. P , T and λ) is also demonstrated.

On the estimation side of adaptive control the current state-of-the-art algorithms are reviewed and shown to suffer from problems such as 'blowup', parameter drift and sensitivity to unmeasurable load disturbances. To overcome these problems two novel estimation algorithms (CLS & DLS) are developed that extend the RLS cost-function to include weighting of estimated parameters. The exploitation of the 'fault detection' properties of CLS is proposed as a more realistic estimation philosophy for adaptive control than the 'continuous retention of adaptivity'.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank all the people who have made the last three years possible, productive, and, most of all, enjoyable. In particular I am grateful to D.W. Clarke for his supervision, guidance and proof-reading of my thesis; Coorous Mohtadi-Haghighi for his constant help, enthusiasm, and good humour (and GPC!); and my brother Martin for getting me into this mess and for adding symmetry to both the self-tuning laboratory and the Varsity pistol team.

Many thanks also to my colleagues in Oxford; Yousef Al-Assaf, Dan McMichael, Terence Tsang, Dave Peel and Dario Boriani, and at Unilever Research; Turaj Tahmassebi, Chris Proudfoot, Dave Gregg, Keith Clark, and all the operators. I am also very grateful to Donald Knuth and Leslie Lamport for the $\text{\LaTeX}$ *document preparation system* used throughout this thesis.

Finally, last but not least, I thank my parents for their constant encouragement and support through the years.

This thesis was funded by the U.K. Science and Engineering Research Council.

Contents

1	Introduction	1
1.1	SELF-TUNING/ADAPTIVE CONTROL	3
1.2	LONG-RANGE PREDICTIVE CONTROL	5
1.2.1	A review of previous algorithms	5
1.2.2	Generalised Predictive Control	7
1.3	MULTIVARIABLE CONTROL	8
1.3.1	A review of previous algorithms	9
1.3.2	Multivariable Generalised Predictive Control	13
1.4	PARAMETER ESTIMATION	14
1.5	PROCESS CONTROL APPLICATIONS	17
1.6	STRUCTURE OF THE THESIS	19
2	Generalised Predictive Control	21
2.1	INTRODUCTION	21
2.2	THE BASIC GPC ALGORITHM	24
2.2.1	The CARIMA process model	24
2.2.2	Long-range prediction	25
2.2.3	The Predictive Control law	26
2.2.4	The Control Horizon	28
2.3	EXTENSIONS TO THE ALGORITHM	29
2.3.1	$P(q^{-1})$: Model following	29
2.3.2	λ : Control weighting	31
2.3.3	$T(q^{-1})$: Load disturbance rejection	32
2.3.4	Feed-forward compensation	34
2.3.5	Pre-specified set points	34
2.3.6	The GPC 'default' specification	35
2.4	CLOSED LOOP ANALYSIS OF GPC	36
2.4.1	Calculation of the GPC characteristic equation	36
2.4.2	Analysis using the GPCTEST Package	38
2.4.3	The Prediction Horizon, NY	40
2.4.4	The Control Horizon, NU	42
2.4.5	The initial prediction horizon, NS	44
2.4.6	The $P(q^{-1})$ transfer function	45
2.4.7	Control weighting, λ	46
2.4.8	The $T(q^{-1})$ polynomial	49
2.5	MULTIVARIABLE GPC	55
2.5.1	'Full' multivariable GPC	56
2.5.2	Distributed GPC	60

2.5.3	Some illustrative simulation studies	62
2.6	CONCLUSIONS	65
3	Towards practical parameter estimation	67
3.1	INTRODUCTION	68
3.2	FUNDAMENTALS OF PRACTICAL ESTIMATION	69
3.2.1	The class of model	69
3.2.2	The estimator criterion	70
3.2.3	The experimental conditions	70
3.2.4	Long-term adaptivity	71
3.2.5	Prior knowledge	72
3.2.6	Implementation and Commissioning	73
3.3	A BASIC ESTIMATION ALGORITHM	73
3.3.1	The CARIMA process model	74
3.3.2	Recursive Least Squares	75
3.3.3	Commissioning of the Algorithm	76
3.3.4	Properties of the algorithm	78
3.3.5	A simulation example	80
3.4	ESTIMATION OF TIME-VARYING PARAMETERS	81
3.4.1	Fixed exponential forgetting	83
3.4.2	Variable exponential forgetting	84
3.4.3	Directional forgetting	86
3.4.4	Alternative techniques	87
3.4.5	Some illustrative simulation studies	88
3.5	RLS WITH PARAMETER WEIGHTING	91
3.5.1	'Constrained' least squares	93
3.5.2	'Damped' least squares	97
3.5.3	Individual parameter weighting	101
3.5.4	The role of fault detection	102
3.5.5	Some illustrative simulation studies	105
3.6	ESTIMATOR JACKETTING	108
3.6.1	Signal conditioning	108
3.6.2	Estimation dead-bands	109
3.6.3	Normalisation and Scaling	110
3.6.4	Parameter constraint	111
3.6.5	Expert systems	113
3.7	CONCLUSIONS	115
4	GPC control of an industrial soap dryer	117
4.1	INTRODUCTION	117
4.2	THE SOAP DRYING PROCESS	120
4.3	HARDWARE AND SOFTWARE	122
4.4	EXPERIMENTAL RESULTS	123
4.4.1	Verification of the on-line moisture measurement	124
4.4.2	Observation of the process step response	125
4.4.3	Identification of the process using PRBS	127
4.4.4	Regulation of soap moisture	130
4.4.5	Regulation during changes in soap throughput	131
4.4.6	Servo control of soap moisture	133

4.4.7	Control of a soap base change	133
4.4.8	The effect of process saturation	135
4.4.9	The effect of temporary process failure	136
4.5	CONCLUSIONS	139
5	Multivariable GPC control of a spray drying tower	142
5.1	INTRODUCTION	142
5.2	THE SPRAY DRYING TOWER	144
5.3	SOME PRELIMINARY TRIALS	146
5.3.1	Observation of the open loop step response	147
5.3.2	Observation of process linearity	149
5.4	MULTIVARIABLE GPC CONTROL	150
5.4.1	General experimental procedure	150
5.4.2	Commissioning and self-tuning	152
5.4.3	Long term regulation	154
5.4.4	Servo control and decoupling	155
5.4.5	Comparison with existing PID control	158
5.4.6	Tower control with hot air	160
5.5	MULTIVARIABLE GPC DESIGN PARAMETERS	161
5.5.1	The effect of $\mathcal{P}(q^{-1})$	162
5.5.2	The effect of $\mathcal{T}(q^{-1})$	163
5.5.3	The effect of control weighting	165
5.6	MULTI-LOOP GPC CONTROL	165
5.6.1	Distributed GPC control, no feed-forward	167
5.6.2	Distributed GPC control, with feed-forward	169
5.6.3	Distributed GPC control, enhanced feed-forward	170
5.7	MULTIVARIABLE PARAMETER ESTIMATION	171
5.7.1	Estimated parameters	174
5.7.2	Adaptivity of the estimator	176
5.7.3	Verification of estimated parameters	178
5.8	DISTRIBUTED PARAMETER ESTIMATION	179
5.8.1	Estimated parameters	180
5.8.2	Adaptivity of the estimators	180
5.8.3	Verification of estimated parameters	182
5.9	CONCLUSIONS	183
6	Conclusions	186
6.1	FURTHER WORK	191
A	Experimental synopses	192
A.1	SYNOPSIS FOR CHAPTER 2	192
A.2	SYNOPSIS FOR CHAPTER 5	194
B	CLS and DLS: Derivation and implementation	196
B.1	DERIVATION OF 'CONSTRAINED' LEAST SQUARES	196
B.2	IMPLEMENTATION OF 'CONSTRAINED' LEAST SQUARES	198
B.2.1	The parameter update	198
B.2.2	The P matrix update	198
B.3	DERIVATION OF 'DAMPED' LEAST SQUARES	199

B.4 IMPLEMENTATION OF ‘DAMPED’ LEAST SQUARES 200

 B.4.1 The parameter update 201

 B.4.2 The P matrix update 201

C Nomenclature and symbols 202

 C.1 GENERAL NOMENCLATURE 202

 C.2 PRINCIPAL SYMBOLS 203

Bibliography 206

Chapter 1

Introduction

There are many problems associated with the automatic control of industrial processes including complicated dynamics (e.g. non-minimum phase transients), fundamental non-linearity; the presence of large time-delays; corruption of measurements by noise and ADC quantisation errors; unmeasurable load disturbances; interaction from other control loops; actuator non-linearities such as stiction and saturation, etc. Depending upon the severity of the above the overall process characteristics can be complex and difficult to control accurately using conventional techniques. This situation is further complicated by the fact that for most processes quantitative values for some or all of the above effects are difficult to obtain, and may be time-varying.

Nonetheless automatic process control is widespread throughout industry, relying until the late 1970's upon two basic techniques (Shinskey, 1979):

- The development of an accurate mathematical process model and the classical design of a fixed control law based upon this.
- The on-line manual tuning of conventional controllers, in particular the PID algorithm, using an iterative experimental approach, i.e. 'trial and error'.

Of these two methods the PID has gained almost universal acceptance, a testament to its general robustness and effectiveness.

It is, however, widely recognised that both methods have their limitations: the cost of accurately modelling the behaviour of a complicated process precludes its use for the majority of control loops, and it may be difficult and/or time-consuming to tune PID controllers for 'sluggish', non-linear or complicated processes (especially those with many

interacting loops). Furthermore the presence of time-delay (often caused by transportation delays where the product has to travel some distance before feedback measurements can be made) often requires the PID to be de-tuned to preserve stability at the expense of less tight control (Clarke, 1984).

More fundamentally both methods contain an implicit assumption that the process characteristics do not change, and thus develop a so-called '*fixed parameter*' control law. If, however, the process does change (e.g. a change in production levels, variations in raw materials, gradual fouling of a heat exchanger, ageing of a catalyst) then control performance may be degraded and the (costly) modelling or (time-consuming) PID tuning would have to be repeated.

SOME COMMERCIAL 'SELF-TUNERS'						
Controller	Manufacturer	Algorithm	GS	AT	CT	FF
Novatune	ASEA	GMV	✓	✓	✓	✓
VUMAN	Vuman Ltd.	LQG	✓	✓	✓	✓
STR	AccuRay Corp.	GMV			✓	✓
DMC	DMC Inc.	LRPC		✓		✓
IDCOM	Set Point Inc.	LRPC		✓	✓	✓
Electromax V	Leeds and Northrup	PID		✓	✓	
Exact	Fozboro	PID		✓	✓	✓
TCS 6355	Turnbull Control Systems	PID		✓	✓	
2071 Microtuner	Goulton West	PID		✓	✓	
UDC 500	Honeywell	PID		✓		
Micron P-200	Process Systems	PID		✓		
CRL 452	Control and Readout Ltd.	PID		✓		
Eurotherm 810	Eurotherm	PID		✓		
Microscan 1300	Taylor	PID	✓			
SLC 3700	Babcock Bristol	PID	✓			
PMS-100	Ferranti	PID	✓	✓		
Maxline	IRCON	PID		✓		
5701	Fenwal	PID		✓		
E5T & E5KN	Omron Electronics	PID		✓		
VeriTrim	Westinghouse	PID	✓			✓
SATT ECA40	Satt Controls	PID	✓	✓		
INTELLICON	Hungarian Sci. Academy	PID			✓	✓
GS—gain scheduling, AT—auto-tuning, CT—continuous tuning, FF—feed-forward						

Table 1.1: Some commercial controllers with 'self-tuning' features.

Awareness of these problems led to the introduction of a new control strategy by Kalman (1958), subsequently known as *Adaptive Control*, which aimed to overcome these problems by the combination of a more sophisticated control law with an automatic tuning mechanism. Unfortunately the increased complexity of this approach meant that its application had to wait until the late 1970's for the unprecedented advances in microprocessor technology and corresponding advances in the theory. Yet its validity is such that only a decade

later a large and increasing number of manufacturers of process control systems (see Table 1.1) now offer some form of adaptive control as a third alternative. This latter point is perhaps the best retrospective justification for adaptive control available.

1.1 SELF-TUNING/ADAPTIVE CONTROL

There have been many definitions of the terms 'self-tuning' and 'adaptive' control, often conflicting, sometimes interchangeable. For clarity in this thesis they are defined as follows:

- *Self-tuning control*: the combination of a sophisticated control law with an automatic tuning mechanism, normally implemented by estimating the parameters of a process model. Once tuned adaptivity is suspended and this becomes a fixed parameter control strategy.
- *Adaptive control*: the extension of self-tuning to the control of processes with significantly time-varying characteristics by the long-term retention of adaptivity in order to track any process changes and thus maintain the quality of control.

The general structure of both of these control strategies is identical and shown in Fig. 1.1, the only difference being the role of adaptivity.

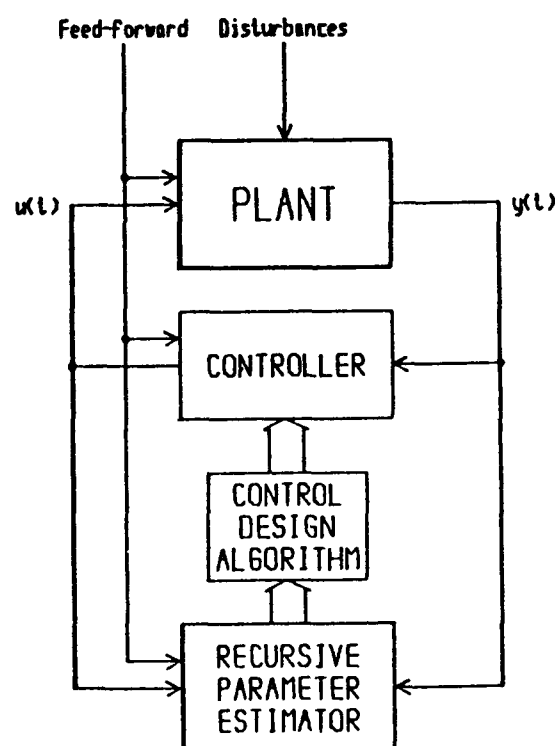


Figure 1.1: General layout of self-tuning/adaptive control.

The main benefit of self-tuning control, apart from a potential increase in the speed of controller commissioning, is that in principle it allows the application of advanced control strategies, such as GPC, that incorporate such important features as dynamic feed-forward and time-delay compensation. Thus it is the control law that aims to solve the specific process problems that were previously insoluble using (say) PID's. The role of adaptivity is simply to enable commissioning of the controller which would otherwise be impossible manually. For adaptive control, however, it is the actual adaptivity that is expected to solve the further control problem of the process being time-varying.

A third category which is often misleadingly known as either 'self-tuning' or 'adaptive' control, is here referred to as:

- *Auto-tuning control*: where self-tuning is used to provide the parameters of a conventional control algorithm, normally PID (Gawthrop, 1982; Åström & Hägglund, 1984)

Despite inherently suffering from the limitations of the conventional controller upon which it is based this retains the benefit of rapid commissioning. More importantly because it uses a controller already familiar to the majority of control engineers (i.e. users!) it is almost certainly more easily acceptable than newer, relatively untried self-tuners. This is highlighted by the relatively large numbers of 'auto-tuners' in Table 1.1 compared to those based upon more advanced algorithms.

This commercial move towards auto-tuning is predominantly motivated by the potential benefit of 'expert tuning by non-expert personnel', i.e. , the widespread use of auto-tuning would be expected to increase the overall quality of control throughout industry by making the quality of control in individual loops less dependent upon the expertise of individual control engineers. This is illustrated by the Foxboro EXACT controller which uses an 'expert systems' approach to tune a PID.

Note that one further immediate advantage of the auto-tuner is that, because of the already existing profusion of normal fixed parameter PID's, one auto-tuner could be used as a field commissioning and start-up tool for any number of individual loops.

1.2 LONG-RANGE PREDICTIVE CONTROL

One of the fundamental difficulties encountered throughout process control, but especially in the chemical industry, is the presence of time-delay (often referred to as 'dead-time'). This time-delay is often a result of flow-rate of material through a process and in many cases is time-varying, e.g. the residence time of material in a dryer represents a time-delay that can change significantly for different input consistencies. In practice this leads to the de-tuning of conventional controllers, such as fixed parameter PID, to ensure stability at the expense of less tight closed loop control (Shinskey, 1979; Clarke, 1984).

Consideration of this problem led to the development of *predictive* control strategies (Kramer, 1960; Seborg *et al*, 1986), whose essential feature was the replacement of the problem of controlling the current system output by that of controlling a predicted value of the output at some time in the future. This represents an attempt at isolating the effect of the time-delay from the actual process dynamics, thus allowing tighter control. The predicted value for the output is calculated from knowledge of the past inputs and a process model.

1.2.1 A review of previous algorithms

One of the earliest such algorithms designed to provide time-delay compensation was the Smith Predictor (Smith, 1959) where a predictive model is used as the basis of a standard control law and, in principle, the effect of the time-delay is removed from the closed loop. This was eventually followed by the Generalised Minimum Variance (GMV) algorithm of Clarke & Gawthrop (1975,79), an algorithm based upon k -step ahead prediction of the process output, where k is the assumed time-delay of the process. Despite gaining widespread acceptance and popularity in the literature this type of algorithm suffers from two major problems: if the initial assumption for k is incorrect (or subsequently becomes incorrect due to a change in the actual time-delay), or the estimated process model is non-minimum phase, they must be de-tuned (by constraining the variance of the input) in order to preserve stability. Ironically this is just like the PID controllers they sought to replace.

There are two major alternatives to this approach, namely *pole-placement* or *long-range prediction*. Because pole-placement (Wellstead *et al*, 1979; Åström & Wittenmark, 1980) is not based upon a cancellation-type of control law it can, in principle, overcome the problem

of time-delay (and/or non-minimum phase systems) simply by overparameterisation of the estimated model. However at least initially the disadvantage of this method was that such overparameterisation could lead to instability because of common factors causing numerical problems in its control calculation.

The latter alternative, long-range predictive control (LRPC), represented a reinterpretation of the significance of the predictive strategy to a more general approach that considered a *range* of future predictions (Dawkins & Briggs, 1965). This is expected to give a de-tuned control action that on one hand may give *slower*, but on the other, more *robust* response. Again because this is no longer a cancellation-type control law it is relatively insensitive to time-delay and/or non-minimum phasedness.

Two important early LRPC algorithms¹ were IDCOM (Richalet *et al*, 1978) and DMC (Cutler & Ramaker, 1980) which used deterministic impulse or step response models respectively. The former subsequently became known as Model Algorithmic Control (Rouhani & Mehra, 1982; Bruijn & Verbruggen, 1984). In terms of the incorporation of long-range prediction into adaptive control these methods have one particular disadvantage, their type of model structure. Whilst they can be easily obtained experimentally, from cross-correlation or reaction curve tests, they are relatively 'extravagant' in their parameterisation and not immediately suitable for modelling integrating (e.g. robotic) or open loop unstable (e.g. high performance aircraft) systems.

This led to the further development of LRPC algorithms based upon transfer-function models such as CARMA (Åström, 1970) which increase parsimony² and can simply represent integrating and/or unstable processes. These later algorithms included those of Peterka (1984) and Mosca *et al* (1984) which are versions of LQG for an input-output model rather than state-space; Ydstie (1982, 84) whose EHAC algorithm is based upon GMV but the output is predicted at a point well beyond the maximum possible dead-time of the process; and De Keyser & van Cauwenberghe (1982,85) whose EPSAC algorithm is a further modification to GMV where the output prediction is extended over a horizon of future values.

¹see Clarke & Zhang (1984) for a comparative study of IDCOM and DMC.

²the least number of parameters required to 'adequately' describe the system.

1.2.2 Generalised Predictive Control

Generalised Predictive Control (Clarke, Mohtadi & Tuffs, 1984; Mohtadi, 1986) is one of the latest members of the LRPC family and represents a generalisation of the earlier algorithms described above, combined with a simpler computational approach.

Using a CARIMA model description to ensure inherent integral action (Clarke & Tuffs, 1985) the open loop system step response is predicted over a horizon of future samples (see Fig. 1.2), an appropriate sequence of future control signals being calculated to take that future response to a pre-specified set-point *trajectory*, subject to a constraint that after a 'control horizon' all control increments are assumed to be zero (from the DMC algorithm of Cutler & Ramaker, 1980). This is therefore a *multi-step* control policy but because only the first control signal from this sequence is implemented, and the calculation repeated at each successive sample interval, it is also referred to as *receding horizon*. The combination of long-range prediction and the control horizon, if chosen correctly (Clarke *et al*, 1985), render GPC relatively insensitive to unknown and/or variable time-delay and non-minimum phase processes respectively.

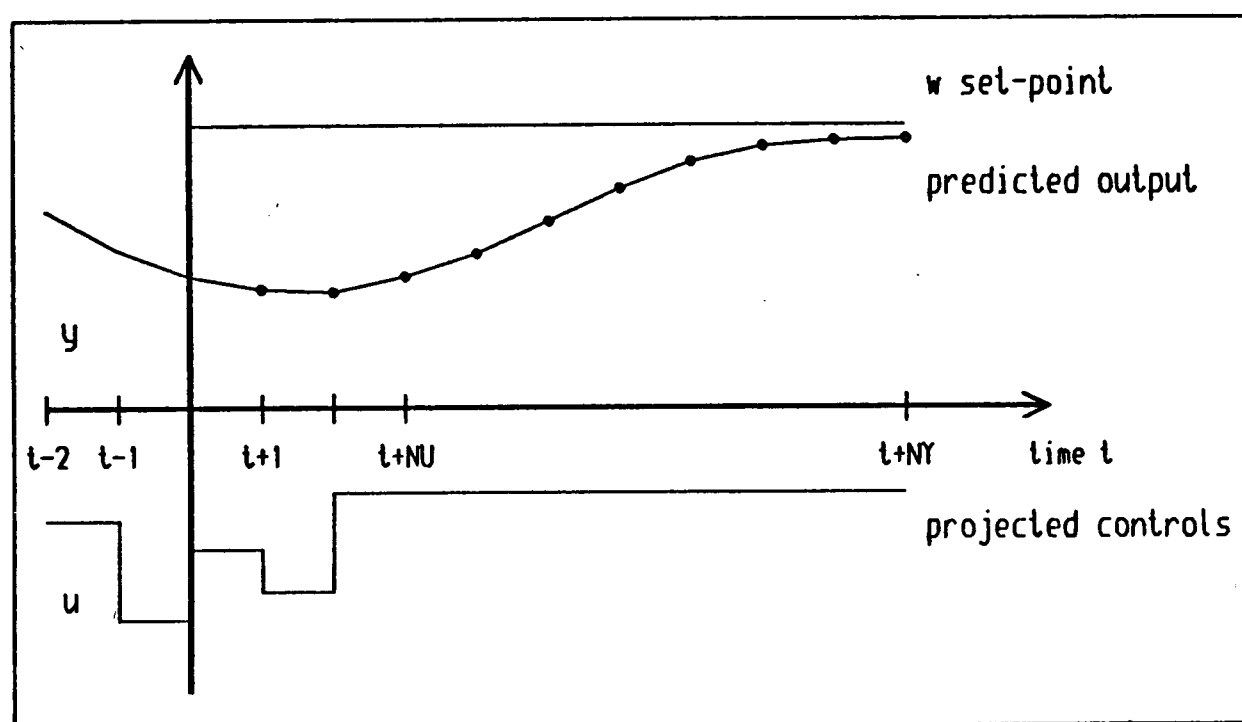


Figure 1.2: The GPC control calculation.

Because GPC also incorporates all the design parameters ('knobs'), i.e. P , T and λ , previously associated with GMV it may be considered as the long-range predictive extension of GMV. It is important to note, however, that it is also a more general form of most of the earlier CARMA model based LRPC algorithms, many of which may be simply

obtained by appropriate choice of GPC's prediction and control horizons (Mohtadi, 1986).

To summarise it is claimed that GPC offers enhanced robustness compared to earlier algorithms, such as GMV or pole-placement, with increased flexibility, and is therefore more suitable for practical application.

1.3 MULTIVARIABLE CONTROL

It is often the case in the process control industry that a process has more than one output measurement to be controlled. A typical example of this, often quoted in the literature, is the control of a binary distillation column (Morris *et al*, 1977,80,81,82) where both 'top' and 'bottom' compositions are required to be controlled to a certain percentage set point. The main control problem for this type of process, as distinct from those already encountered in SISO³ process control, is that often a given output may depend upon more than one input, resulting in possibly undesirable 'interaction' between separate loops that degrades the overall control performance. If each output depended only upon one particular input then obviously the problem reduces simply(?) to n parallel SISO control loops.

A multivariable process, whether already in operation or in the design stage, could perhaps be redesigned to remove or reduce these interactions between loops but in practice this would often be either impossible or involve prohibitively high capital costs. This leaves a requirement for a specifically multivariable control structure that attempts to remove these interactions, a so-called 'de-coupling' control strategy.

Historically multivariable processes have been controlled by one of three control strategies :

- Development of an accurate linear multivariable process model and the classical design of a de-coupling controller based upon this model (Rosenbrock, 1974).
- Development of n independent SISO control loops using such classical control strategies as PID's. Although the effect of interactions is ignored this 'multi-loop' strategy is capable of providing steady-state decoupling via the load disturbance rejection properties of individual loop controllers.
- Development of n SISO control loops linked to other interacting loops by means of simple *feed-forward* terms. These feed-forward terms are almost always simply static

³Single-Input Single-Output

rather than dynamic, e.g. a ratioed value from the control signal of another loop (Shinskey, 1979).

Whilst all of these have been successfully applied in practice they do suffer from a variety of problems. The requirement for an accurate model in the first of these approaches can be prohibitive for complex processes with long time delays; significant non-linearity; or whose dynamics vary with time. Thus *adaptive* multivariable control would seem to be justified for exactly the same reasons as for the SISO case considered earlier. Multi-loop controllers, where the controller for each loop is tuned independently with no information about other interacting loops, have been demonstrated in McFarlane (1972) to often lead to instability, with the interaction appearing as effectively unmeasurable load disturbances. In practice this often results in the detuning of such loops simply in order to retain stability, with a corresponding loss of control performance. Finally the choice of input-output pairings for such a control structure is critical as it determines the overall control performance. Whilst various techniques have been developed to assist the control engineer in this choice, for example Bristol's Relative Gain Array (Bristol, 1966; Shinskey, 1979), this is not always clear.

Thus the control of multivariable processes would appear to be an ideal area for the application of modern model based adaptive control algorithms such as minimum variance, pole-placement, LQG etc. Indeed most of the recent adaptive multivariable controllers since, and including, Borison's multivariable extension of the minimum variance algorithm (Borison, 1979), are often relatively straightforward extensions of their SISO counterparts.

1.3.1 A review of previous algorithms

In this section a review of the main developments in adaptive or self-tuning multivariable control algorithms is presented, starting with the multivariable generalisation of the CARMA (Åström, 1970) process model:

$$\mathcal{A}(q^{-1})\mathbf{y}(t) = \mathcal{B}(q^{-1})\mathbf{u}(t - k) + \mathcal{C}(q^{-1})\boldsymbol{\xi}(t) \quad (1.1)$$

where $\mathbf{y}$, $\mathbf{u}$, and $\boldsymbol{\xi}$ are the output, input, and noise *vectors* respectively, all of dimension n . $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are all *matrices* of dimension $n \times n$, whose elements are polynomials in the backward shift operator, e.g.,

$$B_{ij}(q^{-1}) = b_0^{ij} + b_1^{ij}q^{-1} + \dots + b_{\delta_B}^{ij}q^{-\delta_B}$$

This may of course be rewritten as a polynomial in q^{-1} where the coefficients are themselves matrices, i.e.

$$\begin{aligned} B(q^{-1}) &= B_0 + B_1 q^{-1} + \dots + B_{\delta B} q^{-\delta B} \\ \mathcal{A}(q^{-1}) &= A_0 + A_1 q^{-1} + \dots + A_{\delta A} q^{-\delta A} \\ \mathcal{C}(q^{-1}) &= C_0 + C_1 q^{-1} + \dots + C_{\delta C} q^{-\delta C} \end{aligned} \quad (1.2)$$

where it is frequently assumed that $A_0 = C_0 = I$, that B_0 is non-singular, and that the roots of $\det C$ lie inside the unit circle. A minimum time-delay, k , common to all the loops of the system is included, further time-delays may be incorporated by extension of the corresponding $B_{ij}(q^{-1})$ polynomial.

Initial attempts to provide an adaptive multivariable controller were based upon the so-called 'Self-Tuning Regulator' of Åström & Wittenmark (1973), with the above assumptions and the further assumption that $\det B(q^{-1})$ has all its roots inside the unit circle (Borison, 1975, 1979; Keviczky et al, 1978; Bayoumi & El-Bagoury, 1979). These assumptions, that the time-delays for each loop are the same and known, and that the process is minimum phase, were already known to be restrictive in the SISO case and had led to the development of the GMV algorithm (Clarke & Gawthrop, 1975, 79) which provided stable control of non-minimum phase processes with time-delay through the use of control weighting. Extension of the GMV control algorithm to the multivariable case was therefore to become one of the major directions of research activity (Koivo, 1980; Keviczky & Kumar, 1981; Bayoumi *et al*, 1981; Scattolini & Clarke, 1985).

Further work basically involved the increased generalisation of adaptive multivariable controllers. New methods allowed the consideration of 'non-square' systems, i.e. those with differing numbers of inputs and outputs (Wong & Bayoumi, 1981; Goodwin, McInnis & Long, 1982; Favier & Hassani, 1982; Grimble & Moir, 1983), others allow a different sampling rate for each output variable (Morris *et al*, 1981, 82; Chien *et al*, 1983; Costin & Buchner, 1983). The algorithms of Grimble & Moir (1983) and McDermott (1984) provide stable control of processes which are both open loop unstable and non-minimum phase.

Throughout this work the concept of time-delay proved to be a significant problem because of the widespread use of minimum-variance based control strategies. This led to the specification of the so-called *interactor matrix* as a general description of multivariable time-delay structure by Goodwin & Long (1980) and Elliot & Wolovich (1982) respectively, based upon earlier work by Wolovich & Falb (1976). The definition of this interactor matrix

follows from one possible definition of time-delay in the SISO case, which states that there exists a scalar function $T_d(q)$, of the form $T_d(q) = q^k$, such that :

$$\lim_{q \rightarrow \infty} T_d(q)A(q^{-1})^{-1}B(q^{-1}) = K \quad (1.3)$$

where K is the first non-zero value after the assumed time-delay.

Similarly in the MIMO case there exists a polynomial 'delay-matrix', $\mathcal{T}_d(q)$, which may be defined as satisfying :

$$\begin{aligned} \text{(i)} \quad & \det \mathcal{T}_d(q) = q^m, & m \text{ is an integer.} \\ \text{(ii)} \quad & \lim_{q \rightarrow \infty} \mathcal{T}_d(q)\mathcal{A}(q^{-1})^{-1}\mathcal{B}(q^{-1}) = \mathcal{K}, & \mathcal{K} \text{ is a non-singular matrix.} \end{aligned} \quad (1.4)$$

This delay matrix is in general non-unique, the *interactor matrix* representing the special case when $\mathcal{T}_d(q)$ is defined as having a lower diagonal structure, i.e.

$$\mathcal{T}_d(q) = \mathcal{H}(q)\mathcal{D}(q) \quad (1.5)$$

where $\mathcal{H}(q)$ is a lower triangular unimodular matrix with 1's on the main diagonal, and $\mathcal{D}(q)$ is of the form :

$$\mathcal{D}(q) = \text{diag} \begin{bmatrix} q^{k_1} & \dots & q^{k_n} \end{bmatrix} \quad (1.6)$$

It is thus apparent that the earlier results of Borison (1979), which assumed a common delay between each input/output combination, and Goodwin *et al* (1980), which assumed a different time-delay for each input/output combination, were merely special cases in which the interactor matrix had been taken to be a scaled identity matrix, and a diagonal matrix, respectively. Subsequently the paper of Dugard *et al* (1984) examined the role of this interactor matrix in adaptive multivariable control and concluded that :

The [multivariable] extension is straightforward when the system interactor matrix is diagonal but presents some unexpected difficulties in the general case. Dugard, Goodwin & Xianya (1984)

The specification of the interactor matrix as a lower diagonal form of delay matrix, with no apparent justification apart from enabling the subsequent control design, means that the actual interactor matrix elements themselves depend upon the *ordering* or *pairing* of the process input and output variables. As the derivation usually relies upon the predictor $\mathcal{T}_d(q)y(t)$ the use of such an interactor matrix in the derivation results in a control law that:

- (i) minimises the variance of $y_1(t)$ about $w_1(t)$.
 - (ii) minimises the variance of $y_2(t)$ about $w_2(t)$,
subject to the constraint of achieving (i) for all time.
 - (iii) minimises the variance of $y_3(t)$ about $w_3(t)$,
subject to the constraint of achieving (i) and (ii) for all time.
- and so on ...

Therefore the variance of a particular output's control error itself depends upon the ordering or pairing of the process inputs and outputs, with $y_1(t)$ achieving the tightest control. Moreover the overall sum of the output control errors may be shown to also depend upon the ordering of the inputs and outputs.

For this reason, and also because of the restriction that the interactor matrix be known *a priori*, inappropriate in an adaptive context, Dugard *et al* (1984) proposed an alternative multivariable control law that does not directly depend upon knowledge of the interactor for its derivation. Instead it involved the multivariable extension of an algorithm by Ydstie (1982) that, in the SISO case, was relatively insensitive to process time-delay, and thus in the MIMO case did not require the assumption of $\det \mathcal{B}(q^{-1})$ being non-singular. Similarly Scattolini & Clarke (1985) use the assumption of a maximum bound on the time-delay between any input/output loop to avoid the use of an interactor matrix, but use the CARIMA plant model description to introduce inherent integral action into the closed loop.

Minimum variance based multivariable control algorithms were, of course, not the only multivariable controllers being developed throughout this period. Prager & Wellstead (1980) developed a multivariable self-tuning regulator based upon the popular SISO pole-placement control algorithm, that was capable of controlling non-minimum phase plant with unknown and/or variable time-delays. The presence of time-delay is incorporated simply by extension of the order of the elements of the polynomial matrix, $\mathcal{B}(q^{-1})$ and the use of an explicit formulation for the control calculation. The problems associated with this method are principally that of complexity, with the control calculation involving the solution of two sets of simultaneous equations at each sample interval, and the estimation of large numbers of parameters in processes with long time-delay. With the rapid acceleration of computing power in recent years the former no longer represents a serious objection, however parameterisation remains a significant issue. Further work in this area includes that of Vogel & Edgar (1982), who achieved a significant computational saving by making the number of parameters in each element of $\mathcal{B}(q^{-1})$ dependent only upon the expected

range for the time-delay of that element, McDermott & Mellichamp (1984) who developed a de-coupling multivariable controller with pole-placement properties and capable of controlling unstable processes, and Elliot *et al* (1984).

Adaptive multivariable controllers based upon the Linear-Quadratic-Gaussian algorithm have also been described (Peterka & Åström, 1973; Kučera, 1979; Åström, 1980; Lam, 1980; Grimble *et al*, 1982; Grimble, 1983). These once again require greatly increased computational power as the control calculation for this algorithm performs repeated spectral factorisations.

1.3.2 Multivariable Generalised Predictive Control

It is against this backdrop that the multivariable version of the Generalised Predictive Control algorithm ('MGPC') is proposed as a practical MIMO adaptive controller. There are four principal justifications for this claim:

- MGPC is expected to extend the practical features of the SISO controller (see section 1.2.2) to the multivariable case, i.e. offset-free regulation because of inherent integral action, the specification of control action suitable for a wide range of process control applications, the incorporation of dynamic feed-forward, on-line specification of a variety of control objectives by variation of GPC's design parameters, ease of commissioning through the use of self-tuning, etc. The latter is especially important for complex multivariable processes with significant interaction.
- The long-range predictive nature of MGPC allows control of processes with unknown and/or varying time-delay in the SISO case, and is thus expected to neatly side-step the requirement for the interactor matrix in the MIMO case. Therefore MGPC may be considered as bringing together the two main areas of multivariable control described above, i.e. those based upon minimum variance and those based upon pole-placement respectively.
- MGPC is insensitive to input-output pairing and thus does not require prior knowledge of *dominant* input-output combinations. These should be automatically identified by the estimation of a 'full' multivariable process model.
- For a $n \times n$ plant MGPC effectively solves n simultaneous equations to calculate the n control signal sequences and thus decoupling of loops that exhibit interaction is

expected to be far quicker than multi-loop control structures that rely upon the load disturbance rejection properties of individual loops for decoupling.

1.4 PARAMETER ESTIMATION

Parameter estimation has been defined as:

The experimental determination of values of parameters that govern the dynamic behaviour of a process, where the structure of the process model is assumed to be known. (Eykhoff, 1974)

For self-tuning or adaptive control application this means that the parameters in an assumed dynamic model are estimated on-line from input/output data. The parameters of the control law are then updated as if the estimated parameters were exact, the so-called *certainty equivalence* principle (Åström, 1975).

The dynamic model has typically been assumed to be a linear difference equation model, such as the well-known CARMA⁴ model (Åström & Wittenmark, 1973) shown below:

$$A(q^{-1})y(t) = B(q^{-1})u(t - k) + C(q^{-1})\xi(t) \quad (1.7)$$

where $\xi(t)$ is a zero-mean disturbance term with Gaussian distribution, k is the integral dead-time of the system, and the 'process parameters' $A(q^{-1})$, $B(q^{-1})$ and $C(q^{-1})$ are polynomials in the delay operator, q^{-1} :

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{\delta A}q^{-\delta A}$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \dots + b_{\delta B}q^{-\delta B}$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_{\delta C}q^{-\delta C}$$

Adaptive control is usually based upon simultaneous model identification and control and thus requires real-time updating of the model parameters rather than batch-wise processing of the input/output data. Algorithms that are suited to such real-time usage employ successive updates of the model parameters and are called *recursive*. They can normally be considered to have the following structure:

$$\begin{bmatrix} \text{New} \\ \text{Estimate} \end{bmatrix} = \begin{bmatrix} \text{Old} \\ \text{Estimate} \end{bmatrix} + \begin{bmatrix} \text{Gain} \end{bmatrix} \times \begin{bmatrix} \text{Direction} \end{bmatrix} \times \begin{bmatrix} \text{Error} \\ \text{Function} \end{bmatrix}$$

⁴CARMA : Controlled Auto-Regressive Moving Average

Of the many parameter estimation techniques that are available recursive least squares and extended least squares (Young, 1968) have been the most widely used, but recursive versions of other methods such as maximum likelihood and instrumental variables have also received attention (Åström *et al*, 1977; Isermann, 1982).

One common feature of these algorithms is that as progressively more data is 'processed' the gain tends rapidly towards zero (typically in 10 to 20 iterations), thus leading to *convergence* of the parameters. This is ideal for self-tuning applications (see page 3) but is obviously not suitable for the adaptive control of time-varying processes where, after self-tuning, some form of adaptivity must still be retained if the estimator is to 'track' any subsequent changes in process parameters.

This has led to the development of several algorithms, most of which are modifications of the basic RLS algorithm, that aim to preserve its sensitivity to process variations by discounting old data whose contribution is assumed to be obsolete. Traditionally this has been implemented by the use of a *fixed exponential forgetting factor* (Peterka, 1970; Åström & Wittenmark, 1973). Whilst this does indeed retain adaptivity it is now well-known that it is not a practical algorithm as the absence of exciting data for prolonged periods can lead to 'blowup' (Åström, 1983; Hodgson, 1982), where the gain of the estimator grows exponentially. This makes the algorithm very sensitive to small perturbations in the data vector (e.g. caused by noise) and hence can cause large changes in the estimated parameters and correspondingly poor control. The latter is often referred to as 'bursting' (Hodgson, 1982).

To overcome this further problem several *variable* forgetting factors have been introduced, the intention being to make the amount of forgetting depend upon both the amount of available data *and* the quality of the current estimated parameters (e.g. Åström, 1980; Fortescue *et al*, 1981; Wellstead & Sanoff, 1981; Ydstie, 1982; Dexter, 1983; Hägglund, 1983). Thus in the absence of exciting data adaptation should be suspended to prevent blow-up, whereas in the event of a sudden parameter change it should automatically be increased. Important alternatives include the 'constant trace' algorithms of Irving (1979) and Goodwin *et al* (1985), which attempt to retain a constant amount of adaptivity and thus avoid blow-up because of the inherent upper bound; and the 'covariance resetting' algorithm of Goodwin & Sin (1984) which by periodically resetting the estimator also retains adaptivity and avoids blow-up.

In terms of practical estimation it is well-known (Clarke, 1981; Seborg *et al*, 1986) that relatively simple *ad hoc* modifications, often referred to as 'estimator jackets', can improve robustness and long-term integrity. These include *estimation dead-bands* (i.e. an ON/OFF mechanism for suspending estimation during periods of low excitation to prevent parameter drift), normalisation of data (i.e. to improve numerical conditioning) and parameter constraint (i.e. to prevent parameters moving into non-allowable regions thus avoiding 'silly' results). Interestingly there has been renewed *theoretical* interest in this area because such modifications have been found to be vital for the proof of convergence and stability results in the presence of bounded noise and unmodelled dynamics (Egardt, 1979; Samson, 1983; Martin-Sanchez, 1984; Praly, 1984; Goodwin *et al*, 1985; Ortega *et al*, 1985; Kreisselmeier & Anderson, 1986).

As an illustration of the state-of-the-art in parameter estimation consider the representative 'Improved Least Squares' algorithm of Sripada & Fisher (1987). Based upon RLS this includes:

An ON/OFF criteria to prevent parameter drift during periods of low excitation; a variable forgetting factor which maintains the trace of the covariance matrix at a user-specified value; data preprocessing [i.e. band-pass filtering] and normalisation to improve numerical accuracy; scaling of the data vector to minimise the condition number of the covariance matrix [again to improve numerical conditioning] and finally parameter projection to constrain the estimates to a priori specified regions. Sripada & Fisher (1987)

All these methods, however, rely for long-term adaptivity upon the concept of the 'continuous retention of a small amount of adaptivity'. Thus inherently they are susceptible to the presence of noise and /or unmeasurable load disturbances, the former causing parameter drift and the latter large incorrect parameter variations (especially for variable forgetting factors which can over-react).

To overcome these problems a more cautious approach is the use of continuous adaptation combined with some form of model evaluation or FDI⁵ procedure. Adaptivity would then be enabled, or increased, only if a *genuine* process change is indicated. Thus, in

⁵ A procedure that seeks to improve the performance of a parameter adaptation algorithm in the presence of sudden large changes in the process is commonly referred to as *Fault Detection and Identification* (Willsky, 1976; Isermann, 1984). The term 'fault' is used irrespective of the nature of the process change and does not necessarily imply an actual physical fault in the process. Thus, for example, the 'fault' could arise due to a sudden change of operating point in a non-linear system.

principle, all apparent process changes would be ‘provisionally’ tracked but new parameters could subsequently be discarded if they did not prove to be clearly superior to earlier values. This is a more *cautious* approach that in principle retains the tracking properties of the other algorithms discussed above but is less susceptible to noise and load disturbances causing incorrect parameters. It is this type of parameter estimation strategy that is adopted in Chapter 3 of this thesis.

1.5 PROCESS CONTROL APPLICATIONS

Rapid developments in the theory of self-tuning/adaptive control alongside increasingly affordable and powerful micro-processor technology led almost immediately to practical applications. Two of the earliest involved the control of paper machines (Cegrell & Hedqvist, 1975) and an ore crusher (Borison & Syding, 1976) and were based upon the ‘self-tuning regulator’ of Åström & Wittenmark (1973). Since then there have been numerous applications of a variety of algorithms (although the majority used GMV) and for a wide range of processes (see the survey papers of Åström *et al*, 1977; Parks *et al*, 1980; Åström, 1983; Isermann, 1985; Seborg *et al*, 1986).

The earliest recorded application of long-range predictive control involved the SISO control of furnace outlet temperature and the MIMO control of a hydrocracker process where it was claimed that the DMC algorithm :

Made it possible to solve complex control problems on a digital computer which are not solvable with traditional PID concepts. Cutler & Ramaker (1980)

The former process was subject to large dead-time and was sensitive to variations in inlet air temperature. It was demonstrated that the use of LRPC, combined with a feed-forward signal from the inlet air temperature, dramatically improved control performance from that obtained with a conventional PID controller. This was, however, not an adaptive implementation.

Table 1.2 summarises the experimental applications of specifically long-range predictive control⁶ to industrial plant. The large number of applications reported of such a relatively recent control strategy is encouraging. Two important points that become clear from these results are the suitability of LRPC for the control of processes with significant *dead-time*

⁶see Seborg *et al*, 1986 for a similar tabular synopsis of practical applications of other self-tuning control strategies such as GMV, Pole-placement and LQ.

(De Keyser & Van Cauwenberghe, 1981; Martin & Van Horn, 1982; Hiram & Kershenbaum, 1985; De Keyser, 1985; Karny *et al*, 1985; Luecke *et al*, 1985; M'Saad *et al*, 1987) and the benefits of using *feed-forward* (Karny *et al*, 1985; Jota, 1987). This confirmed the earlier results of Cutler & Ramaker (1980).

Applications of Long-Range Predictive Control			
Type of process	Reported by	Controller	
Absorbtion Plants	Ydstie (1982) [†]	SISO	EHAC
	Chesna & Ydstie (1985) [†]	SISO	EHAC
	Ydstie <i>et al</i> (1985) [†]	SISO	EHAC
	Hiram & Kershenbaum (1985) [†]	MISO	EHAC
Engine test-bed	Tacey (Ricardo Ltd.) (1987)	SISO	GPC
	Hesketh & Sandoz (1987)	MIMO	LQ
Distillation columns	Richalet <i>et al</i> (1978)	SISO/MIMO	IDCOM
	Luecke <i>et al</i> (1985)	SISO	DMC
Heating systems	De Keyser (1985)	SISO	EPSAC
	Chesna & Ydstie (1985)	SISO	EHAC
	De Keyser & Van Cauwenberghe (1985)	SISO	EPSAC
	Jota (1987) [†]	SISO/MIMO	GPC
Robot manipulators	Lambert, M. (1987)	SISO	GPC
	Lelić & Wellstead (1987)	SISO/MIMO	GPP
Chemical reactors	Cutler & Ramaker (1980)	MIMO	DMC
	Engrand (1980)	?	IDCOM
	Martin & Van Horn (1982)	MIMO	IDCOM
	Cutler & Hawkins (1987)	MIMO	DMC
Extruder	M'Saad <i>et al</i> (1987) [†]	SISO	LQ, GPC
Tracking system	Favier (1987)	MIMO	GPC
Furnace	Richalet <i>et al</i> (1978)	MIMO	IDCOM
	Cutler & Ramaker (1980)	SISO	DMC
	De Keyser & Van Cauwenberghe (1981)	SISO	EPSAC
	Karny <i>et al</i> (1985)	SISO	LQ
Steam boiler	Richalet <i>et al</i> (1978)	MIMO	IDCOM
	Karny <i>et al</i> (1985)	SISO	LQ
Dryers	Lambert, E. (1987)	SISO	GPC
	Lambert, E. (1987)	MIMO	GPC
Cement mill	Al-Assaf (1987)	SISO	GPC
Rolling mill	Karny <i>et al</i> (1985)	SISO	LQ
Dredging ship	De Keyser (1986)	SISO	EPSAC
	De Keyser & Van Cauwenberghe (1985)	SISO	EPSAC

Table 1.2: Synopsis of applications using Long-Range Predictive Control

Of particular interest is one novel application of LRPC is described in De Keyser & Van Cauwenberghe (1981) where for the control of a blast furnace their LRPC algorithm was used, not in closed-loop, but as a self-tuning predictor based *operator guide*. Unassisted manual control was previously difficult because of significant dead-time and complicated dynamics and was dramatically improved by this scheme.

[†] denotes the experimental application to laboratory-scale plant

At Oxford, continuing the effort to take self-tuning out of the laboratories and into industry that began with SESAME (Clarke & Gawthrop, 1981; Hodgson, 1982) and later with FAUST (Tuffs, 1984) there have been several recent applications. In Al-Assaf (1987) the application of GPC to the SISO control of temperature in a cement grinding mill is reported. Using a simple step-test commissioning procedure and subsequently fixed-parameter control GPC was shown to completely avoid overshoot on start-up, as compared to the plant's current PI controller which was slower to achieve the operating point and gave a considerable overshoot. In Lambert, M. (1987) GPC was applied to the high performance positioning of a real flexible single-link robot. The higher order of the self-tuned control law combined with the use of pre-specified set point trajectories (commonly available in robotic applications) reliably provided far superior control to that of a 'well-tuned' PID, with almost exact set-point following. Of particular interest in this application was the multi-tasking implementation of GPC, which allowed sample rates in excess of 100Hz, and the use of continuous adaptivity to maintain the quality of control despite changes of inertial load. Chapters 4 and 5 describe the application of SISO and MIMO GPC to the control of industrial spray dryers.

1.6 STRUCTURE OF THE THESIS

This thesis is predominantly concerned with the practical application of long-range predictive control, in the form of the GPC algorithm (Clarke, Mohtadi & Tuffs, 1984,87). This is a self-tuning/adaptive control strategy and thus depends upon two elements, the control law and the parameter adaptation algorithm. These are considered separately before GPC is applied to two actual industrial control problems.

The structure of the thesis is outlined below. A more detailed introduction is given at the start of each individual chapter.

Chapter 2 : reviews the algorithmic details and development of GPC. The closed-loop characteristic equation is derived and, using a GPCTEST package, the closed-loop pole locations are examined for a variety of control scenarios. The results verify earlier analytic and intuitive results (Mohtadi, 1986) and provide an insight into GPC's specification in practice. It is shown how GPC may be extended to the control of multivariable systems and two control structures developed: 'full' multivariable GPC and multi-loop GPC.

Chapter 3 : seeks to identify the fundamental areas and problems of parameter estimation in practice. The well-known RLS algorithm is selected as a good ‘prototype’ and several modified algorithms that have been designed to extend it to the tracking of time-varying systems are outlined and compared in simulation. Two novel estimation algorithms, CLS and DLS, are introduced which use parameter weighting to overcome such practical problems as ‘blowup’ and parameter drift. When combined with an inherent fault detection mechanism, the former is shown to be particularly suitable for estimation in the presence of load disturbances.

Chapter 4 : describes the successful application of single-input single-output GPC to the control of a Lever Brothers industrial soap drying process. These trials were conducted during full-scale soap production and demonstrated a significant improvement in control of the soap moisture compared to manual control. Despite severe production constraints commissioning of GPC was straightforward using PRBS assisted self-tuning, with most results obtained using a ‘default’ GPC specification.

Chapter 5 : reports a further successful application of GPC to the multivariable (2×2) inner loop control of a Unilever Research spray drying tower. The results demonstrated that the quality of individual loop control, and overall decoupling, using ‘default’ multivariable GPC was dramatically improved compared to that achieved by the multi-loop gain-scheduled PID control system used in normal operation. Commissioning was again straightforward using PRBS assisted self-tuning and did not require any prior knowledge of input-output pairing or time-delay structure. The effect of GPC’s design parameters (P , T and λ) upon control performance was examined. Finally a variety of alternative (simplified) multi-loop GPC controllers were applied and it was demonstrated that the use of feed-forward gave decoupling performance indistinguishable from that of full multivariable GPC.

Chapter 6 : draws conclusions from the results presented in the thesis and presents a series of suggestions for further work in this area.

Chapter 2

Generalised Predictive Control

An adaptive control algorithm is defined in Chapter 1 to consist of two distinct procedures: on-line *identification* of a process model using a parameter adaptation algorithm, and a *control* strategy that uses this model as the basis for closed loop control (Fig. 2.1). Generalised Predictive Control, as its name suggests, falls into the latter category and it is the development and interpretation of this algorithm that forms the subject of this chapter. The estimation aspect of adaptive control is considered separately in Chapter 3.

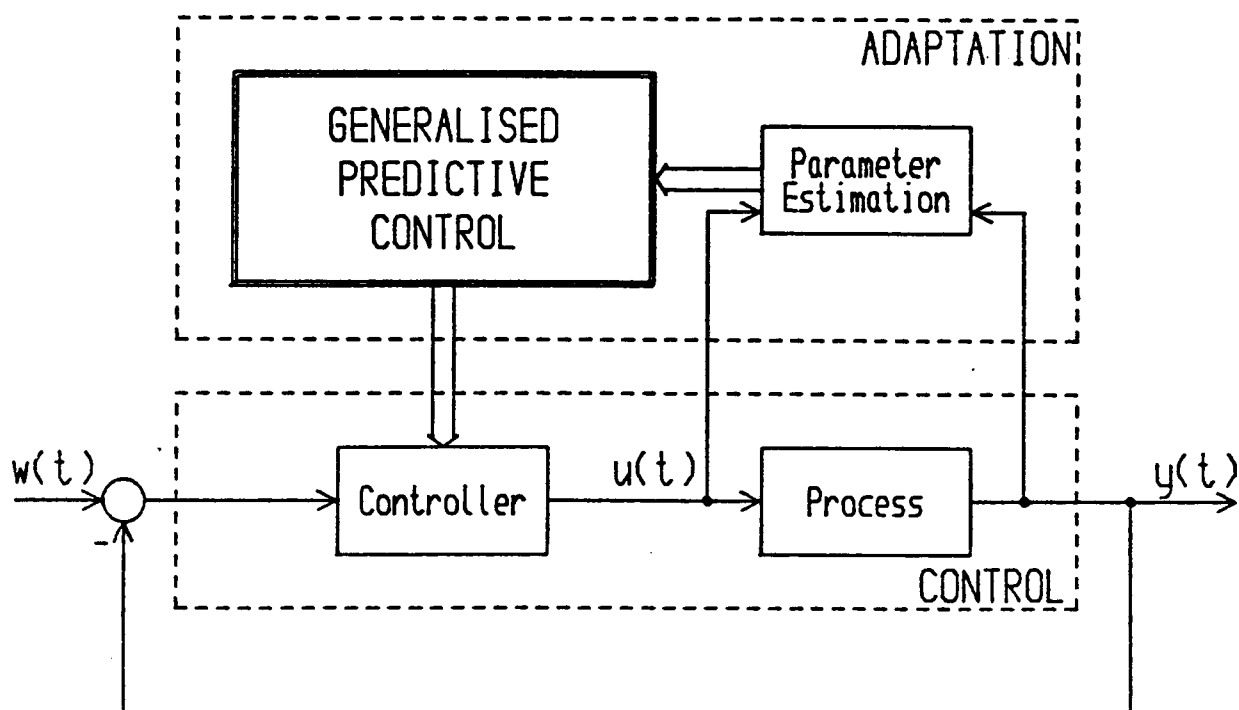


Figure 2.1: Adaptive Control: the control algorithm

2.1 INTRODUCTION

Generalised Predictive Control (GPC) is one of the most recent self-tuning controllers to be reported (Clarke *et al*, 1984,87; Mohtadi, 1986; Lambert, M., 1987; Favier, 1987; M'Saad

et al, 1987), and combines several important features:

- The use of *long-range prediction* (Richalet *et al*, 1978) where instead of directly controlling the output, the output is predicted over a horizon of future samples (from NS to NY samples in the future) and a sequence of future control signals generated to take these values to a set point trajectory (Fig. 1.2). This is expected to increase robustness of the controller to the presence of unknown, or varying, time-delay.
- The use of a CARIMA process model (Tuffs, 1984) to provide these predictions in an incremental form. This inherently introduces integral action into the closed loop and guarantees offset-free control without the requirement for knowledge of d.c. levels associated with earlier CARMA based controllers (Hodgson, 1982).
- The assumption of a *control horizon*, NU , after which all projected control increments are assumed to be zero (Cutler *et al*, 1980). By constraining the calculation for the control signal in this fashion GPC provides stable control of non-minimum phase processes without the use of arbitrary values of control weighting (i.e. λ) associated with GMV (Clarke & Gawthrop, 1975, 79; Clarke, 1984).
- The use of a computationally efficient recursion of the Diophantine equation to calculate the predicted outputs.
- The incorporation of all the design parameters (or so-called 'tuning knobs') of the earlier GMV controller, namely P , T and λ . Combined with the horizons associated with long-range prediction, this allows the specification of a wide range of closed loop control objectives.

Whilst all these features (and especially the 'horizons') are expected to enhance the robustness and versatility of the GPC algorithm there is however a problem because closed loop analysis of the GPC algorithm, in order to determine such useful quantities as pole locations, stability margins etc., becomes less straightforward¹ than earlier algorithms such as GMV or pole-placement (Wellstead *et al*, 1979). Thus the derivation of a meaningful expression for the closed-loop is precluded, except for certain special cases² which are not likely to be encountered in practice. Nonetheless for a process that is assumed to be known exactly, with a given set of parameter estimates, and for a particular specification of the

¹ see Lelić & Zarrop (1987) for an illustrative worked example.

² e.g. when GPC is commissioned to resemble GMV or pole-placement (Mohtadi, 1986).

GPC algorithm, it is possible to express the control law in a transfer function form. This opens up two important possibilities:

- Development of a fixed-parameter transfer function implementation of GPC suitable for high-speed application.
- The application of a variety of standard analytic techniques, such as Nyquist diagrams and root loci.

It is the latter that is of particular interest in this chapter. Knowledge of how GPC performs under ideal conditions is an essential pre-requisite for its application: it not only verifies the predictions of performance obtained from theoretical analysis (Mohtadi, 1986), but also provides an insight into practical considerations such as (for example) the specification of GPC that would be required for the control of ‘difficult processes’, or how to increase robustness in the presence of model-plant mismatch.

Thus in section 2.2 the development of the basic GPC algorithm is described followed in 2.3 by the incorporation of all the extensions previously associated with GMV. The main bulk of the chapter is found in section 2.4 where an algorithm for calculating GPC’s closed loop characteristic equation is developed and a GPCTEST program is used to illustrate the effect of GPC’s design parameters upon the closed loop. By generating discrete closed loop pole loci and system step responses for different GPC specifications it is verified that such control strategies as ‘minimum-variance’, ‘state dead-beat’, and ‘mean-level’ can be obtained by appropriate choices for design parameters. Where appropriate these results are further supported by corresponding simulation studies. Included in this section is the effect of model-plant mismatch upon GPC for the well-known Rohrs’ example (Rohrs *et al*, 1984) and how the T polynomial may be used under such circumstances as an aid to robustness.

Finally section 2.5 considers the multivariable extension of GPC and describes the development (Mohtadi *et al*, 1986) of a ‘full’ MGPC algorithm (so-called because of its basis upon a ‘full’ multivariable process model). A selection of alternative controllers, based upon SISO GPC controllers in a multi-loop structure (DGPC), with or without feed-forward, are then introduced. Based upon the control of a 2×2 spray drying tower simulation it is demonstrated that, once self-tuned, both MGPC and DGPC are capable of tight control with neither requiring prior knowledge of the system interactor matrix. In particular it is shown that MGPC is insensitive to input-output ordering, but that

decoupling is sensitive to scaling of the data in individual loops. The use of feed-forward is shown to dramatically improve decoupling using DGPC which is otherwise unaffected by scaling.

2.2 THE BASIC GPC ALGORITHM

The development of the Generalised Predictive Control algorithm proceeds initially by definition of an appropriate model, suitable for the prediction of future process outputs. Then, as GPC is a long-range predictive algorithm, this is extended to produce a range of future predictions. Finally a quadratic cost-function is minimised to generate the required control signals.

2.2.1 The CARIMA process model

The Controlled Auto-Regressive *Integrated* Moving Average process model description is shown below:

$$A(q^{-1})y(t) = B(q^{-1})u(t - k) + \frac{C(q^{-1})}{\Delta}\xi(t) \quad (2.1)$$

where $y(t)$ and $u(t - k)$ are the process output and input respectively, the latter delayed by an assumed time-delay, k ; Δ is the difference operator, $\Delta = 1 - q^{-1}$, $\xi(t)$ is an uncorrelated random noise sequence with zero mean and A and B are polynomials in the backward shift operator q^{-1} , i.e.

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{\delta A}q^{-\delta A} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + b_{\delta B}q^{-\delta B} \end{aligned}$$

Although $C(q^{-1})$ is another polynomial in the delay operator for simplicity will be chosen as 1. Thus:

$$A(q^{-1})y(t) = B(q^{-1})u(t - k) + \frac{\xi(t)}{\Delta} \quad (2.2)$$

There are many reasons for the choice of this type of model structure, the linear difference equation, but essentially it is simple, suitable for discrete computer implementation, and readily manipulated into a predictive form. The particular choice of the CARIMA model, instead of earlier models such as CARMA (see section 1.4), which used positional data, is made on the basis that the noise term $C(q^{-1})\xi(t)/\Delta$ is a more realistic model of typical

process load disturbances, admitting interpretation as either random steps at random intervals, e.g. a sudden change of load for a robotic manipulator (Lambert, M., 1987); or as drifts, e.g. quality variations in feed-flow (De Keyser, 1986). In practice, however, the main justification for the use of this model structure is that it forces the use of incremental data in the control calculation. Use of a CARMA model requires the estimation of a d.c. offset term to guarantee offset free control, which has been found to be unsatisfactory (Hodgson, 1982). The incremental CARIMA model avoids this requirement and has been demonstrated to correspond to the inherent inclusion of an integrator in the closed-loop (Clarke, 1982). This follows from the *Internal Model Principle* of Francis & Wonham (1976).

Although the model is linear-in-the-parameters its use, especially in the context of adaptive control, may be justified as a form of local linearisation of the process about its operating point. It will be assumed throughout the remainder of this chapter that the parameters of this model are known.

2.2.2 Long-range prediction

A long-range predictive controller requires a series of predictions of future values of the output, i.e. $\{y(t+j), j=1,2,\dots\}$, given information up to time t and assumptions about future control activity. Thus, following the standard derivation of Clarke *et al* (1987) the CARIMA plant model description of (2.1) is rearranged into a form suitable for the generation of j -step ahead predictions, using the so-called Diophantine identity:

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) \quad (2.3)$$

where E_j and F_j are polynomials of degree $j-1$ and $\delta A - 1$ respectively, and uniquely defined given $A(q^{-1})$ and the prediction interval j . Substitution of this identity into (2.1) gives a prediction equation for $y(t+j)$ given measured data up to time t :

$$\hat{y}(t+j | t) = G_j(q^{-1})\Delta u(t+j-1) + F_j(q^{-1})y(t) \quad (2.4)$$

where $G_j(q^{-1}) = E_j(q^{-1})B_j(q^{-1})$ and the associated *prediction error* is,

$$e(t+j | t) = E_j(q^{-1})\xi(t+j) \quad (2.5)$$

Thus, in principle, it is now possible to numerically solve the Diophantine equation (2.3) for each prediction interval, j , and substitute into (2.4) to obtain the required predictions.

In practice it is far more efficient to solve (2.1) for initial values of E_i and F_i and then to calculate subsequent values of E_{i+1} and F_{i+1} using recursion of the diophantine equation.

Alternatively the prediction equation (2.4) may be rewritten as:

$$\hat{y}(t+j) = \tilde{G}_j(q^{-1})\Delta u(t+j-1) + \tilde{G}_j(q^{-1})\Delta u(t-1) + F_j(q^{-1})y(t) \quad (2.6)$$

$$= \tilde{G}_j(q^{-1})\Delta u(t+j-1) + \hat{y}(t+j | t) \quad (2.7)$$

thus the prediction $\hat{y}(t+j)$ may be split into two distinct parts :

- $\hat{y}(t+j | t)$, the prediction of the output assuming no future changes in the control signal. This corresponds to the ordinates of the process 'free-response'.
- $\tilde{G}_j(q^{-1})\Delta u(t+j-1)$, the prediction of the output that is dependent upon the future control increments yet to be calculated. This represents a convolution of the future control increments and the 'step-response' of the process and thus the individual $\tilde{G}$ parameters correspond to the appropriate ordinates of the process step-response.

2.2.3 The Predictive Control law

Knowledge of the prediction equation (2.4) not only allows the calculation of future predictions given future control signals, but more importantly, allows the calculation of future control signals given the desired value of future predictions. For GPC this calculation proceeds by definition of a quadratic cost-function representing the difference between future predictions of the output and a prespecified set-point *trajectory* $\{w(t+j), j = 1, 2, \dots\}$:

$$J_{GPC} = E \left\{ \sum_{j=NS}^{NY} [y(t+j) - w(t+j)]^2 + \lambda(j) \sum_{j=1}^{NY} \Delta u(t+j-1)^2 | t \right\} \quad (2.8)$$

where:

- NS is the *minimum prediction horizon*
- NY is the *maximum prediction horizon*, and
- $\lambda(j)$ is a *weighting* upon future control increments.

From this it is clear that GPC minimises the prediction error over a range of future predictions, starting NS samples into the future and ending after NY samples. The minimum prediction horizon, NS , is normally assumed to be 1 but if a lower bound upon the process time-delay is known, i.e. k_{min} , then it should be chosen equal to or greater

than k_{min} in order to minimise computation for the same overall range of predictions. In general NY , referred to from now on as the 'prediction horizon', should be chosen to extend across most of the process response that is expected to be affected by any change in the current control. Thus, by reference to the CARIMA process model it should be chosen at least greater than the degree of $B(q^{-1})$ and if possible should correspond to the rise-time of the plant itself.

The derivation of the GPC control law from this cost-function is already extensively described elsewhere (Mohtadi, 1986; Clarke, Mohtadi & Tuffs, 1987; Lambert M., 1987). Suffice it to say that the derivation proceeds by minimisation of the cost-function with respect to the vector of future controls and the resulting control law (assuming $NS = 1$ for simplicity) is given by:

$$\bar{\mathbf{u}}(t) = [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f}) \quad (2.9)$$

where all the vectors are of dimension $NY \times 1$, i.e.

$$\begin{aligned} \mathbf{w} &= \begin{bmatrix} w(t+1) & w(t+2) & \dots & w(t+NY) \end{bmatrix}^T \\ \mathbf{f} &= \begin{bmatrix} f(t+1) & f(t+2) & \dots & f(t+NY) \end{bmatrix}^T \\ \bar{\mathbf{u}} &= \begin{bmatrix} \Delta u(t) & \Delta u(t+1) & \dots & \Delta u(t+NY-1) \end{bmatrix}^T \end{aligned}$$

and $f(t+j)$ represents that component of $y(t+j)$ that is known at time t , e.g.

$$\begin{aligned} f(t+1) &= (G_1(q^{-1}) - g_0)\Delta u(t) + F_1(q^{-1})y(t) = \hat{y}(t+1 | t) \\ f(t+2) &= q(G_2(q^{-1}) + q^{-1}g_1 - g_0)\Delta u(t) + F_2(q^{-1})y(t) = \hat{y}(t+2 | t) \\ &\text{etc.} \end{aligned} \quad (2.10)$$

and $\mathbf{G}$ is a matrix of dimension $NY \times NY$:

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ g_{NY-1} & g_{NY-2} & \dots & g_0 \end{bmatrix} \quad (2.11)$$

where the individual elements correspond to the first NY elements of $G_{NY} = E_{NY}B$ (from iteration of the Diophantine equation) or $\tilde{G}_{NY}$ (the step response of the estimated model). Because it is a *receding-horizon* control policy only the first element of the control sequence is actually implemented, therefore it is only necessary to calculate the first row of $[\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]^{-1} \mathbf{G}^T$ at each sample interval.

2.2.4 The Control Horizon

The specification of a control horizon, NU , is an idea borrowed from the DMC method of Cutler & Ramaker (1980) and represents an assumption that after an interval of NU future samples the projected control increments are zero, i.e.

$$\Delta u(t+j-1) = 0 \quad \text{when} \quad j > NU \quad (2.12)$$

In terms of the cost-function this may be considered as being equivalent to placing infinite weights upon future control increments occurring after NU future samples, i.e.

$$\lambda(j) = \infty \quad \text{when} \quad j > NU \quad (2.13)$$

The incorporation of a control horizon into the GPC algorithm is a key modification, offering two immediate benefits :

- *Simplification of the control calculation:* From the prediction equation of (2.4) and the definition of the effect of NU in (2.12), the $\mathbf{G}$ matrix may be shown to be of a reduced dimension $NY \times NU$:

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ g_{NY-1} & g_{NY-2} & \cdots & g_{NY-NU} \end{bmatrix} \quad (2.14)$$

which means that the matrix to be inverted $[\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]$ is itself of dimension $NU \times NU$. In particular if $NU = 1$ (as in the EPSAC algorithm of De Keyser & Van Cauwenberghe, 1985) this reduces to a scalar computation.

- *Inherently stable control of NMP processes:* The assumption of a control horizon represents a constraint upon the control calculation that allows stable control of non-minimum phase processes without the use of λ as required by the GMV algorithm. The control signal is prevented from growing uncontrollably and thus unstable modes are suppressed (Clarke, 1984).

As a GPC design parameter the interpretation of the control horizon follows from the latter point where it may be used to constrain the control calculation by restricting the degrees of freedom in the solution for the control signal. A large value of NU implies more

degrees of freedom and thus potentially more active control. It is important to note that at the extreme when $NU = NY$ this represents, in the absence of control weighting, a minimum variance type of control strategy (Clarke *et al*, 1987a).

2.3 EXTENSIONS TO THE ALGORITHM

The previous section described the fundamental concepts of the GPC algorithm and developed the basic equations for the calculation of the control signal. In this section the algorithm is extended with a series of user chosen transfer-functions and polynomials with the intention of providing more versatile specification of the desired closed-loop performance. This further complication of the control law is justified because, in principle, it allows a control engineer to approach widely differing control problems using the same basic control framework, i.e. GPC. These extra *design parameters* may then be used for application specific tuning of the control law whereby the user can specify 'slow' or 'fast' control for a variety of different processes, from relatively simple chemical processes with time-constants of several minutes to complicated robots with time-constants less than a second.

The design parameters considered include a transfer-function for specifying a desired closed-loop, a polynomial for tailoring the controlled response to load disturbances, and scalar control weighting. These are P , T and λ respectively and very close parallels may be drawn between these and the similarly named extensions to the basic GMV algorithm (Clarke & Gawthrop, 1975) that later led to interpretations of the GMV algorithm such as de-tuned model reference, and optimal Smith predictor (Gawthrop, 1977; Clarke, 1982; Tuffs, 1984).

2.3.1 $P(q^{-1})$: Model following

In section 2.2 the GPC control calculation was described as being based upon the minimisation of the errors between a series of future predictions of the output and the corresponding projected set-points. The $P(q^{-1})$ transfer function is implemented by generation of an *auxiliary* output and the use of this output, instead of the normal predicted output, in the cost-function. The minimisation then proceeds using the prediction errors associated with the auxiliary output, $\psi(t)$:

$$J_{GPC} = E \left\{ \sum_{j=NS}^{NY} [\psi(t+j) - w(t+j)]^2 + \lambda(j) \sum_{j=1}^{NY} \Delta u(t+j-1)^2 \mid t \right\} \quad (2.15)$$

where

$$\psi(t) = P(q^{-1})y(t) \quad \text{and} \quad P(q^{-1}) = \frac{P_n(q^{-1})}{P_d(q^{-1})} \quad (2.16)$$

Thus $P(q^{-1})$ is a transfer function based upon two polynomials P_n and P_d in the backward shift operator q^{-1} with $P(1)$ set to unity to ensure offset-free control with $\psi(t) = y(t)$ at steady-state.

Development of the extended algorithm proceeds exactly as before but with the prediction equations for $\psi(t+j)$ now based upon the modified Diophantine equation :

$$\frac{P_n}{P_d} = E_j A \Delta + q^{-1} \frac{F_j}{P_d} \quad \text{for} \quad j = 1, \dots, NY \quad (2.17)$$

giving a prediction equation,

$$\hat{\psi}(t+j \mid t) = G_j \Delta u(t+j-1) + \frac{F_j y(t)}{P_d} \quad (2.18)$$

where $G_j = E_j B$ and it is again possible to calculate successive values of E_j and F_j by recursion of the Diophantine equation (Clarke *et al*, 1987b), the only difference being the initial values for E_1 and F_1 .

The interpretation of the transfer-function $P(q^{-1})$ is to a large extent application-specific. For process control, when the prediction horizon is typically long (i.e. $NY = 10$) and the control horizon short (i.e. $NU = 1$), it may be most meaningfully interpreted as a means of introducing phase advance into the closed-loop. Thus choice of $P(q^{-1})$ as the unity gain transfer-function $(1 - \alpha q^{-1})/(1 - \alpha)$ forces the signal $\psi(t)$ to lead $y(t)$ by an amount depending upon the value of α , the larger the value of α the greater the phase advance. The use of closed loop phase advance is an established control strategy and is commonly used as an aid to controller robustness, potentially overcoming such problems as overshoot and insufficient loop stability margins.

For higher performance applications, such as the control of flexible robots, the control horizon, NU , may be chosen greater than unity. Under this condition, as for GMV with no control de-tuning, $P(q^{-1})$ is best interpreted as the *approximate inverse closed-loop model*, a description that is exact when $\lambda = 0$ and $NY = NU \geq k$. This implies that, in the noise-free case, that the closed-loop response is given by

$$y(t) = \frac{1}{P(q^{-1})} w(t-k) \quad (2.19)$$

Increasing either λ or the difference $NY - NU$ de-tunes this model following property, but for large NU the difference is slight.

It is important to note that it would also be possible to achieve model following properties by appropriate pre-filtering of the set-point $w(t)$, but this would have no effect on the overall closed loop properties such as load disturbance rejection and thus is of limited use.

2.3.2 λ : Control weighting

One of the most important features of the GMV control algorithm (Clarke & Gawthrop, 1979) was the inclusion, in the controller cost-function, of a term containing the control signal to be calculated, weighted by a user-chosen scalar, λ . A similar term was included in the cost function of the GPC algorithm (2.8), i.e.

$$\lambda(j) \sum_{j=1}^{NY} \Delta u(t+j-1)^2 \quad (2.20)$$

where now it is *increments*³ in the future control signal sequence that are weighted. As in GMV the effect of λ as a design feature is straightforward, with increase of λ increasing the penalisation of control increments in the control calculation and thus reducing overall control activity.

In GMV the use of λ was often found to be essential for robust stable control of non-minimum phase processes (Gawthrop, 1987), and also in the presence of incorrect initial assumptions about time-delay. However, in the GPC control strategy it is almost redundant: the existence of the alternative horizon-based design features associated with long range prediction, NU and NY , rendering GPC inherently more robust than GMV and removing the requirement for λ . Another reason for avoiding the use of λ is the arbitrary nature of its choice. This latter point can be further complicated by gain variations in the process requiring λ to be scaled by $B(1)^2$ in an attempt to retain a consistent de-tuning effect for a given value of λ . This in practice has led to the choice of a relative value of λ_{rel} (Lam, 1980) at the commissioning stage of control, the actual λ used in the control calculation being given by:

$$\lambda = \lambda_{rel} [B(1)]^2 \quad (2.21)$$

³Thus avoiding the problem of λ offset (Hodgson, 1982)

In this section, and subsequently in the trials of section 2.4.7, λ is included simply for completeness.

2.3.3 $T(q^{-1})$: Load disturbance rejection

In the CARIMA plant model description (2.1) a term $C(q^{-1})\xi(t)/\Delta$ is assumed that describes the noise affecting the plant as a zero-mean *white* noise sequence $\{\xi(t), t = 1, 2, \dots\}$ that is *coloured* by $C(q^{-1})$ and then *integrated*. Like the rest of the CARIMA model this is only an approximation and will never, in practice, represent an exact description of the noise structure. Nonetheless it has been found to be useful because the assumption of a noise model of this form leads, via the use of incremental data in the control calculation, to a control law with guaranteed integrating action.

The relation of the $T(q^{-1})$ polynomial to the $C(q^{-1})$ polynomial is best illustrated by consideration of the *filtered* CARIMA model given below:

$$\frac{A(q^{-1})}{T(q^{-1})}\Delta y(t) = \frac{B(q^{-1})}{T(q^{-1})}\Delta u(t-k) + \frac{C(q^{-1})}{T(q^{-1})}\xi(t) \quad (2.22)$$

Obviously if $T(q^{-1}) = C(q^{-1})$ then the prediction errors resulting will have the minimum variance possible for this particular model structure. However, on-line estimation of C has been found in practice to be difficult (Friedlander, 1982; De Larminat, 1985) because of the time-varying nature of C and the fact that $\xi(t)$ is never available and must be *proxied* by either *a priori* or *a posteriori* prediction errors (Ljung & Söderström, 1983). Thus T may best be regarded as a fixed estimate of C .

The incorporation of the $T(q^{-1})$ polynomial into the control calculation follows directly from (2.22), the prediction equations of (2.4) being re-written :

$$\hat{y}'(t+j | t) = G_j \Delta u'(t+j-1) + F_j y'(t) \quad (2.23)$$

where the nomenclature $\Delta u'$ and y' denotes that these quantities have been filtered by $1/T$. This equation corresponds to a modified Diophantine equation of the form

$$T = E_j A \Delta + q^{-j} F_j \quad (2.24)$$

Since the introduction of T as a design polynomial in the GMV control algorithm (Clarke & Gawthrop, 1979) its importance has been increasingly recognised and a variety of interpretations proposed. From a practical engineering point of view perhaps the most appropriate of these is that of T as a *filter*, primarily because of the incremental nature of

the GPC control law. The differencing of data associated with this approach represents a form of high-pass filtering that will often significantly degrade the signal to noise ratio of the process measurement. In particular specification of $1/T$ as a 2nd order low pass filter, when cascaded with the differencing operation, results effectively in *band-pass filtering* of the data where the low frequency data is removed to prevent offsets in control, and the high frequencies attenuated to reduce the effect of high frequency disturbances.

Such high frequency disturbances may be considered as arising in two distinct types of situations :

- GPC control in the presence of high frequency unmodelled dynamics, i.e. model-plant mismatch.
- GPC control in the presence of high frequency unmeasurable load disturbances.

The low-pass filtering effect of T upon the control calculation is expected to increase the robustness of GPC control for both the above circumstances (see section 2.4.8). As a GPC design parameter, however, it is the latter application of T that has received the most attention. This is because if there are no unmodelled dynamics (i.e. the estimated model corresponds exactly to that of the process) the effect of T is confined to the load disturbance rejection properties of the closed loop and has no effect upon the servo performance. Thus in principle T may be used to de-tune response to unmeasurable load disturbances, independent of the set point following. It is this property that has often led to the description of T as a *load disturbance tailoring polynomial*, where the 'slower' the specification of $1/T$ the 'slower' the control signal response to sudden high frequency load disturbances, independent of the set-point following characteristics. The reason for this effect may be stated intuitively as,

What the controller cannot see it cannot respond to !

An oft-quoted example of the potential usefulness of such a design feature as T in the GPC algorithm is that of the rolling of sheet metal (Tuffs, 1984). In this process transient variations in the incoming sheet thickness are often much faster than that which can be obtained from changes in the roll force, i.e. the actuation is sluggish compared to the variation in the load disturbances. De-tuning of the load disturbance rejection could be used to prevent otherwise excessive control activity and possible over-compensation.

2.3.4 Feed-forward compensation

There are of course many examples in the process control industry of load disturbances that are measurable (see below). Rather than relying upon the load disturbance rejection properties of the controller such measurable disturbances can be incorporated as feed-forward terms, a strategy that has been shown to be particularly important for the control of processes with significant time-delay and/or sluggish response (Allidina *et al*, 1981). For GPC feed-forward is incorporated simply by extending the CARIMA model, i.e.

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t - k) + D(q^{-1})\Delta v(t - k_v) + \xi(t) \quad (2.25)$$

where $v(t)$ is the measurable disturbance; $D(q^{-1})$ is a polynomial in q^{-1} that may be estimated in exactly the same manner as A or B and thus enables *dynamic* feed-forward compensation; and $\hat{k}_v$ is the assumed integral time-delay specific to the feed-forward measurement. This gives an extended prediction equation of the form:

$$\hat{y}(t + j | t) = E_j B \Delta u(t + j - k) + E_j D \Delta v(t + j - k_v) + F_j y(t) \quad (2.26)$$

As for $\xi(t)$ the feed-forward signal is assumed to consist of step changes at random times, i.e. $\Delta v(t + j) = 0$ for $j \geq 1$, this being considered the most realistic internal model of a measurable disturbance likely to be encountered in process control (e.g. a sudden measurable change in the moisture content of material entering a drying process, a step-wise control signal change in an interacting control loop, etc.). Alternatively more specific models could be used for particular types of measurable disturbances, e.g. a ramp-wise model for disturbances such as steadily increasing load during engine control, or a sinusoidal model for periodic disturbances such as waves in ship-steering problems.

In section 2.5 of this chapter, and later in Chapter 5, this feature of feed-forward compensation is used to improve decoupling during the control of a multivariable process using SISO GPC controllers with feed-forward from the control signals of other loops.

2.3.5 Pre-specified set points

The use of long-range prediction in the GPC control law enables a control strategy not previously available in such single-stage algorithms such as GMV or pole-placement, namely *pre-specified set points*. Instead of minimising the difference between a range of future predictions and a constant set-point, i.e.

$$w(t) = w \quad \text{for} \quad t = 1, \dots, NY \quad (2.27)$$

a *trajectory* of future set points may be specified where

$$w(t) = \begin{bmatrix} w_1 & w_2 & \dots & w_{NY} \end{bmatrix}^T \quad (2.28)$$

Thus at time t the control calculation may be forced to take into account a set point change occurring (for example) five samples in the future, simply by changing the corresponding elements $w(t+5)$ to $w(t+NY)$ in the control calculation.

Such prior knowledge of set point changes is often available in process control applications operating in a batch or sequential mode and in robot control applications (Lambert M., 1987; Lelić & Wellstead, 1987). The most important potential benefits are expected to be an increase of speed in high performance trajectory following and improved decoupling in multivariable applications.

2.3.6 The GPC 'default' specification

In the next section reference will often be made to the 'default settings' of GPC design parameters. These are given in Table 2.1 and represent standard specifications that are expected to provide adequate low-gain control performance for a wide variety of process control applications.

GPC Specification	
Feature	DEFAULT
NY	10
NU	1
NS	1
$P(q^{-1})$	1
$T(q^{-1})$	$1 - 0.8q^{-1}$
λ	10^{-6}

Table 2.1: The default settings of the GPC algorithm

Thus, for an appropriate choice of sample interval, the prediction horizon is 'long', the control horizon is 'short' and GPC tends towards a cautious mean-level control strategy.

2.4 CLOSED LOOP ANALYSIS OF GPC

This section describes a package (GPCTEST) which is used to calculate the characteristic equation of GPC's closed loop given the *actual* process model, the *estimated* process model and the specifications of GPC's design parameters (NS , NY , NU , P , T , and λ). This enables a discrete *loci* of closed loop poles to be plotted with respect to variations in any of the above and thus in turn allows detailed quantitative observation of the effect of the GPC design parameters upon the closed loop.

2.4.1 Calculation of the GPC characteristic equation

Even in the fixed parameter (i.e. non-adaptive) case the long range predictive nature of GPC prevents the derivation, in closed loop, of a meaningful analytic equation relating the output to the set-point in terms of design parameters (Lelić & Zarrop, 1987). An alternative indirect strategy is therefore employed, whereby an *equivalent* closed loop equation is defined and then calculated in terms of the controller parameters F and G . Using this technique closed loop analysis may then proceed for a *specific* choice of GPC design features, such as NU and NY .

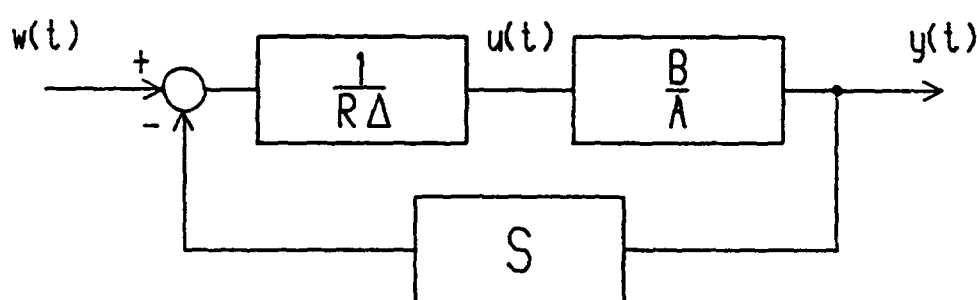


Figure 2.2: A general closed loop system.

A system in closed loop may always be described in the general form shown in Fig. 2.2, where R and S are polynomials in the backward shift operator, q^{-1} , and the time-delay is either assumed to be unity or incorporated as zero leading elements of the B polynomial. The characteristic equation of this general closed loop system is given by:

$$R(q^{-1})A(q^{-1})\Delta + q^{-1}B(q^{-1})S(q^{-1}) = 0 \quad (2.29)$$

Obviously if it is possible to cast the GPC control calculation in a similar form to (2.29) then 'equivalent' R and S polynomials could be calculated and analysis of the closed loop performed.

The control calculation for the general loop may be expressed as:

$$R(q^{-1})\Delta u(t) = w(t) - S(q^{-1})y(t) \quad (2.30)$$

whereas the GPC control calculation is given by a more complicated expression, stated earlier in (2.9) as :

$$\Delta u(t) = [\mathbf{G}^T \mathbf{G} + \lambda I]^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f})$$

Denoting the top row of the matrix $[\mathbf{G}^T \mathbf{G} + \lambda I]^{-1} \mathbf{G}^T$ as the vector $\bar{\mathbf{g}}$ where:

$$\bar{\mathbf{g}}^T = \begin{bmatrix} \bar{g}_1 & \bar{g}_2 & \dots & \bar{g}_{NY} \end{bmatrix} \quad (2.31)$$

and recalling that:

$$\begin{aligned} f(t+1) &= (G_1 - g_0)\Delta u(t) + F_1 y(t) &= \Gamma_1 \Delta u(t-1) + F_1 y(t) \\ f(t+2) &= q(G_2 - g_1 q^{-1} - g_0)\Delta u(t) + F_2 y(t) &= \Gamma_2 \Delta u(t-1) + F_2 y(t) \\ &\text{etc.} \end{aligned}$$

then by substitution into (2.9) :

$$\Delta u(t) = \begin{bmatrix} \bar{g}_1 & \bar{g}_2 & \dots & \bar{g}_{NY} \end{bmatrix} \begin{bmatrix} w - \Gamma_1 \Delta u(t-1) - F_1 y(t) \\ w - \Gamma_2 \Delta u(t-1) - F_2 y(t) \\ \vdots \\ w - \Gamma_{NY} \Delta u(t-1) - F_{NY} y(t) \end{bmatrix} \quad (2.32)$$

which may be multiplied out,

$$\Delta u(t) = \left(\sum_{i=1}^{NY} \bar{g}_i \right) w - \left(\sum_{i=1}^{NY} \bar{g}_i \Gamma_i \right) \Delta u(t-1) - \left(\sum_{i=1}^{NY} \bar{g}_i F_i \right) y(t) \quad (2.33)$$

and (ommitting limits for clarity) this may be rearranged to give :

$$\begin{aligned} \left(\frac{1}{\sum \bar{g}_i} \right) \Delta u(t) + \left(\frac{\sum \bar{g}_i (\Gamma_i)_1}{\sum \bar{g}_i} \right) \Delta u(t-1) + \dots + \left(\frac{\sum \bar{g}_i (\Gamma_i)_{\delta \Gamma}}{\sum \bar{g}_i} \right) \Delta u(t - \delta \Gamma) \\ = w - \left(\frac{\sum \bar{g}_i (F_i)_1}{\sum \bar{g}_i} \right) y(t) - \dots - \left(\frac{\sum \bar{g}_i (F_i)_{\delta F}}{\sum \bar{g}_i} \right) y(t - \delta F) \end{aligned} \quad (2.34)$$

This is equivalent in form to:

$$\begin{aligned} r_0 \Delta u(t) + r_1 \Delta u(t-1) + \dots + r_{\delta R} \Delta u(t - \delta R) \\ = w - s_0 y(t) - s_1 y(t-1) - \dots - s_{\delta S} y(t - \delta S) \end{aligned}$$

The *equivalent* closed loop characteristic polynomials, R and S , are therefore given by :

$$R(q^{-1}) = \frac{1}{\sum \bar{g}_i} \left(1 + \sum_{i=1}^{NY} \bar{g}_i \Gamma_i(q^{-1}) \right) \quad (2.35)$$

$$S(q^{-1}) = \frac{1}{\sum \bar{g}_i} \left(\sum_{i=1}^{NY} \bar{g}_i F_i(q^{-1}) \right) \quad (2.36)$$

and may simply be substituted into (2.29) to obtain the closed loop characteristic equation.

2.4.2 Analysis using the GPCTEST Package

The GPCTEST package was designed as an aid to the analysis of the GPC algorithm, calculating the closed-loop characteristic equation for a variety of user-selected control *scenarios* and displaying the result in the form of discrete loci of pole variation or system step responses. It is based upon the earlier FAUST software package⁴ (Tuffs, 1984) previously introduced to provide a software environment for the implementation, application and evaluation of a variety of self-tuning and adaptive control algorithms. In its function it is similar to Nazer's Root Locus Design package (Nazer, 1981) which plots the z -domain system root locus, with respect to variation in λ , for the GMV algorithm.

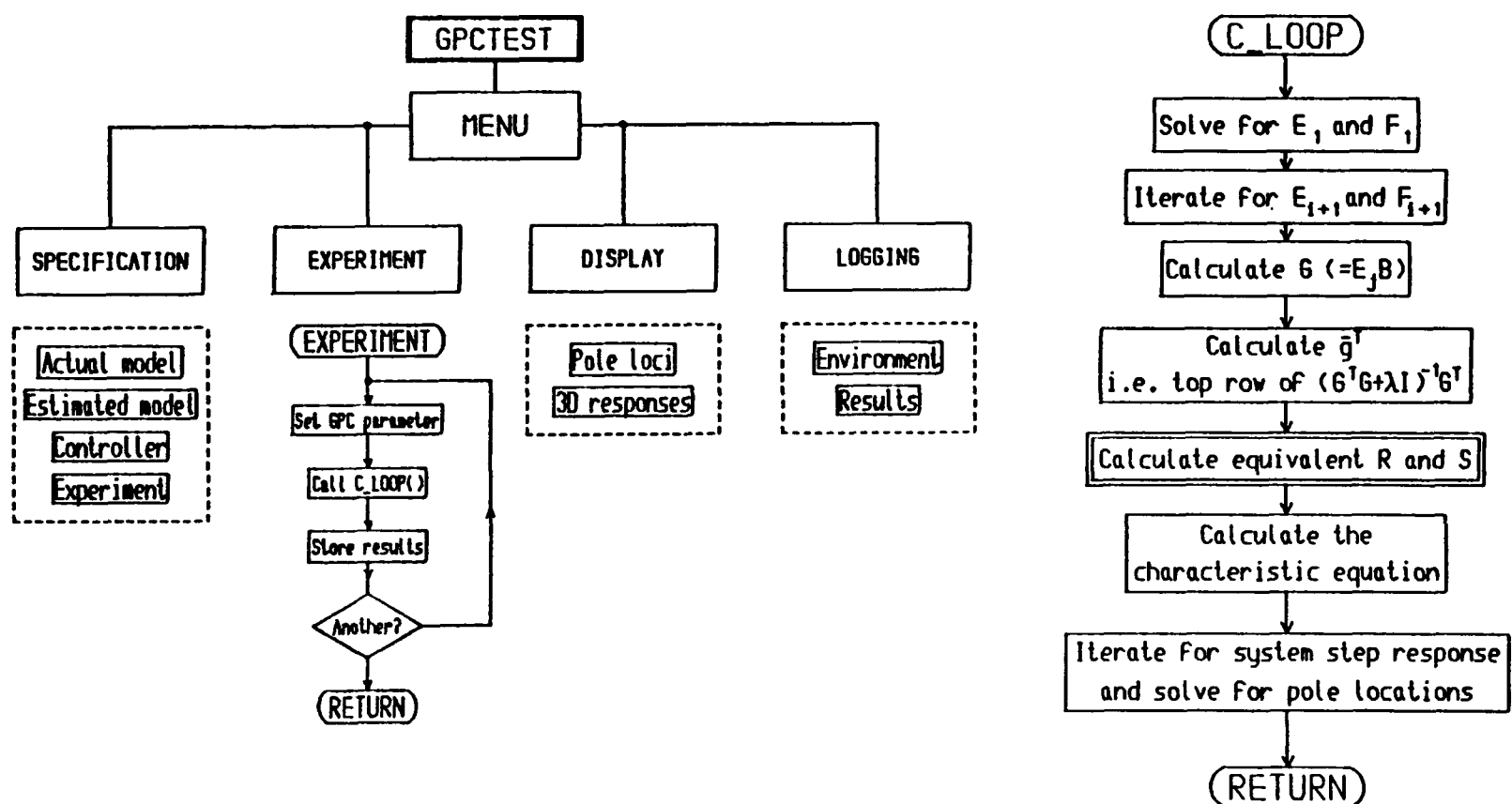


Figure 2.3: The GPCTEST package.

Written in Fortran the GPCTEST package comprises four distinct modules (Fig. 2.3)

⁴itself a successor to the successful CONTROL BASIC self-tuner of Hodgson (1982).

accessed by means of a menu system. The function of the individual modules is as follows:

• SPECIFICATION

This module is itself menu-driven and enables specification of :

- *THE PROCESS*: where models of the *actual* and *estimated* process may be input separately and need not correspond. Thus the effects of model-plant mismatch (MPM), upon the closed loop, may be quantitatively examined.
- *THE CONTROLLER*: where all the design features of the GPC algorithm may be specified exactly as in normal operation and their effect upon the closed loop thus observed.
- *THE EXPERIMENT*: where the nature of the experiment to be conducted is specified. Any of the GPC controller design parameters can be selected as varying and the range of variation, number of steps etc. defined by the user. Alternatively either of the two process models, *actual* or *estimated*, may be varied over a specified range.

• EXPERIMENT

Once the above specifications are complete then it is possible, by definition of an equivalent closed loop expression, to determine a closed loop characteristic equation for the deterministic case. From this the closed loop pole position and the system step response is calculated. The experiment then proceeds by varying one of GPC's design features, or one of the actual or estimated models, as specified above, and repeating the calculations.

• DISPLAY

Once the calculations have been completed the results may be plotted on the terminal using standard GINO-2D and GINO-3D routines. The user may choose to display the results in one of two fashions (see Fig. 2.5); either as a discrete loci of the closed loop pole locations (with the stability boundary of unit radius also shown for reference); or as a three-dimensional mesh plot of the closed loop system step response; both being plotted with respect to variations in a selected GPC design parameter (e.g. for $NY = 1 \rightarrow 10$). The latter represents a useful and compact verification of the significance of the pole loci plot. The package also includes a hardcopy facility. Alternatively the pole locations may be printed out in complex number format.

• STORAGE

This facility enables the saving of all the current specifications, including the GPC controller specification and the parameters of the actual and estimated models. This *environment* may be restored at any time in order to repeat, or continue, analysis of that particular situation. The results of an experiment may also be stored and thus made available to other packages such as Ctrl-C (User's Guide, 1986).

In the following five sections GPCTEST is used to evaluate each of the GPC design parameters in turn.

2.4.3 The Prediction Horizon, NY

The prediction horizon, NY , is perhaps the most fundamental design parameter of the GPC algorithm and thus its effect is the first to be examined. The theory of section 2.2 indicated that, depending upon the specification of the other GPC design parameters, the length of the prediction horizon determines the speed of the set point response. The further this horizon is extended, the slower the response, tending in the extreme (i.e. as $NY \rightarrow \infty$) to mean-level control⁵.

X In order to verify this interpretation the GPCTEST package of section 2.4.2 is used to examine the variation of the closed loop system pole locations and system step responses with variation in NY , for a particular process example.

Example 2.1 *A first-order process with a time constant of 10 minutes, unity gain, and a time-delay of 75 seconds may be written as:*

$$G(s) = \frac{e^{-1.25s}}{1 + 10s} \quad (2.37)$$

which for a sample interval of 2 minutes gives the discretised model:

$$G(q^{-1}) \approx \frac{q^{-1}(0.0723 + 0.109q^{-1})}{1 - 0.82q^{-1}} \quad (2.38)$$

The open-loop pole and zero locations of (2.38) are plotted in Fig. 2.4 which shows clearly that the presence of *fractional* dead-time has introduced a discrete zero outside the stability boundary at $q = -1.5$, a so-called *non-minimum phase*⁶ zero. The fact that the process model is non-minimum phase is important because it can lead to instability

⁵i.e. the set-point response time is equal to that of the open loop.

⁶see Åström *et al* (1984) for a description of this phenomenon.

if controlled by elementary self-tuning algorithms based upon pole-zero cancellation, such as minimum variance.

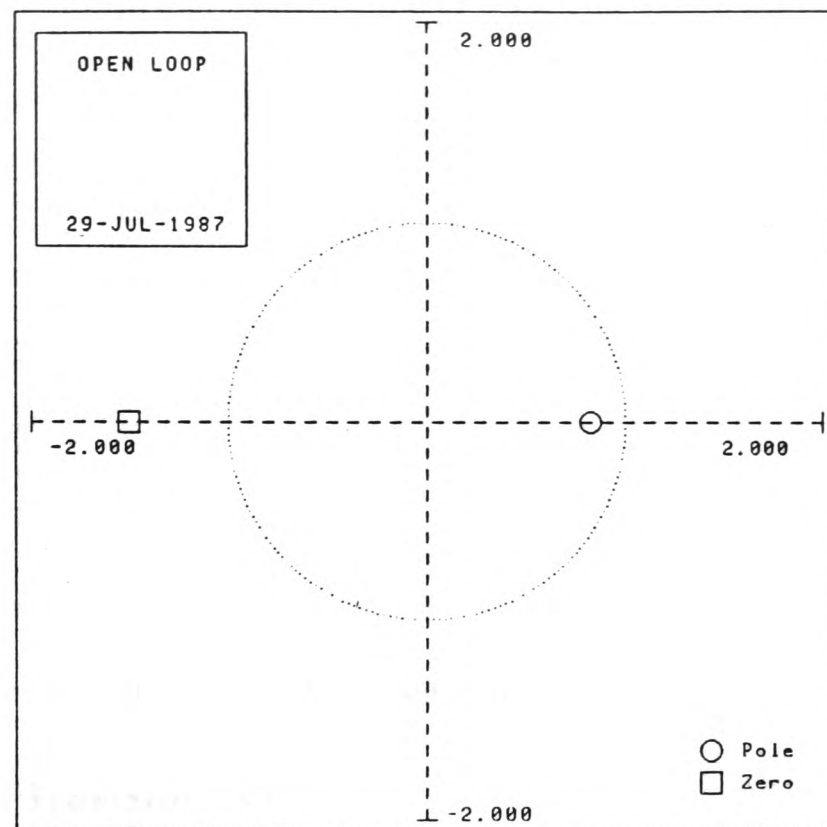


Figure 2.4: Open loop pole and zero locations for Example 2.1.

Fig. 2.5 shows the effect upon the closed loop when NY is increased from 1 to 15 in unit steps, the other GPC design parameters being set to their 'default' values (see Table 2.1). When $NY = 1$ GPC is a single-step-ahead control algorithm corresponding to minimum variance. This is a cancellation control law and thus places the closed loop pole at $q = -1.5$ to cancel the open loop zero. Even in the deterministic case this will result in instability. For $NY = 2$ the system is stabilised and as NY is progressively increased the closed loop pole is seen to migrate⁷ towards its open loop position, becoming progressively slower. The movement of the pole is rapid initially with later increments in NY having a far reduced effect as the prediction horizon exceeds the rise-time of the process.

The effect of varying NY upon the system step response (corresponding to the response of the overall closed loop system to a unit set point demand) qualitatively confirms the results inferred from the above. For $NY = 1$ the response is immediate but subsequently unstable but successive values of NY stabilise the process and progressively lengthen the rise-time.

⁷the exact pole locations from this experiment, and subsequent experiments, may be found in Appendix A (section A.1).

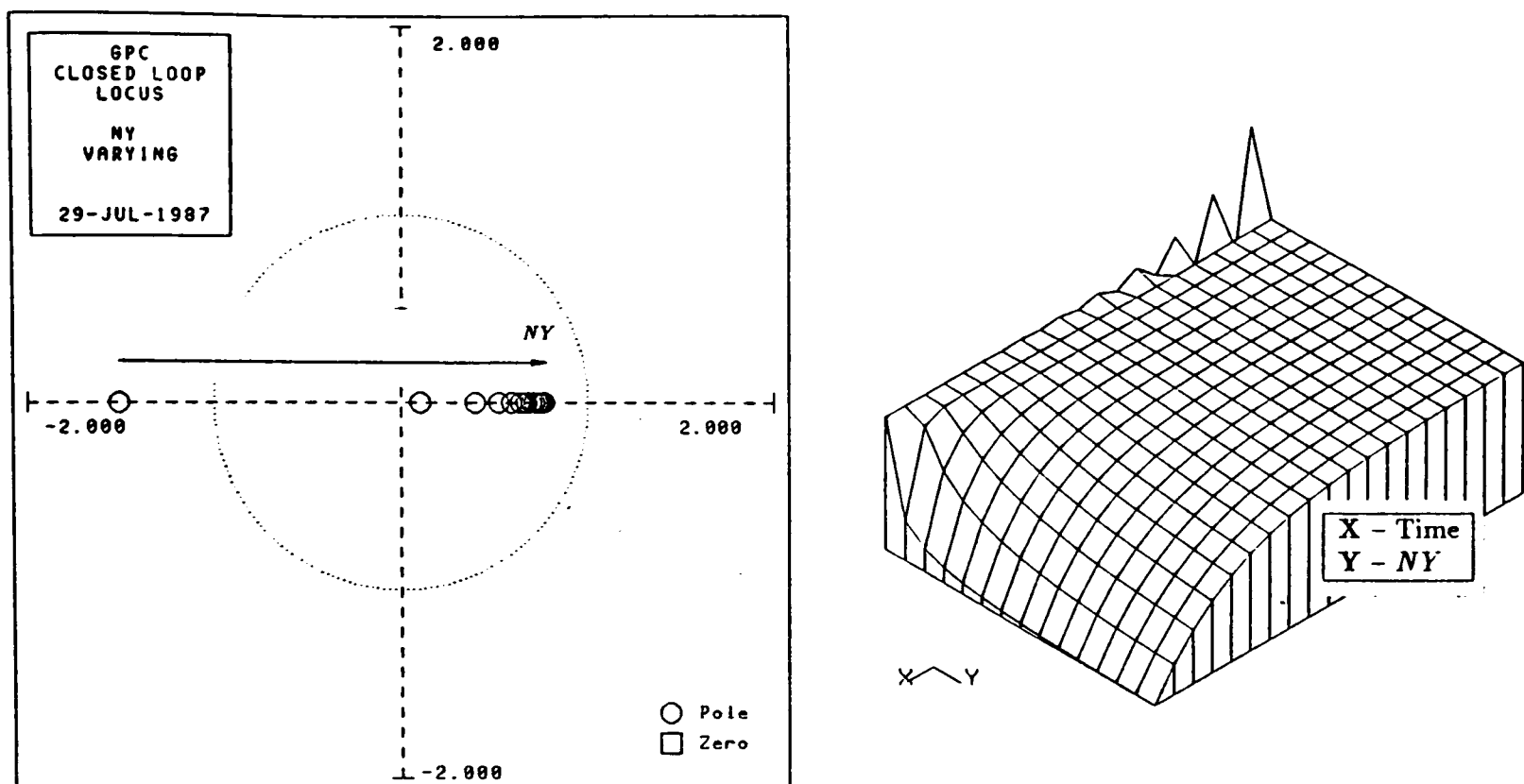


Figure 2.5: The effect upon the closed loop of the Prediction Horizon, NY

2.4.4 The Control Horizon, NU

X The control horizon, NU , is described in section 2.2.4 as defining the degrees of freedom in the calculation of the GPC control signal. Intuitively this allows an interpretation of NU as the *settling time* of the control signal in response to a set point demand, i.e. the number of samples of control activity required to bring the process to the set point. It is however important that, because of the receding horizon implementation, this is only an approximation. Nonetheless NU may still be regarded as a 'coarse' design parameter governing control signal activity.

To verify these interpretations the first order system of Example 2.1 was used as the basis for a GPCTEST analysis experiment where the prediction horizon was set to $NY = 10$ and the control horizon increased from $NU = 1$ to $NU = 10$. The results are shown in Fig. 2.6, and include a simulation run obtained using the FAUST package (Tuffs, 1984) for the above variation of NU to illustrate the special effect of NU upon the control signal.

When $NU = 1$ the closed loop pole is located near to the open loop pole location (see also Figs. 2.4, 2.5) and thus the system response is expected to be slow. This is vindicated by the simulation where there is only one major control signal change followed by a series of relatively minor adjustments, thus tending towards 'mean level' control. For $NU = 2$ the pole migrates to the left hand plane and thus the response is immediate but oscillatory,

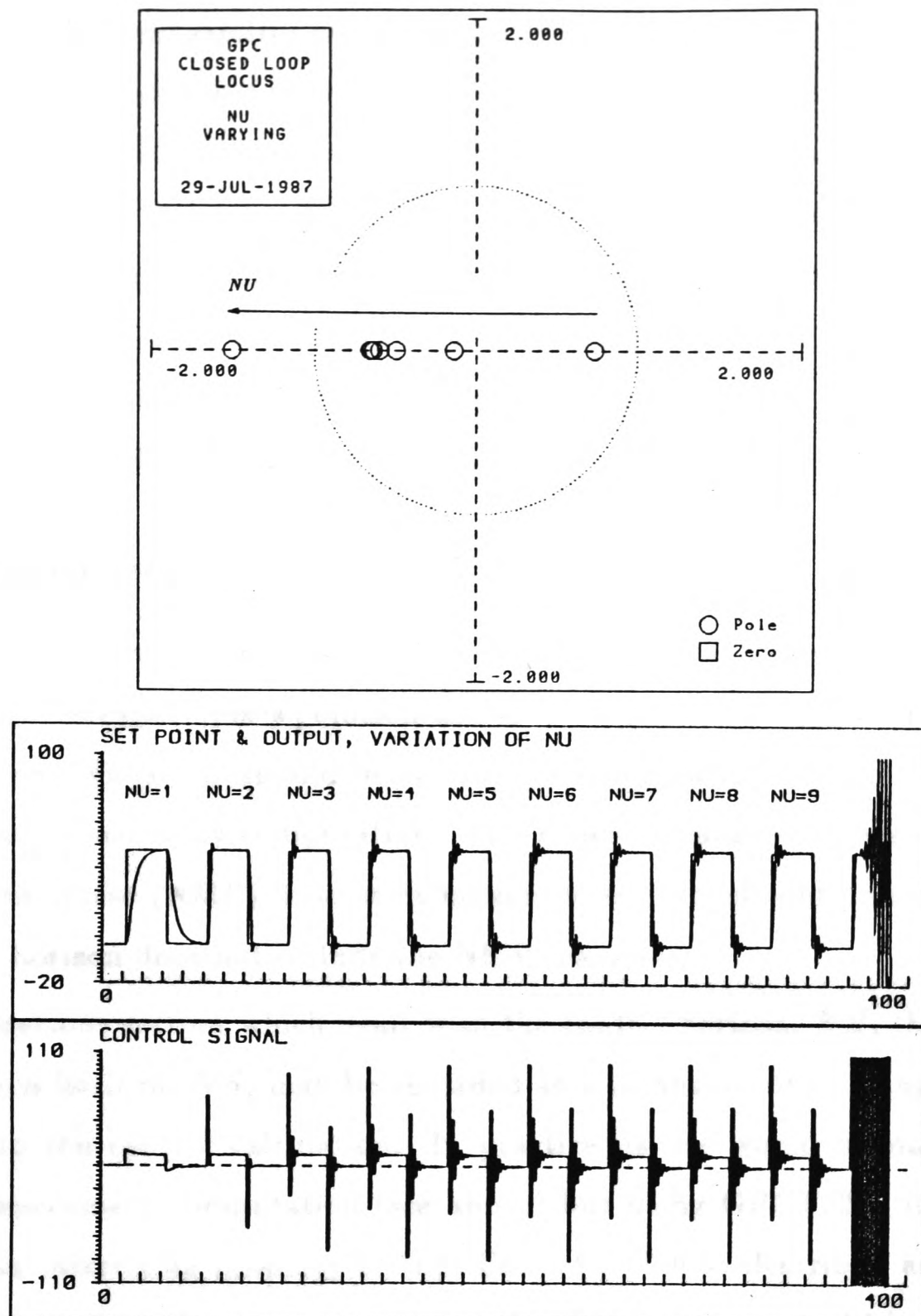


Figure 2.6: The effect upon the closed loop of the Control Horizon, NU

with two distinct control signal changes after the set point change, followed once again by relatively minor adjustments. This, and the subsequent response for $NU = 3$ which exhibits three major control signal excursions, clearly verifies the interpretation of NU as the approximate settling time of the control signal. The additional minor control signal changes are due to the receding horizon implementation of GPC.

Subsequent increase in NU does increase the activity of the control signal and the decay-time of the oscillations but, as with NY in the previous experiment, this increase becomes progressively more difficult to distinguish. The speed of response remains virtually unchanged once $NU > 1$ because, from state space considerations, the example system

requires only two different control inputs to achieve minimum-time response.

As $NU \rightarrow NY$ the poles are seen to be tending towards a fixed stable pole location at $q = -0.67$ which is in fact the *reflection* of the open loop zero about the unit circle. In particular when $NU = 9$ (i.e. $NU = NY - 1$) the GPC algorithm corresponds closely to the stable 'constrained minimum variance' control strategy (Peterka, 1984; Mohtadi, 1986) for a first order system. When $NU = NY = 10$ GPC becomes a minimum variance⁸ design, the pole is located outside the unit circle in an attempt to cancel the open loop zero, and as expected the closed loop becomes unstable.

2.4.5 The initial prediction horizon, NS

If an upper bound upon the process dead-time is known, then beginning the prediction horizon at a point less than this is pointless as the corresponding outputs cannot be affected by control action. Thus NS should be set equal to or greater than the system delay, if known, to avoid unnecessary computation. This observation also applies in the presence of non-minimum phase (NMP) process behaviour where NS should be chosen such that the prediction horizon does not include the NMP transient.

As in the previous section which dealt with the control horizon, NU , the choice of the initial prediction horizon, NS , may be regarded as a means of introducing extra degrees of freedom into the control calculation. In practice its use would normally be limited to avoiding unnecessary computation (see above) but using GPCTEST it is possible to demonstrate an interesting property of the effect of NU first described as a theorem in Mohtadi (1986). In the subsequent experiment the model of Example 2.1 is again used and both NU and NS varied from 1 to 2 respectively with default settings for NY , $P(q^{-1})$ and λ . The results are plotted in Fig. 2.7, with the open loop response shown for reference.

For $NU = 1$, $NS = 1 \rightarrow 2$ the pole remains at the position $q \approx 0.73$, i.e. stable but slow. For $NU = 2$, however, the choice of $NS = NU = 2$ results in a pole located at the origin giving so-called *state dead-beat* control. This is verified by the accompanying simulation run where the corresponding set point response is immediate and free of oscillation. Thus, after a number of samples equal to the order of the states of the system, the output is at the set point and the incremental states are zero.

⁸strictly minimum prototype in the deterministic case but the terms are used here interchangeably.

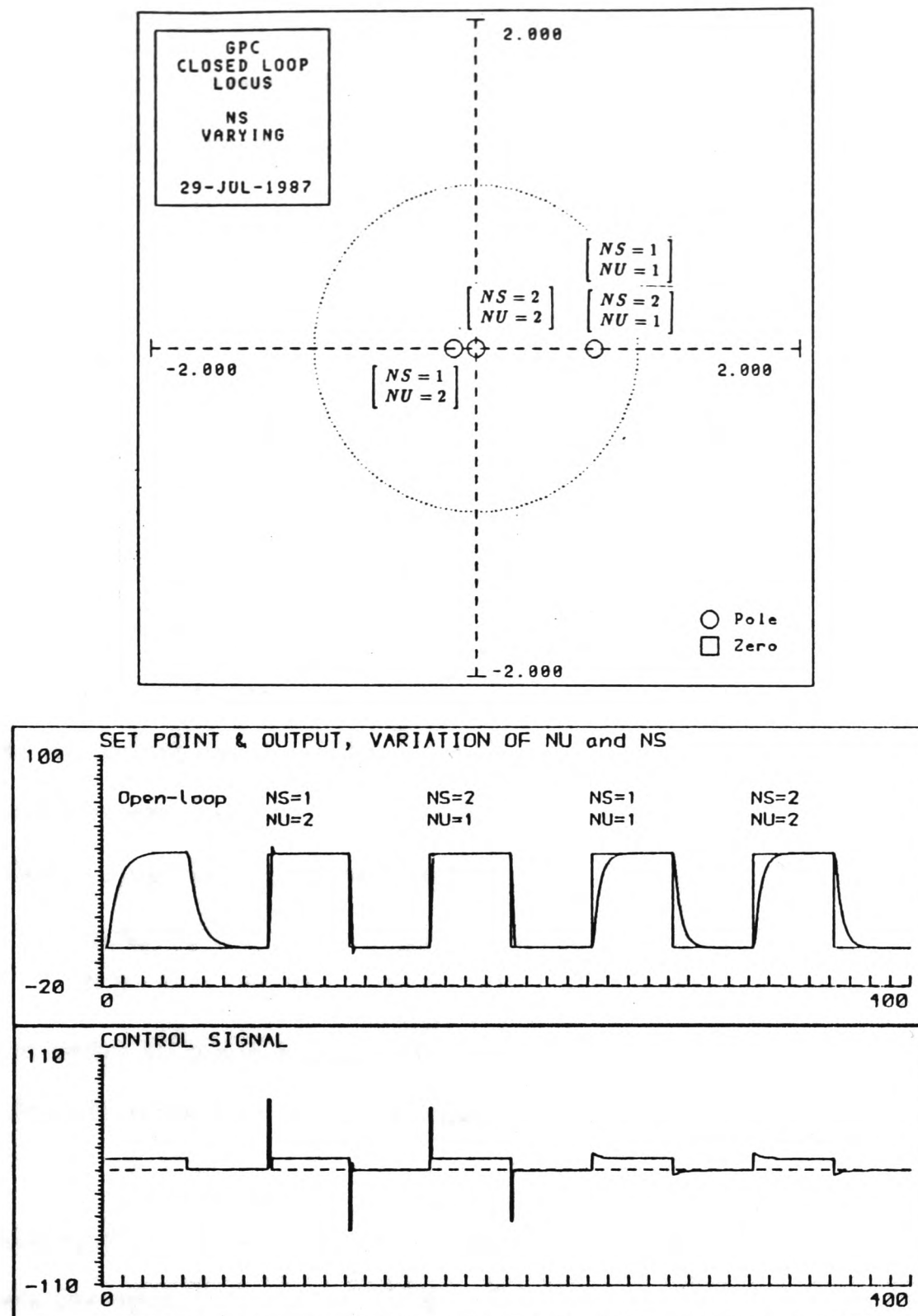


Figure 2.7: The effect upon the closed loop of the initial prediction horizon, NS

2.4.6 The $P(q^{-1})$ transfer function

When a $P(q^{-1})$ transfer function is defined then the corresponding equivalent closed loop equation becomes :

$$R(q^{-1})A(q^{-1})\Delta P_d + q^{-1}B(q^{-1})S(q^{-1}) \quad (2.39)$$

where the degrees of the $R(q^{-1})$ and $S(q^{-1})$ polynomials are now given by :

$$\delta R = \max(\delta B, \delta P_d)$$

$$\delta S = \delta F \quad (2.40)$$

The effect of $P(q^{-1})$ upon closed loop performance is described in section 2.3.1 as admitting two distinct interpretations, namely as a means of achieving pole placement or the introduction of phase advance into the control law. The relative significance of these two possible interpretations is determined by the values of the other GPC design parameters that govern the activity of the control signal.

From previous sections it is clear that if $NS = NU = NY = 1$ and $\lambda = 0$ then the GPC algorithm corresponds to a minimum variance strategy that seeks to fulfil its one-step-ahead control objective exactly. Specification of a $P(q^{-1})$ transfer function under these circumstances should result in exact pole-placement, the closed loop pole(s) location given by the root(s) of $P_n(q^{-1})$. Any modification of a horizon, or λ , that would lead to a reduction in control activity (sections 2.4.3–2.4.5) would therefore be expected to ‘de-tune’ the control performance, reducing the effect of $P(q^{-1})$ and thus rendering the pole-placement no longer exact. In the extreme as $NU = 1$ and $NY \rightarrow \infty$ the location of the pole(s) would be expected to return to that of the open loop.

In this section, using the model of Example 2.1, a ‘model-following’ transfer function,

$$P(q^{-1}) = \frac{1 - 0.5q^{-1}}{0.5} \quad (2.41)$$

was specified in order to place a relatively fast pole at $q = 0.5$. NY was then varied from 1 to 15 with default values for the remaining control parameters. The results are shown in Fig. 2.8.

The presence of $P(q^{-1})$ increases the number of closed loop poles to two. For $NY = 1$ one pole cancels the open loop zero and the other pole is placed at $q = 0.5$, thus the model following aspect of $P(q^{-1})$ is exact but, unfortunately, the closed loop is unstable. This corresponds to GMV control with $\lambda = 0.0$ and $P(q^{-1})$ specified as in 2.41. Subsequent increases in NY are seen to result in the pole locations tending to the origin and to $q = 0.8$, which represent the unit time delay of the process⁹ and the open loop respectively.

2.4.7 Control weighting, λ

Penalisation of future control signal increments in the cost function of GPC, weighted by a scalar λ , provides a further method of ‘de-tuning’ the control performance of the algorithm. The larger the value of λ , the greater the significance of the weighted term in

⁹due to the zero order hold of the sampling operation

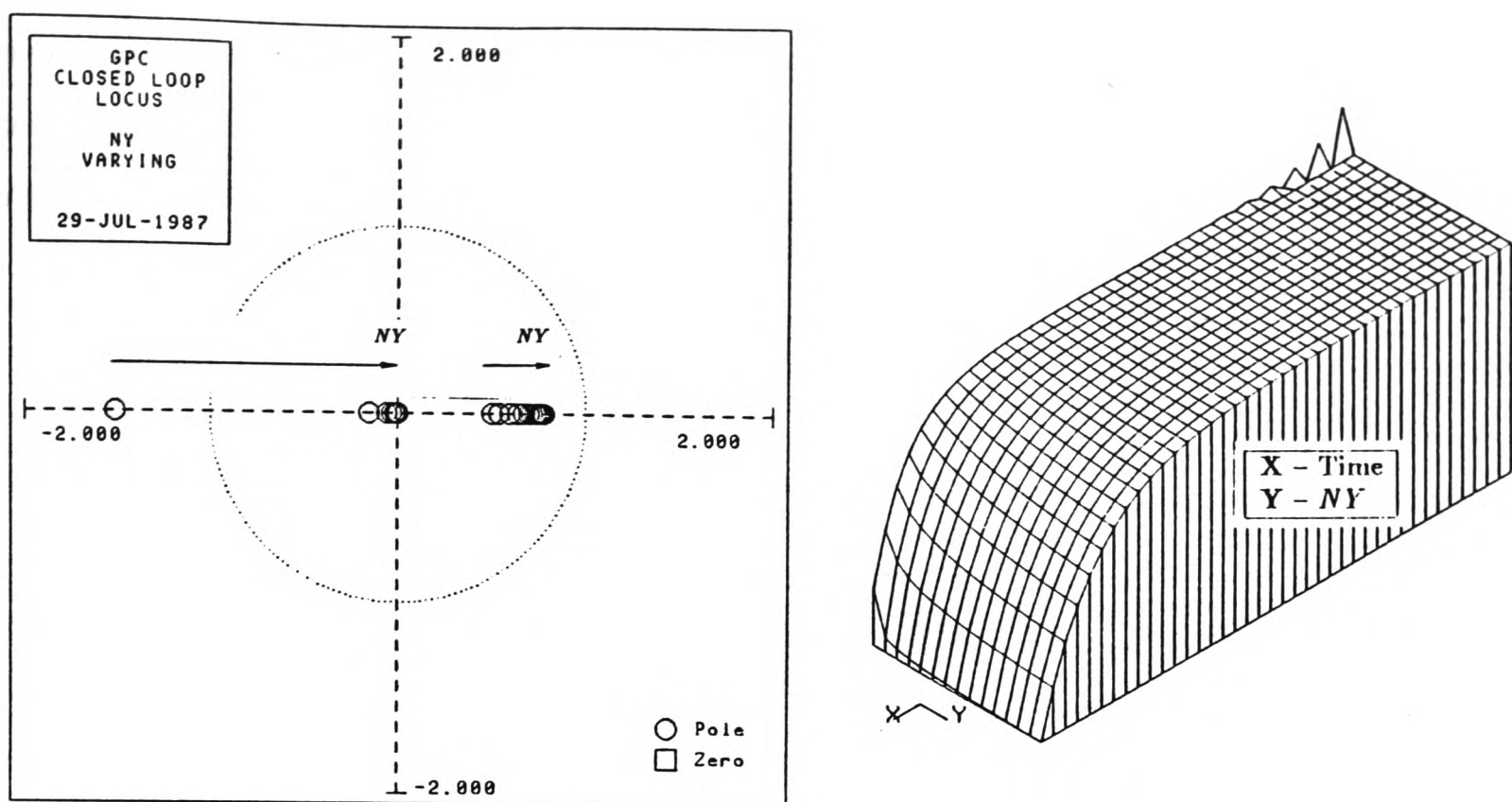


Figure 2.8: The effect upon the closed loop of the $P(q^{-1})$ transfer function.

the cost function, thus reducing control signal activity. However the de-tuning effect of λ differs in two important respects from that of the horizons previously described :

- In the extreme case of de-tuning using horizons the control performance tends to that of the open loop whereas extremely large values of λ prevent *any* control activity and thus effectively switch control off.
- For GPC the practical choice of λ is not particularly meaningful, its effect upon the closed loop depending, for example, upon the process gain. Attempts to render it more meaningful by scaling the specified value λ by the estimated process gain to give a relative λ_{rel} are complicated by the long range predictive nature of the algorithm (Lambert, M., 1987).

Thus the role of λ , previously so critical for the stable operation of the well known GMV algorithm, is secondary in GPC and is included simply for completeness in this section. One possible circumstance where it could be of potential benefit has been suggested by Clarke (1982) to be during the preliminary identification phase of adaptive control where it could be used to arbitrarily de-tune the controller, preventing otherwise excessive control signals due to poor parameter estimates. Once the parameters have been observed to converge λ could then be relaxed and normal control operation applied.

In order to illustrate the effect of control weighting upon the closed loop control of the process given in Example 2.1 the $P(q^{-1})$ transfer function of (2.41) was specified and λ varied through a selection of values from $\lambda = 10^{-4}$ to $\lambda = 100$. A simulation run was also conducted for selected values of λ in that range. The results are shown in Fig. 2.9.

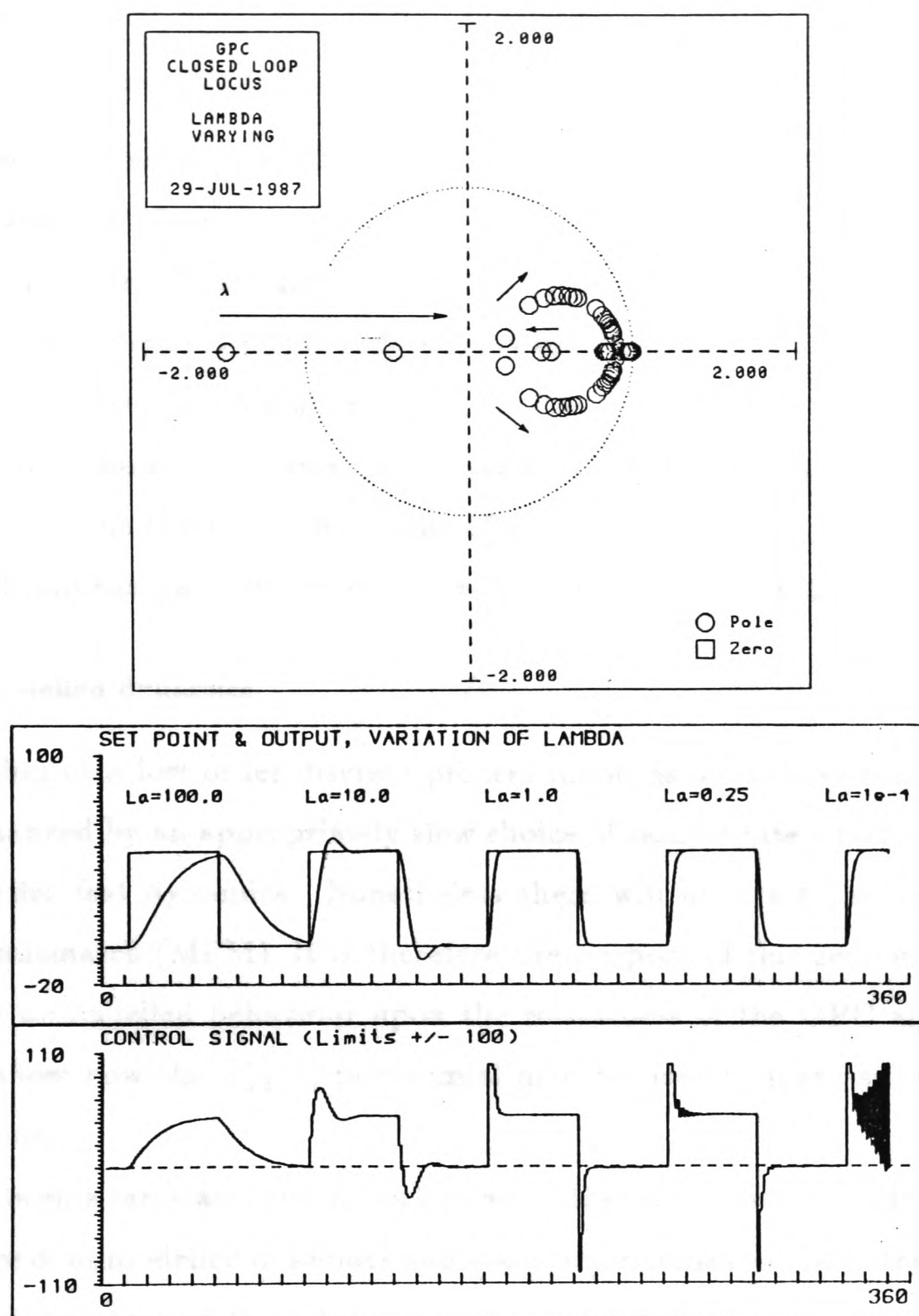


Figure 2.9: The effect upon the closed loop of control weighting, λ .

For $\lambda = 10^{-4}$ there is effectively no de-tuning of the control activity resulting in exact 'model following' with corresponding instability due to the cancellation of the open loop NMP zero. With increasing λ the loop is stabilised, $\lambda = 0.25$ being borderline stable and $\lambda = 1.0$ giving two poles, one at $q = -0.5$ and the other at $q = 0.68$, with correspondingly good control. However when the control weighting is increased to $\lambda = 10.0$ the poles

are slower but have migrated to a complex pair structure. Thus the set point response is slower but overshoots and is oscillatory. At the extreme value of $\lambda = 100.0$ the poles have returned to the real axis and are tending towards $q = 0.82$ and $q = 1.0$, with control correspondingly much slower.

2.4.8 The $T(q^{-1})$ polynomial

The $T(q^{-1})$ polynomial differs from the previous five GPC design parameters in that it has no effect upon the closed loop performance if the estimated model corresponds exactly to the actual process model and if there are no disturbances. Since the GPC control calculation is based upon a linear difference equation model of typically low order, and few industrial processes are disturbance-free, neither of these will be the case in practice. On the basis that model-plant mismatch may be considered separately for the dynamic and noise parts of the CARIMA model the effect of $T(q^{-1})$ in the presence of unmodelled dynamics and unmeasurable disturbances will itself be examined in separate sections.

2.4.8.1 Unmodelled dynamics

The assumption of a low order discrete process model is an approximation that may of course be enhanced by an appropriately slow choice of sample rate which effectively masks out higher order fast dynamics. Nonetheless there will always be a certain amount of model-plant mismatch (MPM). It is therefore the purpose of this section to examine the effect of such unmodelled behaviour upon the robustness of the GPC algorithm and, in addition, to show how the $T(q^{-1})$ polynomial may be used to increase robustness under these conditions.

There has been a large amount of work done on the performance of adaptive controllers in the presence of unmodelled dynamics and many algorithms have been tested for a variety of MPM examples. One of these latter examples stands out however and has been used extensively as a benchmark throughout the literature (e.g. Åström, 1984; Johnson *et al*, 1984; Kosut & Friedlander, 1985; Cluett, 1986.). This is the so-called *Rohrs' example*, first described in Rohrs (1982), which considered a Model Reference Adaptive Control strategy based upon a first order model, and a simulated example of a third order model that contained second order high frequency dynamics.

Example 2.2 Rohrs' model is given by:

$$G(s) = \frac{2}{s+1} \frac{229}{s^2 + 30s + 229} \quad (2.42)$$

Following the analysis of Rohrs et al (1984) this plant is considered for the sample rate $T_s = 0.04$. The resulting discrete model is shown below:

$$G_{0.04s}(q^{-1}) = \frac{10^{-2}q^{-1}(0.3612 + 1.0684q^{-1} + 0.1943q^{-2})}{1 - 2.055q^{-1} + 1.352q^{-2} - 0.2894q^{-3}} \quad (2.43)$$

In order to simulate the effect of unmodelled dynamics the estimated model is specified to correspond to the dominant first order element, $2/(1+s)$, i.e.

$$\hat{G}(q^{-1}) = \frac{0.0784q^{-1}}{1 - 0.9608q^{-1}} \quad (2.44)$$

Thus the estimated model is of a reduced order two less than the actual process model and contains no information about the higher order, high frequency dynamics.

The open loop pole and zero locations of (2.43) are shown in Fig. 2.10, with a non-minimum phase zero, a dominant slow pole at $q = 0.96$ and a faster complex pair of poles corresponding to the higher frequency (and subsequently unmodelled) components of Rohrs' model.

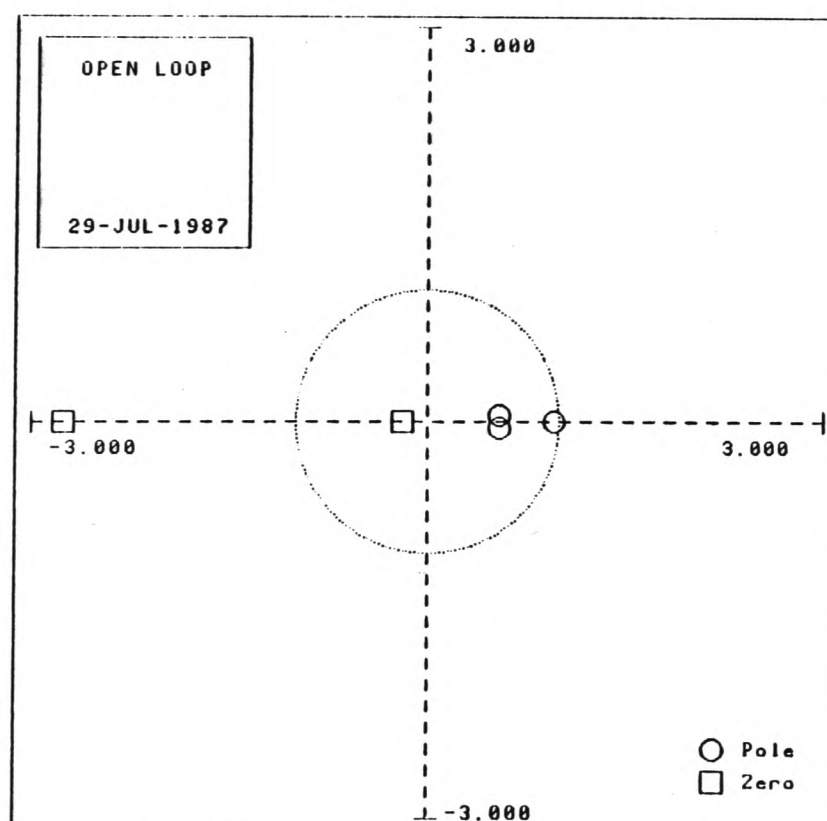


Figure 2.10: Open loop pole and zero locations for Example 2.2.

Based upon the interpretation of T as a filter in the GPC prediction equation (2.23) that roughly determines the bandwidth of the control calculation a reasonable choice for the cut-off point for T would be such that it corresponds to the rise-time of the dominant first order mode. Thus any effects due to unmodelled dynamics of higher frequency than the estimated plant should be attenuated. T is therefore chosen in the subsequent experiments to be :

$$T = (1 - 0.95q^{-1})^n \quad (2.45)$$

where n is the order of the filter and determines the rate of roll-off. The frequency responses of $G(q^{-1})$, $\hat{G}(q^{-1})$ and the band-pass filters, Δ/T , are shown in Fig. 2.11. It can be seen that the second-order filter, when cascaded with the differencing operation, introduces proper band-pass filtering, and that the cutoff frequency of both filters coincides with the point where the reduced order model (2.44) is becoming less accurate.

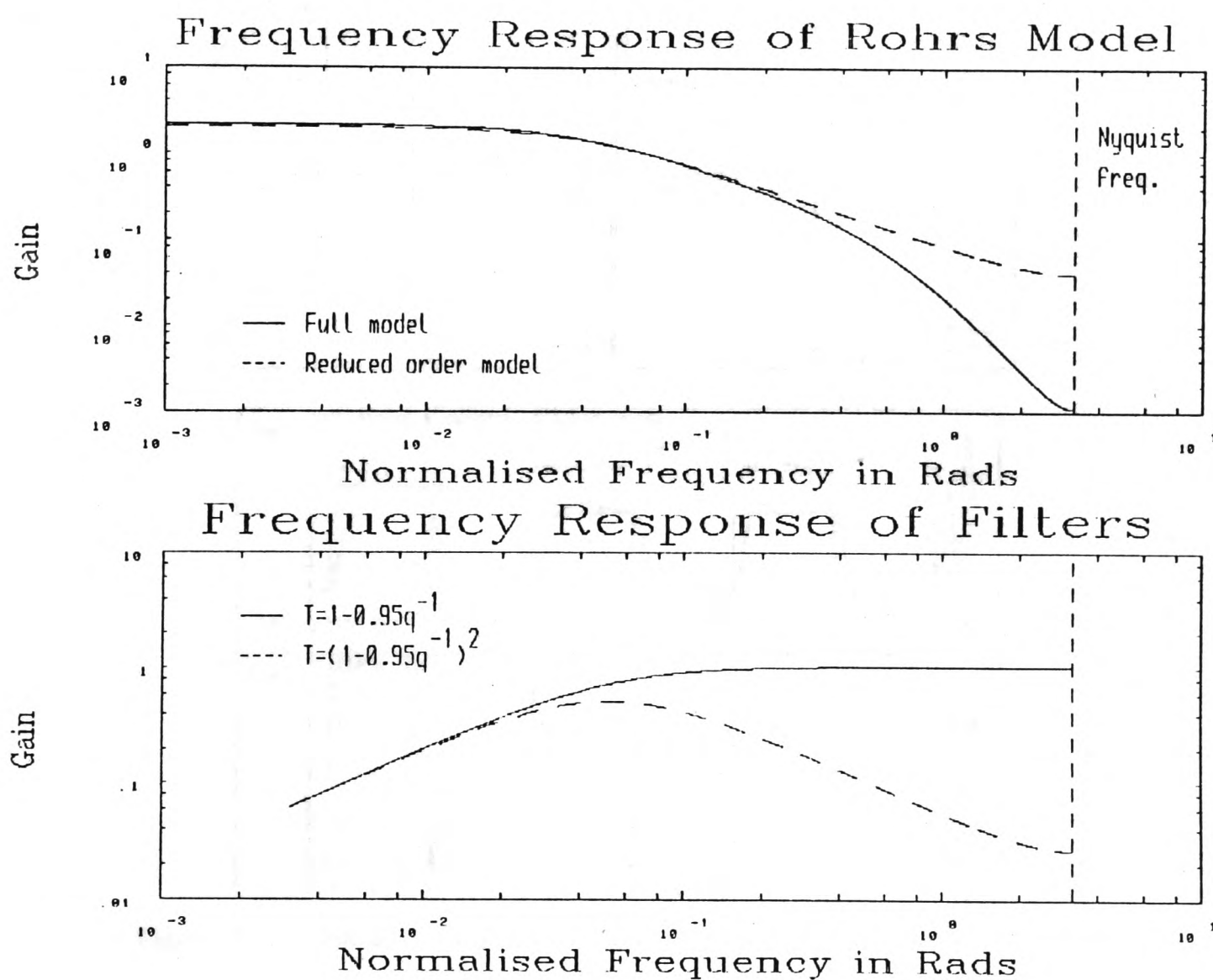


Figure 2.11: Frequency responses for Rohrs' model and associated filters.

As the experiment now uses an estimated model (2.44) and an actual model (2.43) that are no longer the same, so the characteristic equation must be rewritten:

$$\hat{R}A\Delta P_d + q^{-1}B\hat{S} = 0 \quad (2.46)$$

where '^' refers to the fact that these polynomials were obtained using the estimated model polynomials $\hat{A}$ and $\hat{B}$. The closed loop pole locations were then observed for various values of T with default settings for the remaining design parameters. A simulation run was performed as verification.

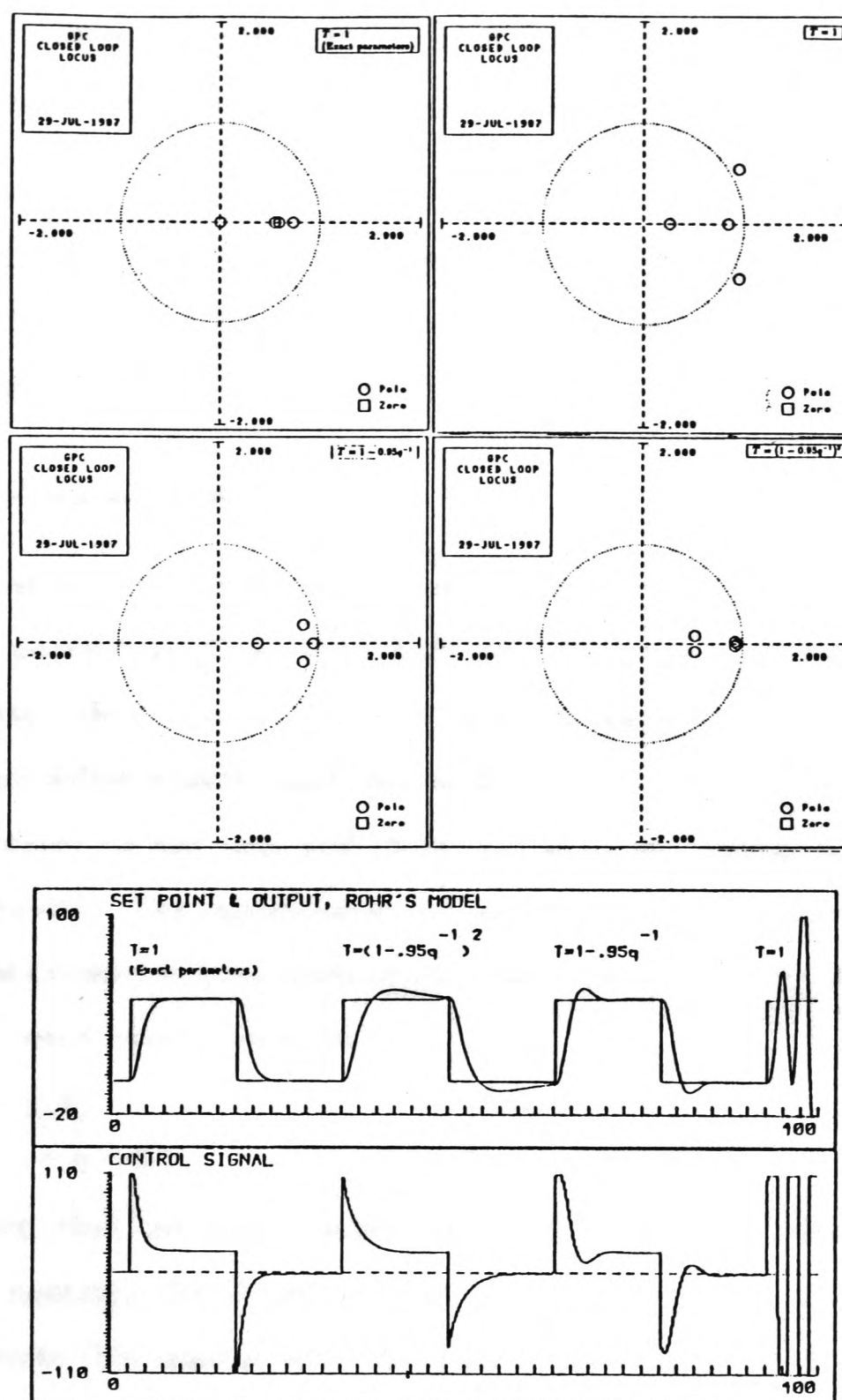


Figure 2.12: The effect of $T(q^{-1})$ in the presence of unmodelled dynamics

The results are plotted in Fig. 2.12 and described below:

- The estimated model is exact (i.e. $\hat{G}(q^{-1}) = G(q^{-1})$) and $T = 1$. This is provided for reference and shows that 'default' GPC is capable of stable control of this example

without oscillation if the estimated parameters are correct.

- *The estimated model is given by (2.44) and $T(q^{-1}) = 1$. The closed loop is unstable, this instability resulting from the presence of the unmodelled dynamics.*
- *The estimated model is given by (2.44) and $T(q^{-1}) = 1 - 0.95q^{-1}$. The closed loop is stabilised but the presence of a complex pair of poles leads to overshoot and oscillation.*
- *The estimated model is given by (2.44) and $T(q^{-1}) = (1 - 0.95q^{-1})^2$. The closed loop remains stable with less overshoot but is significantly slower.*

Thus the importance of T as a means of increasing the robustness of GPC in the presence of unmodelled dynamics has been demonstrated for this example, at least a first order T being required to stabilise the system for what would superficially appear to be a reasonable reduced order model.

2.4.8.2 Load disturbance rejection

The CARIMA plant model of (2.1) assumes an integrated white noise structure, filtered by a noise ‘colouring’ polynomial $C(q^{-1})$ of which $T(q^{-1})$ may be regarded as a fixed estimate. For the special case, when $T(q^{-1}) = C(q^{-1})$ and the assumed noise structure is correct, then the noise is modelled exactly and the residual noise, after filtering by T , is *white*. In practice this will never be the case and there will exist mis-match between the assumed and actual noise models. The significance of T under these conditions, and its effect upon the load disturbance rejection properties of the closed loop, is best understood by reference to the equivalent closed loop expression:

$$y(t) = \frac{BTP_d}{RA\Delta P_d + q^{-1}BS}w(t-1) + \frac{RP_d}{RA\Delta P_d + q^{-1}BS}C\xi(t) \quad (2.47)$$

It can be shown that for exact parameters the closed loop characteristic equation $RA\Delta P_d + q^{-1}BS$ contains the T polynomial as a factor. In the term corresponding to the set point response this cancels with the T of the numerator and thus should have no effect upon the servo performance of the closed loop. However, the factor of T remains in the denominator of the disturbance term and thus specification of $1/T$ as a low pass filter is expected to reduce the effect of high frequency load disturbances. This may also be seen from the actual control calculation itself (2.9),

$$\Delta \mathbf{u}(t) = [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f}) \quad (2.48)$$

where the elements of $\mathbf{f}$ are themselves filtered by T , e.g. $y(t)/T(q^{-1})$ (see 2.23). As the bandwidth of the inputs to the control calculation is limited it follows that the bandwidth of the control signals must also be limited by the cut-off frequency of a low-pass $1/T$.

The ability to specify control signal activity during load disturbance rejection independently from that required during servo operation is known as 'load disturbance tailoring' and represents a very desirable feature for a general purpose practical controller. The degree to which this separation can be achieved in practice depends upon the quality of the estimated model, but nonetheless it is expected to be effective for most industrial processes control applications.

In order to observe the potential benefits of using T for independent load disturbance tailoring a simulation run was performed using the simple 1st order model of Example 2.1, with the estimated parameters fixed throughout the experiment at their correct values, and an initial value of $T_0 = 1$. The experiment consists of a set point change of 50 units magnitude followed by the addition, to the output variable, of a step load disturbance of magnitude 10 units. After 40 samples this is removed and the set-point subsequently returned to its original value. This sequence is repeated three more times for different values of T , namely $T_1 = 1 - 0.4q^{-1}$, $T_2 = 1 - 0.8q^{-1}$, and $T_3 = (1 - 0.8q^{-1})^2$ respectively. The results of the simulation run are plotted in Fig. 2.13 and the closed loop pole locations given in Table 2.2.

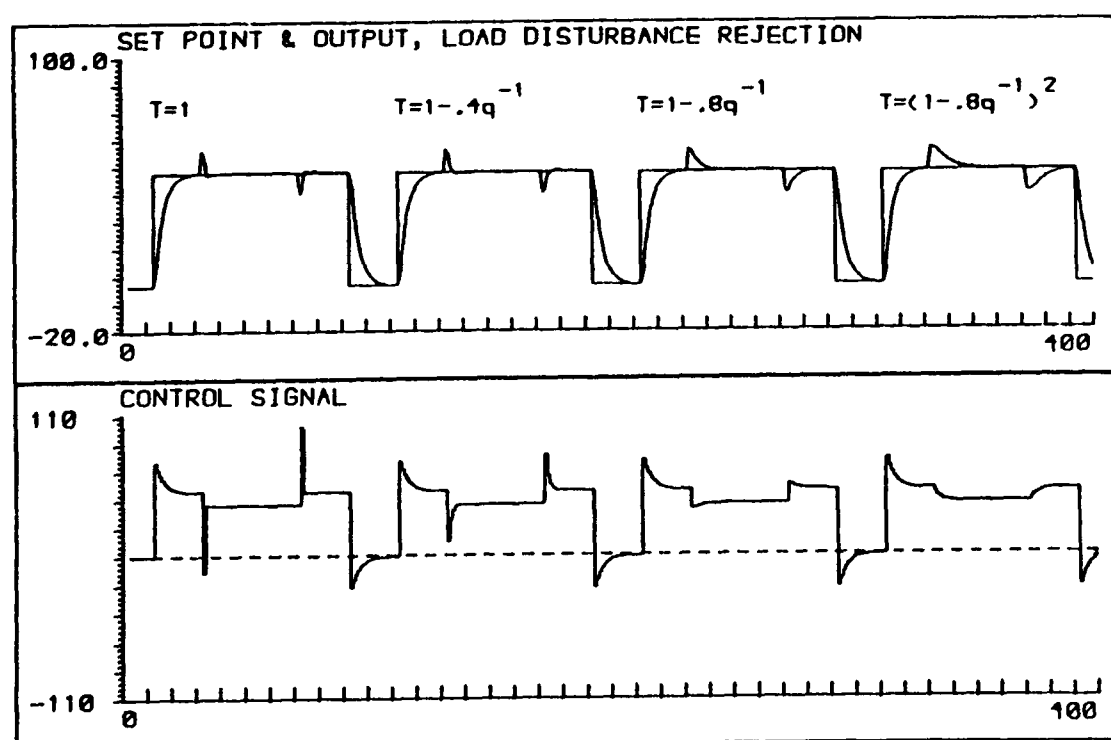


Figure 2.13: The effect of $T(q^{-1})$ upon load disturbance rejection

Pole locations $T(q^{-1})$	Fig.2.13 Roots
1	0.729
$1 - 0.4q^{-1}$	0.729 0.400
$1 - 0.8q^{-1}$	0.729 0.800
$(1 - 0.8q^{-1})^2$	0.729 0.801 0.799

Table 2.2: The effect of $T(q^{-1})$: pole locations.

With exact parameters the variations in T appear to have no effect upon the set point response which remains unaltered throughout the experiment. This is explained above and confirmed in Table 2.2 where the roots of T appear as poles and are thus cancelled out. The load disturbance rejection is, however, significantly affected and becomes progressively slower. For $T_0 = 1$ the rejection of the load disturbance is very rapid with the output being restored to the set point in approximately 2 samples with correspondingly active control. T_1 represents a filter with a bandwidth approximately double that of the process, and T_2 a filter of roughly the same bandwidth of the process. Both of them have a significant effect, progressively reducing the control activity and slowing down rejection. The most dramatic effect is seen, however, when the second order T_3 is introduced, a filter that although of the same bandwidth of the previous T_2 has twice the *roll-off* at high frequency (see Fig. 2.11). The effect of the load disturbance persists for substantially longer (≈ 15 samples) because of the very ‘gentle’ control action.

The results of this experiment show the potential usefulness of the $T(q^{-1})$ polynomial as a GPC design parameter that determines the severity of load disturbance rejection independently of the set-point response.

2.5 MULTIVARIABLE GPC

In Chapter 1 (1.3.2) the multivariable extension of GPC was suggested as being particularly promising for two reasons:

- *Insensitivity to time-delay*: In principle the presence of unknown and/or time-varying time-delay may be incorporated in GPC by overparameterisation of the B parameters of the estimated model. If a time-delay is present in excess of that assumed, i.e.

$k > \hat{k}$, then early B parameters would be expected to go to zero. This relies upon the fact that for a long prediction horizon GPC is not a cancellation control strategy (see section 2.4.3) and thus would not become unstable if, for example, $b_0 \rightarrow 0$. For the more complicated multivariable case this removes the requirement for prior knowledge of a *delay matrix* — a severe practical disadvantage of many earlier methods (Borison, 1979; Koivo, 1980; Goodwin & Long, 1980; Keviczky & Kumar, 1981; Elliot & Wolovich, 1982; Dugard & Dion, 1983).

- *Insensitivity to input-output pairing:* For a ‘full’ multivariable GPC implementation there is no requirement for prior knowledge of dominant input-output combinations as these may be automatically identified by the estimation of a ‘full’ model of the system. In practice, for processes with large interactions between loops such a choice may not be clear, with an incorrect pairing possibly leading to poor control or even instability (Shinskey, 1979).

In this section two GPC control structures are considered, ‘full’ multivariable GPC which is based upon a full $n \times n$ process model and thus effectively solves n simultaneous equations to calculate the control signals for individual loops; and ‘distributed’ GPC where separate SISO controllers are implemented in a multi-loop structure which use feed-forward connections to account for interaction and thus improve decoupling. The relative performance of these strategies are illustrated for a simulation example in section 2.5.3 and, in practice, for the control of a spray drying tower in Chapter 5.

2.5.1 ‘Full’ multivariable GPC

The development of the multivariable GPC control law closely follows that of the SISO algorithm, previously described in sections 2.2 and 2.3. Thus the development begins with the definition of a n -input, n -output multivariable CARIMA process model:

$$\mathcal{A}(q^{-1})\Delta\mathbf{y}(t) = \mathcal{B}(q^{-1})\Delta\mathbf{u}(t - k) + \mathcal{C}(q^{-1})\mathbf{e}(t) \quad (2.49)$$

where $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ are now polynomial matrices of dimension $n \times n$, e.g. (for a 2×2 process):

$$\mathcal{A}(q^{-1}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} (a_1)_{11} & (a_1)_{12} \\ (a_1)_{21} & (a_1)_{22} \end{bmatrix} q^{-1} + \dots + \begin{bmatrix} (a_{\delta A})_{11} & (a_{\delta A})_{12} \\ (a_{\delta A})_{21} & (a_{\delta A})_{22} \end{bmatrix} q^{-\delta A}$$

The quantities in bold, $\Delta \mathbf{y}$, $\Delta \mathbf{u}$ and $\mathbf{e}$, are the differenced output, differenced input and noise $n \times 1$ dimensional vectors respectively. The cost-function for multivariable GPC is shown below:

$$J_{MGPC} = \sum_{j=1}^{NY} [\hat{\Psi}(t+j) - \mathbf{w}(t+j)]^T [\hat{\Psi}(t+j) - \mathbf{w}(t+j)] + \sum_{j=1}^{NU} \Delta \mathbf{u}(t+j-1)^T \Lambda \Delta \mathbf{u}(t+j-1) \quad (2.50)$$

where, instead of using expectations as in (2.8), predictions are used directly, and $\hat{\Psi}$ is a vector of *auxiliary* predicted outputs given by:

$$\hat{\Psi}(t+j) = \mathcal{P}(q^{-1}) \hat{\mathbf{y}}(t+j) = \begin{bmatrix} P_1(q^{-1}) & & \\ & \ddots & \\ & & P_n(q^{-1}) \end{bmatrix} \begin{bmatrix} \hat{y}_1(t+j) \\ \vdots \\ \hat{y}_n(t+j) \end{bmatrix} \quad (2.51)$$

$\mathcal{P}(q^{-1})$ being chosen as a diagonal transfer function matrix with entries corresponding to the desired inverse model between the set-point and the output in individual channels. $\mathbf{w}$ is a vector of set-points, and Λ is a diagonal matrix enabling specification of different values of control weighting for each input independently:

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \quad (2.52)$$

The development of a prediction equation that predicts the auxiliary outputs $\hat{\Psi}(t+j)$ is achieved via the extended Diophantine equation shown below:

$$\mathcal{P}_n \mathcal{P}_d^{-1} \mathcal{T} = \mathcal{E}_j \mathcal{A} \Delta + q^{-j} \mathcal{F}_j \mathcal{P}_d \quad (2.53)$$

where it is important to note that, as in the SISO development, $\mathcal{C}$ is not used directly but a diagonal observer polynomial matrix, $\mathcal{T}$, is considered instead:

$$\mathcal{T}(q^{-1}) = \text{diag} \begin{bmatrix} T_1 & \dots & T_n \end{bmatrix} \quad (2.54)$$

When combined with the process model of (2.49), this gives:

$$\hat{\Psi}(t+j) = \mathcal{T}^{-1} \left[\mathcal{G}_j(q^{-1}) \Delta \mathbf{u}(t+j-1) + \mathcal{F}_j(q^{-1}) \mathcal{P}_d^{-1} \mathbf{y}(t) \right] \quad (2.55)$$

with $\mathcal{G}_j = \mathcal{E}_j \mathcal{B}$. This may be rewritten as:

$$\hat{\Psi}(t+j) = \tilde{\mathcal{G}}_j(q^{-1})\Delta\mathbf{u}'(t+j-1) + \tilde{\mathcal{G}}_j(q^{-1})\Delta\mathbf{u}'(t-1) + \mathcal{F}_j(q^{-1})\mathbf{y}'(t) \quad (2.56)$$

where, $\mathbf{u}'$ and $\mathbf{y}'$ indicate that the individual elements of the vector have been filtered by the corresponding diagonal element of the $\mathcal{T}$ matrix. This may be further rearranged to:

$$\hat{\Psi}(t+j) = \tilde{\mathcal{G}}_j(q^{-1})\Delta\mathbf{u}'(t+j-1) + \hat{\Psi}'(t+j | t) \quad (2.57)$$

where, as in the SISO case 2.4, the prediction $\hat{\Psi}(t+j)$ may be split into two distinct parts:

- $\hat{\Psi}(t+j | t)$, the prediction of the auxiliary outputs assuming no future changes in the control signal.
- $\tilde{\mathcal{G}}_j(q^{-1})\Delta\mathbf{u}'(t+j-1)$, the prediction of the output that is dependent upon the future control increments yet to be calculated.

Thus, with reference to the SISO case (2.6) it is possible to calculate $\hat{\Psi}(t+j | t)$ simply by iterating the estimated plant model for the 'free-responses' of each auxiliary output. The elements of the polynomial matrix, $\tilde{\mathcal{G}}_j$, correspond to the appropriate elements of the step-responses of individual channels and may also be obtained by iteration of the estimated plant model.

Again following the standard derivation of Mohtadi (1986) minimisation of the cost-function of (2.50) leads to the multivariable control calculation given by:

$$\Delta\mathbf{u}(t) = \begin{bmatrix} I_n & 0 & \dots & 0 \end{bmatrix} [\mathcal{G}^T \mathcal{G} + \Lambda]^{-1} \mathcal{G}^T (\mathbf{w} - \mathbf{f}) \quad (2.58)$$

If, for simplicity of notation, the control horizon NU is assumed equal for all channels, then the matrix $\mathcal{G}$ is of dimension $(n \times NY) \times (n \times NU)$, i.e.

$$\mathcal{G} = \begin{bmatrix} G_0 & 0 & \dots & 0 \\ G_1 & G_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ G_{NY-1} & G_{NY-2} & \dots & G_{NY-NU} \end{bmatrix} \quad (2.59)$$

whose elements, G_i , consist of the elements of the polynomial matrix $\tilde{\mathcal{G}}(q^{-1})$, and

$$\begin{aligned} \mathbf{w} &= [w_1(t+1) \quad \dots \quad w_n(t+1) \quad w_1(t+2) \quad \dots \\ &\quad w_n(t+2) \quad \dots \quad w_1(t+j) \quad \dots \quad w_n(t+j)]^T \\ \mathbf{f} &= [\hat{\psi}_1(t+1 | t) \quad \dots \quad \hat{\psi}_n(t+1 | t) \quad \hat{\psi}_1(t+2 | t) \quad \dots \\ &\quad \hat{\psi}_n(t+2 | t) \quad \dots \quad \hat{\psi}_1(t+j | t) \quad \dots \quad \hat{\psi}_n(t+j | t)]^T \end{aligned}$$

Since GPC is a receding-horizon control strategy only the first control increment is required for each input, hence the premultiplication of (2.58) by $[I_n \ 0 \ \dots \ 0]$.

It is obvious from the above that the multivariable GPC control calculation proceeds in exactly the same fashion to that of the SISO algorithm, the only difference being the increased order of calculation at each stage with polynomial matrices replacing polynomials, vectors replacing scalars etc. The assumption of diagonal $\mathcal{P}$ and $\mathcal{T}$ polynomial matrices allow specification of closed loop behaviour in a similar fashion to that of the SISO algorithm, but are commutative and thus simplify the control derivation. Similarly the presence of a control horizon, $NU \ll NY$, greatly simplifies the control calculation by reducing the order of the matrix inversion in (2.58) that could otherwise be prohibitive for large values of n and/or NY .

The properties of this multivariable GPC algorithm, including the specification and effect of the 'horizons' and the design parameters for individual loops, P_n , T_n and λ_n , are in general the same as the corresponding features of the SISO algorithm. If the GPC design parameters corresponding to each loop are specified the same, then the symmetry of the control law is, in principle, retained and input/output ordering or pairing has no effect upon the control performance.

It is important to note, however, that the quadratic cost-function of MGPC only considers set-point tracking and *does not specifically consider decoupling*. This may be illustrated by rearranging the MGPC cost-function (2.50) for a 2×2 example:

$$J_{MGPC} = \sum_{j=1}^{NY} \overbrace{[\hat{y}_1(t+j) - w_1(t+j)]^2}^{\text{loop 1}} + \sum_{j=1}^{NY} \overbrace{[\hat{y}_2(t+j) - w_2(t+j)]^2}^{\text{loop 2}} + \sum_{j=1}^{NU} \Delta \mathbf{u}(t+j-1)^T \Lambda(j) \Delta \mathbf{u}(t+j-1) \quad (2.60)$$

where it is clear that the minimisation, and thus the control calculation, depends upon the relative magnitude of $\hat{y}_1$ and $\hat{y}_2$. Exact decoupling is therefore only possible if (2.60) can be brought to zero, something that will rarely be the case in practice. The practical significance of this observation is two-fold:

- To ensure 'symmetric' control performance in individual loops using MGPC care has to be taken with the scaling of the process measurements, thus perhaps requiring some form of 'normalisation' such that under typical operating conditions their internal¹⁰

¹⁰ where 'internal' values are those actually used within the control calculation after ADC scaling and/or normalisation.

values are of a similar magnitude.

- Decoupling of a particular loop may be enhanced by weighting the corresponding output significantly more than the others, thus decoupling that output from set-point changes in the other loops (Maurath *et al*, 1986).

This may therefore be regarded either as a problem with this type of multivariable control strategy, or as an extra design parameter that may be used to determine decoupling performance.

2.5.2 Distributed GPC

An alternative to the 'full' multivariable control structure is the concept of *de-centralised* or *distributed* control where for a n -input n -output process $n \times$ SISO controllers are used, one for each of the control loops defined. Implementation of relatively complex adaptive algorithms as GPC in such a structure has been made possible by the continuing rise in the reliability and capability of microprocessors along with a decline in the cost (Davison, 1976; Costin & Buchner, 1983; Peel *et al*, 1985). The adaptive multi-loop control of a simulated cooling coil reported in (Jota & Dexter, 1986; Jota, 1987) is of particular interest as it includes a distributed GPC implementation.

Although multi-loop control suffers potentially from problems of loop pairing selection and reduced decoupling performance, it can in practice offer two significant advantages:

- **SIMPLICITY:** For a given process the adaptive implementation of a distributed GPC structure requires the estimation of less parameters than the corresponding full multivariable GPC controller. As it is well known that the performance of typical on-line estimation schemes, such as RLS deteriorates, rapidly with increase in the number of estimated parameters (Ljung & Söderström, 1983) this can represent a very strong justification for the use of distributed control, especially for multivariable processes of large dimension. This is often referred to in the literature as the 'parameterisation issue' and is acknowledged to be at present the most serious limitation in the practical application of full multivariable control strategies (Elliot & Wolovich, 1982). Note that additional benefits of a distributed GPC control structure are that, because of the separate specification of cost-functions for each loop, the overall control performance is independent of relative measurement scaling; and that it is possible

to specify different sample rates for individual loops (Morris *et al*, 1982), vital in the presence of loops with significantly different speed of response and difficult to achieve with MGPC.

- **INTEGRITY:** The distribution of control tasks between several semi-autonomous GPC controllers should, in principle, increase the robustness of the overall control system. In particular such a control strategy may be implemented in a distributed hardware sense (i.e. with individual low-cost microprocessors for individual loops) and thus the consequences of failures, whether in the hardware (e.g. sensor failure) or in the controller (e.g. poor estimation), should more easily be confined to 'local' loops (Patel & Munro, 1982; Jota *et al*, 1986).

For these reasons a GPC control strategy is developed in this section that is suitable both for multivariable control and for distributed implementation. This algorithm is referred to as DGPC.

Compared to the full-A full-B representation of the multivariable CARIMA process model given below for a 2×2 system:

$$\overbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}^A \begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \overbrace{\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}}^B \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} \quad (2.61)$$

the parameterisation for DGPC, with and without the inclusion of feed-forward terms, may be written as:

$$\begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} \quad (2.62)$$

and

$$\begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} \quad (2.63)$$

respectively. The off-diagonal terms in the B matrix of (2.62) correspond to feed-forward $\mathcal{D}$ parameters (2.25) but are written as B 's for comparison with the parameterisation of (2.61). This corresponds to the 'P-canonical' representation of a multivariable process (Isermann, 1981; Peel, Morris & Tham, 1985) which assumes that all interactions are feed-forward couplings. Without feed-forward the representation of (2.63) effectively ignores the potential presence of interaction in the multivariable system to be controlled and relies

instead upon the load disturbance rejection properties of the individual loop controllers to decouple the system.

Although for most multivariable process control applications the P-canonical representation is expected to be adequate its feed-forward implementation means that, in principle, its decoupling properties should be inferior to that of a full multivariable algorithm. This is because the latter effectively solves a set of n simultaneous equations in order to calculate the control signals for each loop and thus can take immediate action to remove the effects of interaction from other loops. For a feed-forward implementation the prediction at time t , for a given loop, is made upon the assumption that the feed-forward signals from other loops remain constant, hence there is a 1-step delay in compensation. For typically slow process control applications this is not expected to make a great difference but could be a significant limitation in critical loops where interaction must be removed quickly, or in high performance applications.

If it is known that the interaction in a multivariable process is predominantly one-way, e.g. for a 2×2 process control activity in one loop causes interaction in another but not *vice versa*, then it is possible to *enhance* the feed-forward structure of (2.62) simply by appropriate ordering of the control calculations. Thus the control signal that causes the interaction would be calculated first and then its *new* value used as the feed-forward term for the second loop's control calculation, rather than being proxied by its previous value. In principle this should improve dynamic decoupling of the second loop because it now takes immediate account of control activity, and thus potential interaction, in the first loop.

2.5.3 Some illustrative simulation studies

In this section the control performance of the various multivariable GPC algorithms, described in the two preceding sections¹¹, are examined in simulation. The process to be simulated consists of two inner loops, air flow and pressure, of a spray drying tower which are controlled by manipulation of inlet (quench) and outlet (exhaust) air dampers. The parameters of the simulated model were obtained using a RLS estimator during GPC control trials upon an actual full-scale tower (see Chapter 5).

¹¹ the DGPC controllers were implemented on one computer and thus were only 'distributed' in an algorithmic sense.

Example 2.3 Consider the simulated model of the inner loops of a spray drying tower:

$$\begin{bmatrix} 1 - 0.6q^{-1} & -0.005q^{-1} \\ 0.0043q^{-1} & 1 - 0.56q^{-1} \end{bmatrix} \begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \begin{bmatrix} 0.05 + 0.6q^{-1} + 0.33q^{-2} & 0.02 + 0.05q^{-1} + 0.015q^{-2} \\ 0.002 + 0.02q^{-1} + 0.01q^{-2} & -0.005 - 0.04q^{-1} - 0.01q^{-2} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} \quad (2.64)$$

where:

y_1 — Air flow u_1 — Quench air input
 y_2 — Pressure u_2 — Exhaust air input

The step response of the simulated system is shown in Fig. 2.14 where it is clear that tower exhibits significant interaction only in one direction, with a decrease in quench air causing a large decrease in pressure. The dynamics of all the channels are very similar.

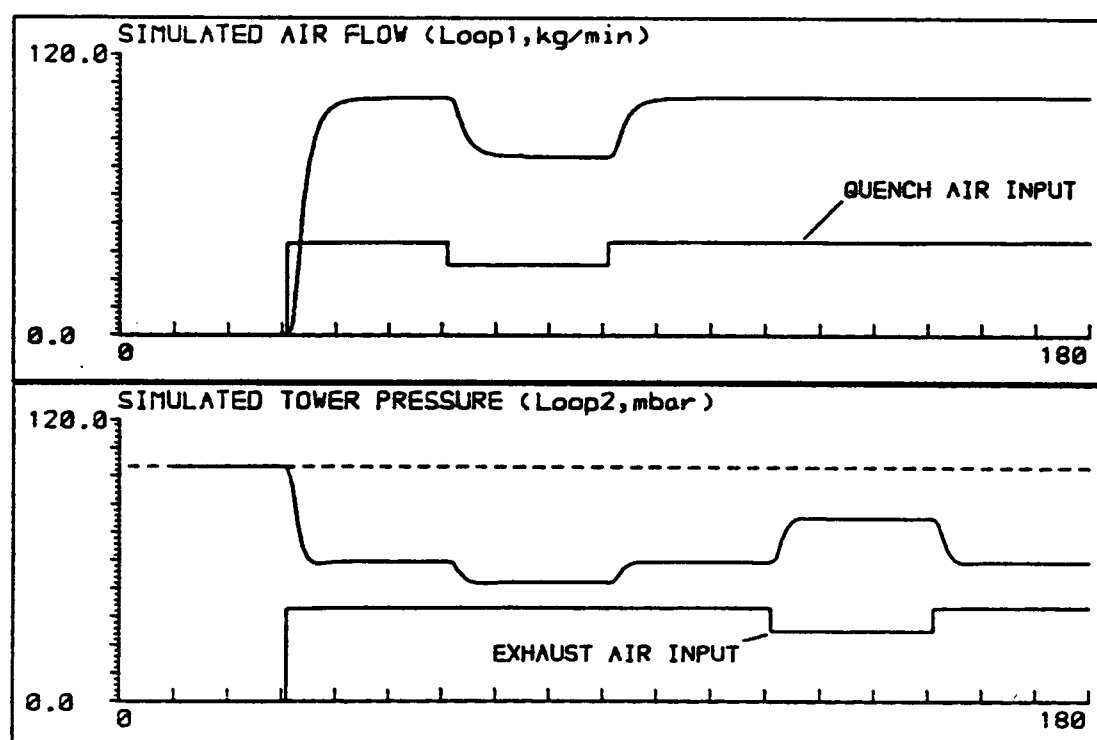


Figure 2.14: Step responses of the simulated spray drying tower of Example 2.3.

The first experiment conducted upon this example demonstrates that *scaling* of the measured variables can be important for symmetrical control performance using full multivariable GPC (see section 2.5.1). The engineering units associated with the spray drying tower measurements mean that at the operating point (100 kg/min, -1mbar) the same percentage error in air flow (e.g. 1% → 1 unit) is far larger numerically than the corresponding error in pressure (e.g. 1% → 0.01 units). Because of the quadratic nature of the multivariable GPC cost function (2.50) such a disparity in control errors is squared

and thus becomes much larger. Hence the air flow output is expected to be controlled far more tightly¹² than tower pressure. Scaling of the pressure measurement before it enters the control calculation so that it becomes numerically similar to that of air flow reduces this disparity and is expected to tighten pressure loop control and, in particular, improve decoupling.

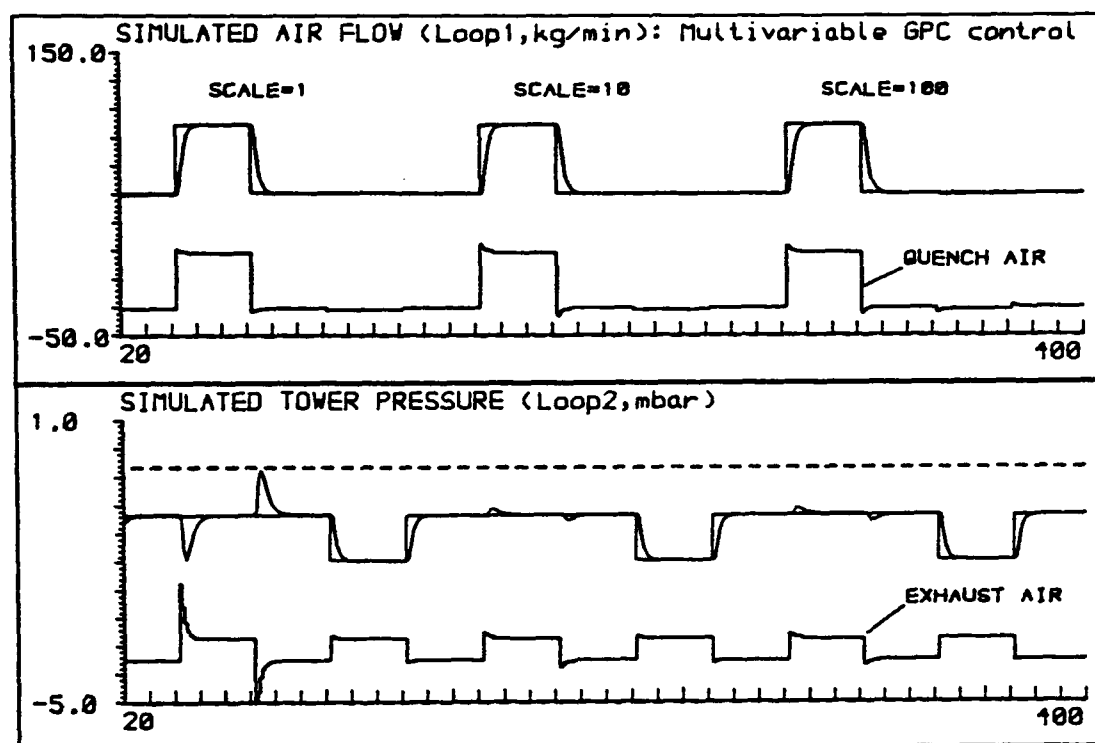


Figure 2.15: Full multivariable GPC control: the effect of scaling.

Thus Fig. 2.15 shows the results of an experiment where full multivariable GPC was used to control the tower simulation, with exact estimated parameters, and default controller parameters. Step-up step-down set point changes were applied to the air flow and tower pressure loops in turn, this procedure being repeated three times with the pressure measurement scaled up by 1, 10 and 100 respectively. From the figure it is immediately clear that control of individual loops is always tight and that decoupling of tower pressure improves dramatically when scaled, the value of 100 giving the best results.

In the next experiment the same procedure was used to demonstrate the role of feed-forward in distributed GPC control (section 2.5.2) of the tower simulation. Thus instead of changing scaling (both loop controllers using 'raw' engineering units in their control calculations) three successive controllers were tested, DGPC with no feed-forward, DGPC with ordinary feed-forward and DGPC with enhanced feed-forward (see page 62). For the latter two controllers it was the quench air input that was fed forward to the pressure loop

¹²in the absence of minimum variance control the minimisation of the cost function does not necessarily lead to the minimisation of the individual loops control errors.

as it is in this direction that interaction is significant (Fig. 2.14). Default GPC control parameters were again specified for both loops.

Note that in order to enable comparison with the results obtained in the preceding experiment the parameterisations of the estimated models for each controller were chosen so as to correspond, as far as possible, to that of the full multivariable model. Thus, without feed-forward, 3 B and 1 A parameters were estimated for both loops but with feed-forward 4 B , 2 A and 4 D parameters were estimated for the pressure loop. The values of the estimated parameters were obtained from a separate identification experiment (not shown) and fixed throughout the experiment.

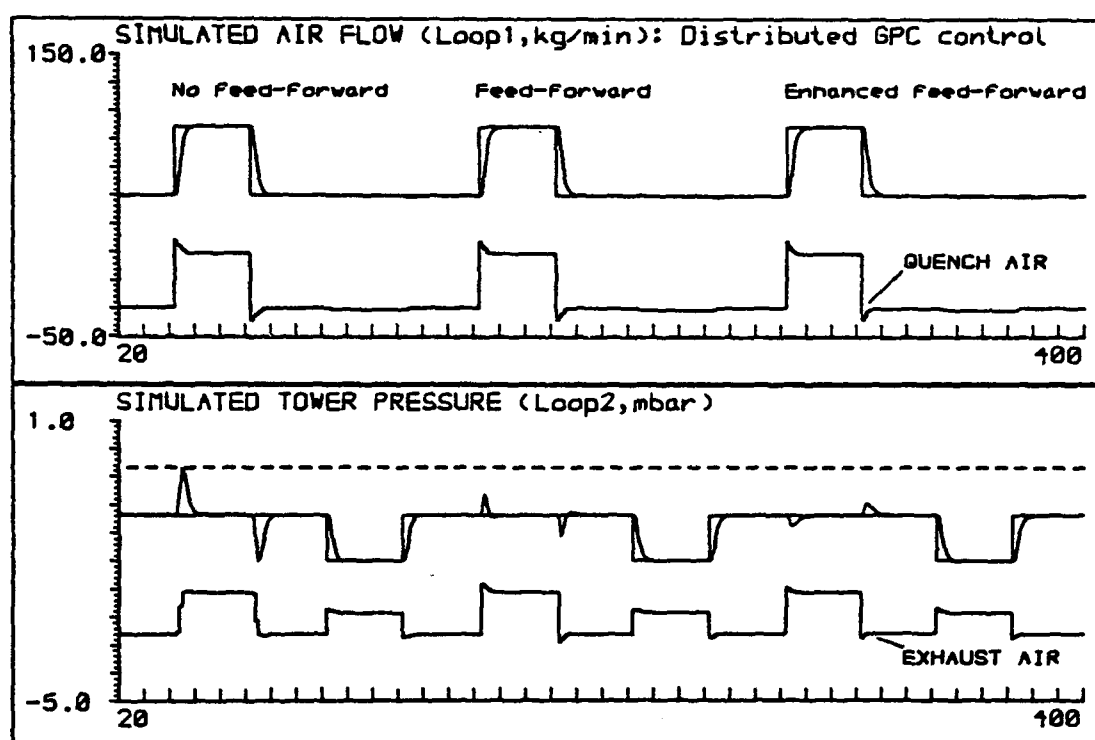


Figure 2.16: Distributed GPC control: the effect of feed-forward.

From the figure it may be seen that the set point following in individual loops is once again tight. Furthermore decoupling of the tower pressure during set point changes in air flow is dramatically improved when feed-forward is used. Enhanced feed-forward again improves this decoupling but only slightly. By comparison with Fig. 2.15 the overall quality of control is almost as good as that achieved using *correctly scaled* full multivariable GPC.

2.6 CONCLUSIONS

This chapter describes the development and interpretation of the GPC algorithm (Clarke *et al*, 1984,87), one of the latest predictive control laws to be used in adaptive control. For a given process model, estimated model and specification of GPC's design parameters

a means of calculating the closed loop characteristic equation was derived. This was incorporated into a GPCTEST package and used to obtain quantitative analysis results in the form of discrete loci of closed loop pole positions.

The results obtained not only verified earlier theoretical and intuitive predictions concerning the expected performance of GPC (Mohtadi, 1986) but also provide insight into how GPC should be specified in practice. In particular the 'default' GPC specification of $NS(1); NU(1); NY(10); P(1); \lambda(0); T(\text{low-pass filter})$ was demonstrated to tend towards a 'mean-level' control strategy that is expected to provide stable, robust control for a wide variety of processes, even in the presence of significant unmodelled dynamics.

Two types of multivariable GPC control structures were then described, including a full multivariable version (MGPC) and an alternative multi-loop structure (DGPC) that uses SISO controllers for individual loops. The performance of both these was illustrated in a simulation exercise. This clearly showed that MGPC and DGPC were both capable of excellent control although for the former it is important that the units of each loop are scaled to be of approximately the same magnitude; and that the latter's decoupling properties may be dramatically improved by using feed-forward.

Chapter 3

Towards practical parameter estimation

The adaptive control configuration (see Fig. 3.1) is a flexible control strategy in that it can accomodate a wide variety of parameter adaptation techniques and controller design strategies. For this reason a large proportion of the research effort in adaptive control has been expended on finding suitable combinations. The work presented in this chapter is no exception. In Chapter 2, GPC was shown to offer a robust and flexible control law in the case when the parameters of the assumed model are known. What is now required, if GPC is to achieve the engineering goal of practical application, is the selection and development of a Parameter Adaptation Algorithm (PAA) that provides these parameters in practice.

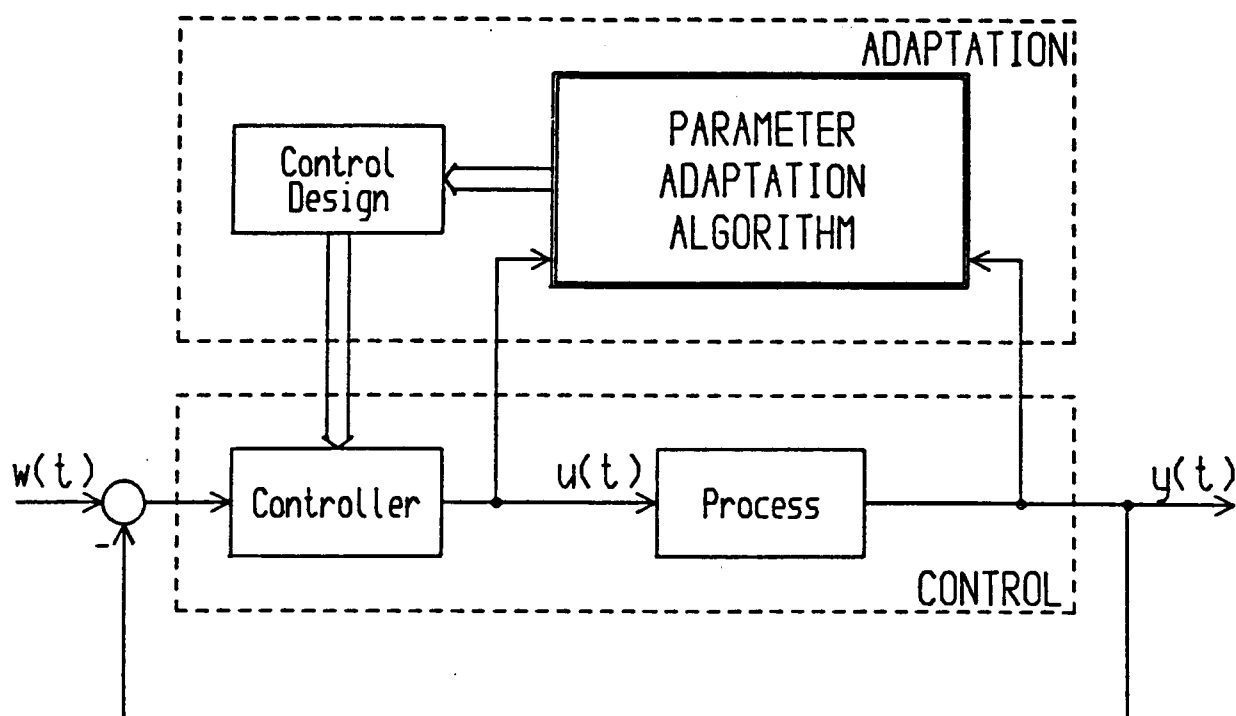


Figure 3.1: Adaptive Control: The adaptation algorithm

3.1 INTRODUCTION

In Chapter 1 a distinction was drawn between *Self-Tuning* and *Adaptive* control: modes of operation which differ only in their use of estimation. A general purpose PAA must be capable of assisting both functions, i.e.

- Initial identification of an ‘adequate’ process model (**Self-tuning**).
- Robust long-term maintenance of an ‘adequate’ model despite process variations (**Adaptation**).

In practice both specifications must be achieved under realistic industrial operating conditions, where problems such as non-linearity and unmeasurable load disturbances will be encountered. This chapter therefore addresses the question:

“What problems do Parameter Adaptation Algorithms encounter during practical application and how can these be overcome for reliable long-term operation ?”

For clarity a preliminary section (3.2) outlines the fundamental areas of the practical estimation problem. The following section (3.3) then selects the well-known recursive least squares (RLS) algorithm as an appropriate prototype estimator and briefly considers its implementation, interpretation and specification. It is demonstrated that whilst this basic algorithm is capable of self-tuning it does not provide long term adaptivity (Åström & Wittenmark, 1973; Tuffs, 1984).

Section (3.4) deals with the requirement for long-term parameter tracking and reviews a representative selection of RLS based algorithms that have been designed to achieve this (Young, 1968; Peterka, 1970; Clarke & Gawthrop, 1975; Fortescue *et al*, 1981; Ydstie, 1982; Kulhavý & Karny, 1984; Goodwin *et al*, 1985). The details of each algorithm are discussed briefly and their performance illustrated with an appropriate simulation study. Also indicated in this section are the principal problems associated with such methods, including ‘blow-up’ (Hodgson, 1982), ‘bursting’ (Åström, 1983), parameter drift etc.

Two novel parameter adaptation algorithms are introduced in section 3.5 that are based on the concept of weighting parameter variations in the cost-function of the RLS estimator. This is shown to overcome the problem of estimator ‘blowup’ whilst successfully retaining long-term adaptivity, to be relatively simple in implementation and commissioning, and to allow meaningful inclusion of detailed *a priori* process knowledge. A further extension

involves the use of 'fault detection' to provide an estimator suitable for use in the presence of large load disturbances. These algorithms are proposed as being particularly suitable for practical application.

Finally in section 3.6 various practical estimator 'jackets' (Clarke, 1981,84) are described. These are heuristic safeguards applied to an estimator in order to enhance its long term integrity. The role of 'expert systems' in their implementation is also considered.

3.2 FUNDAMENTALS OF PRACTICAL ESTIMATION

In order to choose an appropriate algorithm suitable for practical application, from the wide range of algorithms already available (Åström & Eykhoff,1970), the problem must first be clearly formulated. Thus this first section seeks to identify the fundamental areas of practical parameter estimation.

3.2.1 The class of model

The choice of model structure is based upon three important requirements:

- It should be sufficiently *flexible* to describe a wide range of industrial processes.
- It should be *compatible* with the control strategy it is to be mated to.
- It should be readily *implementable* on a digital computer.

A model structure that has commonly been suggested as satisfying all these requirements is the linear difference equation model, which arises naturally due to the implementation of discrete computer-based control algorithms inevitably involving a sampling operation. For a dynamical system with input signal $\{u(t)\}$ and output signal $\{y(t)\}$, these signals being sampled in discrete time, $t = 1, 2, 3, \dots$, this type of model may be written:

$$A(q^{-1})y(t) = B(q^{-1})u(t - k) + e(t) \quad (3.1)$$

where $e(t)$ is a disturbance term of unspecified character, k is the integral dead-time of the system, and $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the delay operator, q^{-1} :

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{\delta A} q^{-\delta A}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_{\delta B} q^{-\delta B}$$

Two possible objections to the use of this class of model are that a complete mathematical description of the process is almost certain to be either impossible or impracticable, and that the sampling operation involves a loss of information. Neither of these are however considered to be severe practical limitations. The former because the need for adaptive control already implies a certain amount of ignorance about the process, and if the sampling interval is approximately 1/10th of the dominant time constant the latter problem has been shown to be insignificant (Clarke,1981; Isermann,1982).

3.2.2 The estimator criterion

The estimator criterion defines the approach used to determine the relative performance of different models. This could therefore be used to choose the model structure, the dimension of the model structure (e.g. the number of parameters in a linear model), or indeed as the basis for the estimation of the unknown parameters. For practical estimation, however, the first two uses are disregarded, with the model structure and the model order normally assumed to be known, or specified *a priori*.

As GPC is based upon long-range prediction of the process output over a horizon, NY , of future samples (Chapter 2), the obvious *criteria of best fit* for a compatible estimator would be the minimisation of the sum of the $1 \rightarrow NY$ step-ahead prediction errors. Unfortunately the algorithmic implementation of this criterion represents a non-linear optimisation problem whose solution is difficult. One possible alternative could be to estimate a bank of NY models, one for each future prediction, but this would be computationally very expensive. Therefore the minimisation of the 1-step-ahead prediction error, using a linear least-squares type of algorithm, is considered to be the next most appropriate criterion because of its relative simplicity.

3.2.3 The experimental conditions

Two distinct classes of experimental conditions may, in general, be defined :

- *Initial Operation*: where process identification has not yet taken place and the process is either in open loop or being controlled by an alternative controller such as a PID.
- *Routine Operation*: where the estimated model is reasonably accurate, the process is in closed loop and the process output is at, or near, its desired value.

A PAA must therefore be capable of operating during both of these phases, with rapid and accurate initial adaptation, and robust long-term parameter tracking (see section 3.2.4).

In the former case it is often advantageous to contrive an artificial experiment so that the improvement of the parameter estimates is rapid, with this period of accelerated learning normally implemented by the injection into the process of an appropriate test signal. Assuming that such an experiment is possible, which it will be in the majority of process control applications (else how were PID controllers tuned?), then the test signal must satisfy two important conditions (Goodwin & Payne, 1977; Ljung & Söderström, 1983) :

- It should excite all *relevant* modes of the system, relevant modes being defined as those likely to be encountered in subsequent control activity (i.e. if regulation is the intended control objective then low frequency information is important, whilst if rapid set point tracking is required then higher frequency information is required).
- It should be chosen such that the process output varies within the normal operating region of the process. This is especially important in the presence of significant non-linearity, inevitable in most industrial processes and occurring in either process characteristics, measuring or actuation devices (see Fig. 3.2).

The introduction of a Pseudo Random Binary Sequence (PRBS) input to the process in open loop, where the clock-period is a multiple of the sample interval, is commonly used to satisfy both these requirements (Everleigh, 1967; Seborg *et al*, 1986). Given the above conditions the PRBS probing signal should be superimposed about a manual input which in normal operation would set the process output to its destined operating point, the magnitude of the PRBS being chosen such that it causes the process output to vary across this operational region whilst subject to any limitations imposed either by production management or by the process itself.

3.2.4 Long-term adaptivity

A fundamental justification for adaptation is the control of processes whose characteristics are significantly time-varying. A practical PAA should therefore always be capable of detecting such process changes and updating the estimated process model accordingly to preserve the quality of closed-loop control. For long-term reliable performance of this parameter tracking property there is thus an inherent requirement for the retention of

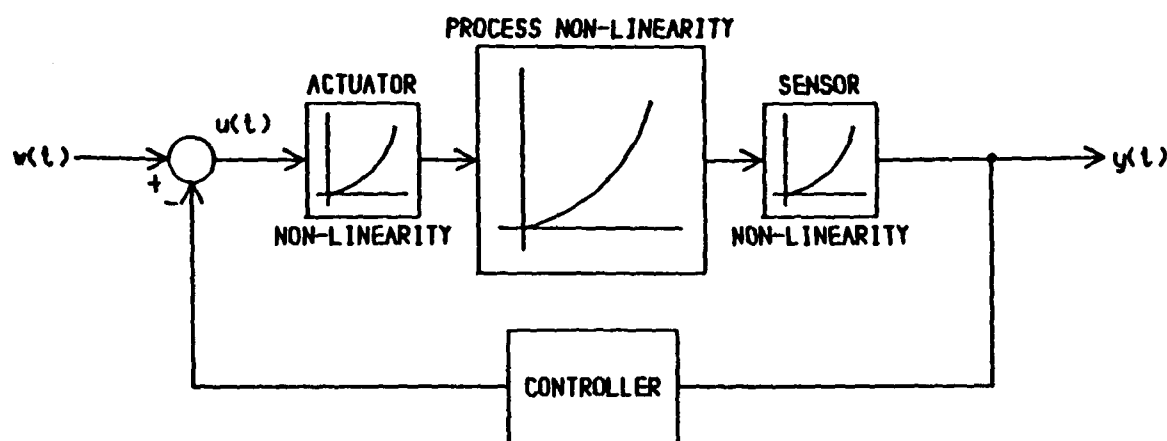


Figure 3.2: Sources of non-linearity.

some form of adaptivity. Unfortunately the price of such persistent alertness is potentially increased sensitivity to random disturbances in the measurements. This can lead to a classic problem where the requirement for performance must be traded-off against that of robustness. Therefore the PAA must be made as robust as possible to the various expected sources of error, including for example, measurement noise, load disturbances, unmodelled dynamics, and even numerical precision.

3.2.5 Prior knowledge

When the concept of self-tuning control was introduced (Kalman, 1958) it was optimistically intended to be a *black-box* solution to the control problem. The ideal self-tuning or adaptive controller was to require no prior knowledge of the process and yet provide good long-term control. Since then a bewildering array of *knobs* (Åström, 1980) have appeared on the surface of the black-box and the importance of *a priori* knowledge has become increasingly recognised, especially for parameter estimation.

The use of such information may be regarded either as a *requirement* for the successful application of parameter adaptation, or as a *bonus* that improves the performance of the particular algorithm used. Its incorporation can be assumed to take place at one of two distinct stages:

- **Estimator commissioning**, where it could be used to define signal conditioning requirements, model complexity, initial values of model parameters etc. The predominant benefits at this stage are the reduction of initial tuning transients and an increase in the speed of commissioning. However the inclusion of accurate information at this early stage can also have significant benefits throughout the entire future use of the parameter estimator.

- **Long-term estimator performance**, where the most basic *a priori* process knowledge may be used in a supervisory manner to increase the robustness of the parameter adaptation algorithm. For example the vast majority of industrial processes are stable by design. If, at any sample, the estimated parameters are updated such that the estimated model is unstable then the update is obviously incorrect and should be disregarded. Failure to disregard such an incorrect result would probably lead to poor control performance. Similarly limits upon process gains, time-delays, speed of response etc. are either known or may be determined. Thus at this stage the use of prior knowledge may be viewed as the imposition of checks upon the allowable results of parameter estimation.

The desire to reduce a parameter estimator's dependence upon prior knowledge is natural in the quest for a generally applicable algorithm. Nonetheless care must be taken to provide algorithms suitable for the inclusion of this knowledge at either of the above stages described above if it is available. Although it appears to abandon the original philosophy of adaptive control it does perhaps represent the easiest way of transferring adaptive control from paper to a practical engineering tool.

3.2.6 Implementation and Commissioning

One of the most fundamental requirements of parameter estimation for adaptive control is that it must be performed in real-time. This demands the use of algorithms that update the estimated parameters at each sample interval on the basis of sequential data, and do not require large amounts of data storage. Such an approach is known throughout the literature as *recursive* parameter estimation.

The PAA should be as simple as possible to set-up, with the specification of any estimator parameters being based upon sound engineering concepts capable of being understood by any process engineer, rather than seemingly arbitrary numbers. If required, the performance of the algorithm should be capable of being modified on-line.

3.3 A BASIC ESTIMATION ALGORITHM

In this section the recursive least squares algorithm, based upon a CARIMA plant model description, is chosen as a simple prototype suitable as the basis of a more practical algorithm.

3.3.1 The CARIMA process model

The CARIMA model (Harris *et al*,1981; Belanger,1983), was described in Chapter 2 as forming the basis for the GPC control algorithm:

$$A(q^{-1})y(t) = B(q^{-1})u(t - k) + \frac{C(q^{-1})\xi(t)}{\Delta} \quad (3.2)$$

where $C(q^{-1})$ is another polynomial in the delay operator; $\xi(t)$ is a random variable with zero mean acting on the process; Δ is the difference operator, $\Delta = 1 - q^{-1}$, and k is the assumed integral time-delay.

There are many justifications for this particular choice of assumed noise structure (Box & Jenkins,1976; Tuffs,1984; Mohtadi,1986) but the major impact of the CARIMA model upon estimation may be seen by consideration of the equivalent process representation given below:

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t - k) + C(q^{-1})\xi(t) \quad (3.3)$$

where the process measurement, $y(t)$, and control signal, $u(t)$, are in a differenced or *incremental* form.

In practice the dynamic response of industrial processes is often non-linear and often takes place about some relatively large value or offset. If the CARIMA model is to satisfy the first estimator requirement of accuracy and flexibility then it must provide an adequate model under such conditions (i.e offset-free prediction). A standard, and generally accepted procedure used in control theory to overcome this modelling problem, is local linearisation. The incremental CARIMA model may therefore be considered as an approximate local linearisation when the point about which the linearisation is to take place, the operating point, is not known.

The alternative approach using full-valued (*positional*) data requires an estimate of an offset term, d ,

$$A(q^{-1})y(t) = B(q^{-1})u(t - k) + C(q^{-1})\xi(t) + d \quad (3.4)$$

to guarantee offset-free prediction, a procedure that was found both in practice and simulation to be unreliable (Hodgson,1982). Whilst incremental estimation has the obvious disadvantage that differencing data with high frequency noise components degrades the signal to noise ratio, it should be possible to overcome this with appropriate signal filtering, and it is thus preferred in practice.

3.3.2 Recursive Least Squares

Least squares (Young, 1968) is only one of the many established prediction error based approaches to the estimation of a linear-in-the-parameters model such as (3.2), the main alternatives being maximum likelihood and Bayesian techniques. It has the practical advantages, however, that it does not require detailed *a priori* information, is a linear calculation, and has a relatively simple recursive implementation.

In order to apply the least squares minimisation the equation of the model given in (3.3) must first be manipulated into an appropriate form, i.e.

$$\Delta y(t) = (1 - A(q^{-1}))\Delta y(t) + B(q^{-1})\Delta u(t - k) + C(q^{-1})\xi(t) \quad (3.5)$$

If an unknown *parameter vector*, θ , is defined as:

$$\theta^T = \begin{bmatrix} a_1 & \dots & a_{\delta A} & b_0 & \dots & b_{\delta B} \end{bmatrix} \quad (3.6)$$

and a *data vector* as:

$$\phi(t-1)^T = \begin{bmatrix} -\Delta y(t-1) & \dots & -\Delta y(t-\delta A) \\ \Delta u(t-k) & \dots & \Delta u(t-k-\delta B) \end{bmatrix} \quad (3.7)$$

Then (3.5) may be rewritten in vector notation as

$$\bar{y}(t) = \Delta y(t) = \phi(t-1)^T \theta + e(t) \quad (3.8)$$

This model describes the measurement increment $\Delta y(t)$ as an unknown linear combination of the components of the observed vector, $\phi(t-1)$, plus a noise term, $e(t)$. Such a model is called a *linear regression* in statistics. Denoting the vector of parameter estimates to be $\hat{\theta}$,

$$\hat{\theta} = \begin{bmatrix} \hat{a}_1 & \dots & \hat{a}_{\delta A} & \hat{b}_0 & \dots & \hat{b}_{\delta B} \end{bmatrix} \quad (3.9)$$

then the least squares algorithm minimises the sum of squares of the *a priori* estimation error, $\varepsilon(t)$, where

$$\bar{y}(t) = \phi(t-1)^T \hat{\theta}(t-1) + \varepsilon(t) \quad (3.10)$$

The quadratic least-squares criterion may thus be written:

$$J_{ls}(\theta) = \frac{1}{2} \sum_{k=t-N}^t [\bar{y}(k) - \phi(k-1)^T \theta]^2 + \frac{1}{2} (\theta - \hat{\theta}(0))^T P(0)^{-1} (\theta - \hat{\theta}(0)) \quad (3.11)$$

The second term of 3.11 is included to account for the initial conditions, with $\hat{\theta}(0)$ is the vector of initial parameter estimates, and $P(0)$ any positive definite matrix. Thus $P(0)$ represents a measure of confidence in the initial estimate $\hat{\theta}(0)$ as its weighting determines the significance of $\hat{\theta}(0)$ in the cost-function.

The development of the actual least squares algorithm from this criterion is not presented here as it is well-known (e.g. Goodwin & Sin, 1984). The resulting recursive parameter update is however shown below, the value of θ satisfying (3.11) at time t being denoted by $\hat{\theta}(t)$:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-2)\phi(t-1)}{1 + \phi(t-1)^T P(t-2)\phi(t-1)} [\bar{y}(t) - \phi(t-1)^T \hat{\theta}(t-1)] \quad (3.12)$$

$$P(t-1) = P(t-2) - \frac{P(t-2)\phi(t-1)\phi(t-1)^T P(t-2)}{1 + \phi(t-1)^T P(t-2)\phi(t-1)} \quad (3.13)$$

where the update may be considered to take place in the above two stages, a *parameter* update (3.12) and a *covariance* update (3.13). It is important to note here that the covariance update is the recursive form of the inverse covariance update

$$P(t-1)^{-1} = P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T \quad (3.14)$$

which uses a matrix inversion lemma (Bierman, 1977) to avoid the otherwise difficult matrix inversion. The estimation procedure at each sample interval is based upon the new measurements $\bar{y}(t)$ and $\phi(t-1)$, and involves the old estimate, $\hat{\theta}(t-1)$, plus a correction term. The term $\bar{y}(t) - \phi(t-1)^T \hat{\theta}(t-1)$ may thus be considered as representing the part of the new output observation that is not accounted for by the input measurements and the last parameter estimate (Eykhoff, 1974).

3.3.3 Commissioning of the Algorithm

Commissioning of the CARIMA model structure involves the specification of six parameters, the model orders δA and δB ; the assumed time-delay $\hat{k}$; the sampling interval Δt ; and initial values for both the covariance matrix, $P(0)$, and the parameter estimates, $\theta(0)$. Whilst the resolution of these parameterisation issues will depend predominantly upon the amount of information available about the process at this commissioning stage it is nonetheless possible to outline some general guidelines:

The choice of the difference equation order, δA , is theoretically process-specific but in practice a compromise must be made between model accuracy and simplicity. For

applications within the process control industry an assumption of a second order model (i.e. $\delta A = 2$) is considered realistic.

For many practical situations, when the effect of noise is appreciable, a delayed 1st or 2nd order system would often provide as elaborate a model as could be justified for the data. Box & Jenkins (1976).

Note that an additional justification for this assumption is the precedent set by the widespread and satisfactory usage of PID controllers throughout the process control industry, which are really only suitable for processes with *predominantly* 2nd order characteristics (Gawthrop, 1986). This assumption should not, however, be regarded as a restriction but rather as a guide.

Once this assumption, corresponding to the order of the underlying continuous-time system, has been made then the order of the $B(q^{-1})$ polynomial must be selected. It has been shown (Ragazzini & Franklin, 1958) that, based upon the properties of the z -transform, a practical choice of δB is given by $\delta A + 1$ where the extra B parameter is required to include the effect of *fractional dead-time* which will almost certainly be present (Gawthrop, 1980; Hodgson, 1982; Åström *et al*, 1984).

If the time-delay is known exactly then it is possible to directly incorporate it into the CARIMA model of (3.3) using $\hat{k}$. However if it is either unknown, or subject to significant variation with time, then the $B(q^{-1})$ polynomial may be further extended, with the unmodelled time-delay expected to manifest itself as zero, or near-zero, leading coefficients of $B(q^{-1})$. Hence if the time delay is known to vary between limits, $k_{min} < k < k_{max}$, then a time-delay $k = k_{min}$ should be assumed and the order of $B(q^{-1})$ increased from δB to $\delta B + k_{max} - k_{min}$. This is preferable to on-line estimation of the time-delay (Kurz & Goedecke, 1981) which has been shown in simulation studies (Lee & Hang, 1985) to be sensitive to noise and unmeasured disturbances. This approach relies however upon the fact that GPC is not a cancellation control strategy¹ and thus relatively insensitive to early B parameters going to zero.

It is clear that the magnitude of the time-delay parameter, k , depends upon the ratio of the actual time-delay to the sampling interval Δt . This inescapable fact can render the choice of sampling interval more complicated than that suggested by Clarke (1981) where, based upon control considerations, a sample interval 1/10th of the dominant process

¹although GPC may be specified to be minimum variance (see Chapter 2) this is very unlikely in practice.

time constant was proposed. If, however, the actual time-delay is comparable to the dominant time constant then the assumed time-delay could be in excess of 10 sample intervals. To accomodate such extreme cases, where this time-delay is either unknown or time-varying, would require a large and potentially inaccurate assumption for $\hat{k}$ or an unrealistic parameterisation of the $B(q^{-1})$ polynomial. Therefore this problem often leads to a compromise being made between a *fast* sample rate that allows tight closed-loop control, and a *slow* sample rate that produces a reasonable model parameterisation (where k is typically 2 or 3). The latter option of slow sampling has the added benefit of improving the robustness of the adaptive controller (Rohrs *et al*, 1984).

Once the details of the model parameterisation have been settled, then initial values for $\hat{\theta}(0)$ and $P(0)$ can be chosen. Once again the degree of process knowledge available *a priori* determines the nature of these choices. If little is known about the process then

$$\hat{\theta}(0) = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \text{ and } P(0) = \lambda I \quad (3.15)$$

is a common choice, where I is the identity matrix and λ is a large value, implying that the confidence in $\hat{\theta}(0)$ is poor, and ensuring a high initial degree of adaptivity with rapid changes in $\hat{\theta}(t)$. Alternatively, if a reasonable set of parameter values are available then $\hat{\theta}(0)$ should be initialised with these values and λ set to some arbitrary small value causing $\hat{\theta}(t)$ to change more slowly (Young, 1968).

3.3.4 Properties of the algorithm

Consideration of estimation from a Bayesian point of view provides many interpretations for the least squares algorithm. For the moment let us consider only two which are immediately relevant to practical applications of the algorithm. If the estimation error, $\varepsilon(t)$, is Gaussian and uncorrelated with the input measurement $\phi(t-1)$, then the estimate $\hat{\theta}(t)$ is said to be asymptotically 'unbiased', i.e.

$$E\{\hat{\theta}(t) | \mathcal{I}_t\} = \theta \quad (3.16)$$

Furthermore, if $\varepsilon(t)$ is assumed to be *white* with variance σ^2 , then

$$E\{\varepsilon(t)^2 | \mathcal{I}_t\} = \sigma^2 \quad (3.17)$$

and

$$\text{Cov}[\hat{\theta}(t)] = \sigma^2 P(t-1) = E\{[\hat{\theta}(t) - \theta][\hat{\theta}(t) - \theta]^T | \mathcal{I}_t\} \quad (3.18)$$

The latter is a particularly significant result as it shows the P matrix to correspond to the actual covariance matrix of the parameter estimates, scaled by the variance of the noise. Thus in practice the size of the P matrix elements is an indication of the accuracy of the parameter estimates, with the individual diagonal elements a measure of the variance of the associated parameters from their true values. Although it is perhaps a little misleading to refer to the P matrix as 'the covariance matrix' this will be the case throughout this chapter, both for compatibility with earlier workers' notation, and to distinguish it from the $P(q^{-1})$ transfer-function of the GPC control algorithm.

It should be stressed, however, that these interpretations only strictly apply when the previous assumptions about $\varepsilon(t)$ are valid, and this particular form of least squares is used. This will not be the case in practice, but nonetheless the interpretations are still useful with the P matrix providing an essential guide to the development of the parameter estimates.

With reference to the criterion-function of the least squares algorithm given in (3.11) it may be seen that the effect of *all* past data is weighted equally, i.e. there is 'no forgetting'. This means that the effect of more recent data, and thus potentially more relevant, is swamped by the presence of all the old previous data. This intuitive result may be confirmed by consideration of the inverse covariance matrix update of (3.14):

$$P(t-1)^{-1} = P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T$$

As this consists of a rank-1 update of a positive definite² matrix then for persistently exciting data (Anderson, 1982),

$$\lim_{t \rightarrow \infty} P(t-1) = 0 \quad (3.19)$$

and as the 'gain' of the parameter update (3.12) depends upon P thus the adaptivity of the estimator will itself tend to zero.

Although in practice persistently exciting data is not guaranteed this does mean that the adaptivity of this RLS algorithm, without forgetting, declines gradually, if only because of the presence of measurement noise in the data vector, $\phi(t-1)$. This well known property (Harris & Billings, 1981; Wittenmark & Åström, 1984) thus limits its usefulness to self-tuning control where the emphasis is only upon preliminary process identification, rather than adaptive control where long-term adaptation is required.

²by definition.

3.3.5 A simulation example

In order to illustrate the characteristics of the *basic* algorithm outlined above an estimation experiment was conducted where RLS with no forgetting was used to estimate the parameters of a simple simulated model:

Example 3.1 *For the sake of clarity the simulated model is chosen to be first order with unity gain:*

$$(1 - 0.8q^{-1})y(t) = 0.2u(t - 1) + \xi(t) \quad (3.20)$$

but subject to a noise disturbance $\xi(t)$ of variance $\sigma_n^2 = 1$.

Estimation took place in open loop with a PRBS input signal of sequence length 127 samples and magnitude of 10 units used to ‘probe’ the system. The experiment consisted of four main sections:

- An initial self-tuning phase where estimation was enabled and the PRBS signal used to probe the system.
- Ramp-wise variation of the b_0 parameter to 0.4 and then back to its original value of 0.2, again with the PRBS input applied. Both these changes took 50 samples to complete and thus simulate a relatively ‘slow’ process variation in the presence of exciting data.
- A period of 150 samples during which the PRBS input was removed and no process variation took place. This simulates a typical period of routine process regulation where the data is non-exciting.
- Step-wise variation of the b_0 parameter to 0.4 during a further period of PRBS input. This simulates a relatively ‘fast’ process change in the presence of exciting data.

The estimated model was parameterised exactly with one B parameter and one A parameter, the initial values of these parameters and the covariance matrix chosen to be:

$$\hat{\theta}(0) = \begin{bmatrix} 1 & 0 \end{bmatrix}^T \quad \text{and} \quad P(0) = 10I \quad (3.21)$$

The result of the experiment is shown in Fig. 3.3 with the actual parameters being shown dotted, the estimated parameters shown as solid lines, and the input/output signals shown for reference. It is clear that the initial identification of the model is both rapid

and accurate but that neither of the later process changes are tracked, despite the PRBS input, because the estimator has 'fallen asleep'. This is confirmed by the constant decline of the trace³ of the covariance matrix.

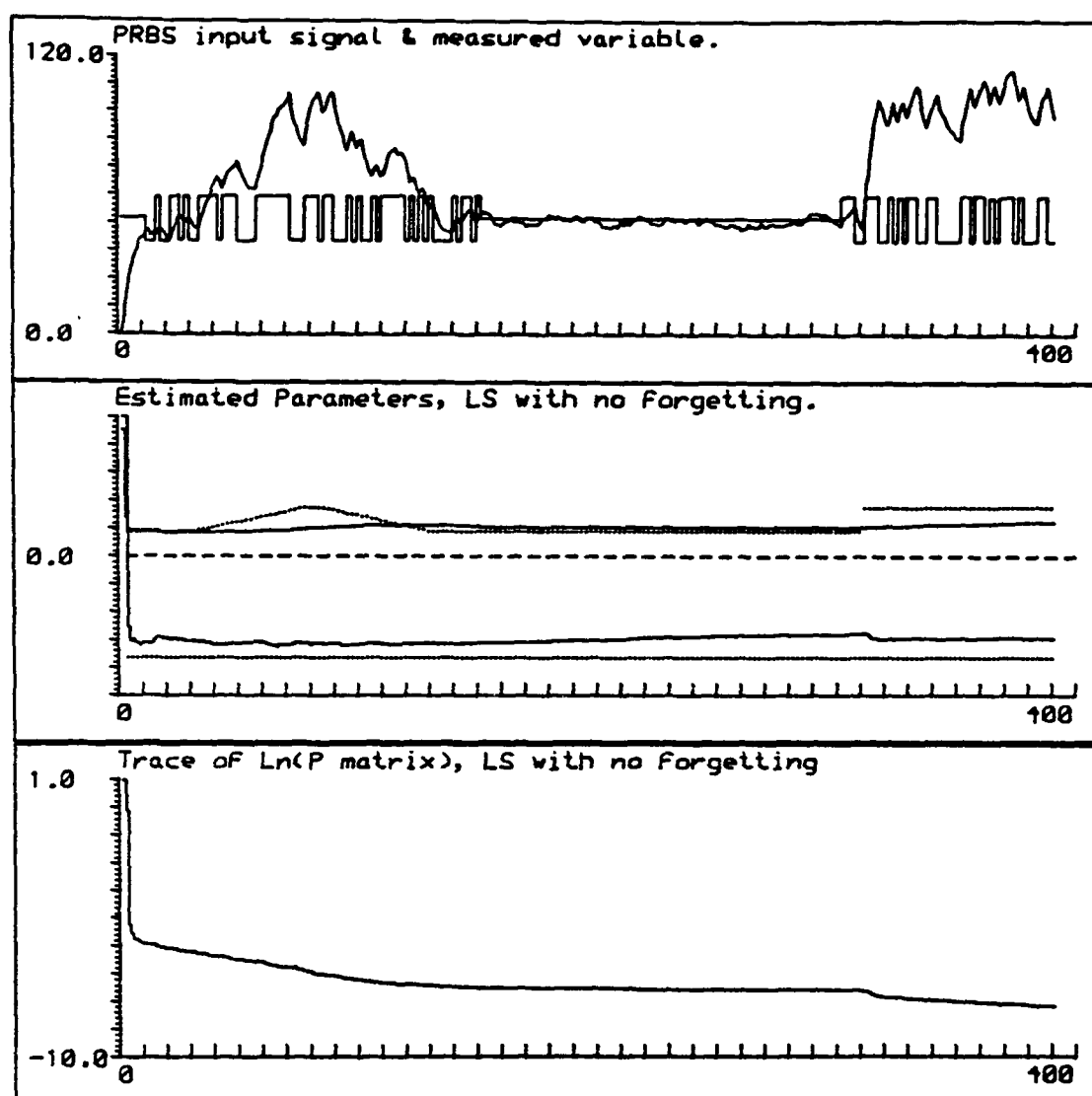


Figure 3.3: Performance of the basic recursive least squares algorithm.

Note that the presence of the noise causes a significant bias on the A parameter, verifying the well-known fact that RLS only provides unbiased parameters when the noise is completely uncorrelated with the data.

3.4 ESTIMATION OF TIME-VARYING PARAMETERS

The basic RLS estimator was shown in the previous section only to be suitable for self-tuning control because of its gradual loss of adaptivity. However the use of such an algorithm, and indeed such a control strategy, contains an implicit assumption about the nature of the process, namely that the parameters $\theta(t)$ are constant but unknown. It is well known, however, that this is not always the case in practice and if, after an initial self-tuning phase the actual process was to change significantly then the estimated model

³the 'trace' of a matrix is the sum of its diagonal elements and is thus a useful measure of the adaptivity of the estimator as it corresponds to the sum of that P matrix eigenvalues.

could become sufficiently invalid that control is degraded. Thus a general-purpose PAA, suitable for adaptive⁴ control, must preserve the validity of the estimated model despite such process changes.

In order to have a reasonable chance of successfully tracking such time-varying process characteristics two assumptions about the actual process parameters must be fulfilled (Hägglund, 1983):

- The parameters $\theta(t)$ vary slowly compared with the states $\phi(t)$.
- The parameters $\theta(t)$ are subject to sudden large changes which are infrequent compared to the time-constants of the system.

If this is not the case then the suitability of the adaptive control strategy is itself questionable.

The former assumption is a commonly encountered industrial phenomenon with examples including long-term fouling of heat exchangers, progressive catalyst degradation, and gradual changes of characteristics during a batch chemical reaction. Tracking such 'slow' parameter changes in closed-loop represents perhaps the most severe problem for an estimator, especially because feedback will act so as to prevent any change having a significant effect upon the measured variable. This is unavoidable and represents a classic paradox, whereby good control can cause poor estimation due to lack of exciting data.

Perhaps the most common approach to this problem of parameter tracking is to modify the recursive least squares algorithm of the previous section such that it always retains a 'limited' amount of adaptivity. Thus, in principle, it should continuously respond to any available data and modify the model accordingly, whilst hopefully not reacting unnecessarily to the presence of noise. This approach is, however, not suitable for the latter type of large or sudden parameter changes because it could result in an unacceptably long period of poor control before the estimated parameters re-tune. What is required under these circumstances is an algorithm that is capable of detecting such large changes and modifying the estimation algorithm such that the parameters are updated more quickly.

Therefore in the following sections selected PAA's are reviewed that have been designed to track time-varying parameters. All of these are based upon the recursive least squares method and represent the best known, or most promising, methods currently available.

⁴see section 3.1 for a the distinction between self-tuning and adaptive control.

3.4.1 Fixed exponential forgetting

The most common modification applied to the least squares algorithm, in an attempt to maintain its adaptivity, is to attach more significance to new data than to old data. This has traditionally been implemented by the inclusion of an exponential weighting factor, widely referred to as a *forgetting factor* (Åström & Wittenmark, 1973; Peterka, 1970), in the criterion function of the basic least squares algorithm (3.11). Thus,

$$J_{ls}(\theta) = \sum_{k=t-N}^t \beta^{t-k} [\bar{y}(k) - \phi(k-1)^T \theta]^2 \quad (3.22)$$

where β is the exponential weighting factor, $0 < \beta \leq 1$, and the initial conditions are not included for simplicity. Note that when $\beta = 1$ all data is weighted equally which corresponds exactly to (3.11), i.e. least squares with no forgetting, but when $\beta < 1$ more weight is placed upon recent measurements than upon older measurements (see Fig. 3.4).

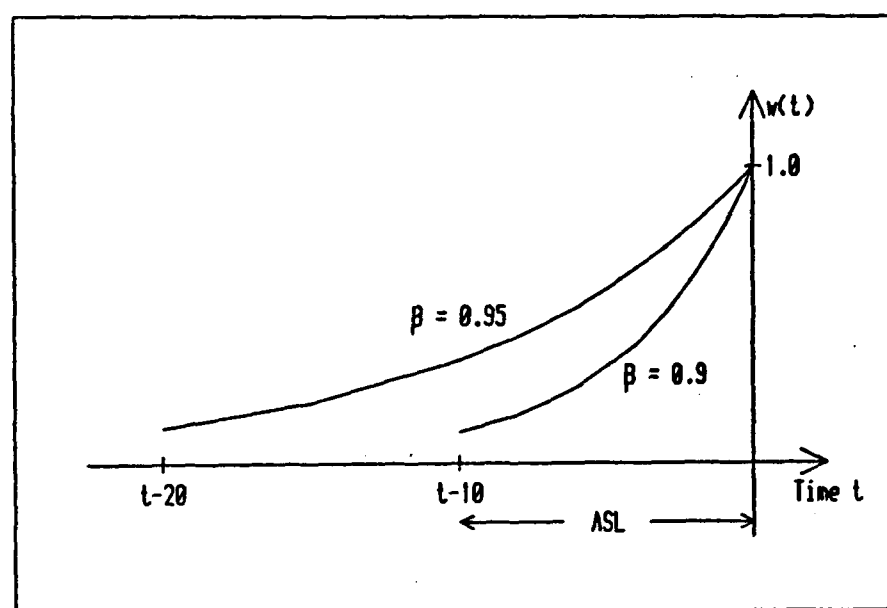


Figure 3.4: Exponential weighting of data

The distance back in time when the data is no longer considered to be important is often referred to as the *data window*, or *asymptotic sample length* (Clarke & Gawthrop, 1975) and may be approximated as $N = 1/(1 - \beta)$. This window length is normally chosen to match the expected time variation of the parameters which should be 'almost constant' over this period (Clarke, 1981; Ljung & Söderström, 1983). For $\beta = 1$ the data window is infinite, with equal weighting for all data, and once again corresponds to the basic algorithm of (3.11).

Following the derivation of Young (1968) for this modified algorithm the update equations of *least squares with exponential forgetting* are given by:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-2)\phi(t-1)}{\beta + \phi(t-1)^T P(t-2)\phi(t-1)} [\bar{y}(t) - \phi(t-1)^T \hat{\theta}(t-1)] \quad (3.23)$$

$$P(t-1) = \frac{1}{\beta} \left[P(t-2) - \frac{P(t-2)\phi(t-1)\phi(t-1)^T P(t-2)}{1 + \phi(t-1)^T P(t-2)\phi(t-1)} \right] \quad (3.24)$$

The interpretation of the P matrix as a guide to the accuracy of the parameter estimates remains, however the inverse P matrix update of (3.24) now becomes,

$$P(t-1)^{-1} = \beta P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T \quad (3.25)$$

which may be rearranged to the more meaningful form,

$$P(t-1)^{-1} = P(t-2)^{-1} - (1-\beta)P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T \quad (3.26)$$

where, at each sample interval, an amount of previous data $(1-\beta)P(t-2)^{-1}$ is discarded (or 'forgotten'), before the new data $\phi(t-1)\phi(t-1)^T$ is added. As $\beta \leq 1$ then this ensures that the elements of $P(t-1)$ are prevented from becoming too small. This may alternatively be considered as artificially reducing the measure of confidence in the current parameter estimates and thereby inflating the gain of the estimator.

Although this algorithm has been widely used for adaptive control, it is now well-known (Hodgson, 1982; Åström, 1983) that it is not a practical modification. When the process is under satisfactory control very little excitation of the process occurs and the elements of the data vector, $\phi(t-1)$, may be close to zero. If this condition continues for a prolonged period then,

$$P(t-1) \approx \frac{1}{\beta} P(t-2) \quad (3.27)$$

and for $\beta < 1$ the elements of the P matrix will grow exponentially large. This is the so-called 'blow-up' which in the extreme can result in excessive sensitivity to any process disturbances, causing large parameter changes and leading, potentially, to momentary instability of the closed loop control. Such 'bursts' in control activity may subsequently result in improved parameter estimation and the restoration of satisfactory control but are obviously very undesirable from a practical point of view.

3.4.2 Variable exponential forgetting

The recognition of this problem led naturally to the development of variable forgetting factors, the intention being to make the amount of exponential data weighting depend

upon both the amount of available data *and* the quality of the current estimated parameters (Åström,1980; Fortescue *et al*,1981; Wellstead & Sanoff,1981; Ydstie,1982). Thus in the absence of exciting data $\beta(t)$ should tend to 1 to prevent blow-up, whereas in the event of a sudden parameter change $\beta(t)$ should automatically be reduced, greatly increasing the significance of new incoming data and thus re-stimulating adaptivity.

Representative of this approach is the variable forgetting factor of Ydstie (1982), a continuation of the work of Fortescue *et al* (1981). In this algorithm the forgetting factor is chosen in order to keep the weighted sum-of-squares of the *a posteriori* estimation errors constant, this being assumed to be representative of the amount of available data and parameter quality. The justification for this choice as an appropriate measure of the *information content* of the estimator may be easily seen, for if the error is small then either,

- The process has not been excited.
- There has been an excitation with a nearly correct set of parameters.
- The estimator is already sufficiently sensitive to reduce the parameter error.

whereas if it is large then,

- The process has been excited with an inaccurate set of parameter values.

The implementation of this algorithm is based upon the least squares criterion of (3.22) but now $\beta(t)$ is time-varying, i.e.

$$S(t) = \sum_{k=t-N}^t \beta^{t-k}(k) \varepsilon(k)^2 \quad (3.28)$$

where the calculation of $\beta(t)$ proceeds via the definition

$$S(t) = S(t-1) = \dots = \Sigma_0 \quad (3.29)$$

and requires the solution, at each sample interval, of the quadratic equation :

$$\beta(t) = \Sigma_0 \left[\Sigma_0 + \frac{\varepsilon(t)^2}{\beta(t) + \phi(t-1)^T P(t-2) \phi(t-1)} \right]^{-1} \quad (3.30)$$

From (3.30) it is clear that if there is little process excitation (i.e. $\varepsilon(t) \approx 0$) for a prolonged period then $\beta \rightarrow 1$ whereas for a large estimation error $\beta \rightarrow 0$, thus corresponding to the specification described above.

The inclusion of the variable forgetting factor does not increase the difficulty of commissioning, with only Σ_0 having to be chosen instead of the previously constant value of β . Although relatively arbitrary Σ_0 should normally be chosen equal to $\sigma_0^2 N_0$, where σ_0^2 is the expected noise variance and N_0 may be thought of as a 'nominal' data window or ASL. The smaller the value of Σ_0 , and thus N_0 , the higher the adaptivity *for a given noise level*.

Herein lies one of the principle objections to the use of this method: that the algorithm contains an implicit assumption that the noise variance σ_0^2 remains constant (Lammers, 1984). Thus, in practice, an increase in noise level or a sudden load disturbance, would, for a fixed value of Σ_0 , lead to an increase in adaptivity as the algorithm attempts to keep $S(t)$ constant by reducing the length of the data window. This action implies that the more recent data is more valid whereas, in the absence of a real process change, the old data is probably more valid as it was gathered in a less noisy environment.

Nonetheless a slightly simplified earlier version of this algorithm, based upon the work of Fortescue *et al* (1981), was successfully used throughout the industrial trials described in Chapter 5 and Chapter 6 of this thesis. Satisfactory performance was obtained through careful specification of Σ_0 and normalisation of the data respectively.

3.4.3 Directional forgetting

At the basis of all the forgetting factor based methods of retaining adaptivity discussed so far has been the concept of discarding old data, which is thus assumed to be *obsolete*. However consideration of the inverse covariance matrix update (3.26) corresponding to the least squares algorithm with a fixed forgetting factor $\beta = 0.9$, reveals an inherent fallacy of this approach.

$$P(t-1)^{-1} = P(t-2)^{-1} - 0.1P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T \quad (3.31)$$

The inverse covariance matrix represents a measure⁵ of the accuracy of the parameter estimates. Yet in the above update, despite the fact that there is new data available only in the 'direction' of $\phi(t-1)$, 10% of all the information obtained from previous data is discarded in *all* directions. Even if there is no information in the new data to suggest that all this old data is obsolete it is nonetheless discarded.

⁵ often referred to as Fisher's information matrix

A more cautious approach would be to consider old data obsolete only in the direction in which new data is available, the so-called 'directional' forgetting factor of Kulhavý & Karny (1984). This gives an inverse covariance matrix update for RLS with directional forgetting as,

$$P(t-1)^{-1} = P(t-2)^{-1} + w(t)\phi(t-1)\phi(t-1)^T \quad (3.32)$$

where the weighting factor $w(t)$ determines the amount of data that is to be discarded [$w(t)$ is negative] or included [$w(t)$ is positive] at each sample interval.

One particular version of this algorithm, where $w(t)$ is calculated as a weighting factor corresponding to the amount of covariance matrix adjustment that would have taken place in the direction of $\phi(t-1)$ if a constant forgetting factor β had been applied in all directions, is known as 'restricted exponential forgetting'. This leads to a simple calculation of $w(t)$, at each sample interval, given by:

$$w(t) = \beta(t) - \frac{1 - \beta(t)}{\phi(t-1)^T P(t-2) \phi(t-1)} \quad (3.33)$$

The introduction of directional forgetting does not by itself completely solve the problem of blow-up. The *balancing act* of forgetting remains because if the amount of information discarded in any one direction consistently exceeds that incoming then the P matrix will still be inflated in that direction.

3.4.4 Alternative techniques

Another important, and popular, approach to the problem of retaining adaptivity in the least squares algorithm is by *direct modification* of the P matrix elements. For example Goodwin & Sin (1984) suggest the addition, at each sample interval, of a constant scaled identity matrix $k_0 I$ with an upper bound applied to P to prevent *blow-up*:

$$P(t-1)^{-1} = P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T + k_0 I \quad (3.34)$$

This matrix, $k_0 I$, may be interpreted as the assumed covariance of random increments in the parameters (i.e. a random walk parameter model) from Kalman filtering. A similar approach involves the weighted sum of P and I , the weights chosen such that the trace of the P matrix is kept constant (Goodwin *et al*, 1985). Egardt (1979) also keeps the trace of the P matrix constant by choice of an appropriate forgetting factor. Most of these types of

algorithms found their conception in the need for parameter adaptation algorithms suitable for analysis but nonetheless do represent important alternatives.

Finally the use of *persistent excitation* to assist estimation is a common approach and has been the subject of much recent work (Vogel,1982). Unfortunately the injection of perturbation signals during routine closed loop operation would probably be unacceptable to process management. For this reason perturbation signals, apart from being used during initial identification experiments (3.2.3), are not considered.

3.4.5 Some illustrative simulation studies

The performance of RLS with fixed exponential forgetting is shown in Fig. 3.5 for the estimation trial described previously in section (3.3.5). A fixed forgetting factor of $\beta = 0.9$ was used, corresponding to a data window of 10 samples, i.e. an expected parameter variation rate twice as slow as the process time-constant. This is not suggested as a realistic value but was chosen to provide an accelerated test.

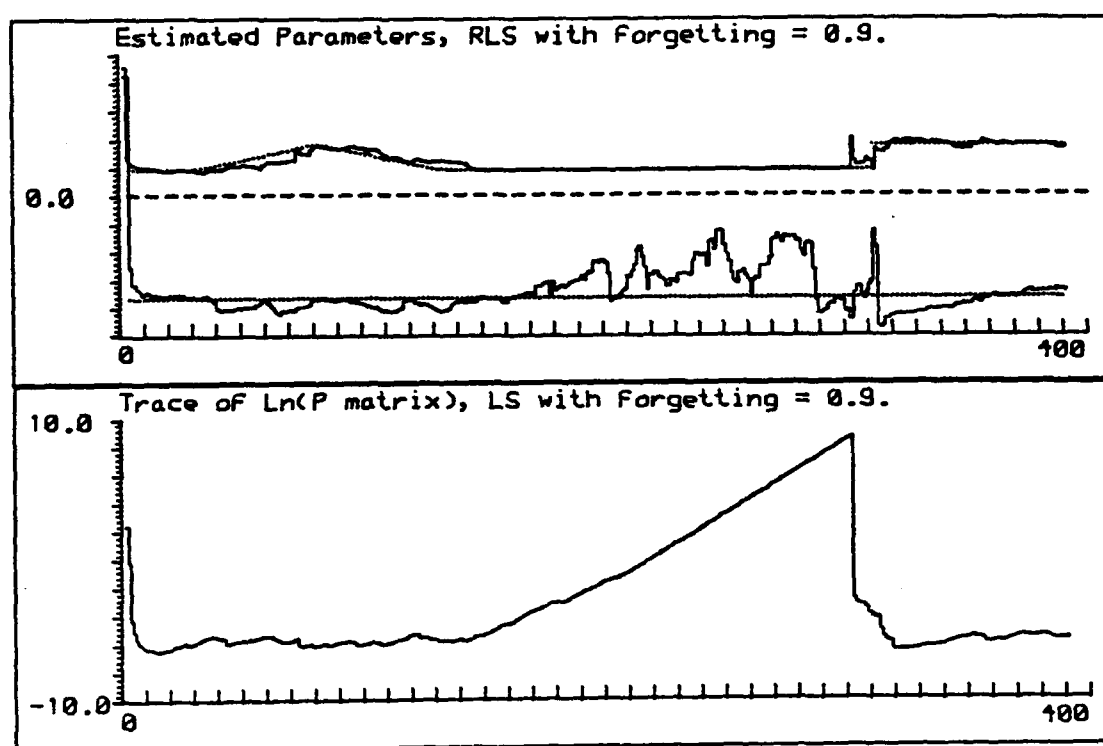


Figure 3.5: Performance of RLS with fixed forgetting.

In the presence of exciting data parameter tracking was both rapid and accurate but when PRBS was not applied, the covariance matrix clearly 'blows-up', the exponential growth of its trace appearing linear because of the logarithmic nature of the plot. This causes excessive adaptivity and extreme sensitivity of the a_1 parameter to noise. The moment that PRBS is reapplied estimation immediately recovers.

The performance of the recursive least squares algorithm with Ydstie's variable forget-

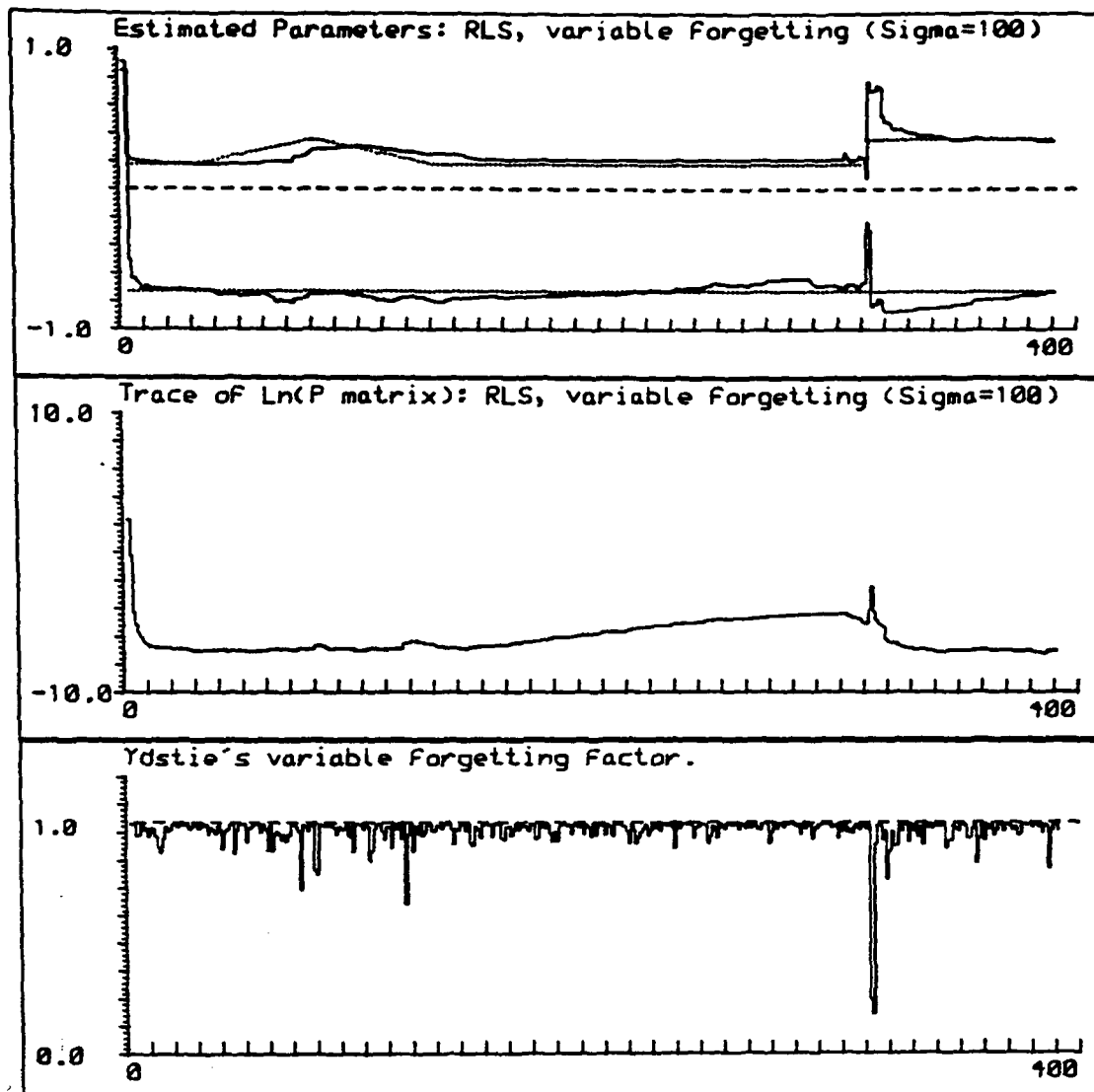


Figure 3.6: Performance of RLS with variable forgetting.

ting factor is illustrated in Fig. 3.6. As the noise variance was known to be unity a value of $\Sigma_0 = 100$ was chosen, corresponding to a 'nominal' data window of 100 samples in order to obtain low-gain adaptation during routine operation. Initial tuning of the parameters is again almost immediate due to the high initial value of the covariance matrix, $P(0) = 10I$. Tracking of the subsequent ramp-wise process changes is adequate but significantly slower than that obtained using fixed forgetting. With no process excitation the forgetting factor remains close to unity (i.e. no forgetting) from $t = 150$ to $t = 300$ and thus successfully avoids blowup and bursting. Tracking of the sudden process change at time $T = 310$ is very active due to adaptivity being restimulated by a temporary drop in the value of the forgetting factor causing a corresponding temporary increase in the elements of the covariance matrix. Note that the excessive activity of the parameter estimates during the latter process change could be simply overcome by the specification of a minimum value for the forgetting factor, i.e. β_{min} .

Fig. 3.7 records the results obtained using the 'restricted exponential forgetting' of Kulhavý & Karny (1984) with, for comparison, a fixed forgetting factor of $\beta = 0.9$. This algorithm provides good initial identification of the process and adequate tracking of both

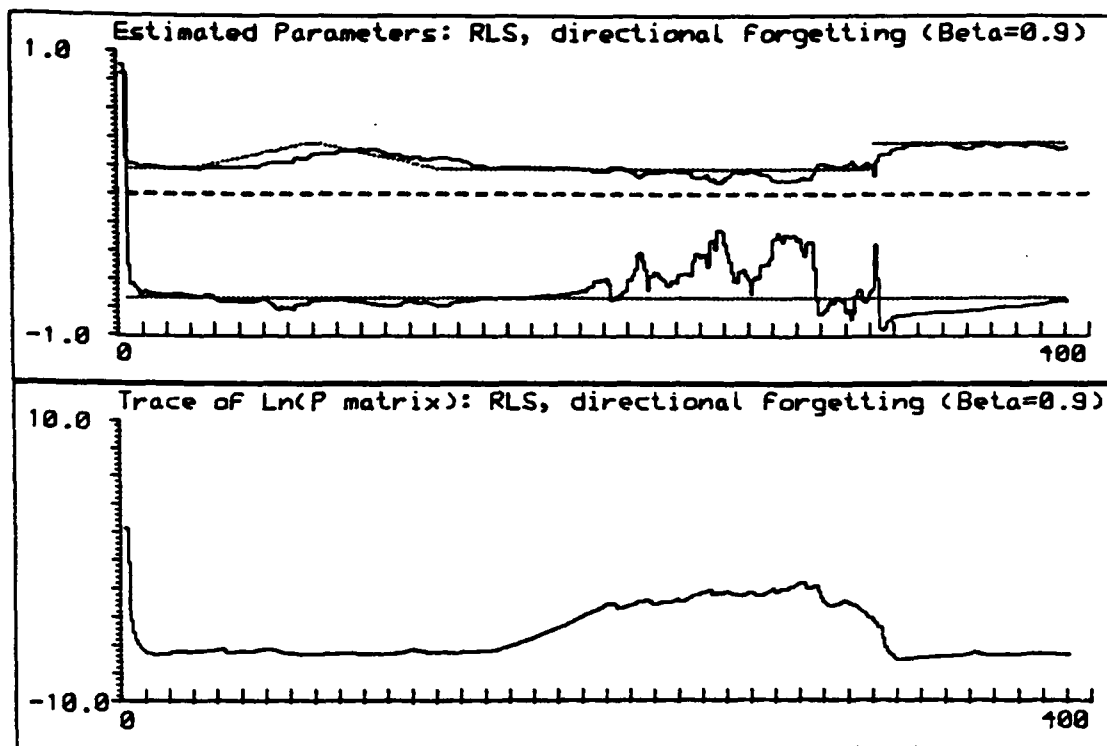


Figure 3.7: Performance of RLS with directional forgetting.

'slow' and 'fast' changes in the b_0 parameter. Compared to the ordinary fixed forgetting factor (Fig. 3.5) its performance is less active but without persistency of excitation ($t = 150$ to $t = 300$) it is superior in that the covariance matrix only partially blows-up. This is seen by the rapid initial growth of the trace of the covariance matrix which then subsequently levels off. The point at which this occurs corresponds to the amount of data being forgotten approximately balancing the amount of incoming data (i.e. the noise). Thus with reference to (3.32) and (3.33),

$$w(t) \approx 0 \quad \text{and} \quad \phi(t-1)^T P(t-2) \phi(t-1) \approx \frac{1-\beta}{\beta} \approx 0.11 \quad (3.35)$$

However even this partial blow-up causes such an increase in adaptivity that estimation of the a_1 parameter is poor during this period because of increased sensitivity to noise. Choice of a larger value for the forgetting factor β would further restrict the amount of blow-up but, as for the normal forgetting factor, this would be at the expense of slower parameter tracking.

The next experiment involved the application of the 'constant trace' algorithm of Goodwin *et al* (1985), as an estimator representative of those that retain adaptivity by direct modification of the covariance matrix elements rather than by using a forgetting factor. In this particular algorithm the covariance matrix update consists of the weighted sum of $P(t-2)$ and I such that the trace is kept to a pre-specified constant value, i.e.

$$P'(t-1) = P(t-1) + \left(\frac{C_0 - \tau}{n} \right) I \quad (3.36)$$

where C_0 is the desired trace, τ is the trace of $P(t-1)$ following a normal covariance of RLS with no forgetting (3.13), and n is the number of parameters. By fixing the trace of the covariance matrix an upper bound is placed upon the maximum eigenvalue (i.e. $\lambda_{\max} \leq \text{trace}(P)$) which prevents blow-up whilst guaranteeing the retention of a certain amount of adaptivity. This modification may be considered as leading to an estimation algorithm whose *search direction* lies somewhere between that of a 'gradient' or 'steepest-descent' algorithm and unmodified recursive least squares.

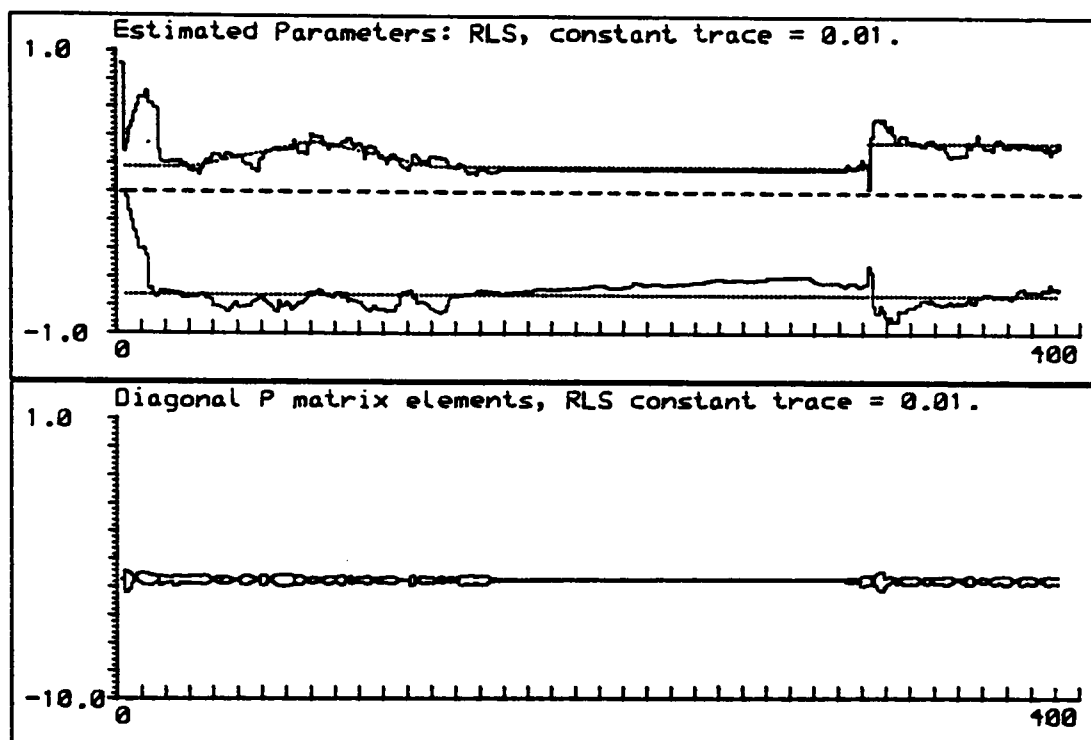


Figure 3.8: Performance of RLS with constant trace.

In the trial shown in Fig. 3.8 the value of the constant trace was specified as $C_0 = 0.01$ and the covariance matrix initialised with $P(0) = 0.005I$. Initial tuning was slow because of the relatively small value for $P(0)$, but both 'slow' and 'fast' process changes are tracked satisfactorily and there is no blow-up in the absence of exciting data. Instead of the trace of the covariance matrix its individual elements are plotted but nonetheless the constant trace is apparent. Whilst these results show the potential of an algorithm of this type the choice of C_0 would in practice be unclear. Although not shown the difficulty of selecting an appropriate value for C_0 was observed in this experiment, a choice of $C_0 = 0.1$ producing excessive parameter variation whilst $C_0 = 1.0 \times 10^{-3}$ tracked process changes far too slowly.

3.5 RLS WITH PARAMETER WEIGHTING

The criterion function that the recursive least squares algorithm with fixed exponential forgetting minimises was given in 3.22 as :

$$J_{ls}(\hat{\theta}) = \sum_{k=t-N}^t \beta^{t-k} [\bar{y}(k) - \phi(k-1)^T \hat{\theta}]^2$$

It is clear from the above that if $\phi(t-1) = 0$ for a prolonged period of time (i.e. during 'good' process regulation at steady-state where the output is near the set-point and control is a constant signal) then,

$$J_{ls}(\hat{\theta}) \approx \sum_{k=t-N}^t \beta^{t-k} \bar{y}(k)^2 \quad (3.37)$$

which is no longer a function of the parameter estimates $\hat{\theta}(t)$. Thus the parameter set which results from the minimisation of this function can lie anywhere in 'parameter space'. This fact is reflected in an exponential increase of the covariance matrix under such conditions (Section 3.4.1, Fig.3.5), corresponding to increasing parameter uncertainty, and the unnecessarily increasing adaptive gains referred to as 'blowup'.

Least squares with parameter weighting is, however, based upon an extended criterion function of the form,

$$J_{ls}(\hat{\theta}) = \sum_{k=t-N}^t \beta^{t-k} [\bar{y}(k) - \phi(k-1)^T \hat{\theta}]^2 + f[\hat{\theta}(t)] \quad (3.38)$$

and thus remains a function of $\hat{\theta}(t)$ even during long periods of no data. This restricts parameter uncertainty in the absence of data and is thus expected to lead to an estimation algorithm which *inherently* avoids blowup.

An additional benefit of this approach is that $f[\hat{\theta}(t)]$ provides a natural mechanism for the long-term incorporation of detailed *a priori* process knowledge, along with a measure of its validity. In section 3.2.5 it was proposed that such information should be used during the commissioning of the ordinary least squares algorithm as initial values for the estimated parameters. However the use of exponential forgetting, in order to retain adaptivity, means that after a few hundred samples the contribution of this information will have been completely discarded, *whether or not it is still valid*.

If, however, this initial model is expected to retain some degree of validity then its long-term incorporation into an estimator may be of benefit in two areas:

- *High performance applications*: where a nominal plant model is available (either from analysis of the system, or from a preliminary identification experiment) throughout the duration of control and all that is required from the PAA is to *trim* these parameters using the incoming data. This requirement occurs naturally in most applications

where estimation is based upon a locally linearised model of a underlying non-linear process.

- *Long-term estimator integrity:* The incorporation of a weighting upon parameter variation can be considered as ‘constraining’ the estimated parameters in some fashion. This may be used to reduce the sensitivity of the estimator to unmeasurable load disturbances, noise etc., and to reduce parameter drift.

In the following sections two algorithms are developed based upon this concept of parameter weighting. These will be referred to as ‘constrained’ and ‘damped’ least squares respectively.

3.5.1 ‘Constrained’ least squares

The ‘constrained’ least squares estimator (CLS) uses a *nominal* model to directly modify the standard least squares criterion function, thus:

$$J_c(\hat{\theta}) = \sum_{k=t-N}^t \beta^{t-k} [\bar{y}(k) - \phi(k-1)^T \hat{\theta}]^2 + \lambda_c(t) \|\hat{\theta}(t) - \theta_0\|^2 \quad (3.39)$$

where θ_0 denotes a nominal parameter vector; $\lambda_c(t)$ is a weighting that determines the significance of this nominal model; and the forgetting factor, $0 < \beta \leq 1$, is defined exactly as in (3.4.1). Minimisation of this criterion function leads to the recursive update equations⁶ given below:

$$\varepsilon(t) = \bar{y}(t) - \phi(t-1)^T \hat{\theta}(t-1) \quad (3.40)$$

$$\begin{aligned} \hat{\theta}(t) = & \hat{\theta}(t-1) + P(t-1)\phi(t-1)\varepsilon(t) \\ & + P(t-1)[\lambda(t) - \beta\lambda(t-1)][\theta_0 - \hat{\theta}(t-1)] \end{aligned} \quad (3.41)$$

$$P(t-1)^{-1} = \beta P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T + [\lambda(t) - \beta\lambda(t-1)]I \quad (3.42)$$

where $P(0) = \lambda_c^{-1}I$. From these equations several important properties of this algorithm may be deduced:

Property C.1 *When $\lambda_c(t) = \lambda_c(t-1) = 0$ these equations correspond exactly to those of standard recursive least squares with exponential forgetting.*

⁶The derivation of these equations and the procedure for the direct update of $P(t-1)$ from $P(t-2)$ may be found in Appendix B.

Furthermore with no-forgetting, i.e. $\beta = 1$, the P matrix may once again be interpreted as proportional to the covariance matrix of the parameter estimates, but with $P(0) \propto 1/\lambda_0$.

Property C.2 *Under conditions of no incoming data the covariance matrix update tends to a pre-specified upper bound and thus avoids the blow-up associated with normal exponentially weighted least squares,*

This may be seen by considering the covariance update equation of (3.42) for $\phi(t-1) = 0$ and constant λ_c :

$$P(t-1)^{-1} = \beta P(t-2)^{-1} + (1-\beta)\lambda_c I \quad (3.43)$$

If $\beta < 1$ then:

$$P(t) \rightarrow \frac{1}{\lambda_c} I \quad (3.44)$$

and thus the covariance matrix P reverts to its original value $P(0)$, an upper bound that prevents blowup.

Property C.3 *In the absence of exciting data the estimated parameters revert to their nominal values at a rate determined by the value of the forgetting factor, β .*

Again this follows from the parameter update (3.41) which becomes,

$$\hat{\theta}(t) \approx \beta \hat{\theta}(t-1) + (1-\beta)\theta_0 \quad (3.45)$$

This therefore represents a ‘cautious’ adaptation strategy that, in the absence of informative data retreats the parameters to their former nominal values. This is similar to the effect of the *ad-hoc* σ -modification of Ioannou & Kokotovic (1984).

During the development special care was taken to write λ_c as a function of time. This enables $\lambda_c(t)$ to be varied on-line, in order that the amount of parameter constraint is available as an estimator ‘tuning-knob’. Thus at any time $\lambda_c(t)$ may be changed, a larger value constraining the parameters to be closer to θ_0 , a smaller value allowing wider, or more active, parameter variation.

Property C.4 *If $\lambda_c(t)$ is set to infinity then the estimated parameters are set immediately to their nominal values.*

This may be seen because for $\lambda_c(t) = \infty$ the covariance matrix becomes

$$P(t-1) \stackrel{\lambda_c(t) \rightarrow \infty}{=} 0 \quad (3.46)$$

which gives a parameter update:

$$\begin{aligned} \hat{\theta}(t) = \hat{\theta}(t-1) &+ \lambda_c(t)^{-1} \phi(t-1) \varepsilon(t) \\ &+ \frac{1}{\lambda_c(t)} [\lambda_c(t) - \beta \lambda_c(t-1)] [\theta_0 - \hat{\theta}(t-1)] \end{aligned} \quad (3.47)$$

As $\lambda_c(t) \gg \lambda_c(t-1)$ then

$$\hat{\theta}(t) \stackrel{\lambda_c(t) \rightarrow \infty}{=} \hat{\theta}(t-1) + \lambda_c(t)^{-1} \lambda_c(t) [\theta_0 - \hat{\theta}(t-1)] = \theta_0 \quad (3.48)$$

Thus in an emergency the parameters may be rapidly forced back to their nominal values at any time by the on-line selection of a sufficiently large value of $\lambda_c(t)$.

Property C.5 *The positive definite nature of the P matrix is not affected by the potential subtraction even if $\lambda_c(t) < \lambda_c(t-1)$.*

This follows immediately from (3.42) as even in the worst case where there is no data,

$$P(t-1)^{-1} = \beta P(t-2)^{-1} - \beta \lambda_c(t-1) I \quad (3.49)$$

Property C.6 *Commissioning of the 'constrained' least squares algorithm is no more complicated than that of ordinary least squares with fixed exponential forgetting.*

The derivation of the above equations imply that the initial covariance matrix should be chosen as $P(0) = \lambda_c^{-1} I$. This is intuitively obvious because the motivation of this approach of constraining the parameter variation, for given initial parameters, is only justified if confidence in those initial parameters is high. Thus a large value of λ_c would not only restrict long term parameter variation, but also cause slow initial adaptivity.

From the point of view of commissioning $\lambda_c I$ may thus be considered as representing the inverse of the initial P matrix and thus with this scheme no more estimator parameters need be chosen at commissioning than those required for ordinary least squares.

Property C.7 *During normal operation the 'constrained' least squares algorithm may be considered as following a search direction that is a compromise between steepest descent and that obtained by normal least squares.*

At initialisation the covariance matrix should be proportional to the identity matrix, i.e. $P(0) = \lambda_c^{-1} I$, and thus at this point the algorithm is said to be a 'gradient' algorithm. If

however $\lambda_c(t)$ is set to zero then the covariance matrix update corresponds to that of RLS with exponential forgetting, where as more information is collected the P matrix becomes close to the Hessian (2nd derivative) of $J_{ls}(\hat{\theta})$ thus producing search directions with a faster rate of convergence than the gradient method (Luenberger, 1973). Constrained least squares, when $\lambda_c \neq 0$, may thus be seen to produce a search direction lying somewhere between the caution of the gradient approach and the rapid adaptation of least squares.

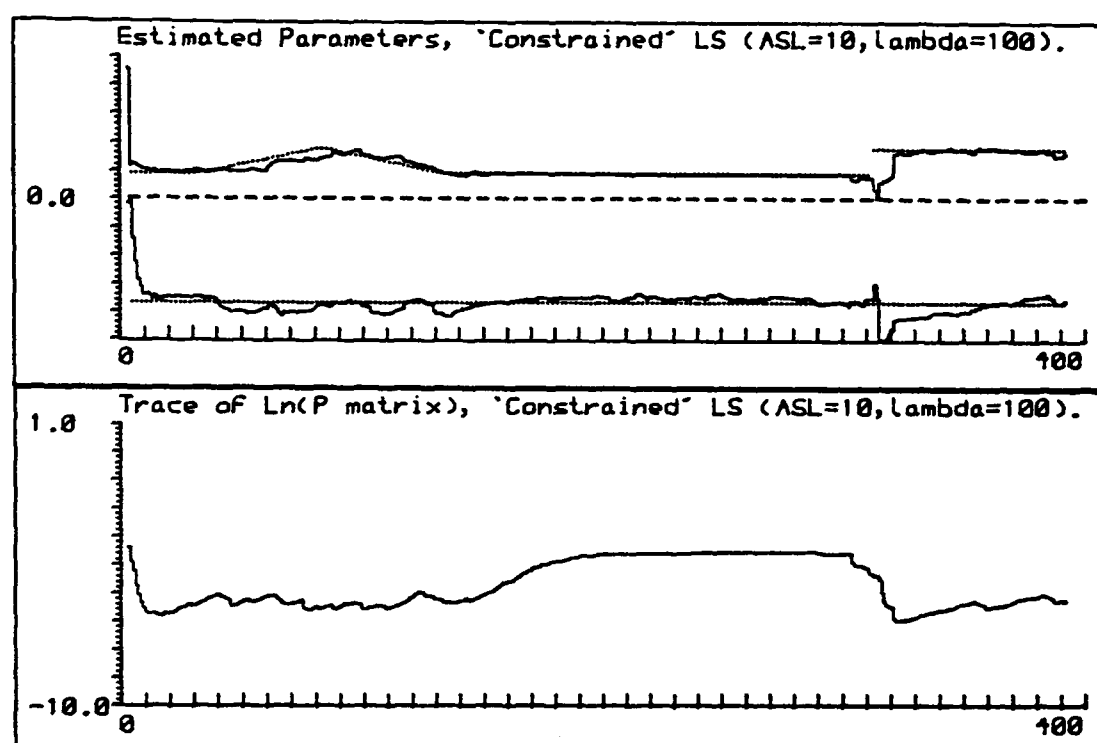


Figure 3.9: Performance of 'constrained' least squares

Fig.3.9 shows the performance of the 'constrained' least squares algorithm described above for the experiment described in section 3.3.5 and used as a benchmark in section 3.4.5. The forgetting factor was chosen as $\beta = 0.9$ and the parameter weighting as $\lambda_c = 100$, the latter thus placing an upper bound upon the trace of the covariance matrix of $\tau_{max} = 0.02$. In order to enable comparison with earlier trials parameter weighting was only enabled at time $t = 20$ after an initial self-tuning phase⁷, the estimated parameters at that point becoming the 'nominal' values.

The performance of the estimator during process excitation is almost indistinguishable from that obtained using RLS with a fixed forgetting factor of $\beta = 0.9$ (see Fig.3.5) with accurate initial self-tuning and excellent parameter tracking despite the constraint. However during the period of no process excitation the advantages of this algorithm are clear with the trace of the covariance matrix levelling off as expected to $\tau = 0.02$, with neither of the estimated parameters suffering from drift due to the noise.

⁷the estimator was thus initialised with $\hat{\theta}(0) = [10]$ and $P(0) = 10I$

Overall the main justification for the use of this type of algorithm occurs when a PAA is required to continuously track the parameters of a process that undergoes temporary changes whilst retaining a long-term basic time-invariance. This could occur due to a temporary change of operating point in a practically time-invariant but significantly non-linear plant. It is a more practical algorithm than ordinary least squares in so far as it inherently imposes an upper bound upon the covariance matrix, thus preventing ‘blow-up’, and provides a very simple mechanism for the controlled incorporation of *a priori* knowledge. The ‘tethering’ of the parameters is also expected to improve the long-term integrity of the algorithm and to reduce parameter drift.

However for processes that undergo slow but permanent changes, such as gradual irreversible fouling of a heat exchanger, then the nominal process parameters, θ_0 , will eventually become invalid and should no longer be used. It will be demonstrated later in this chapter (3.5.4) how the constrained least squares algorithm can be extended to overcome this problem by the periodic updating of θ_0 using fault detection techniques.

3.5.2 ‘Damped’ least squares

Instead of weighting the difference between the estimated parameters and a nominal parameter set as in constrained least squares, the ‘damped’ least squares estimator (DLS) weights *increments* in the estimated parameter vector, i.e.

$$J_d(\hat{\theta}) = \sum_{k=t-N}^t \beta^{t-k} [\bar{y}(k) - \phi(k-1)^T \hat{\theta}]^2 + \lambda_d(t) \|\hat{\theta}(t) - \hat{\theta}(t-1)\|^2 \quad (3.50)$$

Thus DLS may be considered as an approximation to the constrained algorithm where θ_0 , the nominal parameters, are either unavailable or subject to permanent changes.

The effect of the above modification to the parameter update equations will, predictably, be similar to that of CLS. However before proceeding into detailed interpretations it is worthwhile to note the remarkable similarity between (3.50) and the cost-function that forms the basis of the well known GMV control algorithm of Clarke & Gawthrop (1979), the latter being shown below in a simplified form:

$$J_{GMV} = [y(t) - w(t)]^2 + \lambda(t) \|u(t) - u(t-1)\|^2 \quad (3.51)$$

where $y(t)$ is the measured variable, $w(t)$ is the set-point, and $u(t)$ is the control signal. It is well known that variation of $\lambda(t)$ in GMV determines the amount of ‘de-tuning’ of the control activity and thus intuitively the penalisation of parameter increments in DLS

would be expected to reduce the adaptivity of the algorithm compared to ordinary RLS with fixed forgetting.

The recursive parameter update equations⁸ of the damped least squares algorithm are given below:

$$\varepsilon(t) = \bar{y}(t) - \phi(t-1)^T \hat{\theta}(t-1) \quad (3.52)$$

$$\begin{aligned} \hat{\theta}(t) = & \hat{\theta}(t-1) + P(t-1)\phi(t-1)\varepsilon(t) \\ & + P(t-1)\beta\lambda_d(t-1) [\hat{\theta}(t-1) - \hat{\theta}(t-2)] \end{aligned} \quad (3.53)$$

$$P(t-1)^{-1} = \beta P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T + [\lambda_d(t) - \beta\lambda_d(t-1)]I \quad (3.54)$$

where the definition of $\varepsilon(t)$ and the covariance matrix update may be seen to be exactly the same as in the constrained algorithm (3.41,3.42), and $P(0) = \lambda_d^{-1}I$.

Some of the most important properties of ‘damped’ least squares are listed below with the emphasis placed on those properties that differ from those already presented for the constrained algorithm:

Property D.1 *When $\lambda_d(t) = \lambda_d(t-1) = 0$ the above equations correspond exactly to those of standard recursive least squares with exponential forgetting.*

This is once again obvious from the criterion function of (3.50).

Property D.2 *In the absence of exciting data the covariance matrix update tends to a pre-specified upper bound and thus avoids the blow-up associated with normal exponentially weighted least squares,*

Because the covariance matrix update is exactly the same for the ‘damped’ and ‘constrained’ least squares algorithms this follows immediately, with the upper bound being given by:

$$P(t) \rightarrow \frac{1}{\lambda_d}I \quad (3.55)$$

Property D.3 *In the absence of exciting data the current parameter increment decays exponentially at a rate determined by the value of the forgetting factor, β .*

⁸once again details of this derivation are given in Appendix B.

This may be shown because as $P(t) \rightarrow \frac{1}{\lambda_d}I$, and $\phi(t-1) = 0$, the parameter update (3.53) tends to,

$$\hat{\theta}(t) \approx \hat{\theta}(t-1) + \beta[\hat{\theta}(t-1) - \hat{\theta}(t-2)] \quad (3.56)$$

and thus

$$\Delta\hat{\theta}(t) \approx \beta\hat{\theta}(t-1) \quad \text{where} \quad \beta < 1 \quad (3.57)$$

Thus this algorithm may be thought of as *low-pass filtering* the estimated parameter increment, a result that is similar to the work reported in Vogel (1982), and expected to reduce undesirably large variations in the estimated parameters due, for example, to the presence of noise.

Property D.4 *If $\lambda_c(t)$ is set to infinity then the estimated parameters are immediately 'frozen' to their current values.*

By comparison with the role of $\lambda(t)$ in the GMV control algorithm, the setting of $\lambda_d(t)$ to infinity effectively prevents parameter increments, thus switching estimation off. This may easily be verified as once again:

$$\lim_{\lambda_d \rightarrow \infty} P(t-1)^{-1} = \lambda_d(t) \quad (3.58)$$

which gives a parameter update,

$$\begin{aligned} \hat{\theta}(t) = \hat{\theta}(t-1) &+ \frac{1}{\lambda_d(t)} \phi(t-1) \varepsilon(t) \\ &+ \frac{\beta}{\lambda_d(t)} \lambda_d(t-1) [\hat{\theta}(t-1) - \hat{\theta}(t-2)] \end{aligned} \quad (3.59)$$

and therefore,

$$\lim_{\lambda_d(t) \rightarrow \infty} \hat{\theta}(t) = \hat{\theta}(t-1) \quad (3.60)$$

Thus in an emergency parameter adaptation may be suspended at any time by the on-line selection of a sufficiently large value of $\lambda_d(t)$. This result is of real practical importance during periods of extraordinary process operation that are known in advance (e.g. routine maintenance) where selection of a finite, but large, value for λ_d could be used to protect the estimated parameters.

Property D.5 *Although strictly the algorithm should be initialised with $P(0) = \lambda_d^{-1}I$, it is possible to include an arbitrary large initial value, P_0 , in order to achieve initial rapid adaptation.*

i.e.

$$P(0) = \left(P_0 + \frac{1}{\lambda_d} \right) I \quad (3.61)$$

In principle this allows an initial period of rapid self-tuning followed by normal operation of the algorithm with a search direction somewhere between the gradient algorithm and least squares. The value of $\lambda_d(t)$ could then be varied on-line at any time to determine the adaptivity of the estimator.

Property D.6 *Commissioning of the 'damped' least squares algorithm is no more complicated than that of ordinary least squares with fixed exponential forgetting.*

Commissioning of DLS only requires the choice of the fixed forgetting factor, β , and the parameter weighting, λ_d , with the latter chosen to represent an upper bound upon the covariance matrix. It is only when there is little prior knowledge available about the process, and thus an initial period of self-tuning is required, that an extra P_0 parameter may have to be chosen to ensure rapid initial adaptation.

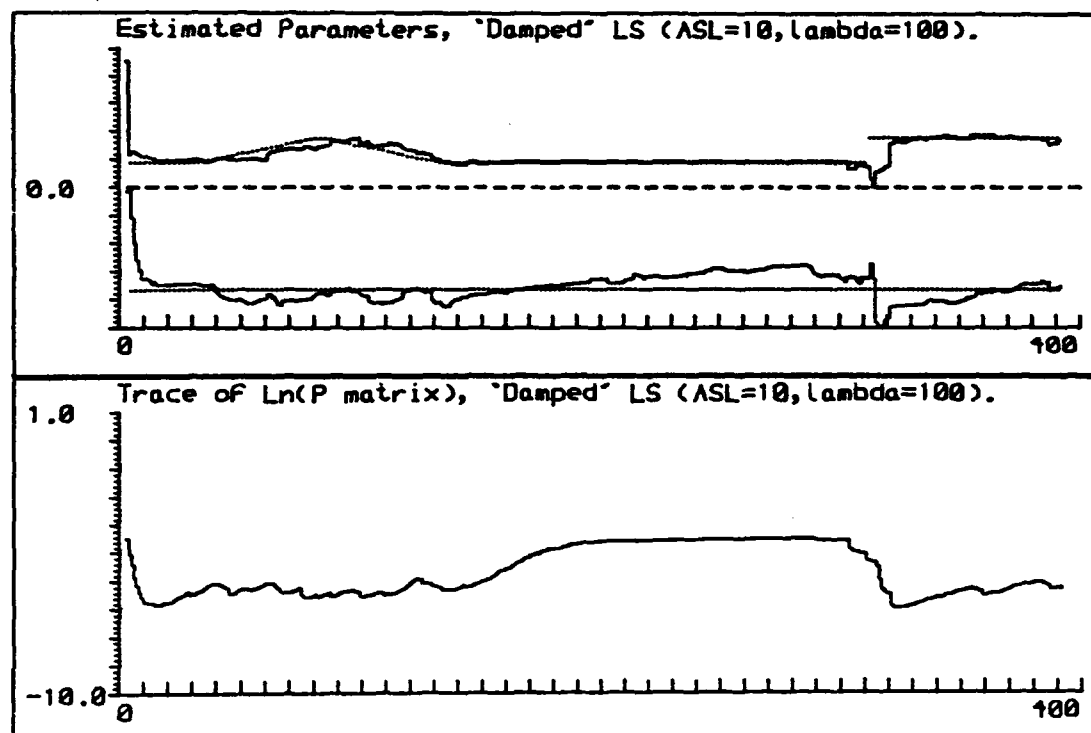


Figure 3.10: Performance of 'damped' least squares

Throughout this chapter all the estimation algorithms described have been tested on the benchmark experiment of section 3.3.5. Thus Fig. 3.10 shows the estimation performance of the 'damped' least squares algorithm for $\beta = 0.9$, $\lambda_d = 100$ and $P(0) = 10I$ for this example. Initial adaptation is rapid and accurate, with subsequent tracking of both slow and fast process changes also being satisfactory. As in the previous trial of CLS (Fig. 3.9)

the trace of the covariance matrix returns to a value of $\tau = 0.02$ during the period of non-exciting data which prevents blow-up and restricts unnecessary parameter drift.

Hence 'damped' least squares has been shown to retain the fundamental properties of the 'constrained' algorithm, namely rapid and accurate parameter tracking without the risk of covariance blowup, whilst also being immediately suitable for the tracking of permanent process changes.

3.5.3 Individual parameter weighting

In the previous section a scalar, λ , was used to weight parameter variation in an extended RLS criterion function, with the value of λ determining *overall* estimator adaptivity especially in the absence of exciting data. This may be considered as a 'tethering' or a 'filtering' of the estimated parameters for the constrained (3.5.1) and damped (3.5.2) algorithms respectively.

However in practice there may be prior information available about *individual* parameter variation. For example it may be known that the dynamics of a particular process are relatively time-invariant but that the gain varies significantly. In this case to assist estimator integrity it would be advisable, once the parameters had converged and satisfactory control achieved, to incorporate this *a priori* knowledge into the estimated model structure by restricting the variation of $\hat{a}$ parameters more so than $\hat{b}$ parameters.

Using parameter weighting such an estimation policy is implemented simply by specification of a diagonal matrix, Λ , of separate weights corresponding to individual parameters, i.e.

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \quad (3.62)$$

where n is the number of parameters estimated. From (3.42) or (3.54) it may easily be shown that specification of such a matrix places potentially different upper bounds upon individual diagonal elements of the associated covariance matrix, thus retaining the property of preventing blowup.

The effect of individual parameter weighting may be seen in Fig. 3.11 where, the experiment of the previous section was repeated but with:

$$\Lambda_d = \begin{bmatrix} \lambda_b & 0 \\ 0 & \lambda_a \end{bmatrix} = \begin{bmatrix} 10^2 & 0 \\ 0 & 10^3 \end{bmatrix} \quad (3.63)$$

It is clear from the results of this experiment, with reference to Fig. 3.10, that the increased weighting of the $\hat{a}$ parameter dramatically reduces the amount of parameter drift due to noise and also the transient variation of $\hat{a}$ during process changes.

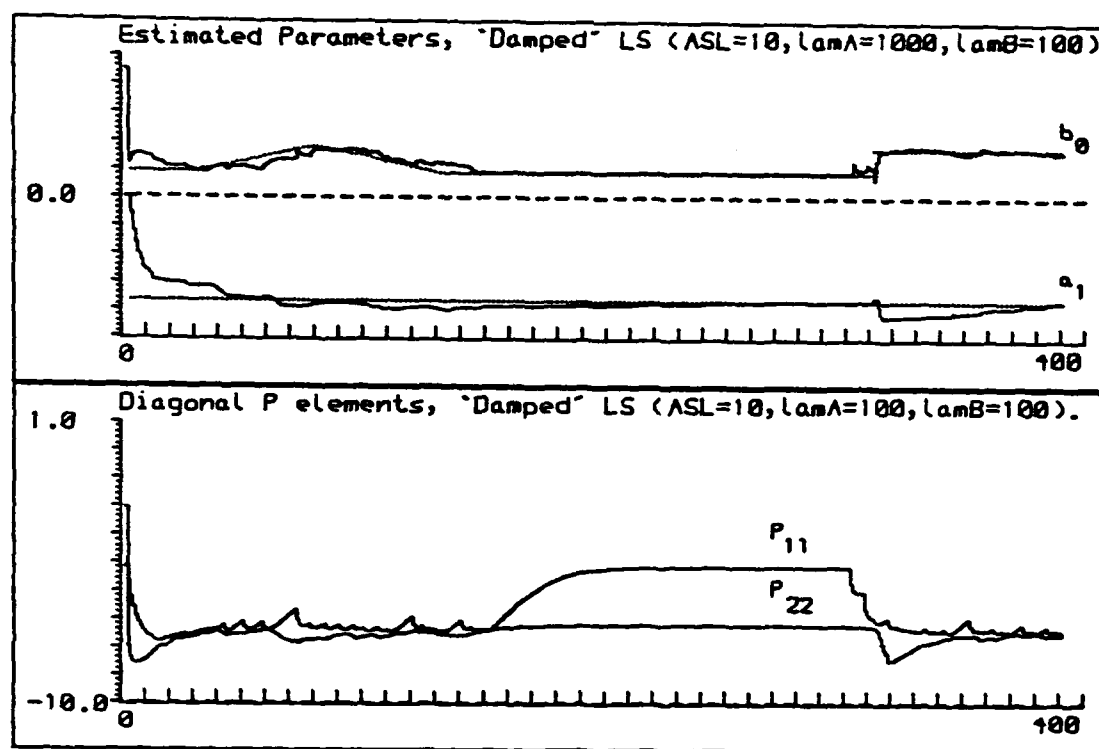


Figure 3.11: 'Damped' least squares with individual parameter weighting.

Because of the linear difference equation basis of the CARIMA model a slight change in an estimated $\hat{a}$ parameter can have a disproportionately large effect upon the estimated gain. As in general the quality of control performance is more sensitive to incorrect $\hat{b}$ parameters than incorrect $\hat{a}$ parameters the facility to deliberately restrict $\hat{a}$ parameter variation is therefore an important feature of the estimation algorithm based upon parameter weighting.

3.5.4 The role of fault detection

The conventional solution to the problem of long-term parameter adaptation is the use of an algorithm that always retains a 'small' amount of adaptivity. This continuous estimation suffers potentially from sensitivity to noise, especially during long periods when there is little exciting data such as routine output regulation, often causing the estimated parameters to drift significantly. More cautious algorithms, for example those using variable forgetting factors, seek to switch estimation off during such periods and only restimulate adaptivity when there is further exciting data. These latter algorithms can indeed reduce

parameter drift but are however liable to over-react in the presence of large unmeasurable load disturbances, causing unnecessarily poor estimation which, in the absence of further exciting data, can result in the estimated parameters remaining incorrect for long periods.

Ideally an estimator should be capable, upon detection of a sudden increase in the estimation error, of quickly determining whether this was due to a load disturbance or a genuine process change. In the former case the corresponding data should be disregarded whilst in the latter adaptivity should be increased to assist parameter tracking. Unfortunately this is unlikely to be possible in practice, if the load disturbances are truly unpredictable or unmeasurable, without the benefit of *hindsight*. There are however two reasonable estimation strategies that may be used:

- The performance of a fixed, or slightly adaptive, estimated model could be monitored using an FDI procedure (see section 1.4) and adaptivity enabled, or increased, only if a *genuine* process change is indicated. This is a cautious approach with the potential disadvantages that by the time a change has been detected it may be too late to track it, and that the delay in updating the model could result in an unacceptably long period of poor control performance.
- The use of continuous adaptation combined with some form of model evaluation or FDI procedure. Thus, in principle, all apparent process changes would be 'provisionally' tracked but new parameters could subsequently be discarded if they did not prove to be clearly superior to earlier values. Related strategies employ several parallel estimators linked by a 'voting' system that determines the final parameters to be used (Lam, 1985). This approach has the potential advantage of rapid tracking of genuine parameter changes.

With reference to the properties outlined in section 3.5.1 CLS appears to be particularly suitable for the latter type of estimation strategy. This is because *sustained* variation of estimated parameters from their nominal values can only occur *if* there is a genuine process change *and* persistently exciting data (see Fig. 3.12). For this reason the estimated parameters themselves provide a realistic fault detection diagnostic, inherently reverting to the nominal parameters under all other circumstances. Furthermore CLS was shown to be capable of retaining relatively high adaptivity, without suffering from blowup or parameter drift, and should therefore track most realistic process changes.

Thus in this section the constrained least squares algorithm is extended, by the use of fault detection, to form an algorithm suitable for the tracking of permanent process changes despite the presence of load disturbances. This is achieved by periodically updating the nominal parameters, θ_0 , when a genuine process change is detected.

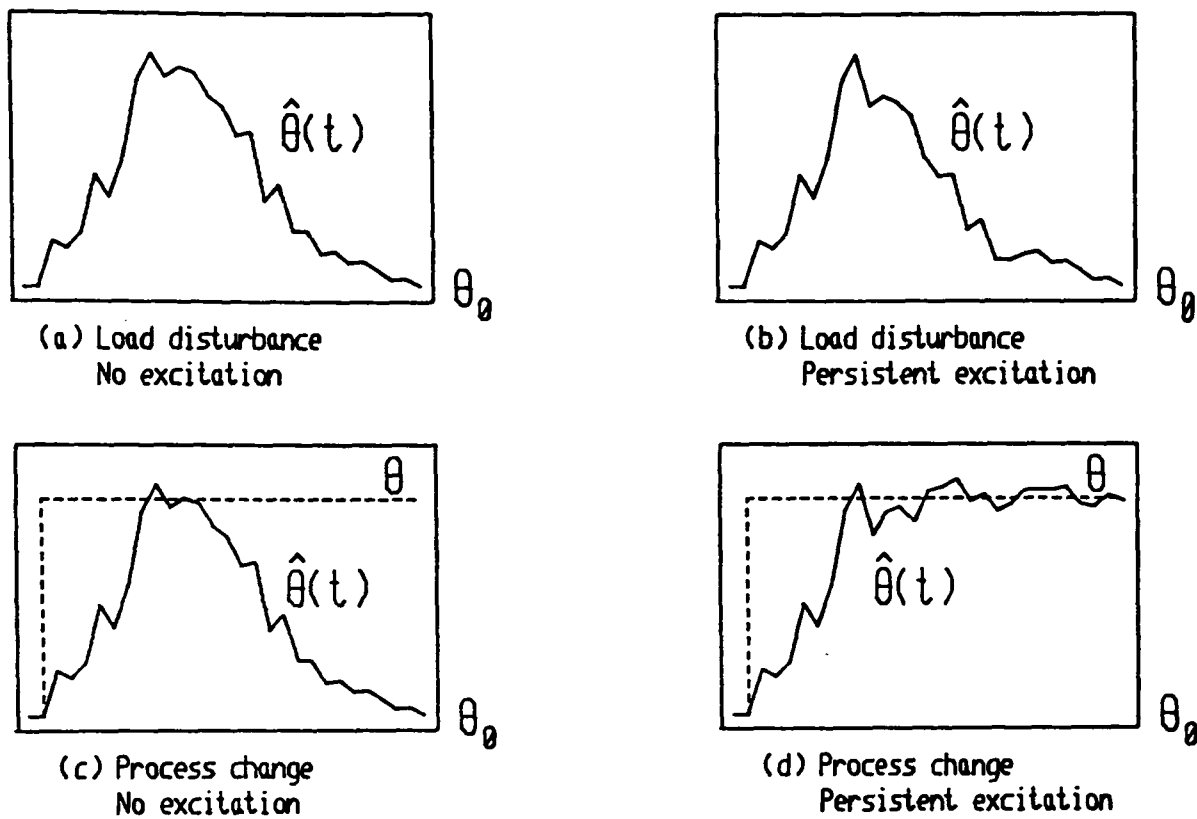


Figure 3.12: Expected parameter variation: Constrained least squares.

In order to distinguish the genuine parameter change in Fig.3.12 from the other *transient* variations in the estimated parameters two separate fault diagnostics are proposed, one related to the magnitude of the variation and the other related to the direction. Thus two series are formed :

$$\begin{aligned}\Delta\theta_1(t) &= |\hat{\theta}(t) - \theta_0| \\ \Delta\theta_2(t) &= \Delta\theta_1(t) - \Delta\theta_1(t-1)\end{aligned}\tag{3.64}$$

In order to make the fault detection independent of the magnitude of the parameter variation a further statistic is formed,

$$s(t) = \begin{cases} 1 & \text{if } \Delta\theta_1 \geq \alpha_1\theta_0 \\ 0 & \text{otherwise} \end{cases}\tag{3.65}$$

where $\alpha_1\theta_0$ represents a pre-specified threshold chosen normally as a 'percentage of nominal'. $\Delta\theta_2(t)$ and $s(t)$ are then low-pass filtered to give,

$$r(t) = \alpha_2 r(t-1) + (1 - \alpha_2) s(t)\tag{3.66}$$

$$q(t) = \alpha_3 r(t-1) + (1 - \alpha_3) \Delta\theta_2(t)\tag{3.67}$$

where $r(t)$ corresponds to an estimate of how often the estimated parameter variation from nominal exceeds the pre-specified threshold, and $q(t)$ is an estimate of the direction of the parameter change. The parameters α_2 and α_3 may be used to specify the 'window-length' over which this fault detection is to take place, normally being chosen as $\alpha_2 = \alpha_3 = \beta$. In general a small value of α_2 or α_3 allows rapid fault detection but at the expense of increased occurrence of false alarms ... another classic trade-off of estimation.

A fault is declared if $r(t)$ exceeds a pre-specified threshold r_T , and confirmed if at that point $q(t) \geq 0$ indicating that the estimated parameter is not returning to its nominal value, i.e.

$$r(t) \geq r_T \quad \text{and} \quad q(t) \geq 0 \quad (3.68)$$

Once a fault is detected then the nominal parameters are updated with the current values of the estimated parameters, $\hat{\theta}(t)$, and the fault detection procedure reset by setting $r(t)$ and $q(t)$ to zero.

3.5.5 Some illustrative simulation studies

Fig. 3.13 shows the open loop input and output variables of a simulation experiment designed to illustrate the potential performance of the extended CLS estimator.

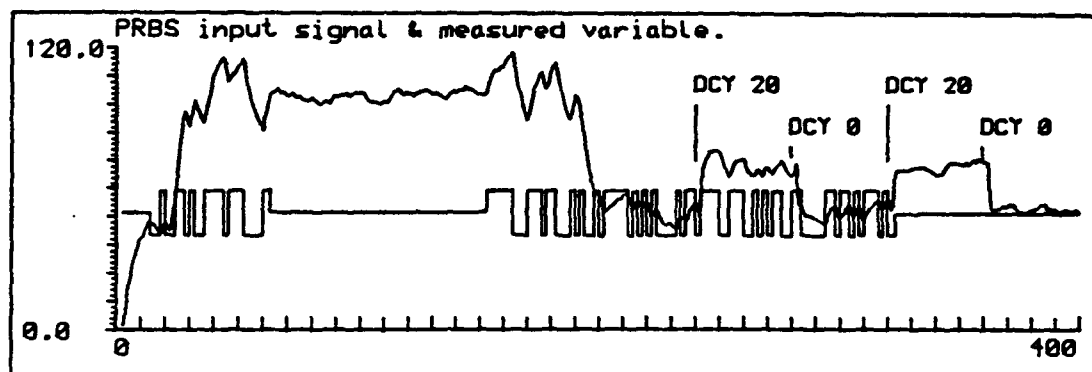


Figure 3.13: Input and output variables

This experiment consisted of estimating the parameters of the model given in Example 3.1 despite large process changes, periods of exciting and non-exciting data, and large step-wise load disturbances upon the measured variable. The results of this experiment are shown in Fig. 3.14.

The step-wise variation in the b_0 parameter at $t = 20$ is tracked accurately with a fault indicated and confirmed at $t = 42$. The nominal parameters are updated accordingly and this ensures that during the following period, when the PRBS input signal is not applied,

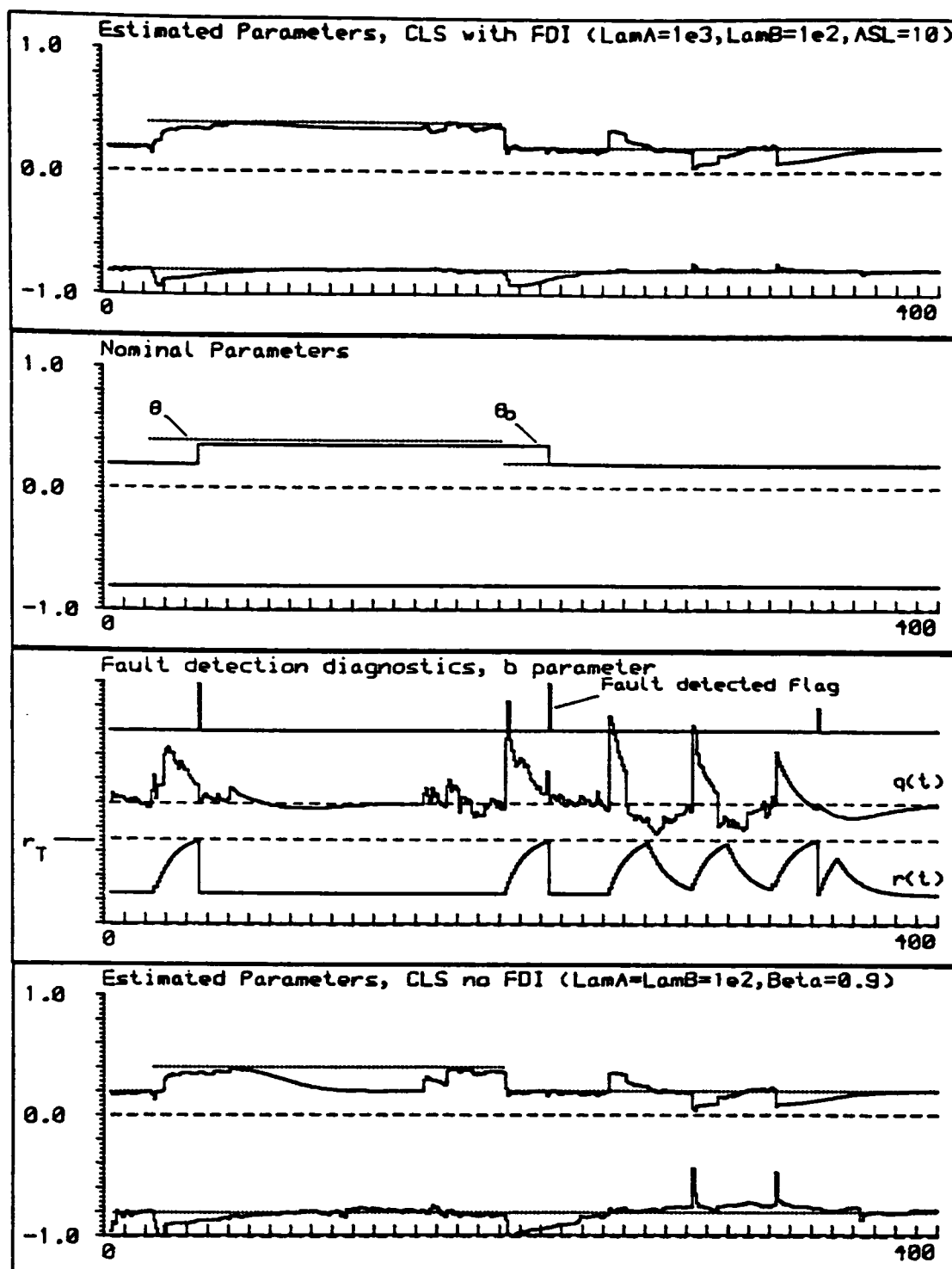


Figure 3.14: Estimation in the presence of load disturbances: CLS with FDI

the estimated parameters do not revert to the earlier nominal values that are now invalid. When b_0 is restored at $t = 190$ this tracking of both estimated and nominal parameters is successfully repeated. The application of load disturbances, simulated by successively adding and removing a constant offset of 20 units to the measured variable, causes only transient variation of the estimated parameters. At one point, after a load disturbance, the fault detection diagnostic $r(t)$ may be seen to exceed its threshold but a fault is not confirmed because $q(t) < 0$, correctly indicating that the estimated parameter is returning to its nominal value. The performance of the algorithm without fault detection is included for reference and confirms Property C.3 because, despite initially tracking the variation of b_0 , $\hat{b}_0$ reverts to its incorrect nominal value once excitation is removed.

During this experiment CLS was commissioned exactly as in section 3.5.3 with $\lambda_a = 10^3$, $\lambda_b = 10^2$ and $\beta = 0.9$. For fault detection the filter constants used were $\alpha_2 = \alpha_3 = \beta$, with $\alpha_1 = 0.2$ representing a '20% from nominal' threshold for parameter variation and $r_T = 0.9$ ensuring that a fault would be indicated if the estimated parameters stay outside this threshold for more than 20 samples.

In order to compare the performance of this algorithm to that of its 'rivals' the above experiment was repeated for four alternative PAA's previously described in 3.4. The results of these further experiments are shown in Fig. 3.15, for RLS with fixed forgetting, DLS, RLS with constant trace and RLS with Ydstie's variable forgetting factor, respectively.

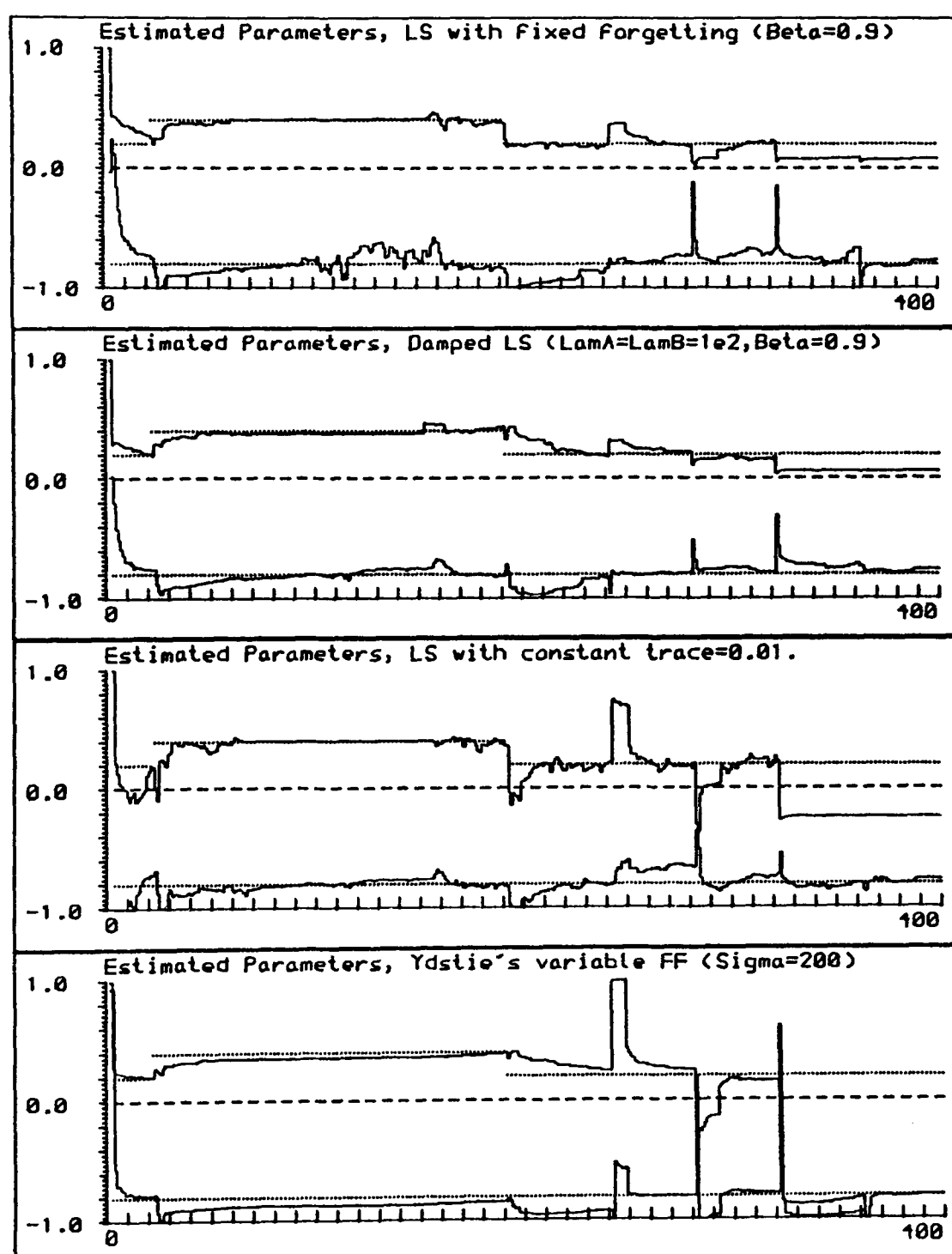


Figure 3.15: Estimation in the presence of load disturbances: Alternative algorithms.

All these algorithms provide good initial self-tuning and successfully track the process

changes but the load disturbances cause large variations in the estimated parameters and offsets which, in the absence of further exciting data, persist. Particularly poor is the performance of the variable forgetting factor which, without a minimum value for the forgetting factor, over-reacts to the load disturbances causing massive parameter variation.

3.6 ESTIMATOR JACKETING

Whilst most RLS-based estimators are capable of satisfactory performance in simulation in practice there are several problems that often cannot be overcome by theory alone, including noise, actuator stiction and saturation, sensor failure etc. There are however several relatively simple process specific modifications that, whilst relatively *ad hoc* have proved to be effective in enhancing general estimator robustness (Clarke, 1981; Seborg *et al*, 1986).

3.6.1 Signal conditioning

The GPC algorithm is based upon a linear difference equation description of the process to be controlled which may be written as :

$$\bar{y}(t) = \underbrace{\phi(t-1)^T \hat{\theta}(t)}_{\text{Estimated model}} + \underbrace{e(t)}_{\text{Error}} \quad (3.69)$$

where $e(t)$ is a disturbance term which is assumed to be Gaussian with zero mean. Obviously this will not be the case in practice where $e(t)$ includes :

- *Measurement noise*: the on-line measurement of a process output is often subject in practice to large amounts of sensor noise, d.c. offsets and even long-term drift.
- *Unmodelled dynamics*: to simplify estimation it was assumed in section 3.3.3 that a second order model structure would be adequate for most process control applications. Any additional higher order (and normally higher frequency) dynamics cannot be fully modelled and thus form part of the disturbance term. Included in this is any model-plant mismatch due to incorrect estimated parameters.
- *Aliasing*: higher frequency model dynamics can appear at a lower frequency due to the sampling operation.

The first can cause unnecessary variation of the estimated parameters and/or bias (3.3.4) and the second can prevent convergence (Rohrs,1982). It is therefore clear that this noise term should either be estimated, and thus incorporated into the model, or removed using signal conditioning, in order to improve the 'signal to noise ratio'.

Signal conditioning for estimation may therefore be defined as the 'elimination of those parts of the disturbance term which cannot be eliminated by the identification method' (Isermann,1980). For the types of signals encountered in process control applications this should take the form of fixed filtering with an analogue anti-aliasing filter, of a time-constant at least twice that of the sample interval, and a digital band-pass filter with a bandwidth approximately the same as the process⁹. For the CARIMA model used in GPC this is implemented by the high pass filtering inherent in the use of differenced data, cascaded with a 1st or preferably 2nd order low-pass filter (Fig. 2.11). The low pass filtering reduces the contribution to $y(t)$ of high frequency noise and unmodelled dynamics. The choice of filter time-constant for individual processes could be made on the basis of a simple step test.

3.6.2 Estimation dead-bands

The inclusion of a mechanism by which parameter adaptation may be suspended has been found to be of crucial importance in the proof of convergence and stability results in estimation (Egardt,1979; Samson,1983; Martin-Sanchez,1984; Goodwin *et al*,1985; Kreisselmeier & Anderson,1986). In practice the incorporation of a dead-band based upon the estimation error, inside which estimation does not take place, provides a simple *ad-hoc* method of freezing estimation when it is not required. This has been found to enhance the robustness of estimators in the presence of noise, unmodelled dynamics and unmodelled non-linearities such as stiction (Seborg *et al*,1986). Its effect is similar to the filtering of the previous section in that it attempts to eliminate unwanted data by forcing the estimator to ignore it.

An estimation dead-band may be readily implemented upon the basis of the *a priori* prediction error, $\epsilon(t)$:

$$\epsilon(t) = y(t) - \overbrace{\phi(t-1)^T \hat{\theta}(t-1)}^{\hat{y}(t|t-1)} \quad (3.70)$$

⁹or with a 3dB point corresponding to the dominant process 3dB point if there are significant high frequency unmodelled dynamics.

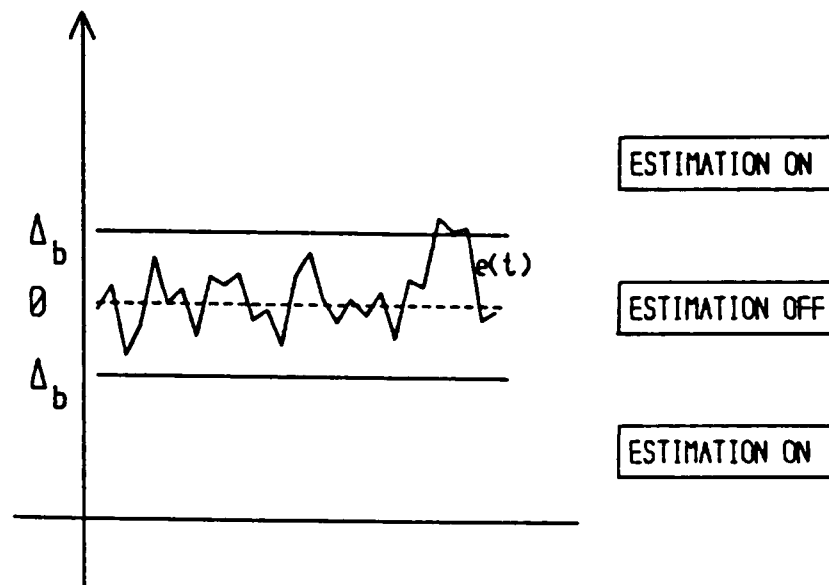


Figure 3.16: Implementation of a simple estimation dead-band.

where

$$\text{Estimation is } \begin{cases} \text{ON} & \text{if } |\varepsilon(t)| > \Delta_b \\ \text{OFF} & \text{otherwise.} \end{cases} \quad (3.71)$$

and Δ_b is the magnitude of the estimation dead-band (Fig. 3.16).

In practice a reasonable value for Δ_b could be chosen by logging the output data, estimating the variance of the noise at steady state, σ_n , and setting $\Delta_b \geq 2\sigma_n$. It is important to note that the use of an estimation dead-band is only appropriate for an adaptive control algorithm whose integral action is inherent and does not depend upon estimation of an offset term (Hodgson, 1982), e.g. GPC.

3.6.3 Normalisation and Scaling

The normalisation of data in estimation is important, like the estimation dead-bands of the previous section, for both the analysis of algorithms and their practical application. For analysis normalisation of data allows modelling errors to be considered as bounded disturbances, thus enabling the development of fundamental convergence and stability results (Egardt, 1979; Praly, 1984; Ortega *et al*, 1985). For practical application normalisation can be used to improve numerical conditioning and to simplify estimator specification.

Normalisation of practical data involves its conversion from the engineering units appropriate to the process (i.e. kg, %, °C, ...) to 'sensible' internal units that will be used in the estimation procedure. To some extent this is inherent in the calibration of the ADC's used to input the data where the data is already normalised between limits corresponding to (say) $-10 \rightarrow 10$ Volts. However it may be advantageous to perform further normali-

sation such as that recommended by Cluett (1986) where a normalisation factor, $\eta(t)$, is specified at each sample interval,

$$\eta(t) = \max \left(\max_{1 \leq i \leq n} |\phi_i(t-1)|, c \right) \quad (3.72)$$

where n is the number of estimated parameters. The normalisation is implemented simply by dividing the measured variable, the data vector, and the prediction error by $\eta(t)$ before calling the estimation routine, i.e.

$$y_\eta(t) = \frac{1}{\eta(t)} y(t) \quad , \quad \phi_\eta(t-1) = \frac{1}{\eta(t)} \phi(t-1) \quad \text{and} \quad e_\eta(t) = \frac{1}{\eta(t)} e(t) \quad (3.73)$$

Alternatively, if it is known *a priori*, a fixed normalisation factor η could be chosen as the largest value expected to be encountered in the data vector.

A practical example of the benefits of normalisation is in the specification of Ydstie's variable forgetting factor, previously described in section 3.4.2. This calculates the value of the forgetting factor on the basis of the weighted sum of the prediction errors squared and thus its specification, and subsequent performance, is sensitive to the noise variance, σ_n^2 . If however the data was normalised by (say) $10\sigma_n^2$, with the variance of the noise being estimated at steady state, then this sensitivity could be reduced.

3.6.4 Parameter constraint

The use of an adaptive algorithm in principle offers long term tight control where the controller parameters are continuously updated to maintain the quality of control despite changes in the process. If, however, the controller parameters are updated incorrectly, because of gross errors in the estimated model parameters, then this can lead to a significant degradation in control and, in the extreme, instability. This can have 'unfortunate' consequences for the output material and/or the process itself and thus must be prevented at all costs¹⁰.

It is well known that for continuously adaptive algorithms significant variations in the estimated parameters can occur because of sudden process changes, or large unmeasurable load disturbances, due to the corresponding large estimation error. To preserve satisfactory control during such periods it is therefore important not to update the estimated model if these new estimated parameters are obviously erroneous. In the absence of 'the perfect estimator' it should be possible to overcome this problem by the application of simple

¹⁰especially for commercial application in the U.S.A. where the legal consequences of controller failure could be expensive !

tests at each sample interval to the new 'provisional' parameter estimates before they are accepted.

Such a procedure implies a requirement for estimator diagnostics that indicate the validity of the estimated parameters, which may easily be calculated, and whose bounds are readily available *a priori*. For process control applications, where the estimated model would normally be assumed to be second order, the following diagnostics are suggested (Vogel,1982; Goodwin & Sin,1984; Seborg *et al*,1986):

- *Gain*: the estimated model should only be updated if the magnitude of the gain indicated by the new parameter estimates, given by

$$|\text{Estimated gain}| = \left| \left(1 - \sum_{i=1}^{\delta A} a_i \right)^{-1} \sum_{i=1}^{\delta B} b_i \right| \quad (3.74)$$

lies within a pre-specified acceptable range. If the estimated gain is too high then this would result in sluggish control whereas if it is too low then this can lead to unnecessarily active and possibly oscillatory control. The lower gain limit also protects the adaptive controller against instability due to the estimation of a process model with gain of the wrong sign.

- *Stability*: if the process is known to be open-loop stable then the model should be updated only if the estimated parameters themselves correspond to a stable model. For a second order system (i.e. 2 $\hat{a}$ parameters) this corresponds to the region in parameter space shown in Fig. 3.17.
- *Rise-time*: as for the gain, if the rise-time of the model given by the new estimated parameters exceeds pre-specified upper and lower bounds then the old model should not be updated.

In practice it should always be possible to obtain approximate values for all these constraints.

An alternative approach (Goodwin & Sin,1984; Seborg *et al*,1986) is to specify a boundary in parameter space, for example a n -dimensional sphere, and to project any estimated parameter sets that lie outside this region onto its surface. This is shown schematically in Fig. 3.17 for the two parameter case, where the modified parameter update for an erroneous estimated parameter set, $\hat{\theta}(t)$, outside the region with centre θ_c and 'radius' K , is given by:

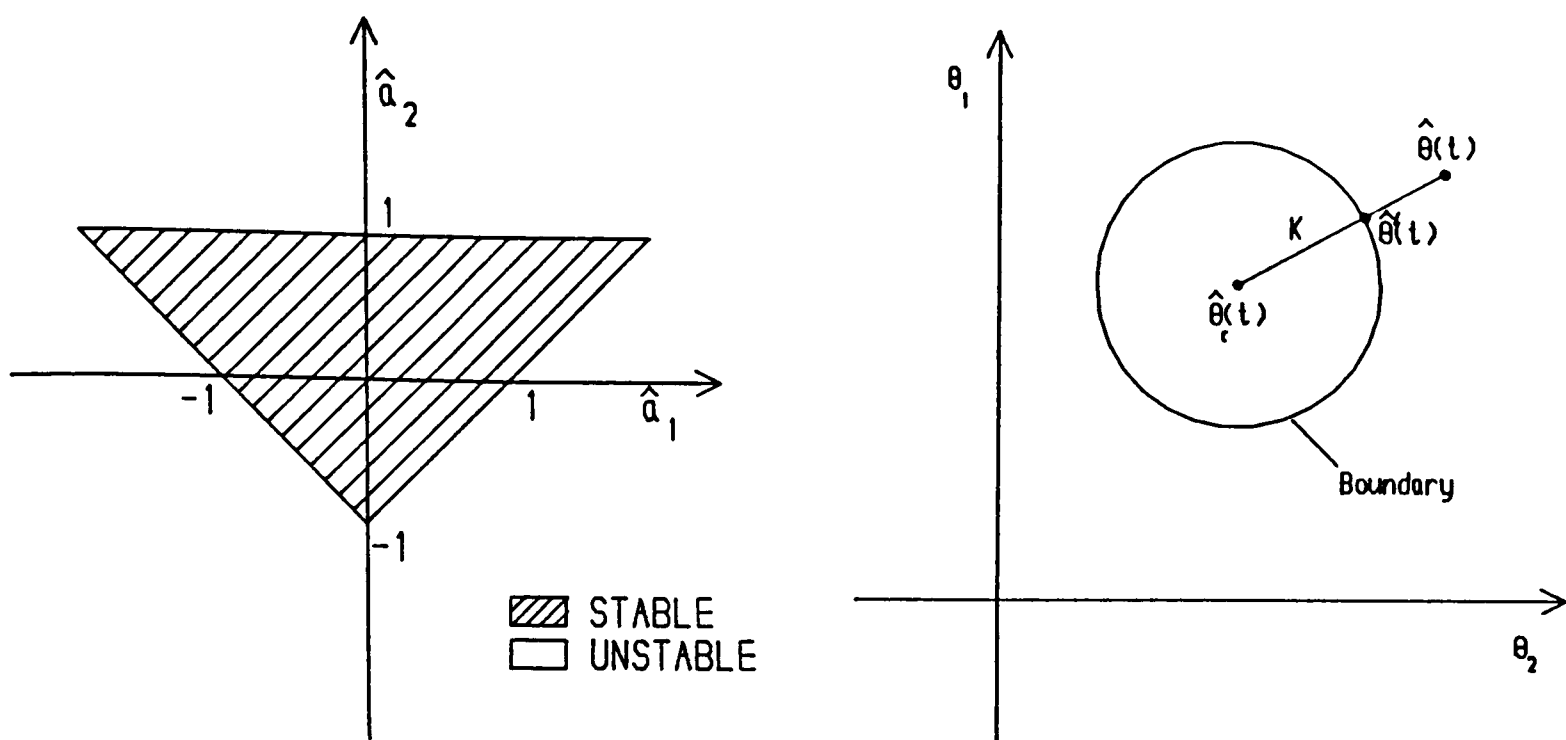


Figure 3.17: Types of parameter constraint.

$$\hat{\theta}'(t) = \theta_c + \underbrace{\frac{(\hat{\theta}(t) - \theta_c)K}{|\hat{\theta}(t) - \theta_c|}}_{\text{normalised projection}} \quad (3.75)$$

Due to the difficulty of meaningfully specifying such a well behaved boundary, and the additional computational burden of the parameter projection, this is principally an analysis tool and less practical than simply discarding the erroneous parameter estimates.

3.6.5 Expert systems

The term 'expert systems' is used to describe an area of Artificial Intelligence (AI) that has expanded rapidly in recent years (Hayes-Roth *et al*, 1983; Waterman, 1986) whose incorporation into adaptive control is claimed to offer substantial benefits (Sanoff & Wellstead, 1984; Åström & Anton, 1985; Årzén, 1986). The practical application of adaptive control normally involves a substantial amount of heuristic logic both at the commissioning stage (i.e. 'choose the sample interval to be 1/10th of the process rise-time'), and the maintenance of good robust performance (i.e. 'disable estimation when the estimation error is small'). The specification of these heuristics depends upon the availability of prior knowledge about the process and can be relatively complicated and unclear for a user unfamiliar with adaptive control theory. The twin goals of an expert system are therefore to provide a meaningful framework in which these heuristics may be embedded and a means of collecting the necessary knowledge upon which they are based. Although this applies

equally to both control and estimation it is the latter that is specifically considered in this section.

Let us first consider commissioning, where the values of algorithmic features such as forgetting factors, and jackets such as dead-bands, must be specified. The choice of these values will depend upon the desired performance of the estimator and prior process knowledge where the latter can be collected in one of two ways; by asking the process operator; or by performing experiments upon the process itself. An experienced process operator will usually have at least qualitative knowledge about the process, perhaps including estimates of the rise-time, static gain, the nature of certain non-linearities etc. Alternatively simple estimation tests can be conducted such as step tests or open loop estimation using a PRBS input signal to probe the system. Despite the fact that information from both these sources is often diverse, uncertain and/or contradictory, the purpose of the expert system should be to exploit and refine this knowledge (Årzén,1986).

An illustrative example of how a expert system might gather such information, by operator interrogation, is shown below (Wellstead & Sanoff,1984) :

Is the process open-loop stable (Yes/No/Help) ? Y

Do you know the rise-time of the process (Yes/No/Help) ? Y

Do you know it precisely (Yes/No/Help) ? N

Please select a rise-time, T_r , from the menu :

1. $T_r \leq 1$ minute
2. $1 \text{ minute} \leq T_r \leq 10 \text{ minutes}$
3. $10 \text{ minutes} \leq T_r \leq 1 \text{ hour}$
4. None of the above

Your selection please ?

In terms of estimator supervision the expert system then uses the knowledge it has gained in order to perform such tasks as fault detection, verification of the estimated parameters, estimator freezing etc., i.e. as described in the sections 3.6.1 – 3.6.4.

Whilst the detailed mechanism by which expert systems are realized is beyond the scope of this thesis, an expert system consists of two basic elements, a *knowledge base* which is a collection of rules and facts pertaining to practical estimation, and an *inference engine* which on the basis of incoming data interprets these rules to determine any appropriate action if necessary. Both of these features should be completely separate from the actual parameter estimator. Because the response time of the estimation algorithm must match

that of the process, whilst rule interpretation is comparatively slow, this leads inherently to a requirement for a multi-tasking implementation with the estimator and expert system as 'foreground' and 'background' tasks respectively. This structure is shown schematically in Fig. 3.18.

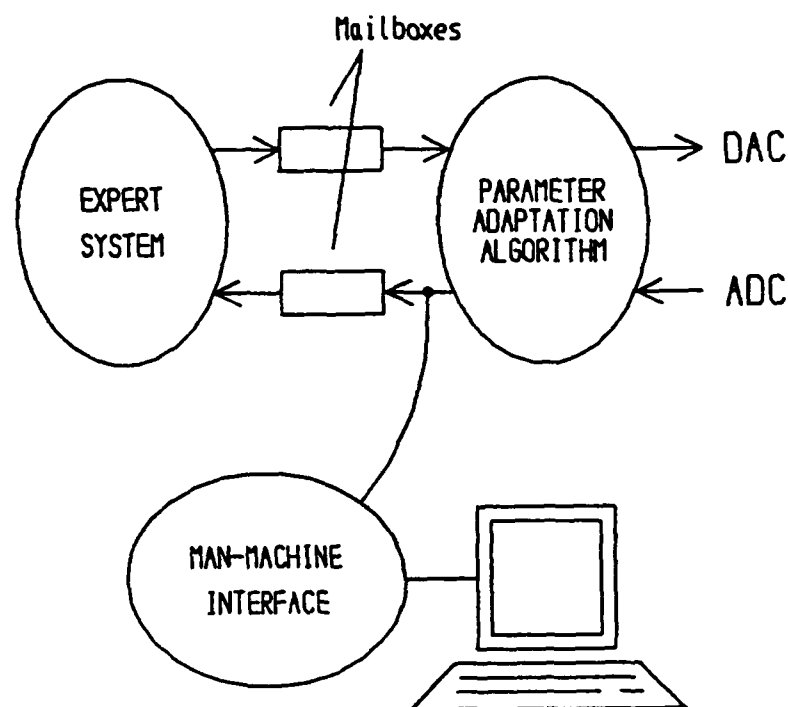


Figure 3.18: The inherent multi-tasking nature of an expert system.

Thus the potential benefits of using expert systems in practical parameter estimation may be summarised as :

- Simplification of the commissioning stage thus enabling a 'complicated' estimator to be used by control engineers unfamiliar with the theory.
- Provision of a formal, flexible mechanism for monitoring and, if necessary, modification of the estimator performance during routine operation in order to guarantee long term integrity. This could be referred to as *estimator management*.

3.7 CONCLUSIONS

In this chapter the fundamental areas of practical parameter estimation were outlined and used to justify the choice of 'recursive least squares' as a particularly appropriate basic algorithm. Whilst a very powerful method ordinary RLS was shown, by analysis and simulation, not to be suitable for the long term tracking of time-varying parameters because it quickly loses adaptivity.

This was followed by a review of a representative selection of currently available algorithms based upon RLS but modified to retain adaptivity by modification of the covariance

or P matrix. The relative merits and demerits of these algorithms were discussed and then demonstrated in simulation. All were shown to provide long-term parameter tracking but at the expense of covariance blow-up (fixed and directional forgetting), parameter drift (fixed and directional forgetting, constant trace) or potential sensitivity to load disturbances (particularly variable forgetting).

To overcome these problems a novel parameter adaptation strategy was introduced that extends RLS to include parameter weighting. Two closely related algorithms were developed, 'constrained' least squares which weights the difference between the estimated parameters and a nominal parameter set, and 'damped' least squares which weights increments in the estimated parameters. Both of these algorithms were shown to have several desirable properties, including the inherent prevention of blowup, reduced parameter drift, a means of on-line specification of individual parameter adaptivity and the provision of a meaningful mechanism for the incorporation of detailed prior process knowledge.

In particular the combination of constrained least squares and a related fault detection algorithm based upon the variation of estimated parameters was shown to be able to track process changes even in the presence of large unmeasurable load disturbances. It is thus suggested that this type of estimation strategy is more realistic than continuous low-gain estimation for long-term practical application.

Finally several important 'jackets' were discussed that attempt to enhance long term estimator integrity. It was argued that the application of the expert system methodology could offer significant benefits in this area as well as for estimator commissioning.

Chapter 4

GPC control of an industrial soap dryer

X This chapter describes the work performed from December 1985 to July 1986 on the first part of a collaborative project between Oxford University Self-Tuning Group and Unilever Research and concerns the application of the Generalised Predictive Control algorithm to the control of an Lever Brothers industrial soap spray drying process.

4.1 INTRODUCTION

One of the last stages in the manufacture of any solid soap product involves cooling and drying the molten soap mixture to form the solid product suitable for moulding, packaging and sale (Woollatt, 1985). The moisture content of this dried soap is important as it determines such basic properties as appearance, texture, lathering and its effective life-time when used. In particular :

- Excessive moisture content can degrade soap consistency and, in the extreme, can contravene trading standards.
- Insufficient moisture content can degrade washing properties, it can prevent successful stamping and mixing during subsequent during later stages of production and, in the extreme, can lead to blockages in the heat exchangers.

In practice, therefore, the control problem consists of maintaining the moisture content as high as possible in order to maximise the number of bars of soap produced per unit ingredient, whilst remaining within the above constraints of quality. This is shown

schematically in Fig.4.1. An additional control objective is that of set-point following because the same soap dryer is used for a variety of input soap bases¹, where the desired moisture content is often different for each type of soap.

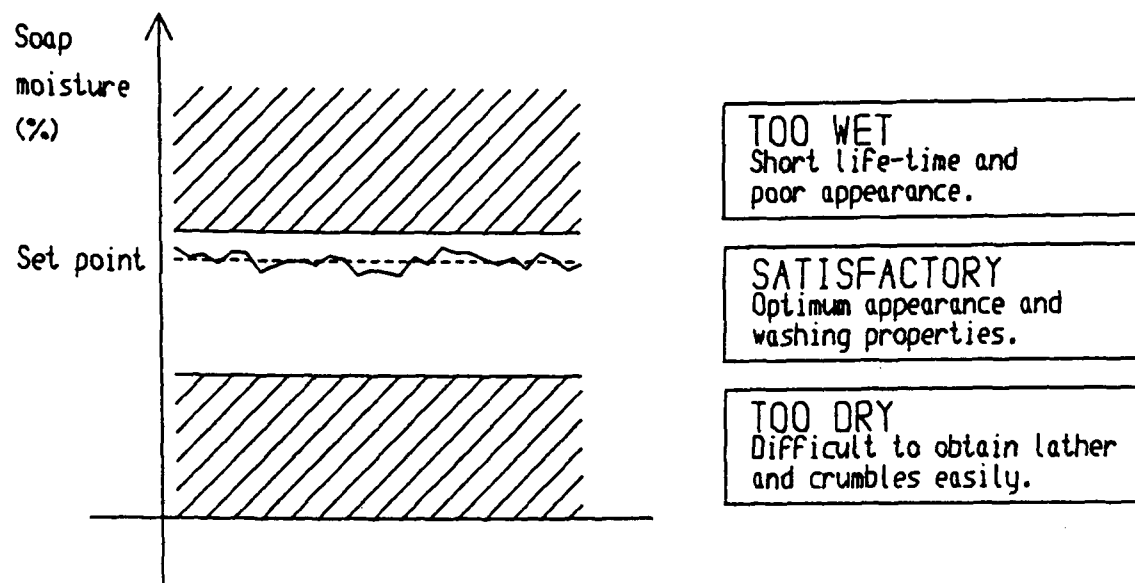


Figure 4.1: The control objective during soap drying.

During normal operation the soap moisture content is controlled manually with off-line analytic moisture tests being conducted periodically and, if necessary, the input signal altered on the basis of past experience. However because of significant variations in the quality and consistency of the input soap 'boils'² this control strategy often results in long periods of offsets in the moisture and poor transient control during soap base changes.

Using an on-line moisture measurement (section 4.3) the principle objective of this work was therefore to demonstrate an overall improvement in moisture control through the use of the Generalised Predictive Control algorithm. In particular, because the main moisture control objective is regulation, the trials aimed to show the inherently accurate and offset free nature of the GPC control law.

It is important to note that all these experiments were to take place during actual full-scale commercial production of soap and thus were subject to the constraint that they should in no way interfere with soap manufacture. This meant that commissioning, control transfer from manual, parameter estimation (including the initial self-tuning) and experimentation, would all have to be performed within the production tolerances shown in Fig.4.1. Whilst this limited the range of experiments that could be conducted, this was considered less important than the practical relevance of the trials. As they took

¹ where 'base' refers to the mixture of ingredients used, each different base being given a code number 1 to 6 corresponding to the various types of soap produced.

² operator terminology.

place under the full rigour of industrial soap manufacture this corresponded exactly to the conditions that would be encountered in actual, rather than experimental, application of a commercial adaptive controller.

Further objectives concerned the specification and commissioning of the GPC controller. The aim here was to show that the 'default' specification proposed in Chapter 2 is indeed suitable for industrial control and that, despite the practical constraints, commissioning is both straightforward and rapid because of the adaptive nature of the algorithm.

Finally it was expected that the use of an adaptive feedback control strategy such as GPC would provide tighter control of soap moisture during soap base changes compared to manual control. If possible it was hoped to shorten the time taken to achieve the new soap moisture set point after the soap was switched over because soap produced during this period may be wasted.

In summary the objectives of the work performed on the Lever Brothers soap dryer were as follows :

- To evaluate the performance of the long range predictive GPC control algorithm under the rigour of full-scale industrial plant operating conditions given the criteria outlined above.
- X • To demonstrate the feasibility of using adaptive control in practice.
- To determine whether a sophisticated modern control strategy, such as the long range prediction based GPC algorithm, could improve the overall control of moisture over that currently achieved by manual control.

The soap drying process is described in more detail in the next section. Section 4.3 provides a brief description of the hardware and software used throughout the trials. A selection of the results obtained from these trials are presented in section 4.4 and demonstrate satisfactory periods of soap moisture regulation and set-point following. Of particular interest is the GPC control of soap moisture during a simultaneous change of input soap base, set-point *and* soap flow rate. The chapter ends with a discussion of the results and concludes that the objectives, outlined above, were all achieved successfully.

4.2 THE SOAP DRYING PROCESS

A schematic diagram of the Lever Brothers full-scale soap drying plant is shown in Fig. 4.2. After the soap ingredients have been mixed and heated, and the chemical reaction is complete, the various soap mixtures are stored in a series of large holding tanks as a molten slurry. The next stage is to dry the soap until it is solid yet soft enough to be moulded into the familiar 'bar of soap'. Thus the molten soap is pumped through a series of two heat exchangers where it is heated further and then forced into a vacuum drying chamber through an atomiser at the top. The combination of the high temperature and the sudden expansion of the soap spray rapidly removes most of the water content of the soap as steam which is subsequently condensed into a separate drum. Rotating blades within the drying chamber scrape the still-moist soap powder off the walls upon which it falls to the base, is compressed, and extruded through a pair of multi-orifice nozzles. As it emerges from the nozzles a further set of rotating blades cut the soap into 'noodles' which drop into a hopper for storage.

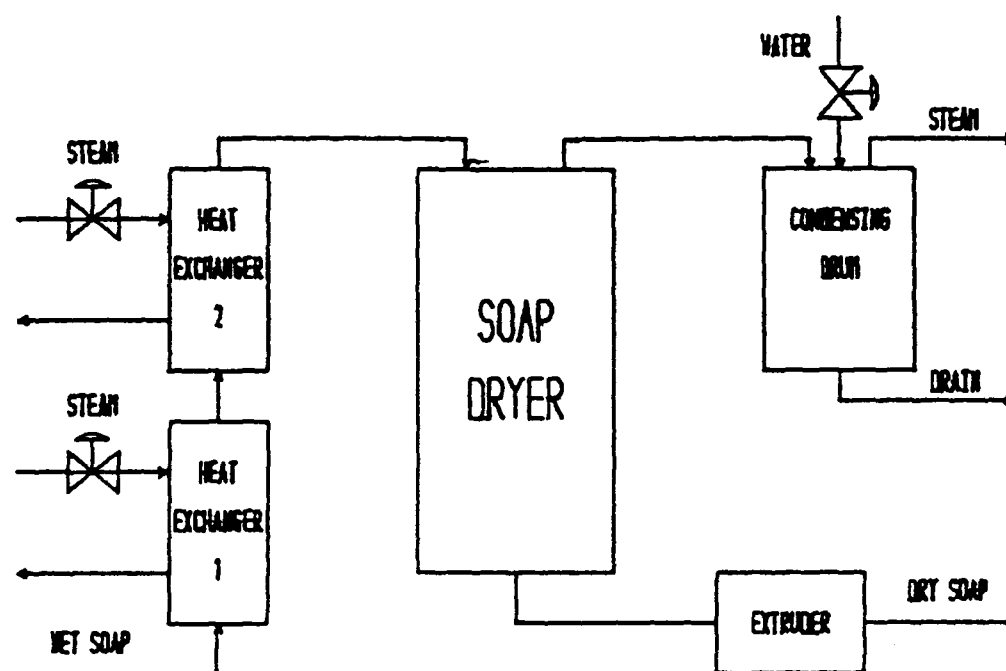


Figure 4.2: Schematic diagram of the soap drying process

Due to the high volume of soap manufacture the dryer is operated almost continuously with the input soap slurry being changed periodically whenever the current input tank is exhausted. As several different types of soap are dried by this plant this changeover can often involve a 'soap base change', where not only does the quality of input material change suddenly but also the desired moisture content of the dried soap.

The moisture content of the soap upon exit from the dryer is determined both by the amount of steam entering the heat exchangers and also the input flowrate of the

molten soap. Thus current control operation depends upon specification of the ratio of the steam flowrate set-point in the first heat exchanger to that of input soap flowrate, referred to as the 'steam/soap ratio'. The set-point for the steam flowrate in the second heat exchanger is itself ratioed directly to that of the first. All the internal flowrates are controlled automatically using PID controllers implemented on a Babcock Bristol Micro-B process control computer.

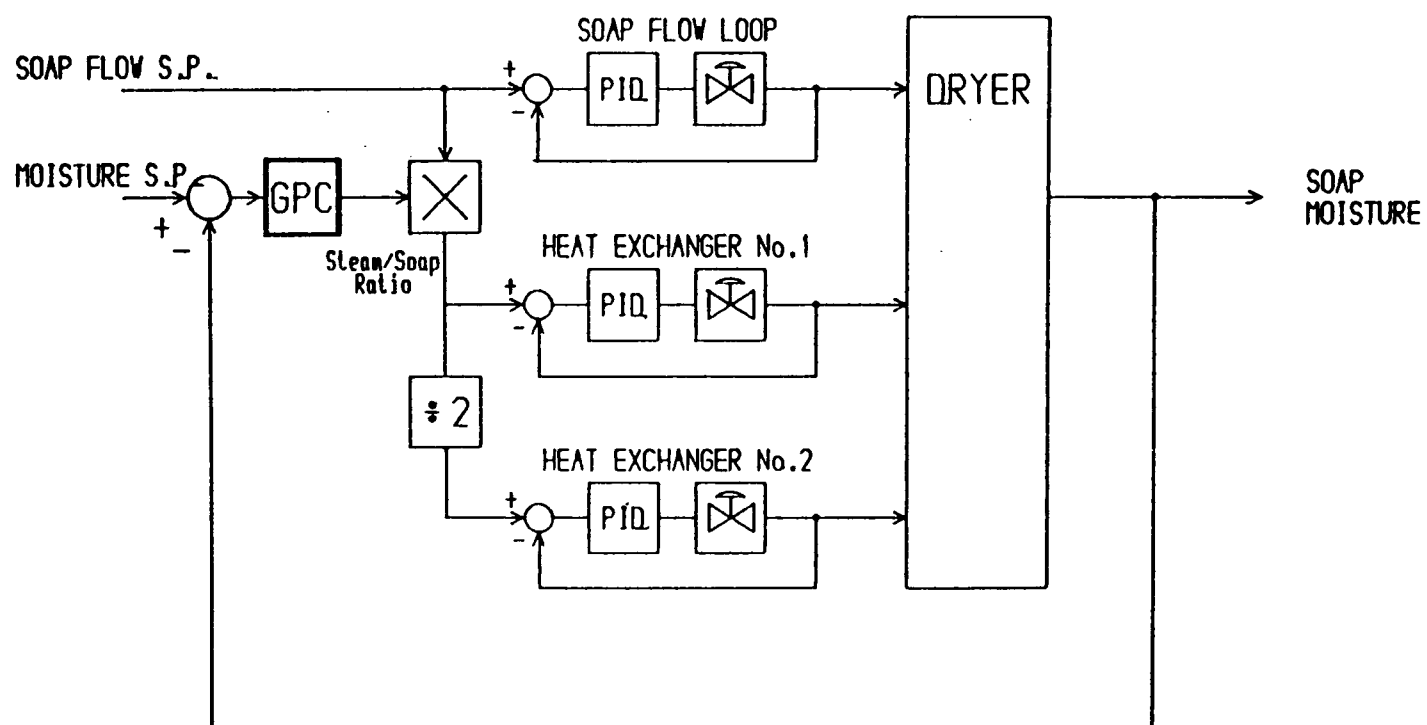


Figure 4.3: Hierarchical GPC control of soap moisture.

It is important to note that although the local flow loops are controlled automatically the actual moisture content depends upon operator manual control by manipulation of the steam/soap ratio. This input would normally be chosen from experience and adjusted on the basis of periodic³ off-line observations of the soap moisture content. These observations consist of taking a sample at the extruder and performing an analytic test to determine moisture content. It is also important to note that the input soap flow rate would normally be set to a constant value pre-specified by production levels and thus its presence in the input ratio is best regarded as a feed-forward term rather than an input.

Thus the control loop that GPC was applied to may be summarised as consisting of a typical industrial drying process whose output or measured variable is soap moisture content, and whose input or controlled variable is the ratio between the steam flow set-point (to the first heat exchanger) and wet soap flow set-point. As such it is a hierarchical SISO control problem, because the control signal generated by GPC is not directly implemented

³ Soap moisture would normally be checked every half an hour.

but forms the set-point for other inner loops controlled by PID's (see Fig. 4.3).

4.3 HARDWARE AND SOFTWARE

The most basic requirement of any automated control strategy based upon feedback is that of a reliable, accurate on-line measurement of the output that is to be controlled. For the process described above this measurement is the percentage-by-mass soap moisture content of the soap as it leaves the dryer. During the trials this measurement was provided by a sensor located just prior to the extrusion stage. This 'strip-line' meter was an experimental model under long-term evaluation and was thus connected to a Commodore PET micro-computer for display, logging and calibration purposes. It operated by passing a microwave signal, in the 10Ghz region, through a U-shaped conductor in continuous contact with the soap. In principle the amount of power lost is proportional to the moisture content of the material sliding over the sensor. Whilst this should be independent of the type of soap base being monitored in practice the fat content of different soaps was found to affect the measurement. This was overcome by a program running on the PET that allowed rapid recalibration of the meter for the various bases simply by selection of the appropriate code. Thus the validity of the moisture measurement could be preserved throughout a change in the soap base.

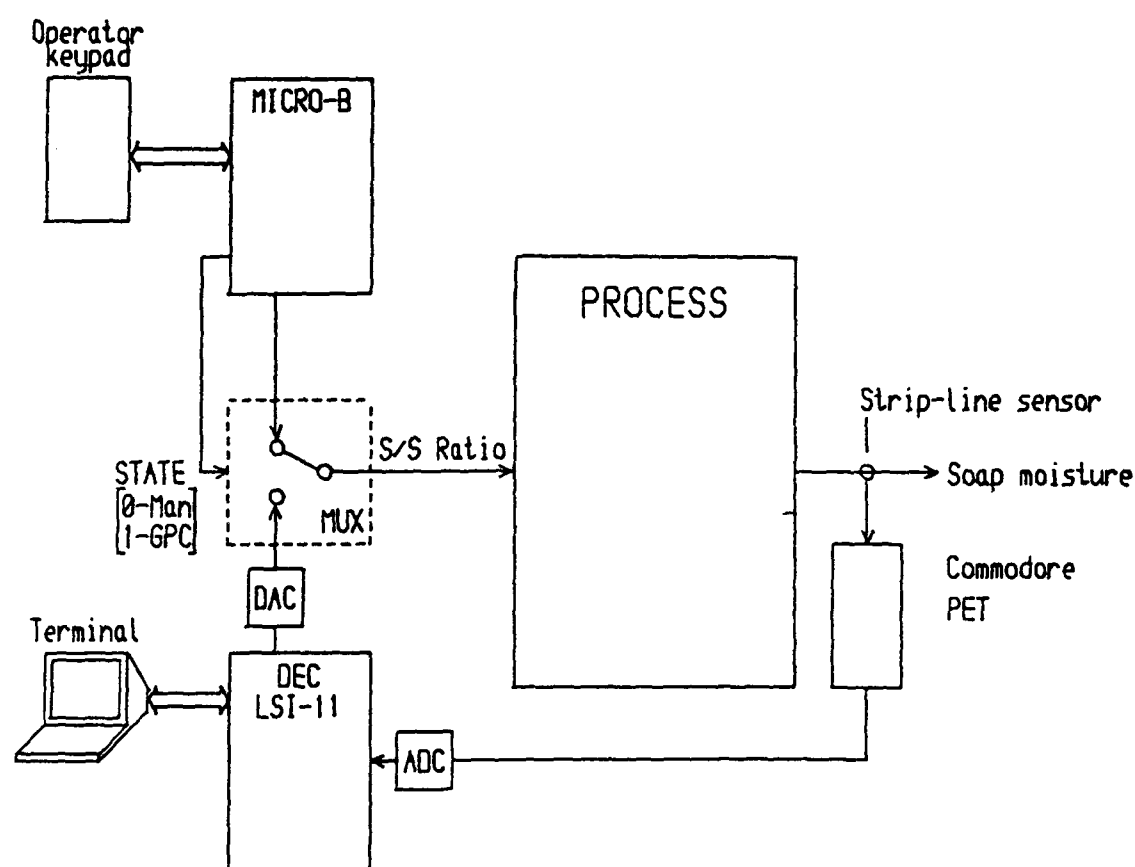


Figure 4.4: Schematic diagram of the hardware.

The apparatus used throughout the trials consisted of a Digital Equipment Corporation DEC LSI-11/73 with a RT-11 real time operating system. Analogue signal interfacing was provided by standard ADC and DAC boards (Analogue Devices RTI-1250 and RTI-1252 respectively) with 12-bit resolution. These were configured for operation over the range 0-10V in order to be compatible with the Lever Brothers hardware. This was linked into the existing control system by means of a simple hard-wired analogue multiplexer arrangement within the MICRO-B, shown schematically in Fig. 4.4. Changing the state of the multiplexer determines whether the manual input, or the control signal calculated by GPC, is applied as the steam/soap ratio.

A single-input single-output implementation of the Generalised Predictive Control algorithm, as described in Chapter 2 and embedded in the FAUST package (Tuffs, 1985; Tuffs & Clarke, 1985), formed the software element used during the trials. FAUST is a package developed by OUEL self-tuning group as a successor to the previous CONTROL BASIC package of Hodgson (1982). Written in FORTRAN IV for machine independence, it provides an adaptive control environment with features such as process simulation, analogue I/O, display handling and data logging. Thus it may be used for both general purpose evaluation of adaptive control algorithms and experimental real-time application.

4.4 EXPERIMENTAL RESULTS

The results presented in this section describe a series of experiments that were performed on the control of a soap dryer using the Generalised Predictive Control algorithm. Where appropriate these results are placed in the order corresponding to the experimental procedure used throughout the actual trials. Thus the early sections describe the procedure used to commission the GPC controller and actually obtain on-line control of soap moisture by manipulation of the steam/soap ratio. This used the method proposed in Chapter 3 where, after verifying the validity of the strip-line moisture measurement (section 4.4.1), a step test was performed (section 4.4.2) to obtain preliminary dynamic information necessary for the specification of the algorithm followed by a brief period of open-loop PRBS assisted self-tuning (section 4.4.3) before initial closing of the loop. It was found that, despite severe production constraints, this straightforward method repeatedly provided successful commissioning.

Upon achieving satisfactory closed loop control the majority of the trials consisted

of long-term regulation of soap moisture content (section 4.4.4). Due to the production commitments of the soap dryer the range of formal experiments that could be conducted was naturally limited. Thus subsequent closed loop experimental procedure consisted of observing the quality of GPC regulation during routine process activities such as changes in the soap flow rate (section 4.4.5) and, more interestingly, a soap base change (section 4.4.7). Nonetheless it was possible to conduct some limited servo control trials whilst keeping the moisture within production limits (section 4.4.6) and also to observe the effect of the T polynomial upon closed loop control activity (section 4.4.5).

Later sections describe some 'unscheduled' results obtained during the trials, which illustrate the effect of practical problems such as temporary process failure (section 4.4.9) and saturation of the PID-controlled inner steam flow loops upon GPC control (section 4.4.8).

4.4.1 Verification of the on-line moisture measurement

Due to the experimental nature of the strip-line moisture sensor it was decided to conduct a preliminary trial to verify the validity of the on-line measurement. Thus the unfiltered output from the strip-line sensor was logged every 10 seconds and compared to an off-line analytic measurement taken manually every 10 minutes.

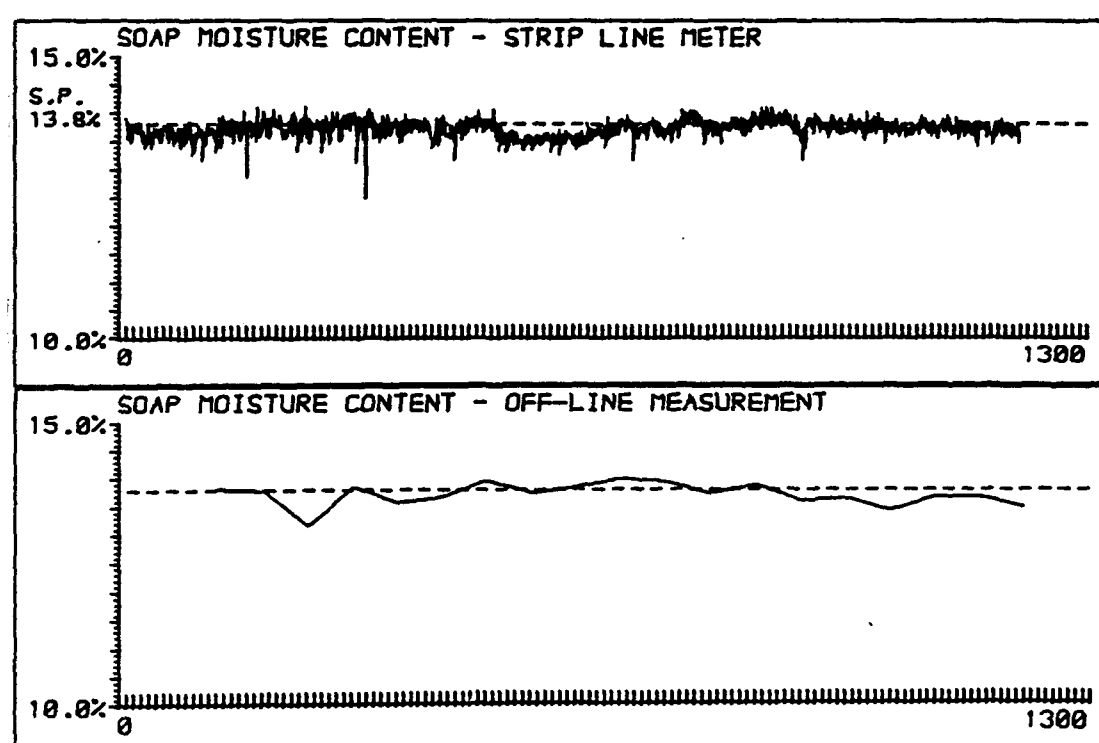


Figure 4.5: Verification of the on-line moisture measurement.

The results of this experiment are plotted in Fig. 4.5 with the estimated mean and variance of both signals given in Table 4.1. Using a simple statistical test based upon the 'normal' distribution, it was calculated that the mean value of each measurement was

the same with almost 80% confidence. Although the strip-line moisture measurement was relatively noisy it was nonetheless considered adequate for use during the experiments.

Comparison of moisture measurements		
	Mean (%)	(rms,%)
Analytic (Off-line)	13.72	± 0.204
Meter (On-line)	13.69	± 0.568

Table 4.1: Verification of ON-LINE moisture measurement.

4.4.2 Observation of the process step response

The application of an adaptive algorithm such as GPC relies upon a limited amount of prior process information which is used by the commissioning engineer in the specification of basic controller parameters such as the sample interval, and in the choice of appropriate filtering. The two most important pieces of information required are approximate values for the dead-time and rise-time of the plant so to obtain these for the soap drying process a simple experiment was conducted. This consisted of applying a series of manual step changes in the input steam/soap ratio, at a typical operating point, whilst observing the moisture output with a sample interval of 20 seconds. The magnitude of these step inputs were chosen large enough to cause a noticeable change in the moisture content but sufficiently small that the soap moisture remained within production tolerances. Note that this test was conducted after control had been transferred to the FAUST package via the multiplexer.

Detailed examination of the step responses plotted in Fig. 4.6 reveal that the soap drying process may be characterised by a dead-time of approximately 120 seconds and a rise-time of about 60 seconds. This relatively large dead-time, twice as long as the rise-time, is mainly due to the *residence time* of the soap in the drying chamber i.e. soap that is already inside the chamber at the time of the input change is not affected by the increased heating effect at the heat exchanger stage. Note that as expected an increase in the steam/soap ratio leads to an increased drying effect and thus the final moisture content is reduced, i.e. the process has negative gain.

On the basis of these observed approximate dynamics the choice of sample interval, T_s ,

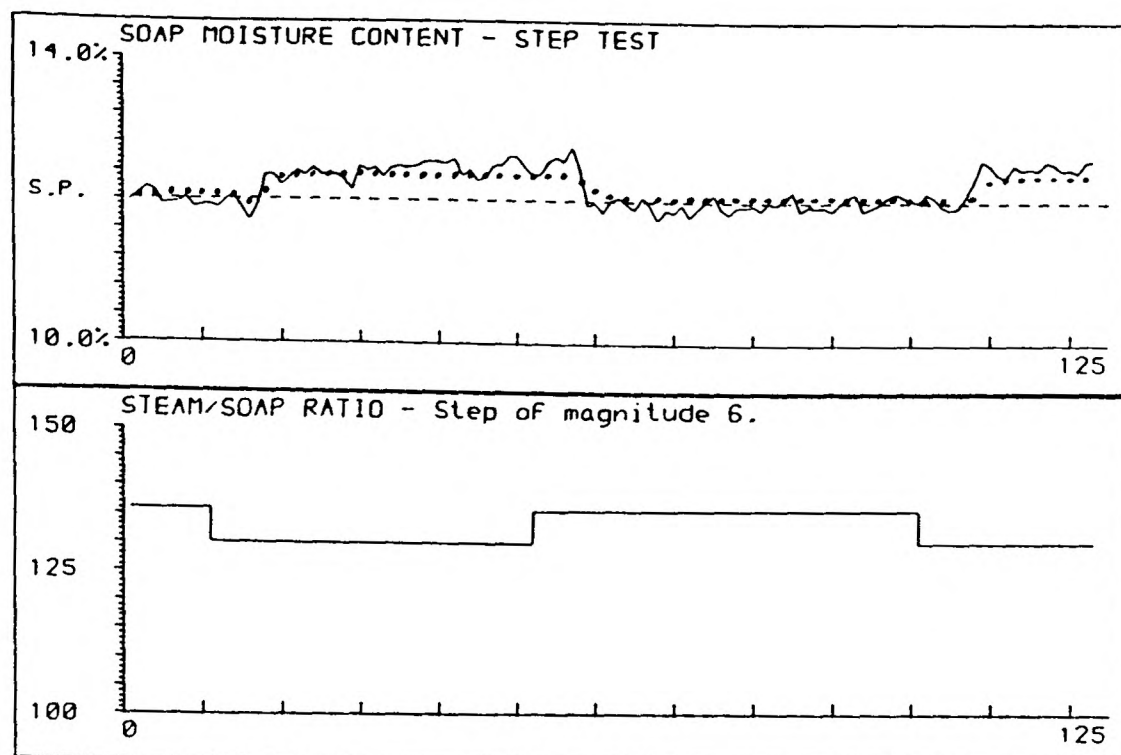


Figure 4.6: Step response of the soap drying process.

may be made. In principle it should correspond to 1/10th of the dominant process time constant (Clarke, 1981; Wittenmark & Åström, 1984) to ensure that the discrete CARIMA model (3.3.1), upon which GPC is based, is capable of accurately modelling the dynamics of the system. However in the presence of time delay of comparable magnitude to the process time constant this can lead to an unrealistic parameterisation of the estimated model (see Chapter 3). Thus in the case of the soap dryer the 'ideal' choice of $T_s = 6$ seconds would require an assumed time delay, $\hat{k}$, of about 18-20 samples and/or the estimation of large numbers of $\hat{b}$ parameters. This is undesirable in practice because the former implies accurate prior knowledge of the actual time-delay and the latter because in the extreme it can degrade the performance of the estimator.

Parameterisation	
Degree of $\hat{B}(q^{-1})$	5
Degree of $\hat{A}(q^{-1})$	2
$\hat{k}$ (samples)	5

Table 4.2: Parameterisation of the estimated model.

The alternative was to sample more slowly and thereby trade off model performance (and control performance) for simplification of the parameterisation, with an additional benefit expected to be increased controller robustness (Rohrs *et al*, 1984). Thus a com-

promise value for the sample interval was chosen as $T_s = 20$ seconds corresponding to sampling approximately three times as fast as the process time-constant and allowing a reasonable parameterisation as shown in Table 4.2.

This parameterisation follows both from observation of the process response and the chosen sampling time, where :

- A second order model was assumed to provide a reasonable description of the process (Box & Jenkins, 1976)
- The assumption of a minimum time-delay, $\hat{k}$, of 5 samples thus corresponded to 100 seconds.
- An extended B polynomial was estimated to incorporate any additional time-delay, and the presence of zeroes due to sampling. The presence of any excess time-delay would thus be incorporated by the early $\hat{b}$ parameters going to zero.

Note that to prevent aliasing a digital filter of time-constant twice that of the sample interval, i.e. $T_f = 40$ seconds, running on the MICRO-B at a high sample rate, was specified for the moisture signal and used throughout all the subsequent experiments.

4.4.3 Identification of the process using PRBS

The next stage in the application of the GPC adaptive control strategy was the on-line identification of a process model by use of a recursive parameter adaptation algorithm. This is the so-called 'self-tuning' phase that distinguishes adaptive control from controllers based upon off-line analysis and design. The parameter adaptation algorithm used in this, and all the subsequent work, is the well known *recursive least squares* algorithm, modified for practical application by incorporation of the variable forgetting factor of Fortescue *et al* (1981). This type of estimator, and its motivation, is described in detail in section 3.4.2 of Chapter 3. Its specification during the experiments is shown in Table 4.3.

As the identification is to start from a position of complete ignorance of the process the initial P matrix, $P(0) = I$, was chosen as a relatively high value to provide active initial estimation. The specification of the A.S.L or 'data-window' and Σ_0 , which determine the performance of the variable forgetting factor, are such that subsequent adaptation should be low-gain and cautious. A small estimation dead-band, approximately equal to the r.m.s.

Estimator specification	
$P(0)$	I , the identity matrix.
A.S.L.	100
Σ_0	100
Deadband	± 0.5

Table 4.3: Specification of the estimator.

variation of the soap moisture measurement at steady state, was implemented to reduce the effect of noise upon the parameters during subsequent regulation.

In order to provide 'exciting' data to assist the identification the process output was perturbed about its normal operating point by a PRBS⁴ signal of magnitude ± 3 units superimposed upon the otherwise constant manual input. With the estimation algorithm running this PRBS signal was used to probe the plant for about 200 samples (see Fig. 4.7). During this period the soap moisture is clearly perturbed but retains its average value and, as desired, never went beyond production tolerances.

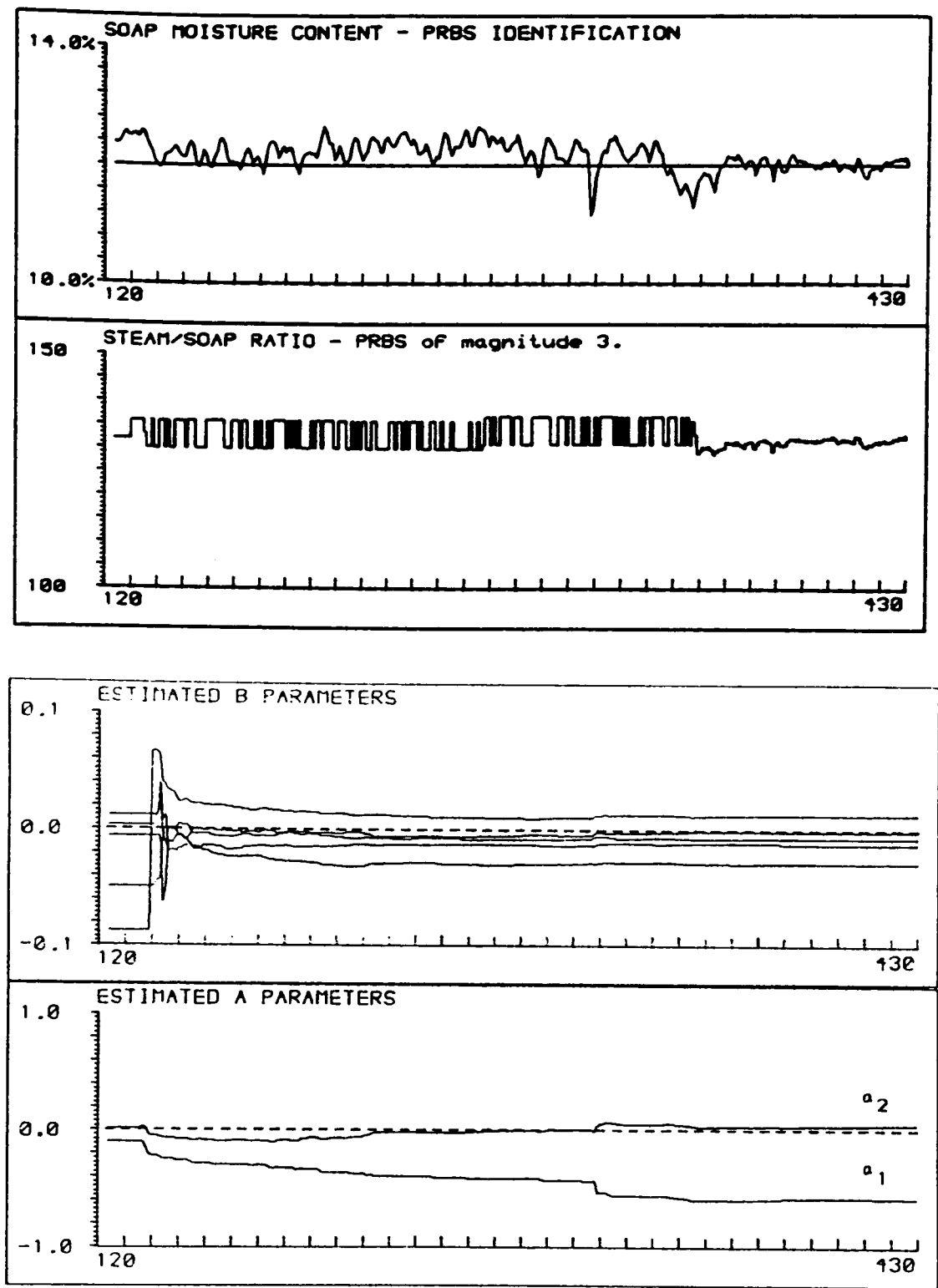
After this period of preliminary identification the estimated parameters were judged to have converged so the loop was closed. The self-tuned GPC controller then quickly removed the offset due to the incorrect manual choice of steam/soap ratio and provided satisfactory regulation for the remainder of the experiment. The evolution of the estimated parameters is also plotted in Fig. 4.7. After an initial rapid adaptation the parameters quickly converge and remain relatively constant throughout the remainder of the trial.

In order to check the validity of this estimated model the step response test of (4.4.2) was repeated for a simulated model using the estimated parameters at the end of the experiment, i.e.

$$\Delta y(t) = \frac{10^{-2} q^{-5} (1.302 - 0.632q^{-1} - 2.817q^{-2} + 1.146q^{-3} - 0.096q^{-4})}{1 - 0.582q^{-1} + 0.046q^{-2}} \Delta u(t-1) \quad (4.1)$$

This is shown superimposed upon the 'real' results in Fig. 4.6 with the simulated response shown dotted. It is clear that the performance of the estimated model corresponds very closely to that of the actual soap dryer, with a time-delay of 100 seconds and a time-constant of approximately 60 seconds.

⁴an 8-bit Pseudo Random Binary Sequence.

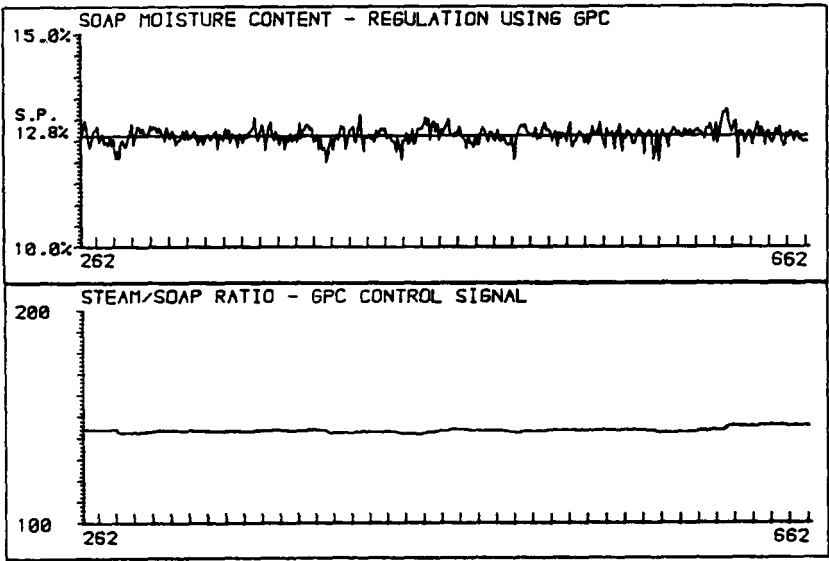


Experiment Synopsis		
Measurement	Units	Value
Soap Base	#	6
Soap Flow	t.p.h	3.8
Set Point	%	12.6
Manual input	ratio	132.0

Figure 4.7: Process identification using a PRBS input.

4.4.4 Regulation of soap moisture

For the majority of chemical processes the control problem consists primarily of regulating the output measured variable to a constant value for a long period of time. Control of the soap dryer is no exception with normal operation involving regulation of the soap moisture to a given set-point for the duration of that particular batch of soap. Thus the ability to perform long-term, offset-free regulation in the presence of noise and unmeasurable load disturbances is one of the most important requirements for a practical adaptive controller.



Experiment Synopsis		
Measurement	Units	Value
Soap Base	#	6
Soap Flow	t.p.h	3.5
Set Point	%	12.8
Manual input	ratio	144.0

Figure 4.8: Soap moisture regulation using GPC.

In this experiment the regulation properties of the adaptive GPC algorithm were examined over a period of 400 samples (i.e. over 2 hours) where the moisture was to be controlled to a set-point of 12.6%. Rather than repeating the self-tuning phase of the previous section, the estimated model was initialised with the parameters given in (4.1) and the corresponding values of the final covariance matrix, P . The specification of the GPC design parameters were as given in Table 4.4 and thus corresponded to the ‘default’ algorithm described in Chapter 2. Parameter estimation was enabled throughout the entire experiment.

GPC Specification			
Feature	Default	Feature	Default
NY	15	$P(q^{-1})$	1
NU	1	$T(q^{-1})$	$1 - 0.8q^{-1}$
NS	1	λ	10^{-6}

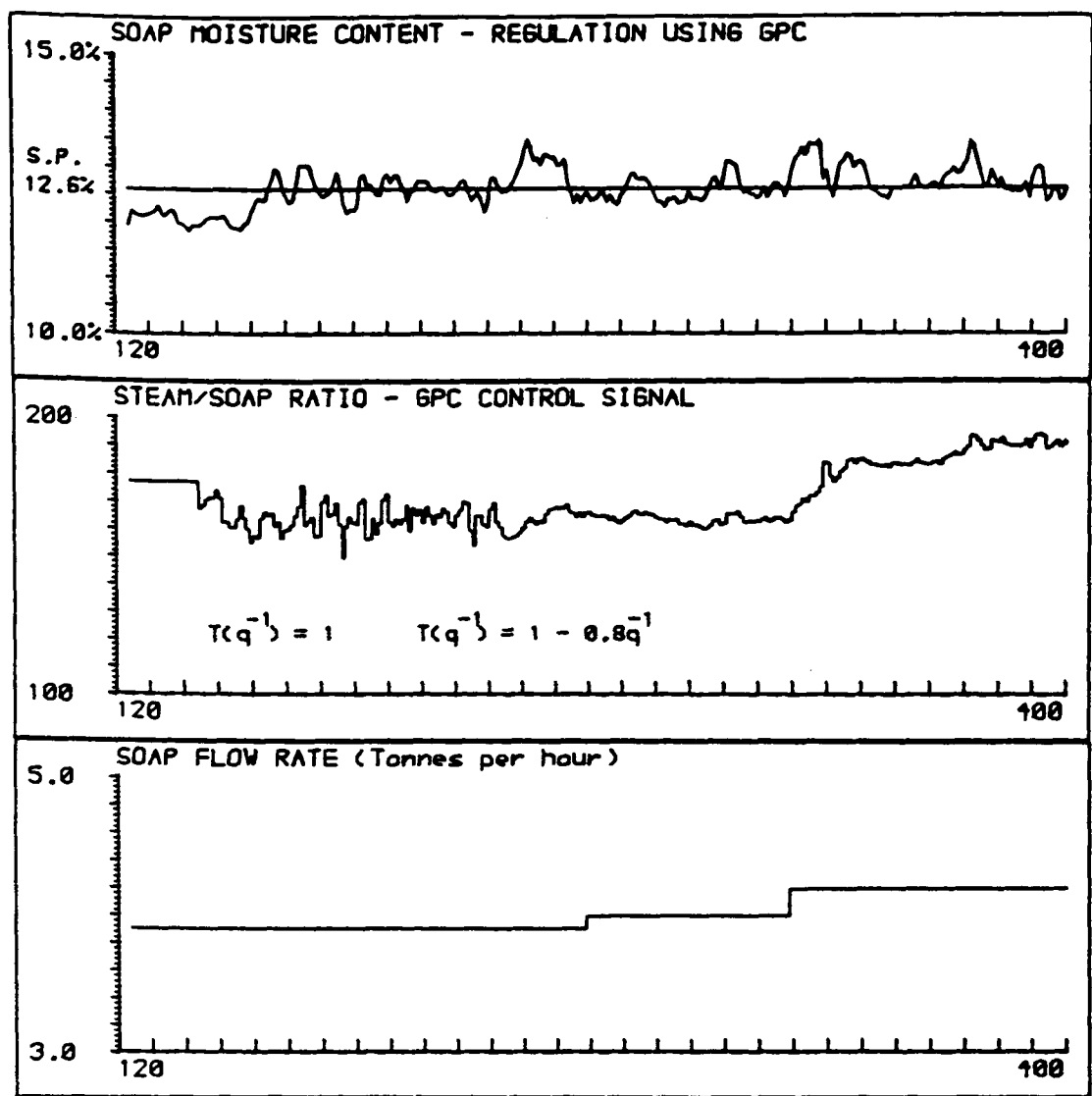
Table 4.4: Commissioning of the GPC algorithm.

The results of this regulation experiment are plotted in Fig. 4.8 and show a prolonged period of satisfactory regulation of the soap moisture content. Despite a noisy measurement the GPC control signal is relatively smooth and the moisture content successfully regulated without an offset. This latter observation, combined with the initial control upon closing the loop in the previous section (Fig. 4.7), clearly demonstrates the inherent integral action of the GPC control law.

4.4.5 Regulation during changes in soap throughput

A common problem in the regulation of this type of unit operation is the maintenance of the desired moisture content despite changing process throughput. Realisation of this problem had led to the current manual specification, not of steam flow-rate, but of a steam/soap ratio as the means of obtaining a desired soap moisture content. Thus this may be regarded as a static feedforward control strategy where the gain of the feed-forward term is manually fixed by the operator. Increase in the soap flow into the process would therefore result in extra steam entering the heat exchangers and thus, in principle, the drying effect would be maintained despite the increased load.

In the experiment conducted in Fig. 4.9 the effectiveness of such a feed-forward strategy, within the overall GPC control structure, is examined. Once again the GPC controller was commissioned with the initial parameters of (4.1) and with the controller specified as in Table 4.4 although initially with $T = 1$. After a period of regulation during which control activity was judged to be too active this was changed to $T = 1 - 0.8q^{-1}$ resulting in significantly smoother control activity. At time $t = 260$ the input soap rate was deliberately changed from 3.9 t.p.h. to 4.0 t.p.h. with no observable effect upon the moisture content. A further increase to 4.2 t.p.h. at time $t = 320$ however required a substantial increase



Experiment Synopsis		
Measurement	Units	Value
Soap Base	#	3
Soap Flow	t.p.h	3.9
Set Point	%	12.6
Manual input	ratio	177.0

Figure 4.9: Regulation in the presence of changing soap throughput

in the steam/soap ratio calculated by GPC in order to preserve regulation. This clearly shows that in this case the GPC controller is acting as a *trim* to the fixed internal ratio control.

Note that compared to the previous experiments the moisture control obtained during this experiment was significantly worse. There are several possible explanations for this; including a fault in the strip line meter, the use of a different soap base, poor estimation or even a ‘bad boil’⁵ of soap, the latter being a relatively common occurrence according to the operators. Unfortunately the nature of such a practical trial makes it virtually impossible to say which, if any, of the above was the cause. Nonetheless the benefit of the *T* polynomial under such circumstances is clearly illustrated with its use preventing the GPC control from responding with excessive control activity.

⁵operator terminology!

There is also an important observation to be made from the results at the beginning of this experiment. During the initial period of manual control the soap moisture content is well below the set point and consequently energy is being wasted in the excessive drying of the soap. However the moment GPC control is applied it immediately decreases the steam/soap ratio to a value that produces the correct steady state performance. This saving of energy (corresponding to a reduction of approximately 20% in steam flow) clearly illustrates the benefits of using automatic control.

4.4.6 Servo control of soap moisture

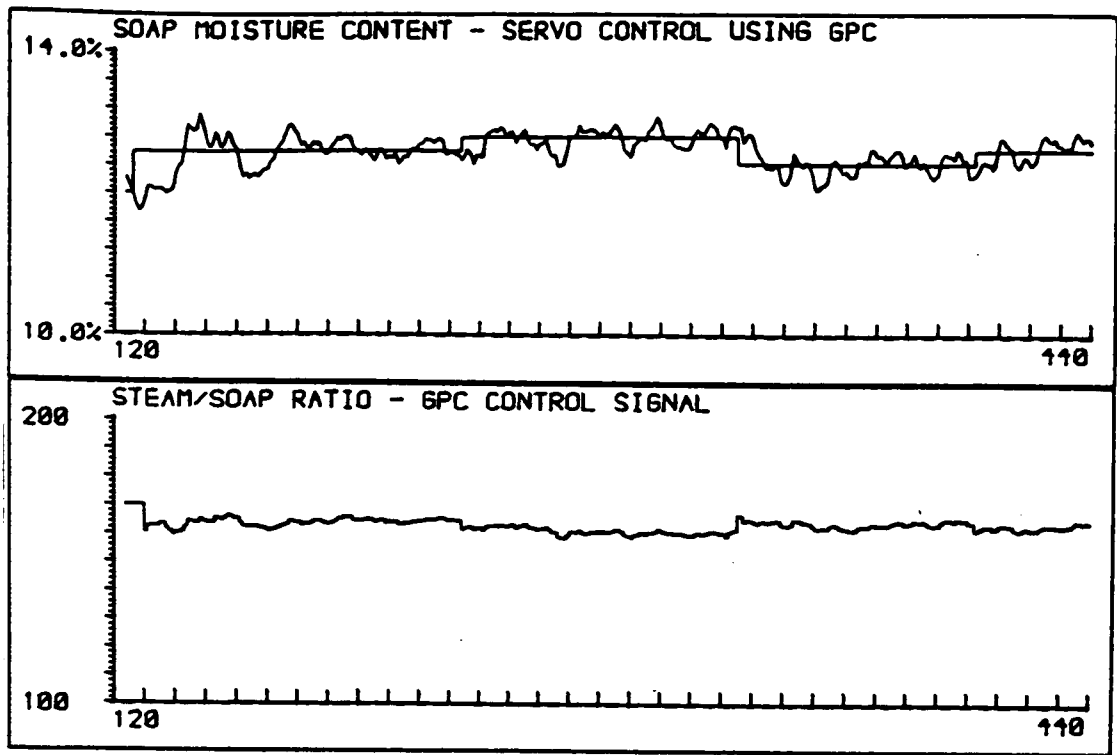
After regulation the ability of a control algorithm to follow changes in the set point is the next most important practical requirement. Thus this section describes an experiment where the *servo* performance of the GPC algorithm was evaluated by implementing a series of changes in the moisture content and observing the controlled response (Fig. 4.10). It must be stressed that there is no practical requirement for changing the moisture set point during normal production for this particular process, the set point only being altered in the event of a soap base change. Throughout this trial GPC was commissioned exactly as in the previous section and estimation enabled.

Unfortunately strict production limits meant that the set point could only be changed from a minimum of 12.4% to a maximum of 12.8%. Because of the large amount of noise present this limited the effectiveness of the trial but nonetheless it is possible to observe adequate set point following with corresponding control activity.

4.4.7 Control of a soap base change

One of the most severe problems associated with the control of soap moisture in this batch type of drying process occurs during soap base changes. During this operation the input to the dryer is suddenly switched from an exhausted vat to a fresh vat containing the new molten soap batch to be dried. If this new batch consists of a different soap base then a new value of soap moisture may be required. Hence soap produced during the transition period between switching over and achieving the new set point could be outside the production constraints outlined in section 4.1 (see Fig. 4.1). Thus it is important that this transition be made as quickly and as accurately as possible.

As these soap base changes occur infrequently and, to some extent, unpredictably, it



Experiment Synopsis		
Measurement	Units	Value
Soap Base	#	3
Soap Flow	t.p.h	4.0
Set Point	%	12.6 $\pm$ 0.2
Manual input	ratio	177.0

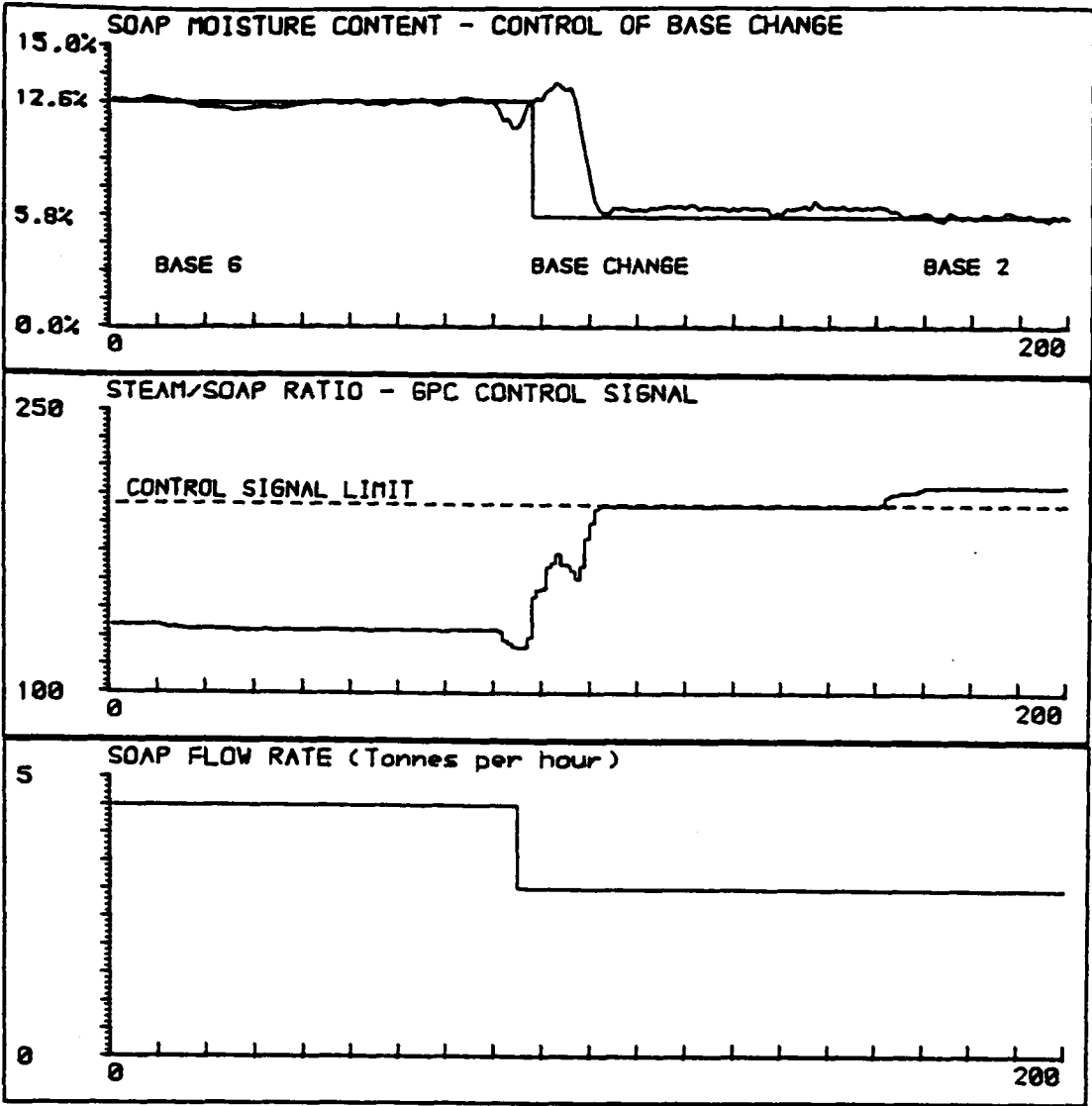
Figure 4.10: Servo control of soap moisture.

was fortuitous that one occurred during a period of GPC closed loop control. However it is a measure of operator confidence in the performance of GPC that they allowed closed loop control during such a critical period. Fig. 4.11 is a record of the control performance during a soap base change from Base 6 to Base 2⁶, where the input soap flow and moisture set point were simultaneously changed to 3 t.p.h. and 5.8% respectively. The strip-line meter was automatically recalibrated for this new type of soap as described in section (4.3). At the time that it occurred the GPC controller was running with the standard design specified in Table 4.4 and estimation was enabled.

After an initial transient drop in the moisture content due to the method of changing the soap base the GPC controller succeeds in bringing the soap moisture content close to the new set-point in about 6 minutes. The control signal then reached an operator imposed saturation limit (200 units) resulting in a small offset remaining upon the soap moisture. This could only be removed by a slight relaxation of the saturation limit to 210 units.

This result is of great practical interest as it demonstrates excellent control of the soap moisture under conditions of extensive process variation. Ignoring the slight offset due to

⁶'Lux' toilet soap and 'Lux' flakes respectively.



Synopsis of Base-Change			
Measurement	Units	From	To
Soap Base	#	6	2
Soap Flow	t.p.h	4.5	3.0
Set Point	%	12.6	5.8

Figure 4.11: Control during a change in soap base.

saturation the speed with which the set point of the new batch was accurately achieved was about three times faster than that routinely obtained by manual control.

4.4.8 The effect of process saturation

In contrast to the previous experiment showing the successful application of GPC the results presented here describe the effect upon GPC of a classic control problem, namely saturation. This was encountered in the previous section where after the soap base change the moisture set point could not quite be achieved despite maximum control effort.

The period of control shown in Fig 4.12 begins with an initial period of operator manual control where, despite application of the ‘appropriate’ steam/soap ratio the moisture content ($\approx 13.25\%$) remained consistently well above the set-point (12.6%). Even when fixed parameter GPC control was applied the control signal merely ramped up and saturated at

a steam/soap ratio of 210 units without any observable effect upon the output moisture content. The reason for this behaviour becomes clear if one examines the lowest trace of Fig 4.12 which shows the set-points and actual values for both of the heat exchanger steam flows. Throughout all of this period of unsatisfactory control neither of the inner steam flow loops were achieving their set-points. Both of the local PID controllers had themselves saturated and the process had literally 'run out of steam'. It was not possible to log these variables on-line and therefore manual inspection was carried out every 10 minutes.

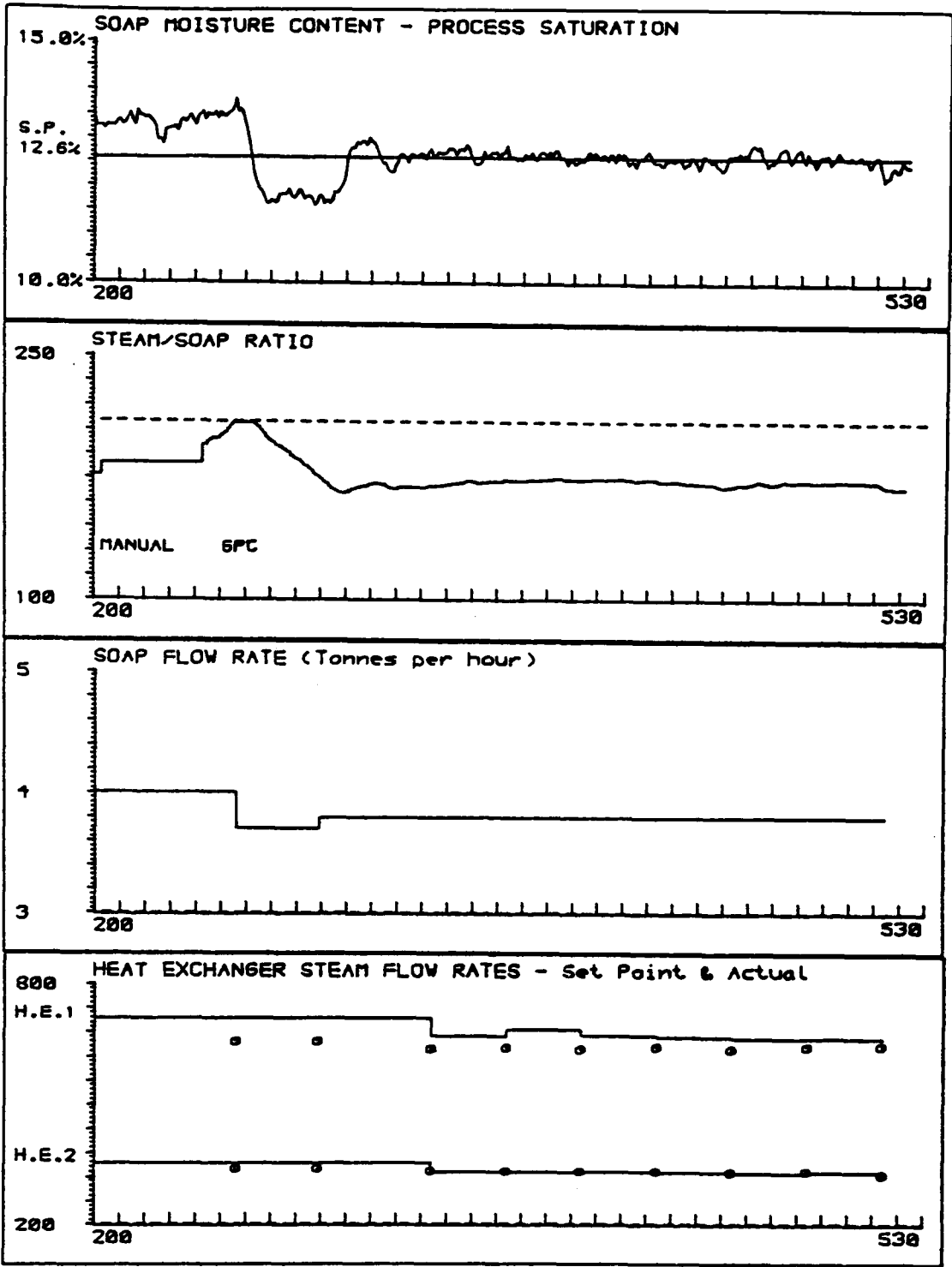
Based upon this observation the soap flowrate input to the process was reduced at time $t = 255$ from 4 t.p.h. to 3.7 t.p.h. This immediately caused an excessive drop in the soap moisture to 11.7%, where it remained despite a gradual reduction in the GPC control signal. This further failure of the steam to respond to the GPC applied steam/soap ratio could be explained if the inner loop PID controllers were implemented without any form of integral desaturation. Eventually, after a further change in the input soap flowrate to 3.8 t.p.h. the steam flow may be seen to begin responding and moisture is restored to the set-point. This is followed by a long period of good, problem-free regulation.

This result is important because it shows a significant problem affecting the successful application of an adaptive controller in practice, especially its application in the hierarchical configuration of Fig. 4.3. This is the problem of *hidden* actuator saturation. In this example there were many periods when although the steam/soap ratio calculated and output by GPC had not reached its maximum allowable value saturation of the inner loops had already occurred. Thus further increase in the steam/soap ratio had no effect which could easily be misinterpreted by the estimator as the process gain going to zero, an error that could lead to instability of the closed loop system.

4.4.9 The effect of temporary process failure

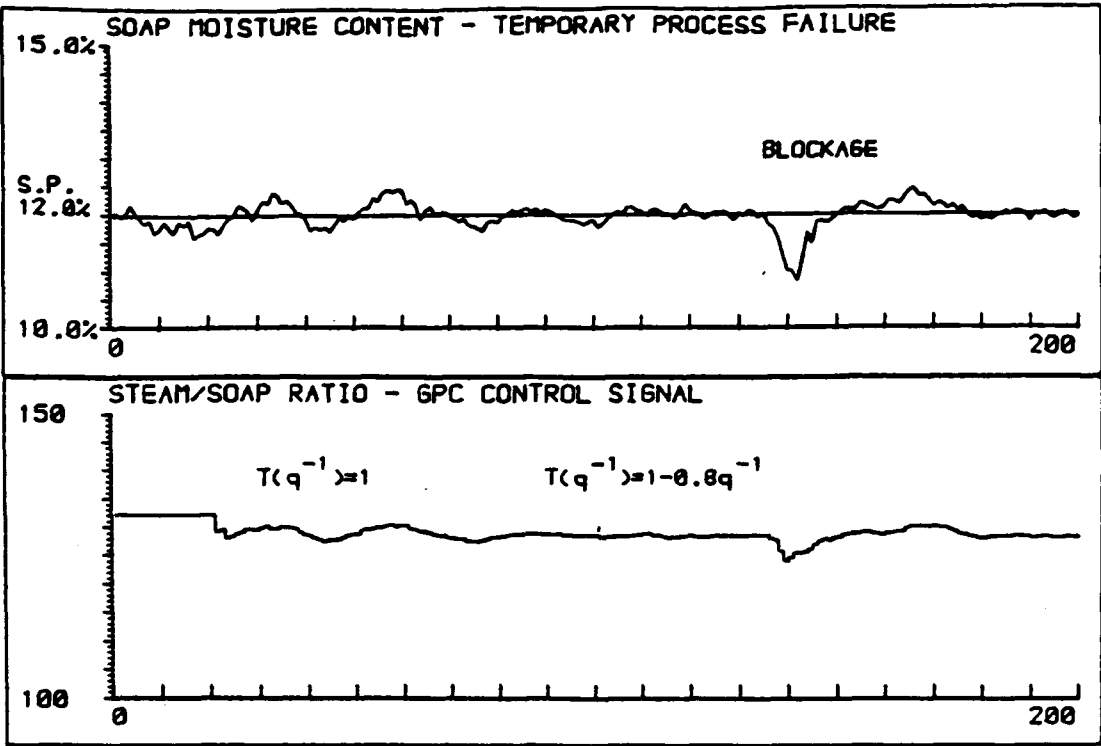
The result described in this section, as in the previous, occurred during a period of apparently routine closed loop GPC regulation of the soap moisture just previous to the soap base change of (4.4.7). It is included here because it demonstrates the effect of T polynomial upon both normal closed loop performance and the rejection of a load disturbance caused by an unscheduled plant failure (Fig. 4.13).

A T polynomial of $T = 1 - 0.8q^{-1}$ had been specified at time $t = 60$ and had successfully reduced the previously excessive control signal activity which in turn had led to quite



Experiment Synopsis		
Measurement	Units	Value
Soap Base	#	3
Soap Flow	t.p.h	4.0
Set Point	%	12.6
Manual input	ratio	177.0

Figure 4.12: The effect of process saturation.



Experiment Synopsis		
Measurement	Units	Value
Soap Base	#	6
Soap Flow	t.p.h	4.5
Set Point	%	12.0
Manual input	ratio	132.0

Figure 4.13: GPC control during a temporary process failure.

dramatically improved regulation. Then at about time $t = 120$ a blockage occurred that prevented the soap ‘noodles’ from falling into the hopper below, threatening a spillage. This problem was remedied by the operator temporarily switching off the pumps governing the soap flow, thus causing the soap flow to cease whilst the blockage was cleared. Because the soap currently in contact with the strip-line meter was stationary this blockage does not immediately show up in Fig. 4.13 the meter reading a relatively constant value. In addition the molten soap that had just entered the heat exchangers prior to the suspension of soap flow was subject to heating for the entire duration of this stoppage.

Thus when the process was restarted, and after a period of time corresponding to the residence time of the dryer, this soap emerged after effectively being dried for about three times as long as normal. This caused a large sudden drop in the measured soap moisture appearing at time $t = 140$, i.e. a load disturbance. Although initially responding with a reduction in the steam/soap ratio the GPC control action succeeds in restoring the moisture of the soap to the set-point within 10 samples with no apparent degradation in control performance.

Thus this result once again demonstrates the integral action of the GPC controller and

the effect of the T polynomial in preventing excessive control activity during the rejection of this 'apparent' load disturbance. This is important because it reduces potential over-compensation when the process itself recovers from the effects of the temporary stoppage and begins to return to normal operation.

4.5 CONCLUSIONS

The trials described in this chapter represent one of the first industrial applications of the promising long range prediction based GPC control algorithm. Of particular importance is the fact that these trials were all conducted during actual full scale soap manufacture. Whilst this limited the range and content of the experiments that could be conducted it also meant that the emphasis was upon the achievement and evaluation of routine closed loop control using GPC under the rigour of operating conditions that would typically be encountered during the *commercial*, rather than *experimental*, application of such an algorithm.

The Generalised Predictive Control algorithm used in these trials was described in Chapter 2 with most of the results obtained using the so-called 'default' values given in Table 4.4 which were suggested as being suitable for the control of relatively slow chemical processes such as the soap dryer. Thus the prediction horizon ($NY = 15$) was 'long' and the control horizon ($NU = 1$) was 'short' with the control performance expected to tend towards mean level control.

In terms of evaluating the GPC controller in practice the trials on the soap dryer provided some very promising and interesting results, such as the accurate control of soap moisture through the critical period of a change in soap base, and closed loop control during a temporary process failure. Thus not only were the potential benefits of the algorithm clearly demonstrated but also some typical problems that threatened its successful application.

The commissioning of the adaptive GPC controller was shown to be both rapid and straightforward if based upon the guidelines given in section (4.4.2). Thus after the performance of a simple step-test, in order to assist specification of the controller and estimator, only a brief period of open loop self-tuning with a PRBS input was required before the loop could be closed and satisfactory moisture regulation obtained. Equally important is that this commissioning phase was successfully conducted without interfering in any way

with soap manufacture.

Subsequent trials showed that GPC was capable of maintaining accurate regulation of the soap moisture, with no offset, for long periods and despite rather noisy output measurements, changes in input soap flow and even large disturbances caused by the temporary stoppage of the drying process. The consistently offset-free nature of the regulation⁷ clearly confirms the inherent integral action of the GPC control law, and is a distinct improvement over that commonly obtained with a manual input updated about every half an hour. Satisfactory regulation was in fact obtained for a total period of 5850 samples (≈ 32 hours) with the longest continuous period being about 1500 samples (≈ 8 hours). Production commitments prevented the testing of GPC regulation over a longer period.

The control of a soap base change in 4.4.7 was very successful and, for this process, represents the major justification for the use of automatic control. The simultaneous changeover of input soap base, input soap flow rate and desired moisture set point was accomplished far more quickly than was previously possible with manual control with a corresponding reduction in the amount of soap wasted. In addition it was a measure of the operators confidence in the ability of the GPC controller that they allowed the test to take place during such a critical period of production.

Finally the importance of the T polynomial, outlined in section (2.3.3), was verified in practice where the presence of large measurement noise, unmeasurable load disturbances and (almost certainly) significant model-plant mismatch sometimes led to relatively poor GPC performance. The specification of T as a low pass filter was seen to improve considerably the quality of control under these circumstances by reduction of unnecessary control signal activity. This result lends practical credence to the interpretation of the T polynomial as a means of increasing the robustness of the GPC control algorithm.

Of equal importance to the above results was the observation of some typical problems that the application of adaptive control alone cannot immediately overcome. In this particular application to soap drying the success of the GPC controller was hampered by the relatively *ad hoc* manner of its implementation in the control structure (see Fig. 4.3) due mainly to Lever Brothers hardware constraints. Thus the unavailability of important inner loop signals (i.e. steam flowrates) combined with the routine operation of the process at, or near to, maximum capacity resulted in actuator saturation that was 'hidden' from the GPC algorithm (section 4.4.8). This prevented GPC control from achieving the set-point

⁷the period of offset in 4.4.7 being due to process saturation.

and was a serious threat to the performance of long term parameter adaptation. A more practical implementation could overcome these problems by the provision of the appropriate measurements so that, for example, estimation could be suspended during process saturation or temporary stoppages. These are typical *engineering* modifications that would be required before GPC would become suitable as a practical control engineer's tool.

To summarise the results obtained from the industrial trials described throughout this chapter it is claimed that all the objectives presented in (4.1) were successfully achieved, i.e.

- The closed loop control performance of the long range predictive GPC algorithm was applied for a substantial period of full scale industrial soap dryer operation and shown to provide consistently high quality of control despite the associated rigours and limitations of these conditions.
- The feasibility of using adaptive control was demonstrated in practice, with the emphasis upon the self-tuning aspect where the GPC controller rapidly achieved satisfactory control without the need for detailed *a priori* process knowledge.
- The use of the modern GPC control strategy was found to offer significant benefits over the current manual control of soap moisture content, namely the provision of long term offset free regulation and tighter control during soap base changes.

Chapter 5

Multivariable GPC control of a spray drying tower

This chapter describes experiments performed from September to November 1986 on the second part of the collaborative project between Oxford University Self-Tuning Control Group and Unilever Research. This work involved the application of the Generalised Predictive Control algorithm to the multivariable inner loop control of an industrial spray drying tower: the first ever practical application of a full multivariable GPC design.

5.1 INTRODUCTION

The spray drying tower at Port Sunlight is a smaller version of production spray dryers located at Warrington and is used periodically for the drying of experimental batches of soap. The moisture content of the soap leaving the tower depends upon the amount of hot air flowing through the tower (i.e. the drying medium) and the tower pressure. Because the soap is produced for research purposes the accuracy of its moisture content is particularly important and thus tight control of hot air flow and tower pressure is critical. This is complicated by the fact that these two inner loops exhibit a significant degree of interaction and thus represent a *multivariable* control problem.

During normal operation the control of these two inner loops is performed by a multi-loop PID system. In the presence of interaction this has two potential disadvantages: it requires *a priori* knowledge of the appropriate input–output pairing, and its ‘decoupling’ performance (i.e. the extent to which it can prevent one loop affecting another) depends upon the load disturbance rejection properties of individual loops. Thus the principal

objective of this application was to demonstrate that this is not the case with full multivariable GPC ('MGPC') because of its basis upon a full multivariable model of the process. This was hoped to provide an overall improvement in control over that of the conventional multi-loop PID, with the emphasis on improved decoupling of the two loops, and to do this with no prior assumption about input-output combinations.

For the spray drying tower the correct input-output pairings are apparent, hence a further series of tests were conducted using alternative simplified controllers which again used GPC but in a *multi-loop* control structure ('DGPC') similar to that of the original PID control system. The aim here was to observe the effectiveness of feed-forward for decoupling of a multivariable process, to compare this with the current PID multi-loop control and that of MGPC, and thus to determine whether this type of control structure offers a viable alternative to the relative complexity of the full multivariable algorithm for processes where input-output pairing is not a problem.

It is important to note that the GPC control experiments took place outside actual production and with the tower running 'cold', i.e. with the heaters off and with no introduction of soap. Thus, whilst they were performed on a *real* industrial process, they were not subject to the production constraints of the previous chapter. This allowed a wider variety of experiments to be conducted without detracting from the 'industrial' validity of the exercise. In particular this made it possible to observe the effect of some of the GPC design parameters upon closed-loop set point following.

The main objectives of this study were therefore:

- To evaluate GPC, in its full multivariable implementation, under the rigour of full-scale industrial plant operating conditions and given the criteria outlined above.
- To demonstrate the feasibility of using adaptive control in practice.
- To determine whether the full multivariable GPC controller could improve the quality of individual inner loop control, and overall decoupling, from that currently obtained using the conventional Unilever Research multi-loop PID control system.
- To evaluate the specification, interpretation and practical usefulness of GPC's closed loop design parameters.
- To test alternative simplified 'multi-loop' control strategies based upon the GPC algorithm and compare their performance to that of full multivariable GPC.

The next section of this chapter describes the operation of the spray drying tower and the controller hardware and software used. The first of five results sections (5.3) deals with the performance of two preliminary experiments designed to obtain basic information about process characteristics such as the approximate dynamics, the presence of interaction, non-linearity, etc. This information is vital for the practical commissioning of the GPC controller

In section 5.4 the self-tuned multivariable GPC controller is shown to provide excellent control performance by a series of basic control trials that include regulation of both loops and set point following in individual loops. The decoupling properties of the multivariable GPC algorithm are also demonstrated and compared to that of the PID control system used during normal operation. Of particular interest in this section are the results obtained during the only test that was performed where hot air was used in the tower.

The effect of three of GPC's design parameters, $\mathcal{P}$, $\mathcal{T}$ and λ , are investigated in section 5.5. This is followed in section 5.6 with a selection of results obtained using alternative multi-loop control strategies (DGPC) where feed-forward is used to dramatically improve decoupling. Finally sections 5.7 and 5.8 present some estimation results where the performance of the parameter adaptation algorithms associated with MGPC and DGPC respectively are examined in detail for typical periods of control.

The chapter ends with section 5.9 which discusses the most important results and concludes that all the objectives of this industrial trial were achieved successfully.

5.2 THE SPRAY DRYING TOWER

Spray dryers are commonly used in industry for the drying of temperature-sensitive liquid solutions, suspensions and pumpable pastes by bringing them into contact, in the form of small droplets, with heated air in a suitably designed enclosure. The relative movement of feedstock and drying medium can be counter-current, co-current, or a combination of both (Williams-Gardner, 1971). The spray drying tower at Port Sunlight falls into the former category. Although referred to as a 'pilot-scale' plant the actual spray drying tower occupies three storeys in a specialised building and is used extensively by Unilever Research for the drying of various experimental batches of soap slurry.

With reference to the simplified schematic diagram of the spray drying tower (Fig.5.1) liquid soap slurry is sprayed into the top of the tower where it atomises, falls under gravity

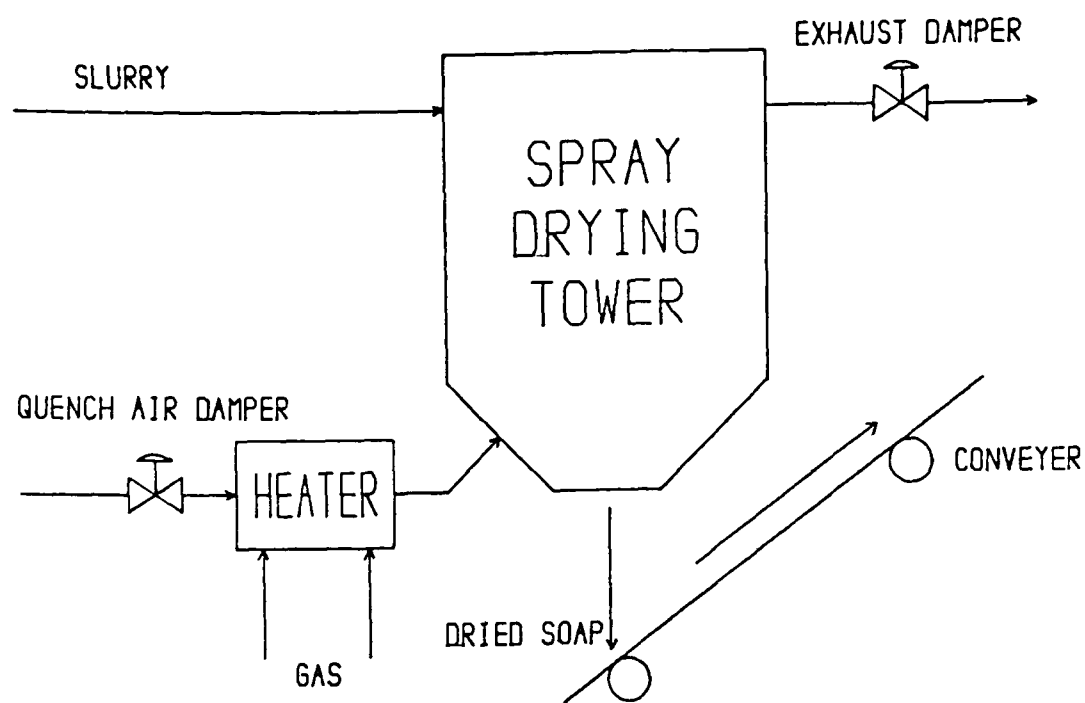


Figure 5.1: Schematic diagram of the soap spray drying tower.

and is dried by a counter-current stream of hot air injected at the tower base. This differs significantly from the Mazzoni spray dryer of Chapter 4 in that the material to be dried, and the drying medium, are actually brought into contact.

Two inner loops, *air flow* and *tower pressure* are critical in determining the final moisture content of the soap, with both affecting the rate of evaporation during soap drying. In normal operation the tower pressure must be regulated to a fixed value slightly below that of atmospheric pressure to prevent a potentially hazardous escape of hot material from the tower, and the hot air flow manipulated to produce a desired soap moisture content.

The air flow and pressure are themselves altered by an 'exhaust' air damper and a 'quench' air damper located at the top and bottom of the tower respectively. The degree to which these are opened determines the effect of fixed capacity fans which blow quench air into the tower and suck exhaust air out of the tower. Because variation of either of these dampers affects both the air flow and tower pressure measurements the control problem is thus multivariable and may be represented schematically by Fig. 5.2.

The control problem to be addressed by the GPC algorithm during these trials may therefore be summarised as the control of air flow and pressure in an industrial spray drying tower by means of quench air and exhaust air actuation. This control task was currently being performed by a Ferranti PMS-100 distributed control system employing independent PID controllers in a multi-loop structure.

The multivariable GPC algorithms to be tested were implemented in the FAUST software package, written in Fortran IV and running on a DEC LSI-11 micro-computer. A

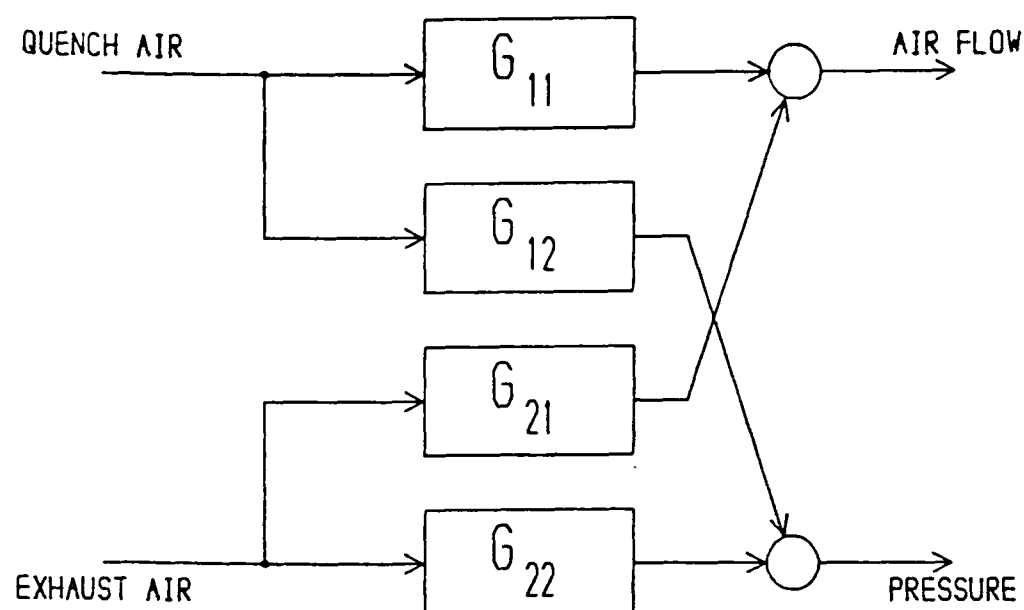


Figure 5.2: Inner loops of the soap spray drying tower.

detailed description of the multivariable GPC algorithm is given in Chapter 2. The associated estimator used the recursive least squares algorithm modified with a multivariable parameterisation and the variable forgetting factor¹ of Fortescue *et al* (1981) in order to retain adaptivity throughout the experiments.

Ref.	Loop	Variables	Description	Units
a	1	Output & Set-point	Air Flow	kg/min
b	1	Control signal	Quench air	% (Open)
c	2	Output & Set-point	Tower Pressure	mbar
d	2	Control signal	Exhaust air	% (Open)

Table 5.1: General layout of experimental results

The experimental results associated with each of these sections are displayed as a graphical plot of the measured variables with the layout given in Table 5.1. A tabular synopsis of the commissioning of the GPC algorithm used in each trial is provided in Appendix A.

5.3 SOME PRELIMINARY TRIALS

Unlike the idealised ‘black-box’ controller of Kalman (1958) the practical commissioning of the adaptive MGPC algorithm requires some basic prior knowledge about the process. A

¹see Chapter 3, section 3.4.2

simple test which was used in the previous chapter to obtain such fundamental dynamical information is the observation of the process step response. For the spray drying tower observation of the step responses of both air flow and pressure loops has the additional benefit that the presence of any interaction between the respective loops will be revealed.

5.3.1 Observation of the open loop step response

The process was first taken to the operating point which was to be used throughout all the subsequent trials by manual manipulation of the quench and exhaust air dampers from FAUST. This was deliberately chosen to corresponded to that used during routine production (see Table. 5.2).

Experimental operating point	
Air flow (kg/min)	100
Tower pressure (mbar)	-1

Table 5.2: The experimental operating point.

Once steady state was achieved, a step change of 10% variation in damper position was then applied to the quench air and the exhaust air in turn and the step responses of both air flow and pressure loops recorded. It is important to note that the size of the input variation was chosen such that each of the output loops were caused to vary over the full range of values that were to be encountered in the later closed loop experiments. The experiment was conducted with a sample interval of 2 seconds, this being the fastest sample rate obtainable with the available hardware and software.

Fig. 5.3² clearly shows that the variation of the inlet quench air damper at the base of the tower causes a large change in both air flow and tower pressure (1). In contrast the exhaust air damper that extracts air from the top of the tower causes a large change in the tower pressure (2), but has a negligible effect upon the air flow. Thus the process exhibits a predominantly *one-way* interaction. More detailed examination of the behaviour of the step responses in individual loops indicates that both loops have an approximate dead-time

²Note that the dotted line on the plot showing the tower pressure measurement represents 0 mbar which, because of the differential nature of the pressure sensor, corresponds to atmospheric pressure. It is shown here for reference but becomes more important in the later control experiments where the tower pressure must not rise above this threshold in order to prevent material escaping from the tower, a safety hazard when the tower is running 'hot'.

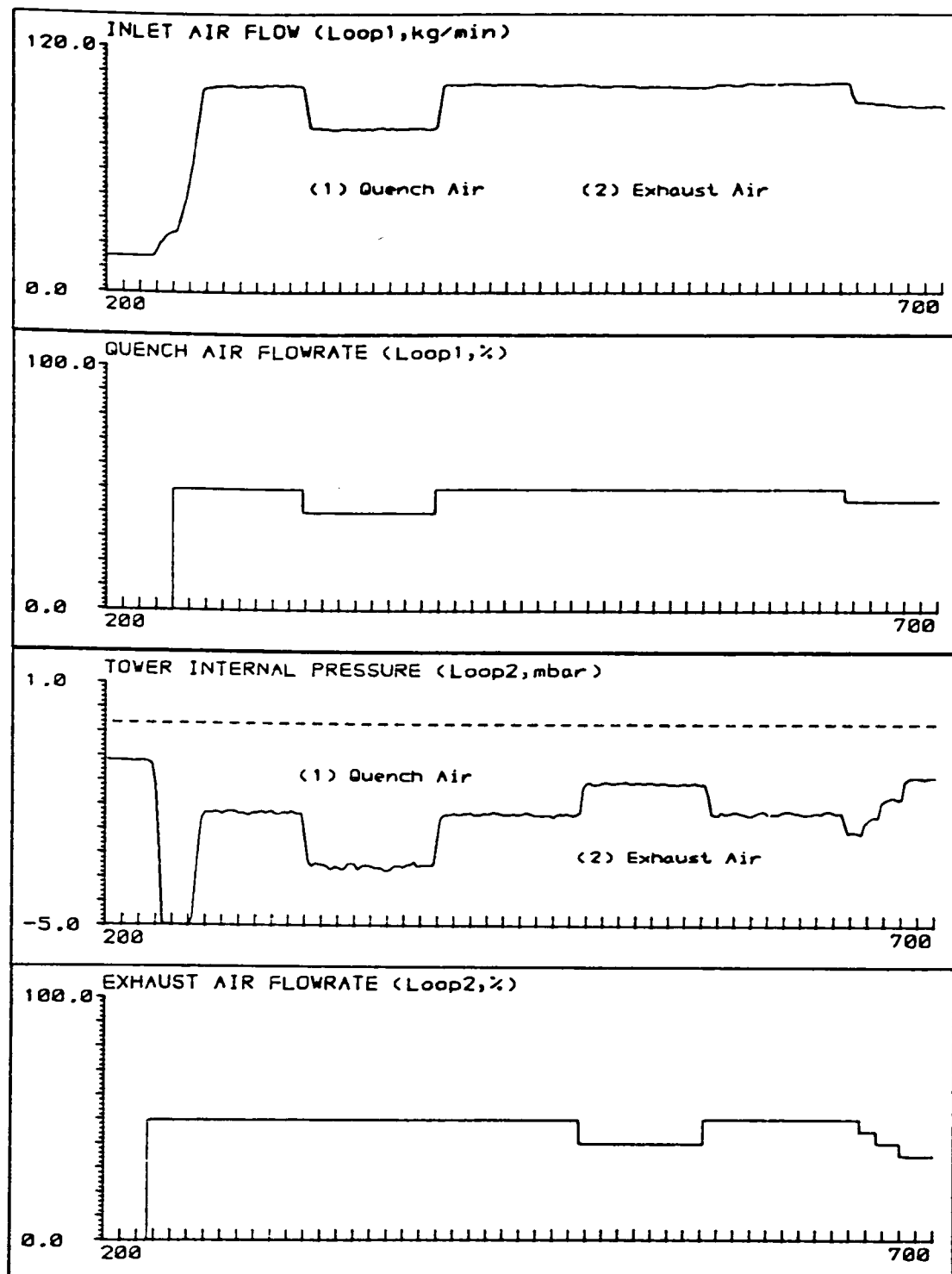


Figure 5.3: Open loop step response.

of 2 seconds, a rise time of approximately 6 seconds and are very well behaved with no overshoot and a high signal to noise ratio. The relatively high speed of the response is typical of gas flow loops.

Based upon these observations, and the limitations of the hardware, a sample interval of 2 seconds, with an anti-aliasing filter of time-constant 4 seconds, was selected and used throughout the remainder of the experiments. The simplicity of the response led to the assumption of a first order model structure for each loop (see section 5.7).

5.3.2 Observation of process linearity

The objective of this experiment was to observe the linearity of the process over the region of operating points that were to be used during subsequent experiments. In order to do this the tower was once again manually controlled to the upper operating point (100kg/min,-1mbar). As later experiments were to involve set-point changes of 100 $\rightarrow$ 80 kg/min on air flow and -1 $\rightarrow$ -2 mbar on tower pressure the tower was taken through this 'envelope' of operating points by the introduction of step-wise ramps in the quench air and exhaust air control signals respectively. The results are shown in Fig. 5.4.

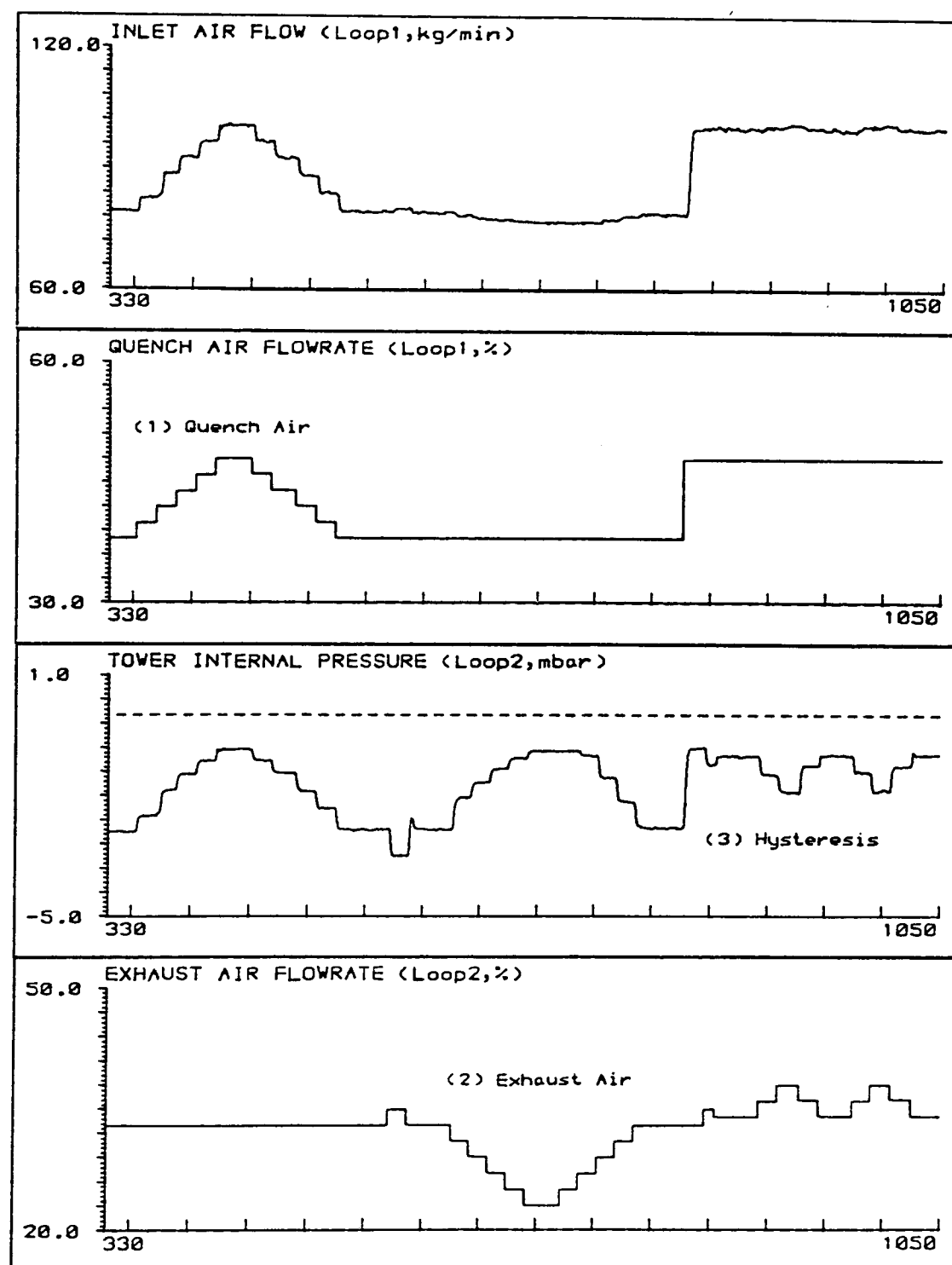


Figure 5.4: Observation of process linearity.

The response of air flow to the variation in quench air appears to be reasonably linear, as does the corresponding response of tower pressure due to interaction (1). However the

response of tower pressure to step-wise variation of the exhaust air is far from linear, with the gain of that loop appearing to vary by a ratio of over 5:1 between operating points (2). Furthermore the presence of *hysteresis* is clearly exhibited by the lack of symmetry of the pressure response as the pressure is returned to -2mbar (3), i.e. the process exhibits direction-dependent gain.

Whilst these observations were of little practical use in the commissioning of the MGPC controller, compared to the step response tests of the previous section, they do provide an added insight into the true control problem.

5.4 MULTIVARIABLE GPC CONTROL

The Generalised Predictive Control algorithm developed in Chapter 2 is one of the most recent adaptive control algorithms to be extended to the control of multivariable systems. In that chapter it was claimed that GPC was particularly suitable for the control of such systems because of its basis upon *long range prediction*, and its *explicit* control formulation, rendering the controller relatively insensitive to unknown and/or varying time-delays. This is expected to overcome the major problem of multivariable time-delay structure that limited the practical usefulness of earlier multivariable controllers based upon GMV, by removing the requirement for detailed prior knowledge of a *delay matrix*. In addition the multivariable implementation of GPC is significantly simpler and computationally less expensive than alternative multivariable algorithms based upon pole-placement or LQG control strategies.

The so-called 'full' multivariable implementation (MGPC) effectively solves two sets of simultaneous equations in order to calculate the control signals for both loops. As this model takes into account the potential presence of interaction between loops MGPC is thus expected to provide good decoupling of the air flow and pressure loops, without requiring any assumptions about input-output pairing.

5.4.1 General experimental procedure

In this section an early period of experimental control using MGPC is presented (Fig. 5.5) that is a typical example of the general experimental procedure used throughout the trials. The main areas of interest during the experiment are described below:

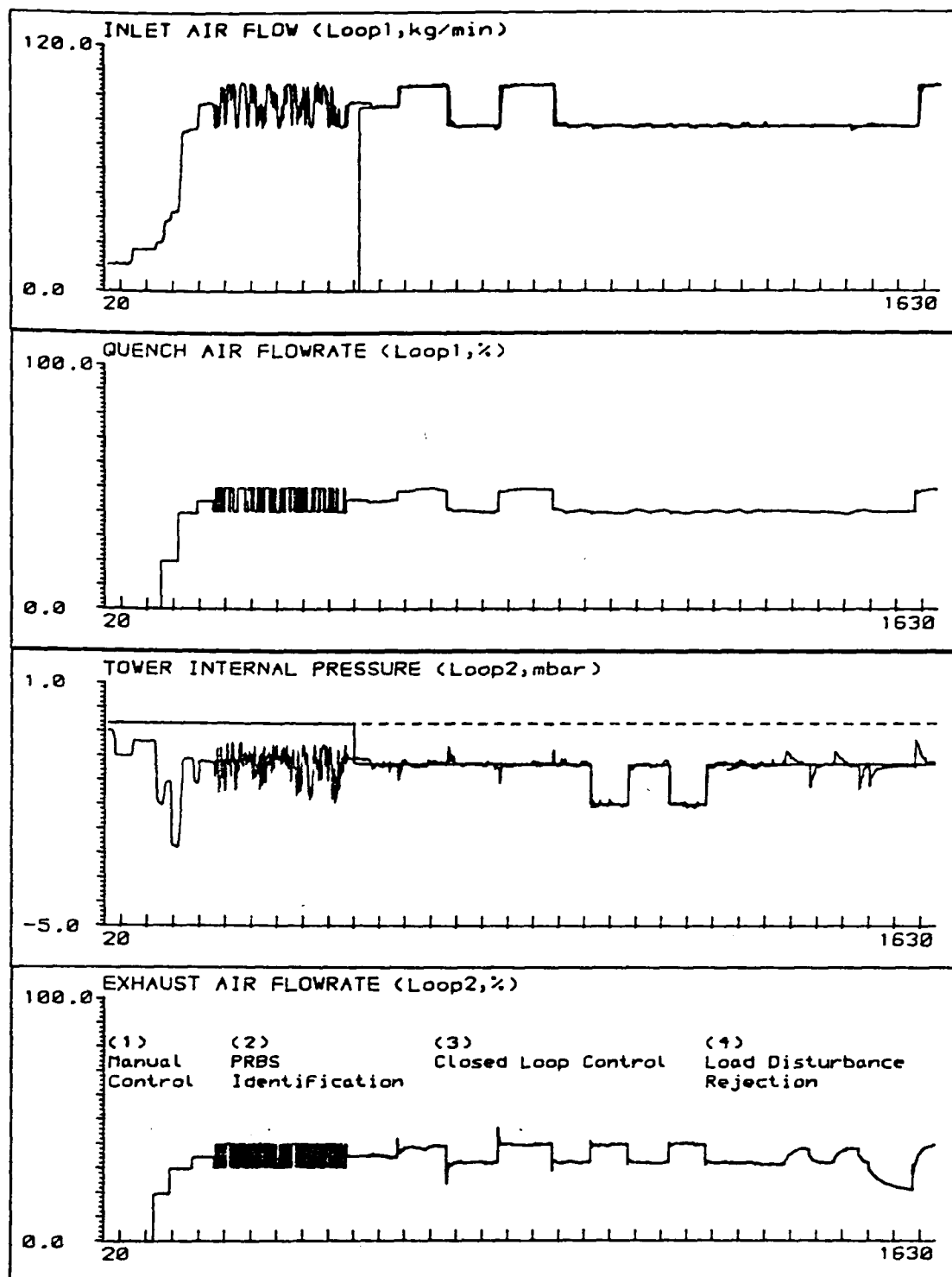


Figure 5.5: General experimental procedure.

1. *Manual Control:* The tower is brought on-line by closing both air dampers and activating the fans. Then the air dampers are 'manually' controlled to take the air flow and tower pressure to the values given in Table 5.2.
2. *PRBS Identification:* Once steady-state has been achieved the RLS estimator is enabled and PRBS sequences superimposed upon the open loop inputs of both loops. This provides a sustained period of 'exciting' data to assist the initial self-tuning of the adaptive MGPC controller whilst maintaining the air flow and pressure, on average, at the operating point.
3. *Closed Loop Control:* Once the parameters of the estimated model have converged the set-points may be specified and the loop closed. The performance of MGPC is

then tested by alternately applying set-point changes to each loop whilst regulating the other. This demonstrates both the set-point tracking performance in individual loops and the overall decoupling performance of the controller.

4. *Load Disturbance Rejection*: The load disturbance rejection properties of MGPC may be tested by the introduction of artificial load disturbances to the tower pressure. This is simply and repeatably achieved by suddenly changing the cross-sectional area of the orifice at the base of the tower.

It is important to note that the 2×2 MGPC cost-function involves the minimisation of the sum of squares of the control error for air flow *plus* the sum of squares of the control errors for tower pressure (see eqn. 2.60). Because of this the control activity in each loop could therefore be affected by the relative magnitudes³ of the two measured variables. This experiment demonstrated that this was not a serious problem for the spray drying tower where, despite the value of the air flow measurement being approximately 100 times that of tower pressure at the operating point, MGPC control based upon these engineering units appeared to be equally tight in both loops.

Nonetheless in order to ensure *symmetrical* control performance the tower pressure measurement was scaled up in subsequent trials by a factor of 100 before entering the control calculation. This was found to slightly improve decoupling of the tower pressure during control activity in the air flow loop (see Figs.5.8,5.12,5.14). The importance of this scaling for estimation is discussed in section 5.7.

5.4.2 Commissioning and self-tuning

The procedure for commissioning the adaptive MGPC controller may in practice be separated into four distinct stages :

- Specification of the control objective by means of the MGPC controller's design parameters (i.e NS , NY , NU , $\mathcal{P}$, Λ and T).
- Specification of an appropriate structure for the estimated model and initialisation of the estimation algorithm.
- The performance of an initial self-tuning experiment.

³see Chapter2, section 2.5.1 for further details of this phenomenon and a simulation example.

- Closing of the loop when satisfactory model performance has been achieved.

As in the application described in Chapter 4 the so-called 'default' controller design parameters were used throughout these trials unless otherwise indicated. However, because of the multivariable nature of the controller these parameters were now diagonal matrices, with individual elements determining control performance for the air flow and tower pressure control loops respectively. In all cases the default settings were identical for both loops.

Specification and initialisation of the estimator is described in section (5.7) which deals specifically with the performance of estimation throughout these trials. It is sufficient here to note that a first order model was estimated for each 'channel' (Fig. 5.2) and that the RLS algorithm was initialised so as to provide rapid initial adaptation.

The results of one of the earliest self-tuning experiments to be carried out on the spray drying tower, following the procedure described above, is shown in Fig. 5.6. After the air flow and tower pressure are brought to the experimental operating point by manual control a PRBS⁴ sequence of magnitude $\pm 5\%$ was superimposed simultaneously upon both the quench and exhaust air actuation. The magnitude of the PRBS input was chosen to guarantee that the two output measurements, air flow and tower pressure, vary through the entire range of operational values that will be used in the later closed loop tests. That it is particularly important to self-tune the controller at the operating point, and not just at any arbitrary point, is indicated by the significant non-linearity of the tower pressure step response observed in section 5.3.2.

After approximately 15 minutes of on-line identification in open loop the PRBS input was removed, the set points specified and the loop closed. The MGPC control immediately achieved the set-point, with no 'bump' in the control signal, and then provided excellent control of a set point change in the air flow at $t = 600$ whilst regulating the tower pressure.

As in the previous chapter this demonstrates that, facilitated by its adaptive implementation, the practical commissioning of MGPC is both straightforward and methodical, even for a multivariable system. Excellent control of air flow and tower pressure was achieved within 20 minutes of bringing the tower on-line and without any detailed prior knowledge of the process or time-consuming manual tuning. The majority of subsequent trials

⁴Note: In order to equally excite all modes of this multivariable plant care must be taken that the PRBS inputs on each input are relatively uncorrelated. This was achieved in practice by the application of the same PRBS sequence to both loops but starting the sequence on the quench air several samples later than that of the exhaust air.

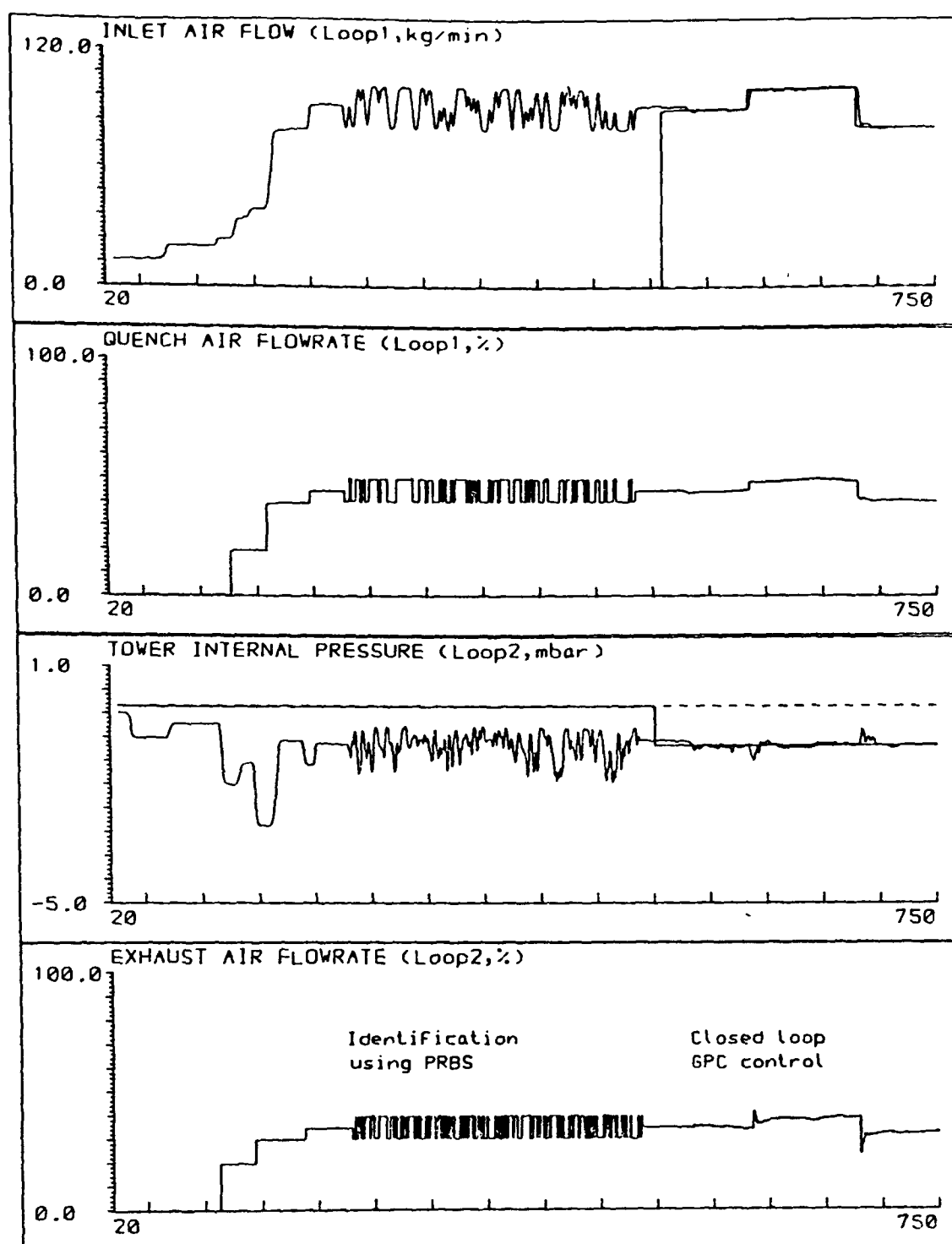


Figure 5.6: On-line process identification using a PRBS input

repeated this self-tuning procedure and it proved to be consistently reliable and successful.

5.4.3 Long term regulation

During normal operation of the spray drying tower, when used for the drying of experimental batches of soap, the principal control objective is the regulation of the soap moisture content to a constant set-point. This in turn implies that for most of that period the hot air flow and the tower pressure must themselves be regulated to constant values.

Thus Fig. 5.7 records a period of closed loop control using MGPC where the air flow and tower pressure were regulated to their set-points for over 3600 samples (i.e. > 2 hours).

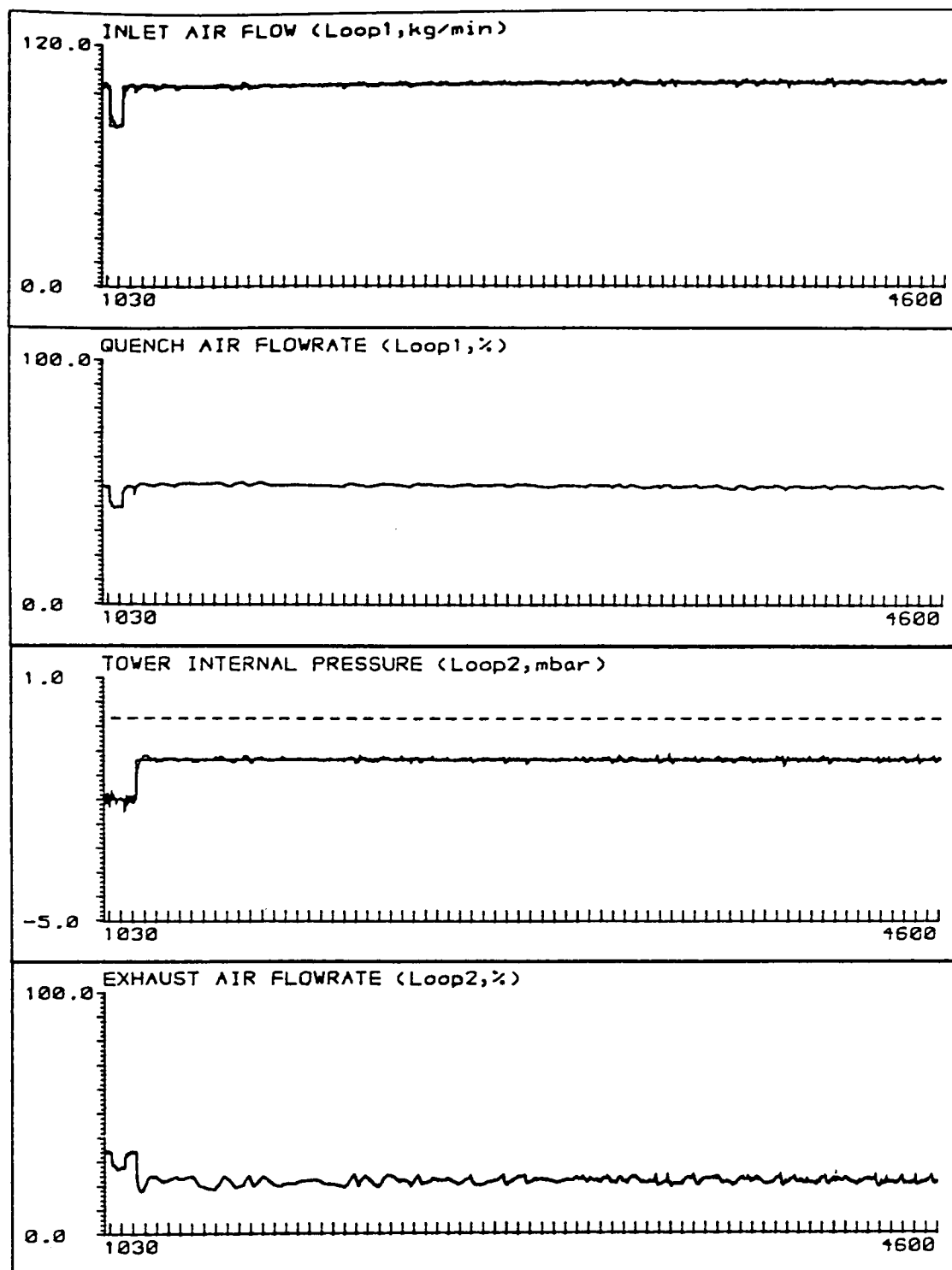


Figure 5.7: Long term MGPC regulation.

MGPC control was very tight⁵ with output r.m.s. errors of only 0.65 kg/min and 0.035 mbar for the air flow and tower pressure respectively, with no observable offset. Whilst little appears to be happening of interest, the success of such an experiment is vital for MGPC to become accepted in practice, as long term regulation is the most fundamental control objective encountered in process control.

5.4.4 Servo control and decoupling

The moisture content of soap leaving the spray drying tower is determined by the flow-rate of hot air through the tower, so that accurate manipulation of the air flow set-point

⁵see section 5.4.5 for a comparison with the existing PID controller's performance.

is an important requirement for inner loop control. This is complicated by the presence of interaction which means that the specification of a set-point for air flow also affects tower pressure. The set point tracking performance of MGPC must therefore be judged with respect to two elements: individual loop set point following, and decoupling.

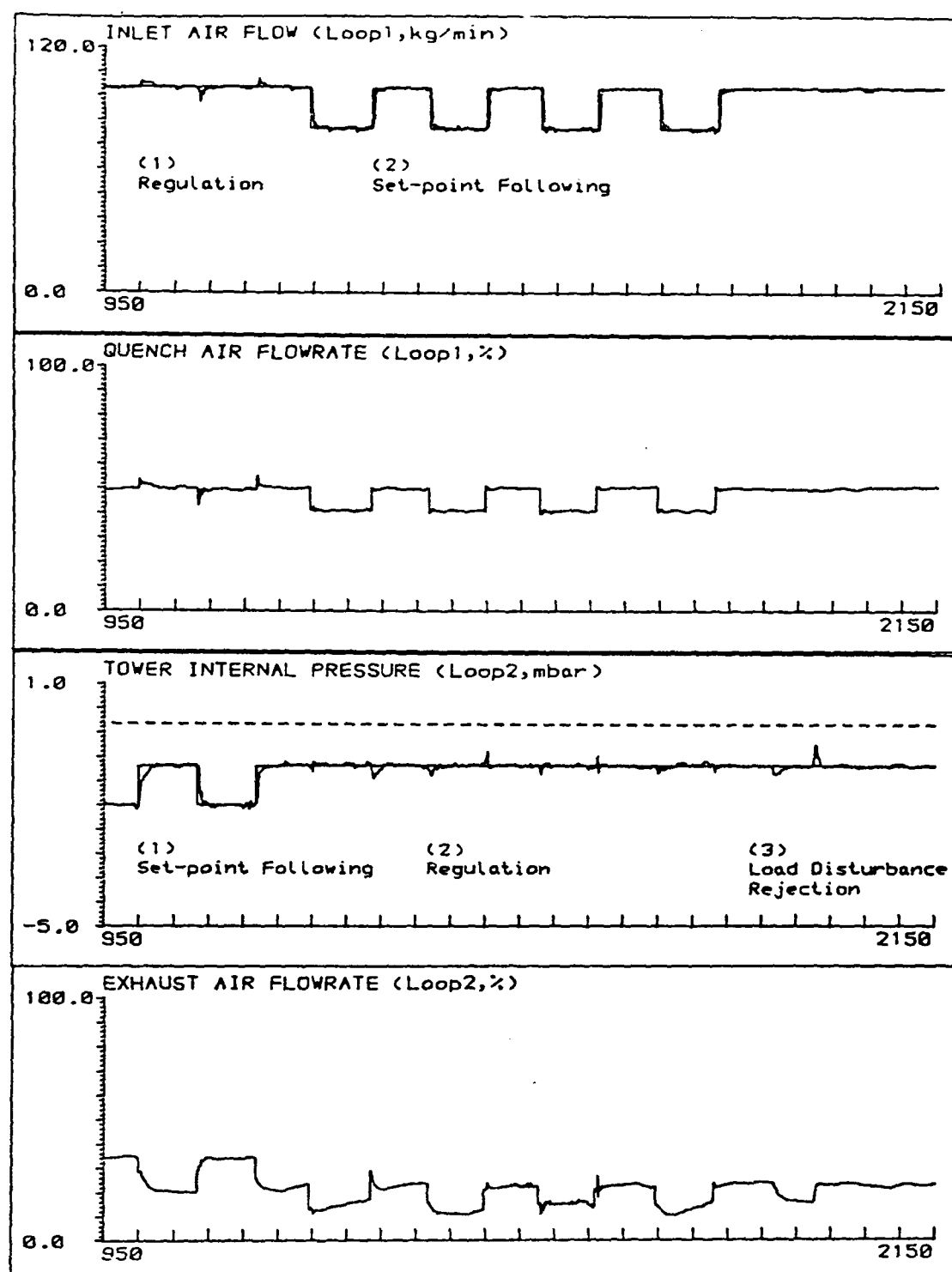


Figure 5.8: Multivariable GPC closed loop control.

Thus the experiment shown in Fig. 5.8 consists of a period of closed loop control using MGPC where a series of 'step-up step-down' set-point changes were applied first to the tower pressure loop (1), and then to the air flow (2) whilst regulating the other loop. This is followed by a period of regulation in both loops during which load disturbances were artificially induced upon the tower pressure (3) in order to evaluate the load disturbance rejection properties of the MGPC closed loop. The results obtained for the separate

elements of this experiment, and numbered in Fig. 5.8, are discussed below:

1. *Servo control of pressure, Regulation of air flow*: The MGPC controlled response of tower pressure to a series of set-point changes between -1 and -2 mbar is both rapid and accurate as a result of step-wise changes in the exhaust air damper position. The effect of interaction from the tower pressure loop upon the air flow is negligible.
2. *Servo control of air flow, Regulation of pressure*: The set point following of the air flow measurement is almost immediate in response to step-wise control signals applied to the quench air damper. Despite the large interaction between quench air and tower pressure, demonstrated in Fig. 5.3, the regulation of tower pressure is maintained by corresponding variations in exhaust air damper position which act so as to decouple the two loops.

Closer inspection of the tower pressure measurement indicates that this decoupling improves with each successive set point change in the air flow loop, as the estimated model gradually improves. It is important to note that despite large and active variations in the air flow (20% of full scale) the tower pressure never once exceeded the critical threshold of atmospheric pressure.

3. *Load Disturbance Rejection*: It was possible to artificially induce disturbances in the tower pressure measurement by suddenly changing the cross-sectional area of an orifice at the base of the tower. Under manual control (i.e. with fixed quench and exhaust air damper positions) this was observed to cause a sustained offset of in tower pressure of approximately 0.5 mbar. However, when two such unmeasurable load disturbances were introduced during MGPC closed loop control the tower pressure was in both cases completely restored to the set-point within 30 seconds and without excessive variation of the quench air damper.

This sequence of results clearly demonstrates that the 'default' MGPC control algorithm is capable of very tight closed loop control of both air flow and pressure inside the tower, including regulation, set-point following, decoupling and load disturbance rejection. The step-like nature of the control signal activity in both loops is explained by the relatively slow sample rate and the specification of MGPC's design parameters to give a mean-level control strategy (see Chapter 2).

It is also interesting to note that although there is no pairing of inputs and outputs

implicit in the MGPC implementation the estimated model may be seen to have automatically identified⁶ the dominant pairings. Thus the quench air damper was manipulated to change air flow and the exhaust air damper manipulated to vary tower pressure.

5.4.5 Comparison with existing PID control

The control of air flow and tower pressure was, in normal Unilever Research operation, being performed by the distributed control section of an extensive Ferranti PMS-100 process control system. This consisted of individual PID controllers applied to each of the two loops, with the quench air damper being used to control the air flow and exhaust air damper used for tower pressure. A schematic diagram of this control structure is shown in Fig. 5.9 which also includes a table of the actual PID settings used. Note that the PID used to control the air flow was gain-scheduled suggesting that process non-linearity was considered a problem during tuning.

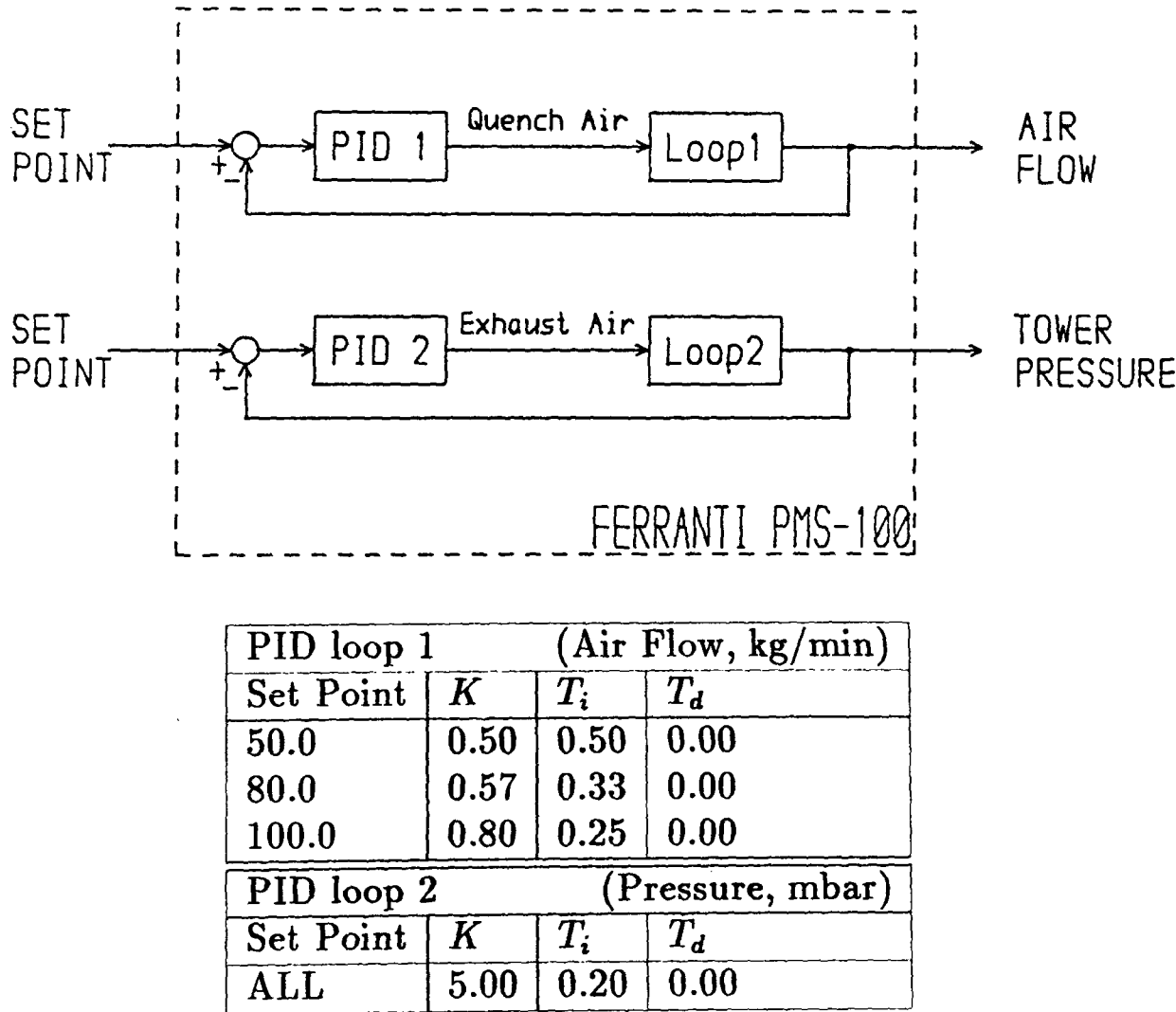


Figure 5.9: The Ferranti distributed PID control structure.

To enable comparison of the MGPC control performance to that of the Ferranti control system the procedure of the previous experiment was repeated whilst under PID control.

⁶see section 5.7.3 for further details.

Unfortunately it was not possible to log the PID control signals and thus only the air flow and tower pressure measurements are shown in Fig. 5.10.

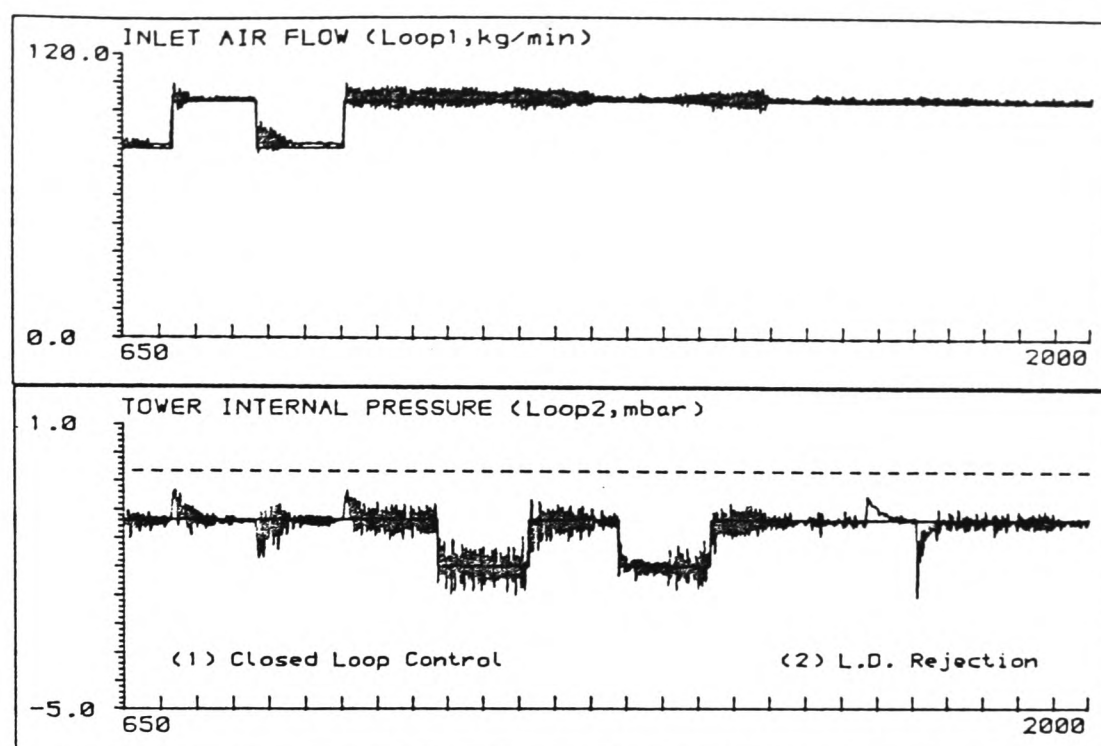


Figure 5.10: Comparison with existing PID control scheme.

From this plot it may be seen that the closed-loop control provided by the multi-loop PID control system is indeed tight with the outputs of both loops definitely following their set-points (1). However the excessive activity of the control signals (not shown) lead to both air flow and tower pressure being unnecessarily oscillatory. This large degree of oscillation is typical of high-gain PID control in the presence of time-delay and/or non-linearity and could only have been removed by 'de-tuning' the PID settings. This would of course have been at the expense of such tight closed loop control.

The decoupling performance of a distributed control structure such as this, which takes no special account of the presence of interaction, relies upon the load disturbance rejection properties of individual loops. This is seen during changes in air flow where the speed with which the interaction is removed from the tower pressure corresponds to the speed with which load disturbances are rejected later in the experiment (2).

By comparison of these results to those obtained using MGPC in section 5.4.3 the overall control performance of the self-tuned MGPC controller is clearly superior to that of the manually tuned multi-loop PID control system (see Table 5.3).

	PID		GPC	
Measurement	mean (kg/min)	error (rms)	mean (kg/min)	error (rms)
Air Flow	101.10	5.59	99.75	0.64
Tower Pressure	-1.03	0.11	-0.9975	0.035

Table 5.3: Mean and variance of control errors: MGPC vs. PID.

5.4.6 Tower control with hot air

The results presented in this section were obtained during the single period of MGPC control that was performed whilst the spray drying tower was ‘running hot’, i.e the heater was switched on (see Fig. 5.1) so that the air entering the tower was at the high temperature used during the actual drying of soap ($\approx 300^{\circ}C$). As it was not possible to actually spray soap into the tower for financial reasons water was instead injected to cool the tower. It is included here because although soap was not being dried the control during this experiment is taking place in conditions that correspond more closely to those encountered in practice than when the tower is running cold.

With reference to the results of earlier experiments it is obvious that with hot air the tower pressure measurement becomes substantially noisier. Nonetheless the overall performance of the default MGPC controller is again satisfactory with characteristically good initial control after only a brief period of PRBS assisted self-tuning. Regulation and set-point following are both accurate with the former being offset-free and the latter giving fast response with no overshoot in both loops, which improves dramatically as the estimated model itself improves. Decoupling of the tower pressure during set point changes in air flow also progressively improves and never once exceeds the threshold of atmospheric pressure.

Note that the increased activity of the MGPC control signal is due mainly to the specification, in this early experiment, of $T = I$. Specification of individual T ’s as low-pass filters would have reduced control signal activity and greatly improved overall control performance (see sections 2.3.3 and 5.5.2).

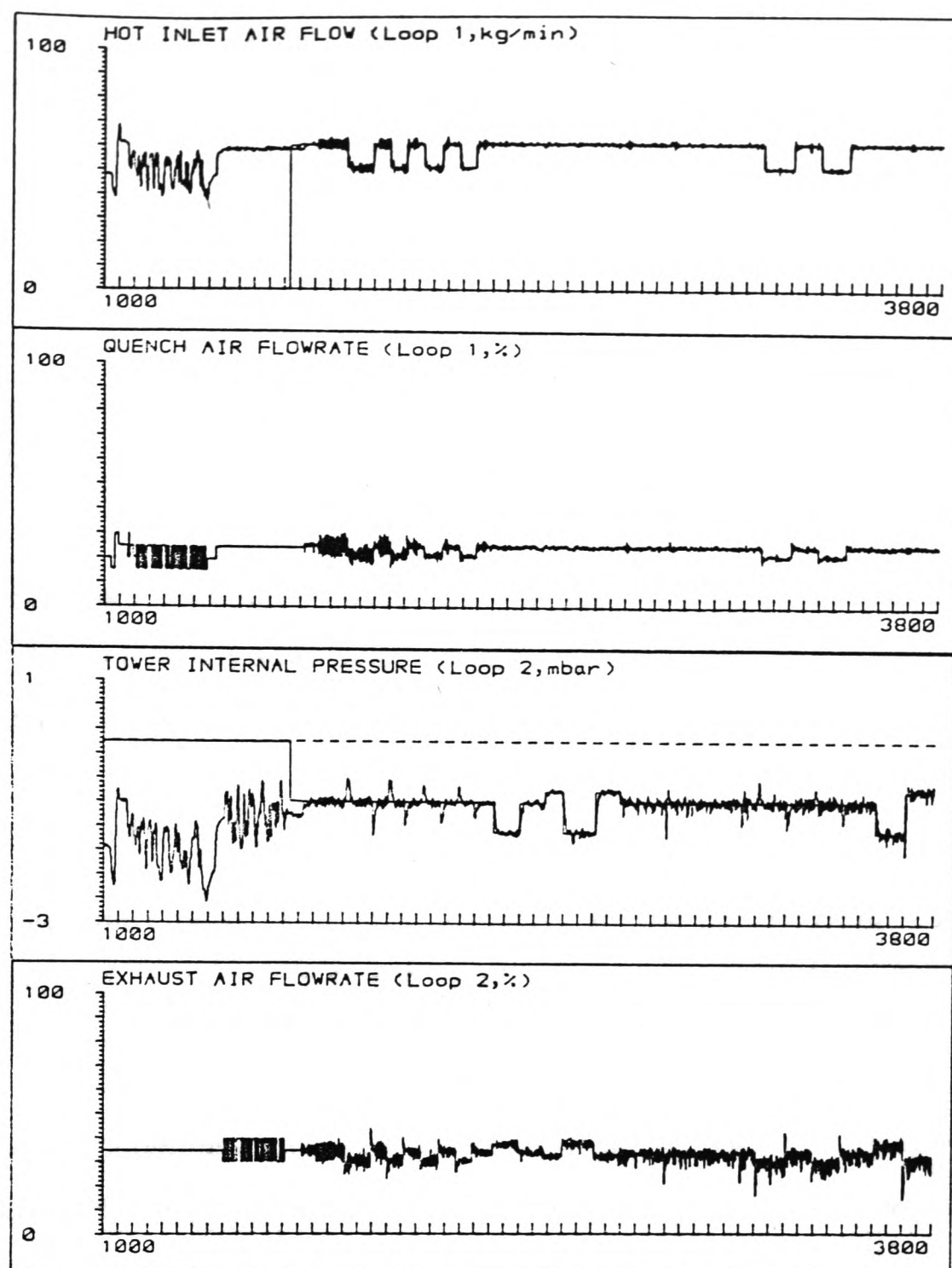


Figure 5.11: Tower control with hot air.

5.5 MULTIVARIABLE GPC DESIGN PARAMETERS

The 'default' specification of the MGPC algorithm, used successfully in the previous section and indeed throughout all the trials described in the previous chapter, is designed to provide satisfactory stable control for a wide range of industrial processes. This is because the prediction horizon is chosen to be relatively long ($NY = 10$), so that the steady-state dominates the control calculation, and the control horizon short ($NU = 1$) in order to restrict the activity of the control signal. This corresponds to a cautious, low-gain type of control law often referred to as *robust* (Rohrs, 1982).

However in practice there may be a requirement for the performance of the controller to be changed from that obtained using the default MGPC specification with, for example,

a faster or slower output response desired by increase or decrease of the control signal activity. The aim of the experiments described in this section was therefore to demonstrate how such process specific tuning of the control could be achieved *on-line* by variation of MGPC's design parameters. This not only represented an attempt at practical verification of the deterministic results outlined in Chapter 2 but also sought to evaluate the industrial usefulness of the design parameters.

Unfortunately, due to the relatively slow sample rate and the speed of the inner loop response, the default MGPC controller was already producing very fast controlled response of air flow and pressure. Thus it was only possible to obtain meaningful results by varying the $\mathcal{P}$, $\mathcal{T}$ and Λ design parameters from their default values, as these de-tune the control response. In order to prevent adaptivity having any effect upon the results estimation was switched off during these trials.

5.5.1 The effect of $\mathcal{P}(q^{-1})$

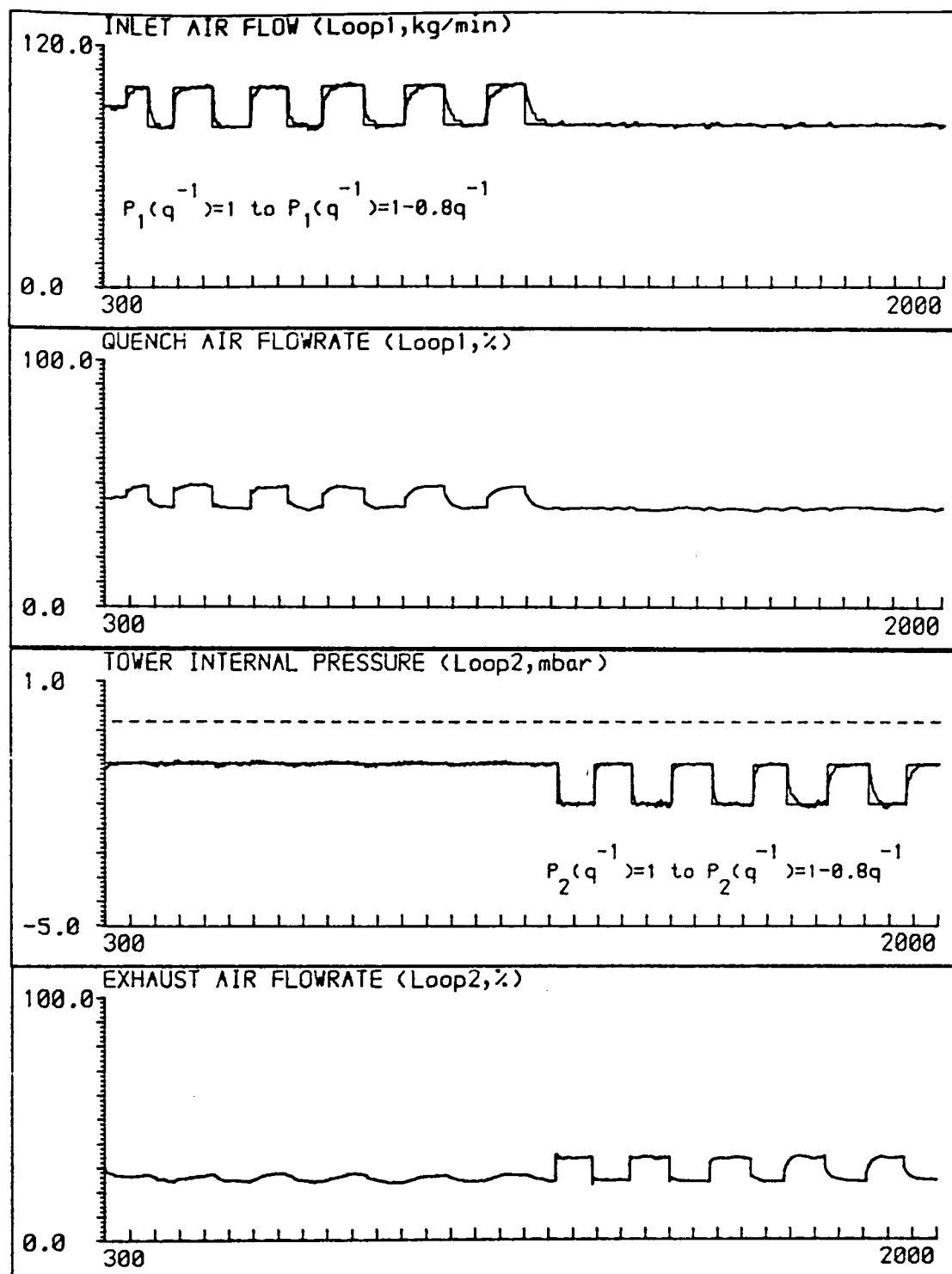
The MGPC design parameter, $\mathcal{P}(q^{-1})$, is a matrix for the multivariable case, i.e.

$$\mathcal{P}(q^{-1}) = \begin{bmatrix} P_1(q^{-1}) & 0 \\ 0 & P_2(q^{-1}) \end{bmatrix} = \begin{bmatrix} (P_1)_n/(P_1)_d & 0 \\ 0 & (P_2)_n/(P_2)_d \end{bmatrix} \quad (5.1)$$

whose elements are *polynomial transfer functions* in the backward shift operator q^{-1} . This allowed separate P transfer functions to be specified for each of the two outputs, with P_1 and P_2 corresponding to air flow and tower pressure respectively.

In Chapter 2 $\mathcal{P}$ was described as a means of adding phase advance into the closed loop for typical process control applications such as the spray drying tower. It was also shown, for the SISO GPC algorithm, to represent an approximation to pole placement which was exact if $NU = NY$ and $\lambda = 0$. In practice however, where estimation is never exact and $NY \gg NU$, the individual P transfer functions may best be regarded as the inverse of low pass filters where the slower the specification of the filter time-constant (i.e. P_n) the slower the control performance in that loop.

In order to verify this Fig. 5.12 shows the results of an experiment where progressively 'slower' P transfer functions were specified, first for the air flow loop, and then for the tower pressure loop. For each successive value of P two set-point changes were applied to the appropriate output to observe the effect upon the closed loop performance. A progressively slower closed loop response is clearly demonstrated in both loops with correspondingly less


 Figure 5.12: The effect of $\mathcal{P}(q^{-1})$.

active control signals.

Thus the MGPC design parameter, $\mathcal{P}$, provided a means of 'de-tuning' the control activity of the closed loop on-line. Its implementation as a diagonal matrix allowed this to be performed independently for each of the two loops.

5.5.2 The effect of $\mathcal{T}(q^{-1})$

The multivariable implementation of the GPC algorithm allows specification of a matrix, $\mathcal{T}$,

$$\mathcal{T}(q^{-1}) = \begin{bmatrix} T_1(q^{-1}) & 0 \\ 0 & T_2(q^{-1}) \end{bmatrix} \quad (5.2)$$

whose diagonal elements are *polynomials* in q^{-1} . Thus for the control of the spray drying tower the choice of T_1 corresponds to the air flow loop and T_2 to the tower pressure. The objective of this experiment was to demonstrate in practice the theoretical claim of Chapter 2 that the choice of $1/T$ for a particular loop as a low-pass filter should de-tune the control activity in that loop for high-frequency transient disturbances. The results are shown in Fig. 5.13, where T_2 was varied from 1 to $1 - 0.8q^{-1}$. For each value of T_2 a load disturbance was artificially induced on the tower pressure, using the technique described in section 5.4, and then removed a few samples later.

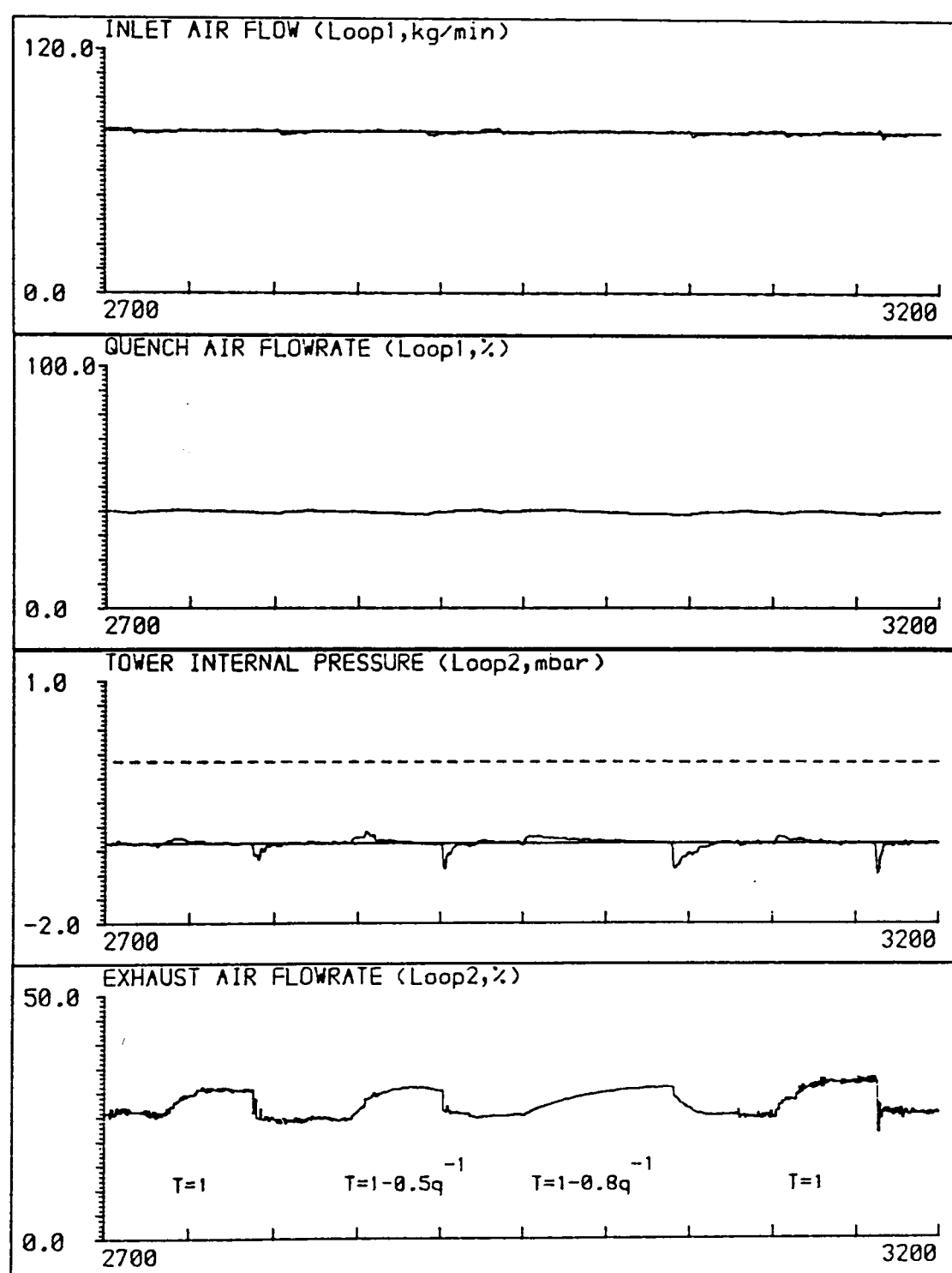


Figure 5.13: The effect of $T(q^{-1})$.

By inspection of either the pressure output, or the exhaust air control signal, it is clear that the specification of $1/T_2$ as a low pass filter does indeed reduce the speed with which

the load disturbance is rejected, with correspondingly less active control. Furthermore in terms of time-constants the 'slower' the specification of $1/T_2$ the slower the load disturbance rejection. It is also worth noting that with $T_2 = 1 - 0.8q^{-1}$ the control even in steady-state is significantly less active than with $T_2 = 1$. This is because the filtering effect of T_2 discards those high frequency components of $y_2(t)$, such as measurement noise and unmodelled dynamics, which could not in any case be controlled (see Chapter 2, section 2.4.8). Thus the control signal, based upon a control calculation using data filtered by T_2 , itself has a lower frequency content.

5.5.3 The effect of control weighting

Penalisation of control signal *increments* in the quadratic cost-function of MGPC (2.8) is expected to result in a de-tuning of the control effort. The amount of reduction in activity for a given control signal is determined by the value of a scalar weighting, λ , associated with that input. In a similar fashion to $\mathcal{P}$ and $\mathcal{T}$ a diagonal matrix of control weights may be specified as,

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (5.3)$$

where the diagonal elements, λ_1 and λ_2 , represent the individual weightings upon the quench air flow and exhaust air flow control signals respectively.

In this experiment the control weighting was varied from 10^{-6} to 100 with the effect of successive values of λ being observed, as in 5.5.1, by two set-point changes being performed for that value. This variation of λ was carried out first in the tower pressure loop and then in the air flow loop.

Fig. 5.14 shows both loops behaving as predicted, with increasingly large values of control weighting resulting in a progressive decrease in speed of the set-point response with a corresponding lessening of control signal activity. It is particularly interesting to note that de-tuning the air flow loop also reduces the effect of interaction upon tower pressure.

5.6 MULTI-LOOP GPC CONTROL

This section describes the application of various alternative *simplified* multivariable GPC controllers to the control of the spray drying tower. Instead of the 'full' MGPC used in

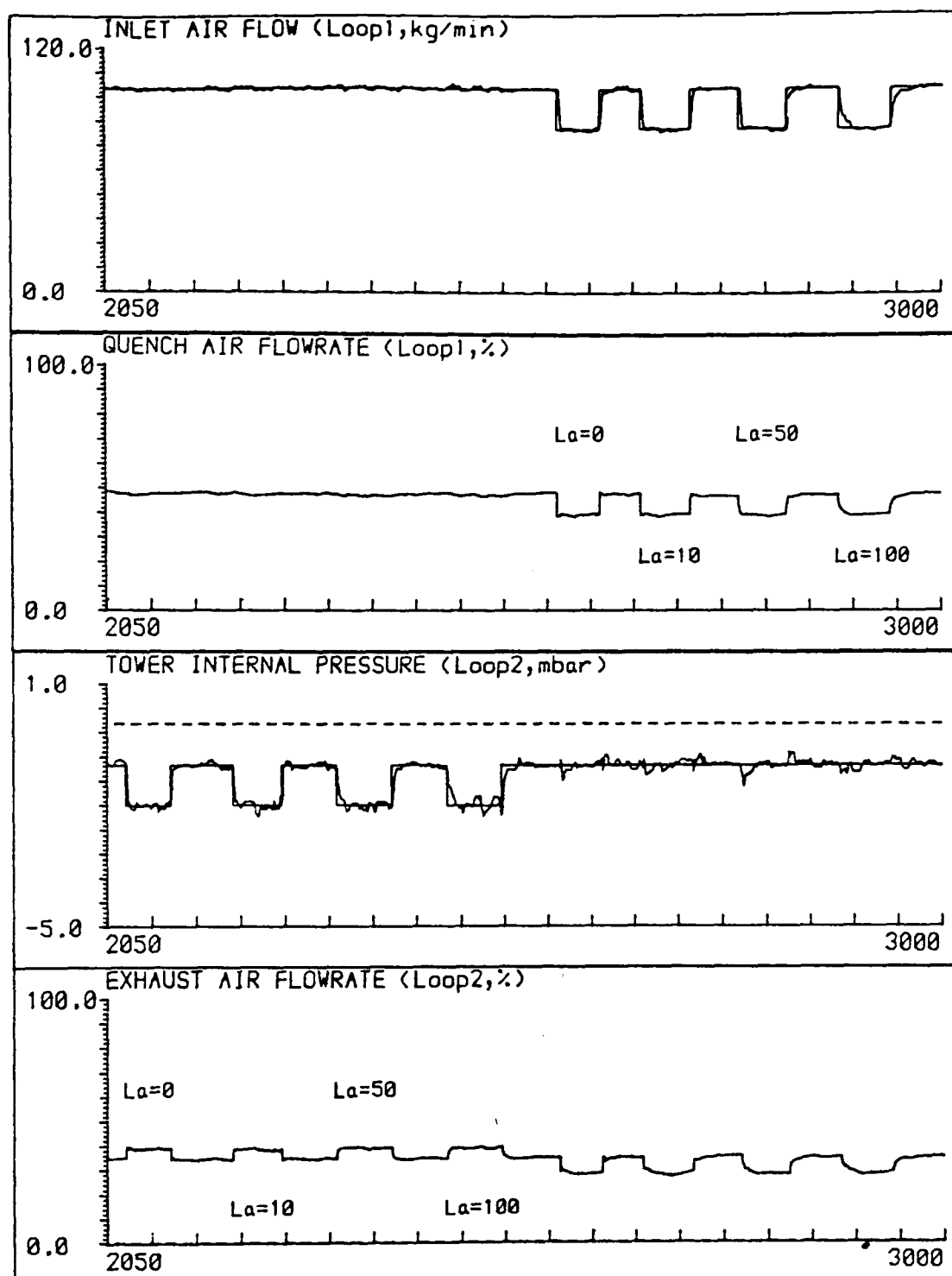


Figure 5.14: The effect of control weighting.

the previous sections these are based upon a multi-loop structure with separate single-input single-output GPC controllers around the air flow and pressure loops. Whilst this control structure was implemented upon a single machine, and is thus not distributed in a hardware sense, it is referred to as *distributed* GPC (DGPC) in order to distinguish it from full MGPC.

The advantage of this distributed GPC approach was claimed in Chapter 2 to be the reduced order, and thus simplification of the estimation and control calculations. In addition the independence of the cost-functions of the separate GPC controllers removes the requirement for normalisation of the data necessary in the full implementation to preserve the symmetry of the algorithm. There are however two potential disadvantages: the in-

herent requirement for *a priori* selection of input–output pairings, and reduced decoupling performance.

The former is only a problem when there is a large amount of interaction between all the process loops and the best input–output combination is not clear. The latter arises because, even if feed–forward is used the measured feed–forward signal (i.e. the control signal of the other loop) is not known at time t and must instead be proxied in the control calculation by its previous value, i.e. at time $t - 1$. Thus the decoupling response to control activity in another interacting loop will always be one sample later than that of the MGPC controller which, by effectively solving a set of simultaneous equations to calculate its control signals, takes *immediate* account of the effect of the other loops' control signal.

In order to enable a fair comparison of DGPC control performance with that of MGPC where possible the estimator was commissioned identically and a first order model estimated. Once again the exact parameterisation and initialisation of the estimator used throughout these particular trials is discussed in a later section (5.7).

5.6.1 Distributed GPC control, no feed-forward

In this experiment the most basic of the distributed GPC control strategies was tested where two independent SISO GPC controllers were employed, one to control air flow by manipulation of the quench air damper and the other to control tower pressure by means of exhaust air actuation. This input–output pairing was selected on the basis of a standard procedure, well known to control engineers, that involves obtaining the system's Relative Gain Array (Bristol, 1966; Shinskey, 1979). This is calculated from the observed dynamic gains, at steady-state, of the various input–output combinations. From Fig. 5.3 and for the assumed pairing of quench air/air flow and exhaust air/tower pressure shown in Fig. 5.2 the RGA for the spray drying tower is given by:

$$\Pi = \begin{bmatrix} 0.92 & 0.08 \\ 0.08 & 0.92 \end{bmatrix} \quad (5.4)$$

The strong diagonal dominance of this matrix indicates both that the interaction is relatively minor (i.e. one-way) and that the assumed input–output pairing is definitely correct.

No feed-forward terms were estimated in either of the two loops and the 'default' specifications were used for each of the DGPC design parameters. The layout of the experiment closely followed the general procedure presented in section 5.4.1.

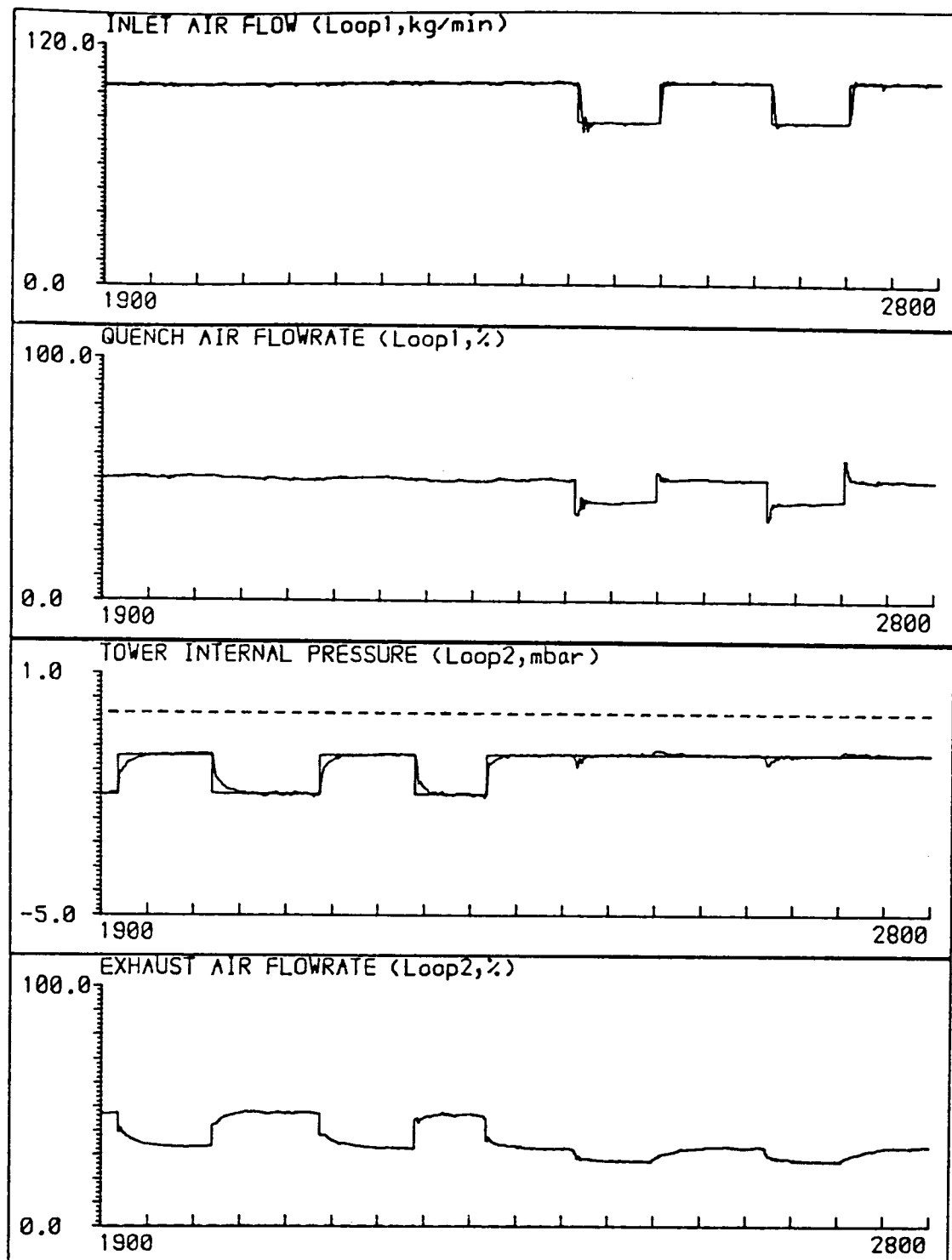


Figure 5.15: Distributed GPC control, no feed-forward.

From the results plotted in Fig. 5.15 it is clear that the overall control performance of this simplified DGPC control strategy is of a very high quality and virtually indistinguishable from that obtained using full MGPC (5.8). In particular despite the assumption, implicit in this control structure, that there is no interaction, the decoupling of the tower pressure measurement from that of air flow is excellent. Variation in quench air damper position (during set-point changes in air flow) still caused a change in tower pressure but this was transient and quickly removed by the load disturbance rejection properties of the GPC controller around the pressure loop.

One other important observation may be made from the results presented in Fig. 5.15. During the earlier trials of MGPC (section 5.4.4) an air flow change of $100 \rightarrow 80$ kg/min

required an approximate 10% change in the exhaust air for steady-state decoupling of the pressure loop. However in this experiment a similar air flow change required only a 5% change in exhaust air, indicating a reduction in magnitude of the interaction by a factor of 2. As intuitively a lower gain process is easier to control (especially in the presence of actuator non-linearity) this both illustrates the hazards of comparing algorithms in practice and also clearly demonstrates that the process is time-varying, one of the major justifications for the use of adaptive control. This apparent variation in process gain may be observed throughout all the trials and whilst it was never satisfactorily explained it was speculated that either the pressure sensor was itself responsible or sensitive to changes in ambient atmospheric conditions.

5.6.2 Distributed GPC control, with feed-forward

This experiment was a repeat of that performed in the previous section but with the GPC controller on the pressure loop modified to include a feed-forward term. This feed-forward connection consisted of the control signal from the air flow loop, the quench air damper position, that was shown to cause interaction in section 5.3.1. It was thus included in the tower pressure control calculation as a *measurable load disturbance*. This feed-forward extension represents the only difference between this controller and that of the previous section, all the DGPC design parameters remaining at their default specifications. The results are shown in Fig. 5.16.

Closed-loop control of both loops is very tight with the interactive effect of quench air flow upon tower pressure being almost completely removed. Note that the controller also succeeded in rapidly eliminating a large unidentified load disturbance⁷ (U.L.D) upon the tower pressure later in the run (1).

Once again absolute comparison with the preceding control strategies must be cautious as in this particular experiment steady state de-coupling of the pressure loop, during a step change in air flow of 20%, requires only an approximate 5% change in exhaust air. Nonetheless the response of the exhaust air actuation is significantly faster and more accurate than that of DGPC without feed-forward. Thus feed-forward has been demonstrated to offer an effective means of overcoming interaction in a multivariable system.

⁷During later trials it became apparent that large transient load disturbances upon tower pressure, similar to that indicated in Fig. 5.16, were apparently being caused by the opening and closing of an emergency door in very close proximity to the base of the tower, similar to the so-called 'open-door' effect of Hodgson (1982).

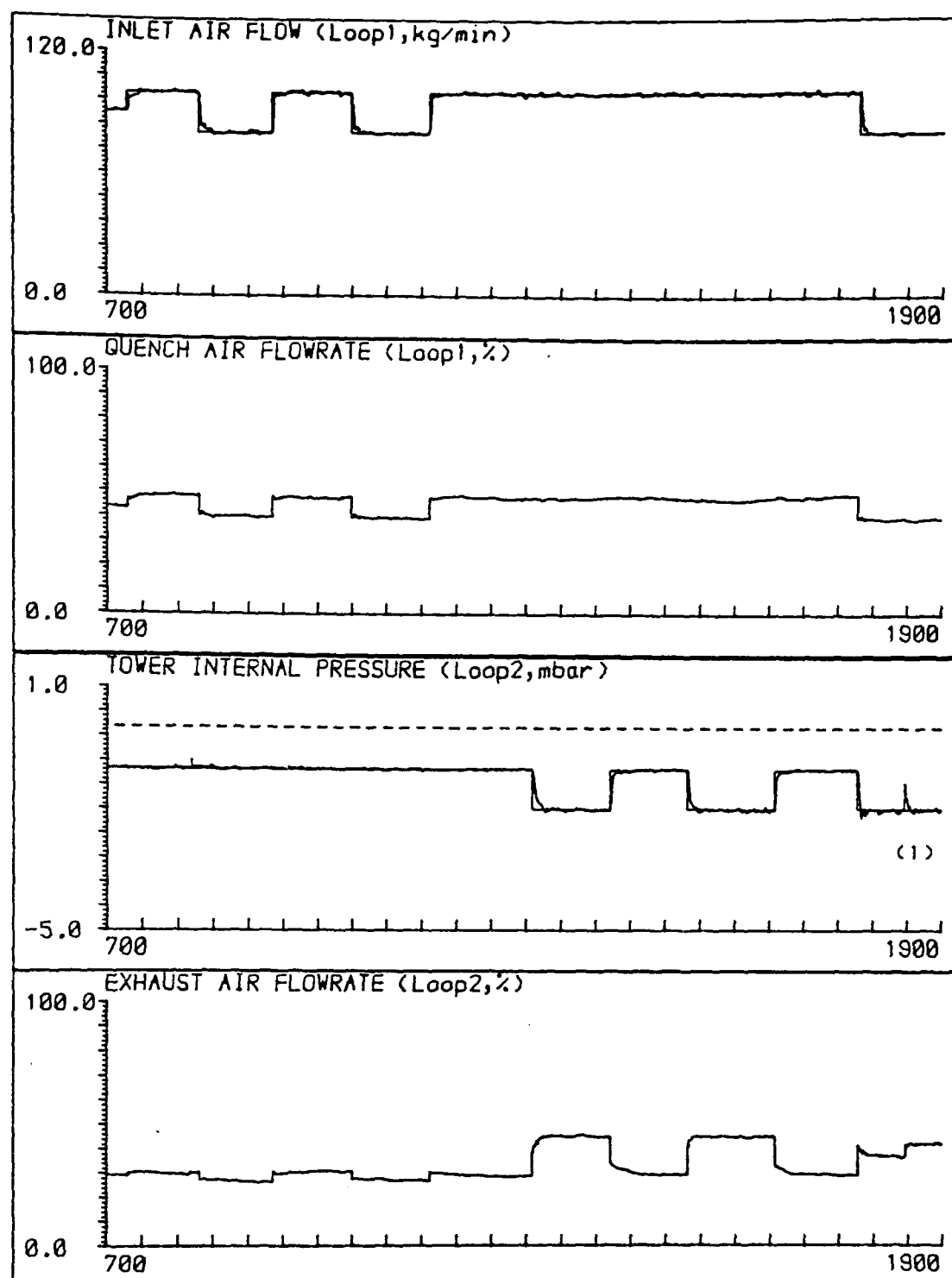


Figure 5.16: Distributed GPC control, with feed-forward.

5.6.3 Distributed GPC control, enhanced feed-forward

The one-way nature of the interaction between the spray drying tower inner loops enables the *enhancement* of the feed-forward structure of the previous section by ordering of the control calculations (see section 2.5.2). Thus the quench air control signal was calculated first and then its *new* value used as the feed-forward term in the exhaust air control calculation, rather than being proxied by its previous value. In principle this should improve dynamic decoupling of the tower pressure because it takes immediate account of changes in the quench air input.

The results obtained using enhanced feed-forward DGPC are shown in Fig. 5.17. Control is again very tight, with almost complete decoupling of both loops. Although it does

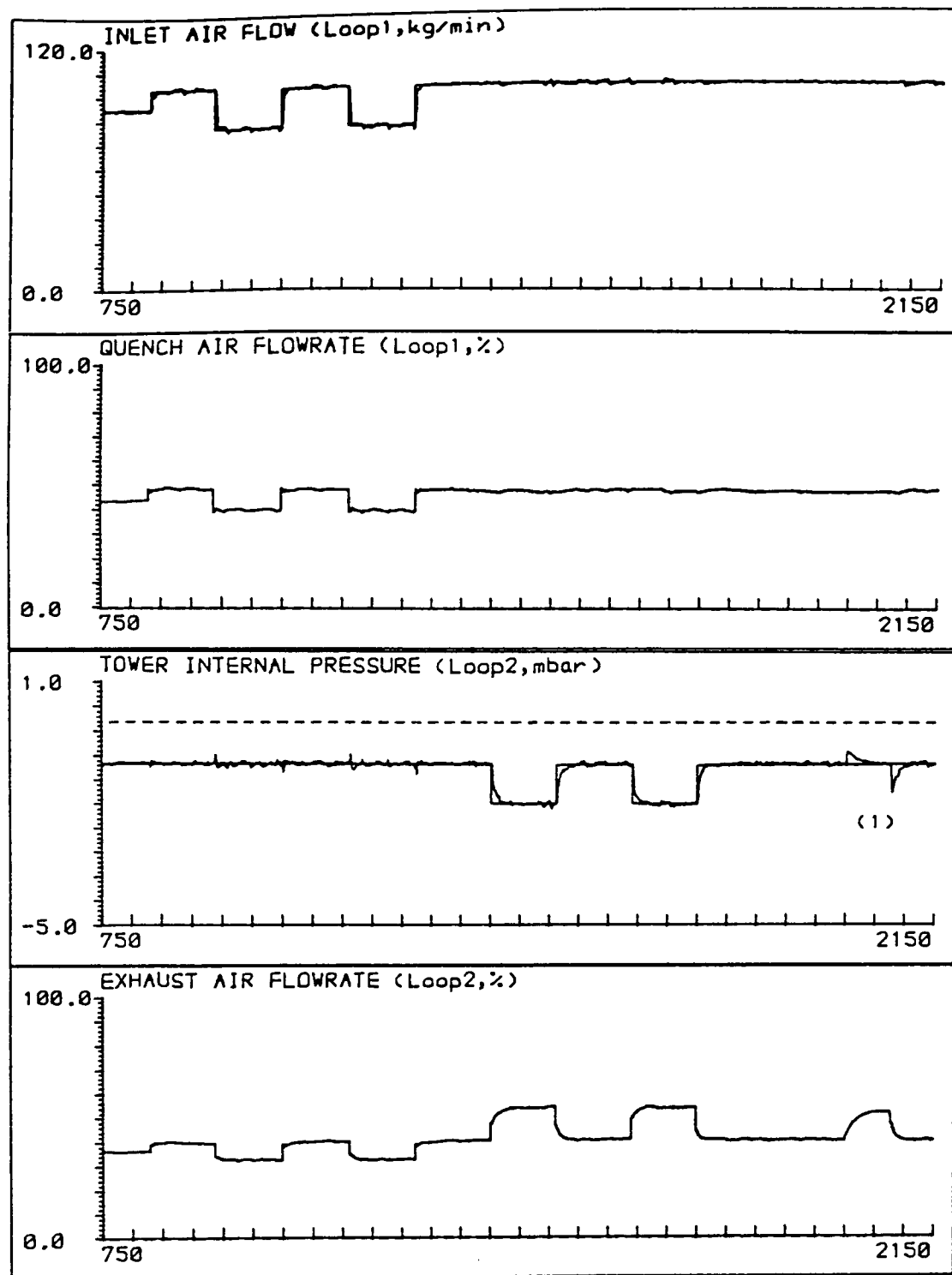


Figure 5.17: Distributed GPC control, with enhanced feed-forward.

not appear to be significantly better than ordinary feed-forward for this process absolute comparison is difficult because decoupling required a step change of $\approx 10\%$, as compared to $\approx 5\%$ during the previous experiment, i.e. the effect of the interaction was greater.

5.7 MULTIVARIABLE PARAMETER ESTIMATION

Up to this point the results presented have emphasised the *control* performance of the adaptive GPC algorithm during practical experiments conducted on the spray drying tower's inner loops of air flow and pressure. The remainder of the results presented in this chapter deal specifically with the performance of *estimation* throughout these trials, i.e. the basis of adaptive control.

The estimator used throughout all the experiments upon the tower was the well-known *recursive least-squares* algorithm (Young, 1968; Harris & Billings, 1981). This was modified by the inclusion of a scalar variable forgetting factor (Fortescue *et al*, 1981) that by attempting to keep the sum of squares of the estimation errors constant over a pre-specified 'window' of data always retains a certain degree of adaptivity in the estimator (see Chapter 2, section 3.4.2 for further details).

The 'full' multivariable GPC controller used was a 2×2 implementation and thus based upon the multivariable CARIMA plant model representation shown below :

$$\overbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}^A \begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \overbrace{\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}}^B \begin{bmatrix} \Delta u_1(t-k) \\ \Delta u_2(t-k) \end{bmatrix} + \begin{bmatrix} \xi_{11} & \xi_{12} \\ \xi_{21} & \xi_{22} \end{bmatrix} \quad (5.5)$$

where for the spray drying tower it was assumed that the measured variables correspond to:

$$\begin{array}{ll} y_1 & \text{— Air flow} \\ y_2 & \text{— Tower pressure} \end{array} \quad \begin{array}{ll} u_1 & \text{— Quench air damper} \\ u_2 & \text{— Exhaust air damper} \end{array}$$

✕ The estimation calculation was split into two separate 'loops' in order to allow different estimator parameters such as filters to be specified for different outputs, thus

$$\hat{A}_{11}\Delta y'_1(t) + \hat{A}_{12}\Delta y'_2(t) = \hat{B}_{11}\Delta u'_1(t-1) + \hat{B}_{12}\Delta u'_2(t-1) \quad (5.6)$$

$$\hat{A}_{21}\Delta y''_1(t) + \hat{A}_{22}\Delta y''_2(t) = \hat{B}_{21}\Delta u''_1(t-1) + \hat{B}_{22}\Delta u''_2(t-1) \quad (5.7)$$

This means that the regression vectors ϕ' and ϕ'' are not necessarily the same and thus effectively there are two estimators, one associated with air flow and the other with tower pressure. In practice the specifications of both estimators were always chosen to be the same as the characteristics of the air flow and tower pressure loops were similar.

On the basis of the step response tests carried out in section 5.3.1 it was decided that a reasonable model parameterisation would be the estimation of 3 B parameters and 1 A parameter for each 'channel' (see Fig. 5.2), i.e.

$$\begin{bmatrix} (1 + \hat{a}_1 q^{-1})_{11} & (\hat{a}_1 q^{-1})_{12} \\ (\hat{a}_1 q^{-1})_{21} & (1 + \hat{a}_1 q^{-1})_{22} \end{bmatrix} \begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \begin{bmatrix} (\hat{b}_0 + \hat{b}_1 q^{-1} + \hat{b}_2 q^{-2})_{11} & (\hat{b}_0 + \hat{b}_1 q^{-1} + \hat{b}_2 q^{-2})_{12} \\ (\hat{b}_0 + \hat{b}_1 q^{-1} + \hat{b}_2 q^{-2})_{21} & (\hat{b}_0 + \hat{b}_1 q^{-1} + \hat{b}_2 q^{-2})_{22} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} \quad (5.8)$$

where no dead-time is directly assumed but the B parameters are extended to cope with a possible unit sample dead-time observed during the aforementioned step response test. The $()_{ij}$ notation is used to distinguish between the polynomials estimated for each of the channels. This parameterisation was used successfully throughout all the control trials using full multivariable GPC.

ESTIMATOR SPECIFICATION			Initial parameters				
	Loop 1	Loop 2	Ch.	b_0	b_1	b_2	a_1
Σ_0	100	100	11	1	0	0	0
β_{min}	0.95	0.95	12	0	0	0	0
T_{est}	$1 - 0.8q^{-1}$	$1 - 0.8q^{-1}$	21	0	0	0	0
$P(0)$	10I	10I	22	1	0	0	0

Table 5.4: Estimator specification and initial parameters.

The detailed specification of the RLS estimation algorithm used during the trial described in this section may be considered as involving two distinct elements :

- Specification of the basic algorithmic parameters that determine estimator performance during normal adaptive operation, such as filters, dead-bands, and forgetting factors.
- Specification of the initial parameter estimates and the initial P matrix.

The exact specifications are given in Table 5.4. The initial values of $P_1(0) = 10I$ and $P_2(0) = 10I$ are relatively large values that ensure active adaptation during the initial period of estimation. Estimator ‘jacketing’ consisted simply of a discrete filter, T_{est} , in both loops that by filtering the data in the regression vector reduces the effect of noise and unmodelled dynamics upon the estimated parameters. It was chosen deliberately to correspond in bandwidth to the T polynomials used in the control calculation. Estimation dead-bands were not considered necessary because of the already high signal to noise ratio of the air flow and pressure measurements.

The constant Σ_0 determines the behaviour of the variable forgetting factor, and thus the adaptivity of the estimator, during long term operation. The value of $\Sigma_0 = 100$ was chosen for both loops to provide relatively slow adaptation and was a cautious practical choice that should, in principle, allow tracking of slow parameter changes whilst not being too

sensitive to measurement noise and/or load disturbances. A minimum forgetting factor of $\beta_{\min} = 0.95$ was also specified to prevent the variable forgetting factor going to zero and causing a division by zero.

These seemingly arbitrary values may be best considered as a *default* estimator specification that, in a similar fashion to the default specification of GPC's design parameters, is expected to provide robust satisfactory performance for a wide variety of processes, with the emphasis upon *rapid* initial adaptation and *robust* long-term operation.

Because of the numbers of parameters estimated and the generally large amount of the data generated by multivariable estimation the estimator performance is only described for one representative trial of full multivariable GPC. Because this particular experiment was used as the basis for the general experimental procedure employed in all subsequent trials, and included all the major elements of adaptive control from self-tuning to load disturbance rejection, the performance of the estimator may also be considered as typical of that encountered throughout the trials.

5.7.1 Estimated parameters

Fig. 5.18 records the parameter evolution that occurred during the full multivariable GPC control of air flow and tower pressure previously described in section 5.4.1, with the estimator specified as above. In the diagram the 3 B parameters and the A parameter estimated for each 'channel' (see Fig. 5.2) are plotted together, with the scales of the plot chosen to be the same as an aid to comparison.

Initially the parameters are constant at their initial values whilst the inner loops of the tower are being manually controlled to their operating points and estimation is not enabled. Then a PRBS signal was simultaneously superimposed upon both the quench and exhaust air damper manual control signals and estimation enabled at time $t = 226$. The response of the parameters is immediate and very active with large transient variations in all channels. However this active adaptation quickly dies away as the parameters converge and after 20 samples there is only minor parameter variation for the remainder of the PRBS assisted self-tuning phase. One notable exception is the $(\hat{a})_{12}$ parameter which did not appear to converge until the loop was closed with GPC control at time $t = 510$.

Subsequent parameter variations during closed loop control were small, with most parameters remaining almost constant despite extensive control activity throughout the ex-

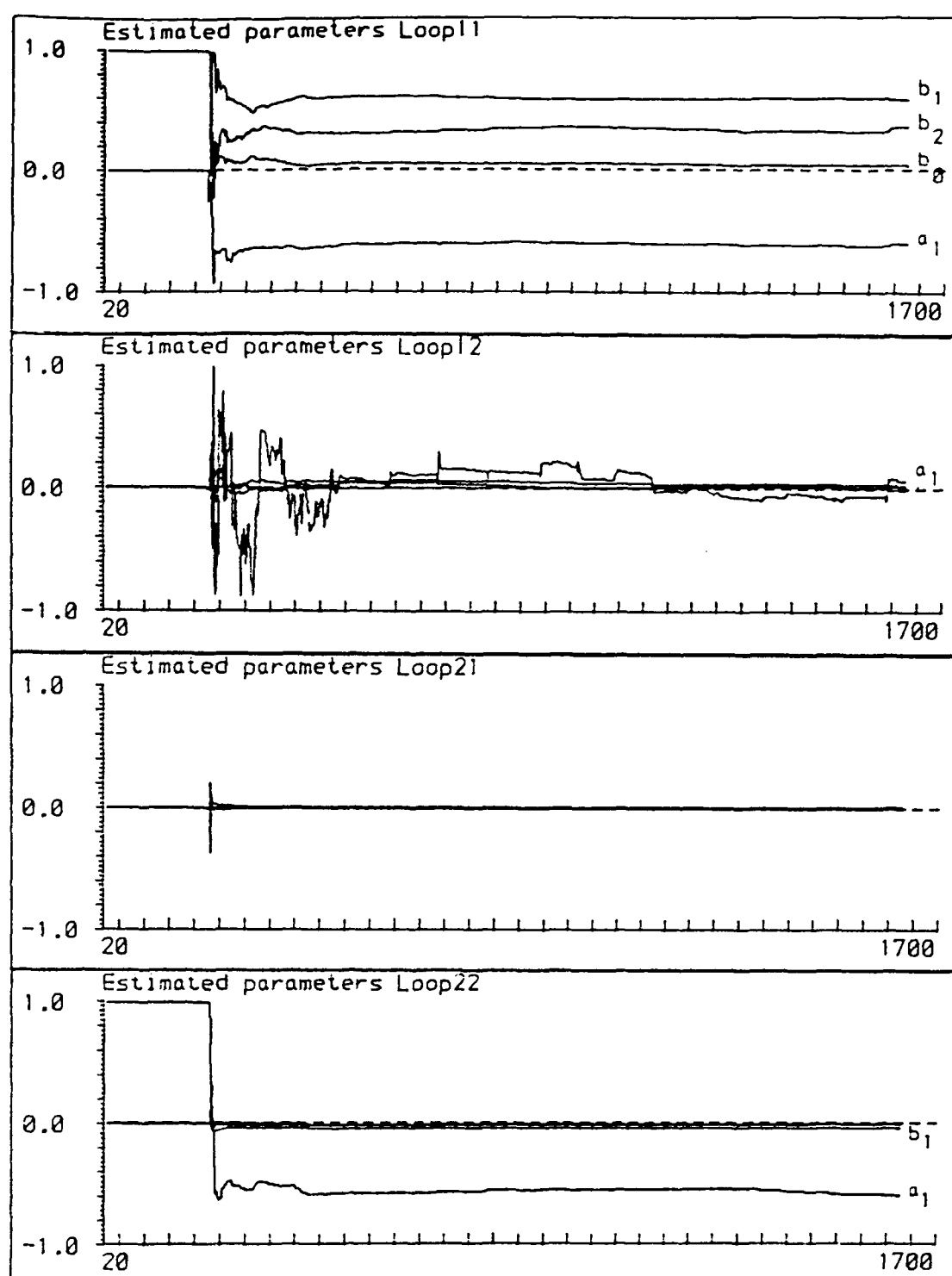


Figure 5.18: Estimated parameters: Full multivariable GPC.

periment which included set-point following for both outputs and the rejection of large load disturbances upon tower pressure (see Fig. 5.5). The average values of the parameters, calculated over the entire period of closed loop control (and thus not including the transient behaviour during initial PRBS self-tuning), are given in Table 5.5.

From this table it may be observed that in the majority of channels the $\hat{b}_0$ parameters are significantly smaller than their corresponding $\hat{b}_1$ and $\hat{b}_2$ parameters. This is a clear indication of the existence of a unit sample time-delay in these channels which verifies an identical observation made on the basis of results obtained during the step response test of section 5.3.1. The quality of control obtained using these parameters shows that extension of the B parameters in the estimated model was effective in dealing with the presence of

Estimated Parameters					
Ch.	$\hat{\theta}$	Average	Ch.	$\hat{\theta}$	Average
11	b_0	4.976×10^{-2}	12	b_0	2.197×10^{-2}
	b_1	5.905×10^{-1}		b_1	5.059×10^{-2}
	b_2	3.285×10^{-1}		b_2	1.535×10^{-2}
	a_1	-5.982×10^{-1}		a_1	5.494×10^{-2}
21	b_0	2.179×10^{-3}	22	b_0	-5.236×10^{-3}
	b_1	1.870×10^{-2}		b_1	-3.916×10^{-2}
	b_2	1.060×10^{-2}		b_2	-1.115×10^{-2}
	a_1	-4.264×10^{-3}		a_1	-5.602×10^{-1}

Table 5.5: Average values of the estimated parameters (MGPC).

unknown time-delay.

5.7.2 Adaptivity of the estimator

Fig. 5.19 shows the determinant⁸ of the P matrix and the value of the variable forgetting factor for each of the two estimators, one associated with air flow (loop1) and the other with tower pressure (loop2). As this diagram shows both determinants were initially very high, corresponding to the period before estimation when the initial parameter estimates (see Table 5.4) are assumed to be very inaccurate. However once estimation is enabled and the parameters begin to converge the determinant of P in both loops, and thus the gain of the estimator, falls rapidly.

Whilst the P matrices of both loops seem to respond in a very similar fashion during this initial PRBS assisted self-tuning phase the variation of the variable forgetting factors are markedly different. $\beta_1(t)$ is very active and consistently less than one, thus preserving the adaptivity of the estimator during this period of 'rich' data, whilst $\beta_2(t)$ is almost constant at unity. The effect of this differing behaviour is shown by the determinant of P_1 matrix remaining significantly higher than that of P_2 . This contrast continues throughout the entire trial with $\beta_1(t)$ reacting to control activity and maintaining a reasonable degree of adaptivity in loop 1 while $\beta_2(t)$ remains almost constant at unity and thus, with no

⁸ a useful measure of the adaptivity or 'gain' of the estimator and the expected accuracy of the parameter estimates (see Chapter 3, section 3.3.4).

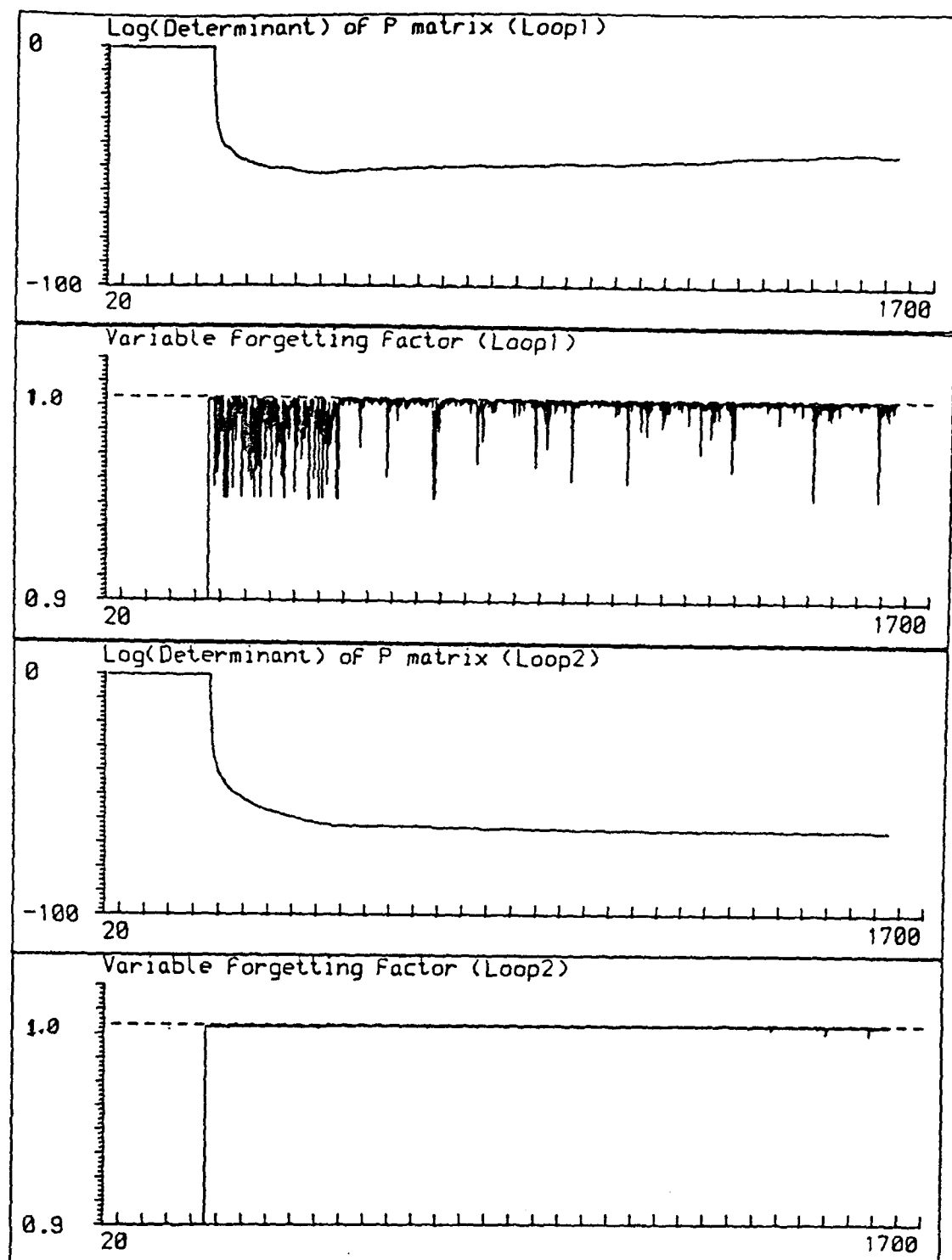


Figure 5.19: Adaptivity of the estimator.

forgetting, the adaptivity of loop 2 continuously declines.

This different behaviour of the forgetting factors in each loop, despite being specified identically at the beginning of the experiment (Table 5.4), was caused by the different magnitudes of the corresponding measured variables. The calculation of $\beta(t)$ for each loop is based upon the estimation errors associated with that loop (Chapter 3, section 3.4.2) and this error is an order of magnitude greater for the air flow loop than the pressure loop simply because of the units used. This causes the desired active variation of $\beta_1(t)$ but because of the far smaller estimation error associated with tower pressure $\beta_2(t)$ fails to respond and thus estimation becomes progressively more ineffective in that loop.

Thus Fig. 5.19 represents an important result as it not only shows the potential effec-

tiveness of RLS estimation with variable forgetting (loop 1) but also demonstrates that specification of such an estimation error based algorithm requires great care, with the same 'default' values for both loops giving widely differing performance (loop 2). Despite this problem the overall estimator performance throughout the trial, with reference also to Fig. 5.18, was satisfactory with the same arbitrary large value of $P(0)$ causing rapid and accurate initial adaptation in both loops that quickly reduced in gain when the parameters had converged.

This problem, although not significantly degrading the performance of MGPC control throughout this relatively brief trial, could have been overcome in two different ways :

- By specification of a much smaller value of Σ_0 for the pressure loop.
- By scaling the measured variables used in the estimation such that the estimation errors driving both loops are of a similar magnitude. Thus the same value for Σ_0 should result in similar estimator performance in both loops.

The latter approach corresponds almost exactly to that required by the full multivariable GPC control calculation in order to preserve the *symmetry* of the control performance for both outputs (section 5.4). For this reason it was considered more appropriate than the former method and a scaling factor of 100 (i.e. the same as for the control calculation) used throughout most of the subsequent GPC trials, where it was found to dramatically improve estimator performance in the tower pressure loop.

5.7.3 Verification of estimated parameters

In order to check the validity of the parameters that were estimated during this trial (Fig. 5.4.1) the average parameter values given in Table 5.5 are used in this section as the model basis of a simulation experiment. To compare qualitatively the characteristics of the estimated model with that of the actual spray drying tower the open loop step response tests of Fig. 5.3 were repeated upon this model using simulation routines provided by the FAUST package. The results of this test are recorded in Fig. 5.20.

It is clear from this diagram that the behaviour of the estimated model corresponds very closely to that observed for the actual spray drying tower (Fig. 5.3). Thus decreasing the quench air input results in a fall in air flow and tower pressure whilst decreasing exhaust air causes a sizeable increase in tower pressure without having any significant effect upon

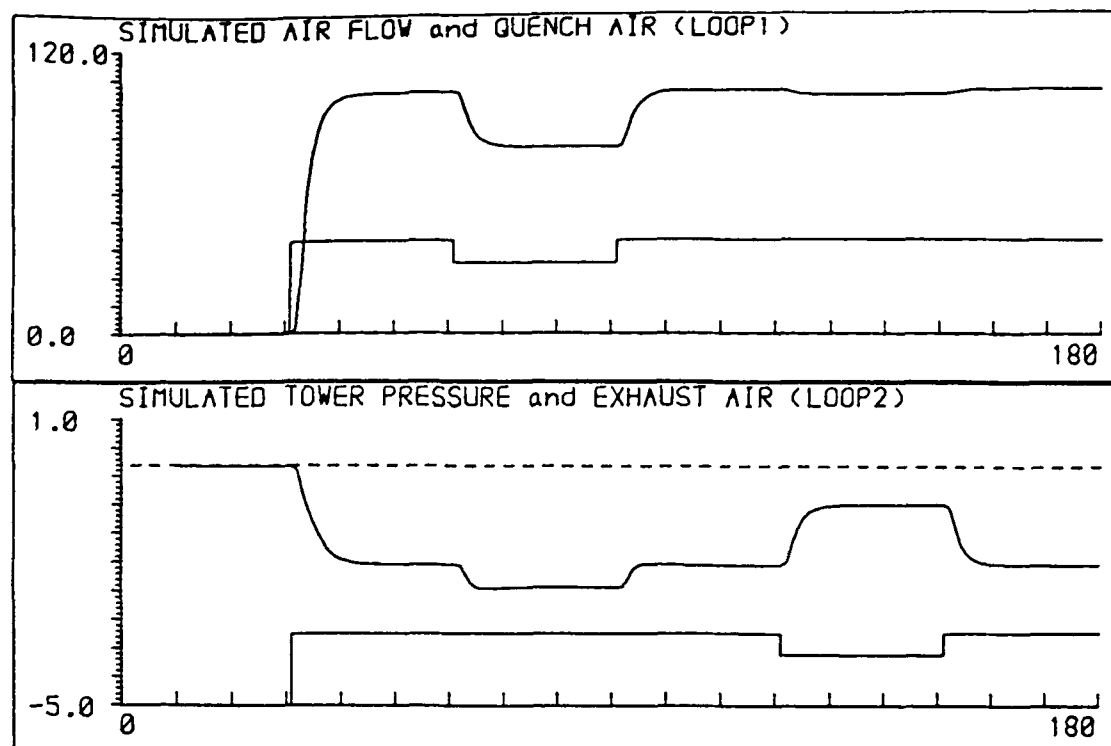


Figure 5.20: Verification of estimated model parameters.

air flow. Therefore the recursive least-squares estimator has successfully identified the fundamental structure and dynamics of the spray drying tower inner loops.

5.8 DISTRIBUTED PARAMETER ESTIMATION

The control performance of the 'distributed' GPC structure, involving the application of two separate SISO GPC controllers to the air flow and tower pressure loops respectively, was described in section 5.6. In this section the performance of the associated distributed parameter estimation is examined for a typical experiment where the quench air damper position was used as a feed-forward signal for the tower pressure controller in an attempt to improve the decoupling performance. As the interaction between exhaust damper position and air flow was insignificant it was ignored and no feed-forward term used in the air flow controller.

As for the multivariable case the dynamics of each channel were assumed to be first order. Thus 3 B and 1 A parameters were estimated for the air flow controller and 3 B , 2 A and 2 D parameters estimated for the tower pressure controller⁹, the D parameters corresponding to feed-forward terms, i.e.

$$\begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \begin{bmatrix} \left(\frac{\hat{B}(q^{-1})}{1 - \hat{a}_1 q^{-1}} \right)_{11} & 0 \\ \left(\frac{\hat{D}(q^{-1})}{1 - \hat{a}_1 q^{-1}} \right)_{21} & \left(\frac{\hat{B}(q^{-1})}{1 - \hat{a}_1 q^{-1}} \right)_{22} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} \quad (5.9)$$

⁹note that for this parameterisation to correspond exactly to that used for MGPC would require the estimation of 4 B 's, 2 A 's and 4 D 's. In practice the reduced order parameterisation given proved satisfactory.

The estimation algorithms used in both of the loops were both identical to the algorithm used for MGPC, i.e. RLS with scalar variable forgetting. In addition, to further facilitate comparison, each of the estimators was specified and initialised exactly as in Table 5.4.

5.8.1 Estimated parameters

Fig. 5.21 records the parameter evolution in the two estimators, with 'loop 1' corresponding to the GPC air flow controller and 'loop 2' the GPC tower pressure controller respectively. These results were obtained using the specifications outlined above, and during the DGPC control trial shown in Fig. 5.16.

Once again all the parameters tune quickly when estimation is enabled at time $t = 307$ and a PRBS sequence used to excite the process, with convergence occurring within 50 samples of estimation. Despite estimation being enabled for the entire duration of the experiment subsequent parameter variations are of a very small magnitude due to the specification of a large Σ_0 corresponding to low-gain adaptation. The average values of the parameters during this period of closed loop DGPC control are given in Table 5.6.

Estimated Parameters					
Loop	Parameter	Value	Loop	Parameter	Value
1	b_0	1.097×10^{-1}	2	b_0	-1.130×10^{-2}
	b_1	5.886×10^{-1}		b_1	-2.784×10^{-2}
	b_2	2.593×10^{-1}		b_2	3.430×10^{-3}
	a_1	-6.224×10^{-1}		a_1	-8.803×10^{-1}
				a_2	2.550×10^{-1}
				d_0	1.461×10^{-3}
				d_1	1.528×10^{-2}

Table 5.6: Average values of the estimated parameters (DGPC).

5.8.2 Adaptivity of the estimators

Fig. 5.22 shows the determinant of the P matrix and the value of the variable forgetting factor, $\beta(t)$, for each of the two DGPC control loops. As in the full multivariable estimation (5.7.2) the initial self-tuning performance of the estimators is similar with a rapid drop

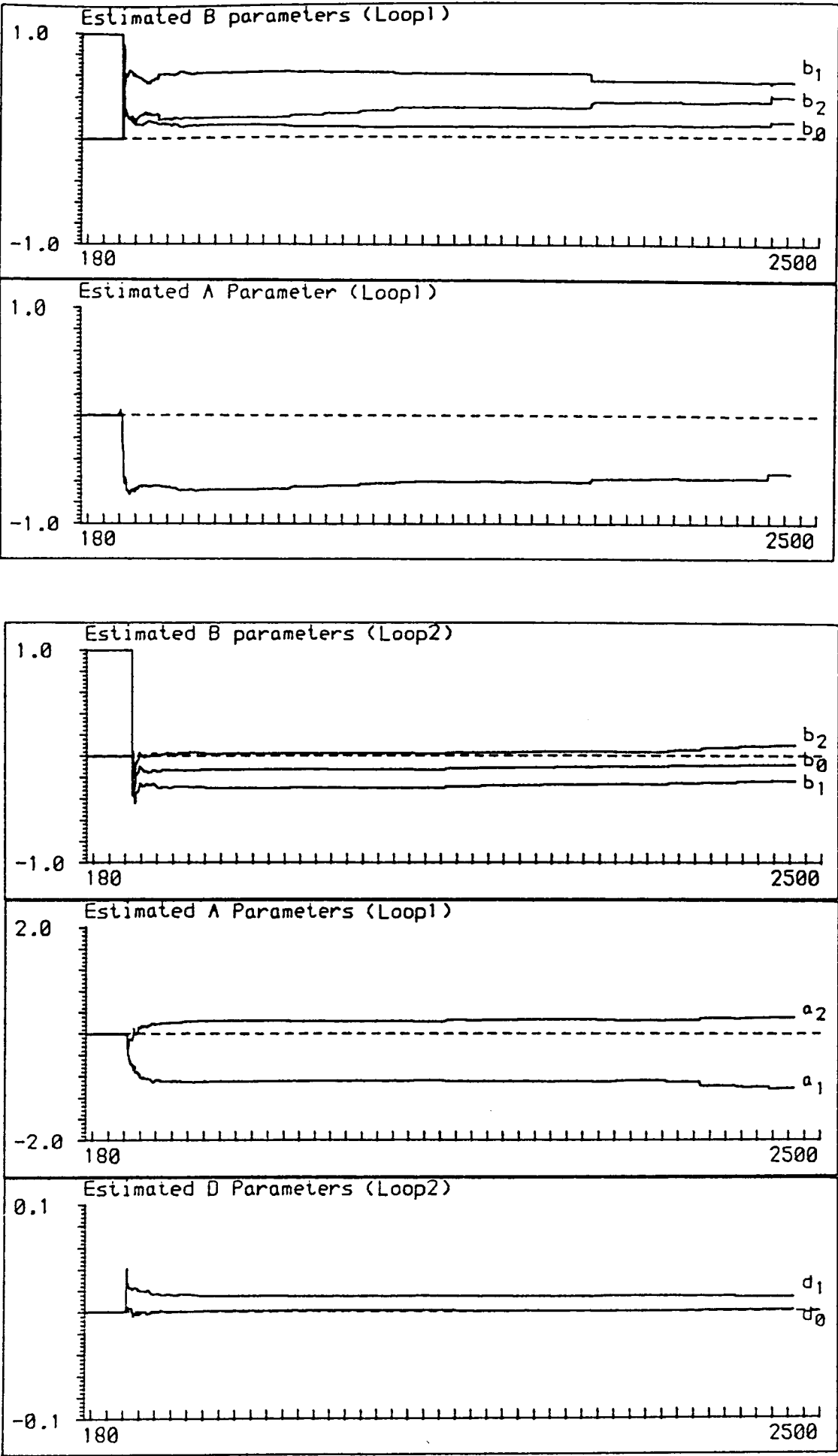


Figure 5.21: Estimated parameters: Distributed GPC with feedforward.

in the P matrix determinant. Subsequently in closed loop, without exciting data, the estimator in loop 1 retains adaptivity whilst that in loop 2 gradually 'falls asleep'. This is once again due to the specification of the same default value for Σ_0 in each of the loops and the different size of the measured variables. In later trials this was overcome by scaling the tower pressure measurement up by a factor of 100.

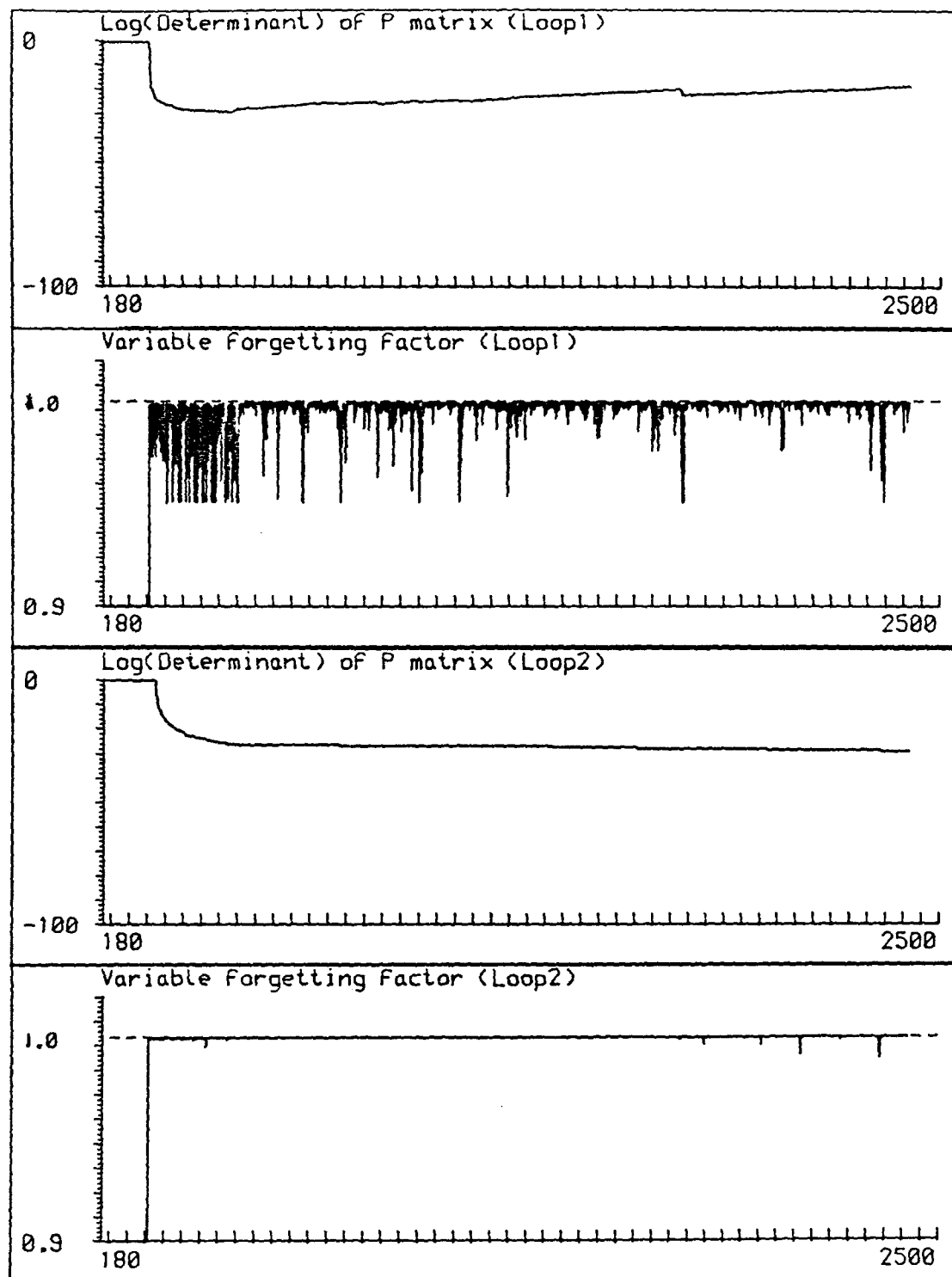


Figure 5.22: Adaptivity of the estimators

5.8.3 Verification of estimated parameters

Using the values for the estimated parameters given in Table 5.6 a simulation was conducted in order to test their validity. The results of this test are plotted in Fig.5.23 and shows that the fast process dynamics and the presence of the one-way interaction have

again been successfully identified.

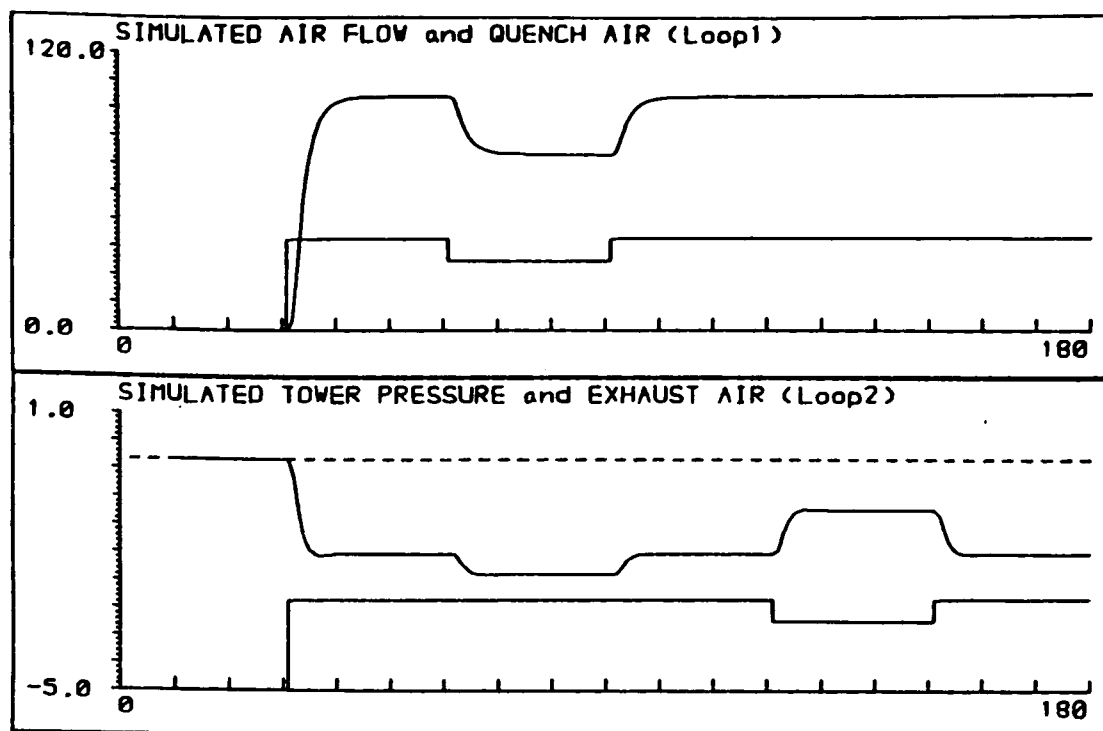


Figure 5.23: Verification of the estimated parameters.

5.9 CONCLUSIONS

The trials described in this chapter represent one of the first applications of an adaptive controller based upon the long range predictive GPC algorithm to the control of a *multivariable* industrial process.

The process used as the basis for these tests was a pilot-scale Unilever Research spray drying tower with air flow and tower pressure the measured variables to be controlled. From results obtained in section 5.3 this process may be summarised as a multivariable control problem with relatively simple dynamics but with a large one-way interaction between loops, time-delay, non-linearity, and with the tower pressure measurement subject to large unmeasurable load disturbances. It also became clear during the trials that the characteristics of the process varied significantly from day to day. For these reasons the control of the spray drying tower was considered to be a realistic industrial test of the multivariable MGPC controller despite the fact that the tower was operated 'cold'¹⁰ during most of the experiments.

During these trials the practical commissioning of MGPC consistently proved to be both straightforward and effective. Initial specification of the controller and estimator parameters was assisted by the performance of a simple step response test in open loop

¹⁰i.e. without the actual spray drying of soap using hot air.

followed by a brief period of PRBS assisted self-tuning before closing the loop with MGPC. Because of the nature of the MGPC algorithm this did not require detailed prior knowledge of the process characteristics (i.e. input/output pairing), or about the process time-delay structure (i.e. the delay-matrix). The presence of time-delay in various channels was successfully incorporated by overparameterisation of the corresponding B parameters in the estimated model.

Once commissioned the closed loop control performance of the 'default' multivariable GPC controller was demonstrated to be of a very high quality, with offset-free long term regulation, accurate set-point following, and rapid load disturbance rejection. The full multivariable structure of the control law was shown to provide effective decoupling of the pressure loop in the presence of interaction caused by control activity in the air flow loop. This control performance was compared to that obtained in routine Unilever operation using a gain scheduled multi-loop PID control scheme, and found to be clearly superior. In addition the air flow and tower pressure were successfully controlled during an experiment which used hot air and the spray injection of water to cool the tower and thus corresponded more closely to the kind of conditions that would be encountered during the actual spray drying of soap. The quality of control, given these conditions, was very similar to that obtained during trials using cold air.

Further experiments evaluated the effect of varying selected GPC control design parameters. It was successfully demonstrated that $\mathcal{P}$ and Λ could be used to 'de-tune' the control activity on-line for individual loops. Specification of $1/T$ as a low-pass filter in both loops improved the overall quality of closed loop control and also enabled tailoring of the load disturbance rejection performance of the closed loop.

Three simplified multivariable DGPC controllers were tested which used two SISO GPC controllers in a multi-loop or distributed control structure with and without feed-forward compensation. All these controllers proved to be capable of tight control of the spray drying tower with the use of feed-forward improving the decoupling performance which otherwise depended upon the load disturbance rejection properties of individual GPC control loops. Indeed the quality of DGPC control was slightly better, with feed-forward, than that obtained using MGPC. This was probably due to the relative simplicity of the estimated model and the independence of the two controllers' cost-functions, with the latter removing the requirement for scaling of the data.

In terms of estimation the Recursive Least Squares algorithm was used reliably and re-

peatedly to estimate the parameters of a model of the process during the self-tuning phase of GPC commissioning. The modification of this algorithm with the variable forgetting factor of Fortescue *et al* (1981) proved successful in retaining adaptivity of the estimator without 'blow-up'. However this was only achieved after scaling of the data in individual estimator loops, the default specification of the forgetting factor previously failing to preserve adaptivity in the tower pressure loop because of the size of the measurement units.

These conclusions may be summarised as follows :

- The adaptive implementation of the GPC algorithm enabled straightforward, rapid and effective commissioning of the controller in practice by the use of *self-tuning*.
- The performance of the 'default' multivariable GPC control algorithm was successfully evaluated under the rigour of full-scale industrial operating conditions and achieved an all-round improvement in control performance over that currently provided by the Unilever Research multi-loop gain-scheduled PID control system. In particular the superior decoupling properties of the algorithm were clearly demonstrated with the tower pressure being successfully regulated despite large variations in air flow.
- Once commissioned the multivariable GPC controller satisfied a useful range of operator specified control objectives through on-line variation of its various design parameters.
- The 'distributed' GPC control structure offers a useful and viable alternative to full multivariable GPC for the control of practical multivariable processes *when the correct input-output pairing is clear*.

Thus all the experimental objectives outlined in the introduction (5.1) were fulfilled during the trials.

Chapter 6

Conclusions

The GPC algorithm is one of the most recent self-tuning/adaptive control strategies to be developed (Clarke *et al*, 1984,87) yet it has already claimed widespread interest (Lambert, M., 1987, Favier, 1987; M'Saad *et al*, 1987; Jota, 1987). Based upon the incremental CARIMA process model (Tuffs, 1984), long-range prediction (Richalet *et al*, 1978), incorporating the control horizon of Cutler & Ramaker (1980) and including all the design parameters of GMV (Clarke & Gawthrop, 1979) it represents an amalgamation of the best features of many earlier algorithms and has thus been claimed to be particularly suitable for process control application (Mohtadi, 1986). The principal objective of the work conducted for this thesis was to investigate this claim, by analysis, simulation, and finally by conducting a series of trials upon the 'real thing', i.e. full-scale industrial processes.

Theoretical analysis of GPC, even for the fixed-parameter case when the process model is known exactly, is difficult because of its long-range predictive nature. This prevents the development of a meaningful expression for the closed-loop transfer function *in terms* of GPC controller parameters. Therefore an alternative approach was instead adopted and an algorithm developed to calculate an equivalent closed loop expression for *particular values* of the GPC controller parameters. Implemented in a software package (GPCTEST) this enabled quantitative investigation of GPC by plotting the loci of closed loop poles for a selection of control *scenarios*. By this means, and supported by simulation studies when appropriate, the effect of individual controller parameters were observed and it was verified that GPC was capable of producing a wide range of control strategies, such as 'minimum variance' and 'state dead-beat' control. Also investigated was the robustness of GPC in the presence of unmodelled dynamics, using the well-known Rohrs' model.

It was demonstrated that band-pass filtering (using the T polynomial) is vital under such circumstances and effective in preserving stability if its time-constant is chosen to correspond to the observed rise-time of the process. These results provide an increased insight into the question of how GPC should be specified in practice, with the so-called 'default' specification of a *long* prediction horizon and a unity control horizon giving a 'mean-level' type of control that, by being cautious, can reasonably be expected to provide stable, robust control for most process control applications.

Because GPC, in its default specification, is not a cancellation control law (i.e. it does not attempt to cancel process zeroes), its extension to the control of multivariable processes (Mohtadi, 1986; Mohtadi *et al*, 1986) is expected to be significantly more robust and practically applicable than earlier, GMV based, multivariable controllers. In particular this property may be used to avoid the previously restrictive requirement for *a priori* knowledge of the system delay matrix (Dugard *et al*, 1984). The development of a 'full' multivariable GPC controller (MGPC) was described as well as an alternative multi-loop, or *distributed* GPC control structure (DGPC).

These were compared for the simulated control of the spray drying tower that was to form the subject of the subsequent practical trials. It was demonstrated that, as predicted, MGPC is insensitive to time-delay and input-output pairing, but is sensitive to the magnitude of the measurements in individual loops. This latter observation follows because, in common with all algorithms based upon minimisation of a quadratic cost-function, MGPC does not directly include a measure of decoupling in its control calculation (Maurath *et al*, 1986). Conversely, however, this allows the decoupling control activity to be altered on-line simply by scaling the measured variables. The alternative, DGPC, was shown to be independent of measurement magnitude (because of the independence of the cost-functions in individual loops) and capable of effective decoupling using feed-forward.

The practical application of GPC relies upon the provision of an adequate process model, not just at the commissioning stage but throughout its entire period of use. By identifying the fundamental areas of practical estimation the well-known RLS algorithm was selected as an appropriate *prototype* estimator because of its simplicity and effectiveness. Whilst immediately suitable for the self-tuning application of GPC the basic RLS algorithm must be modified for adaptive control in the presence of time-varying process behaviour, such that it retains some form of long-term adaptivity. A selection of the state-of-the-art modifications currently available were described and their features and performance

demonstrated in simulation. All proved capable of long-term parameter tracking but at the expense of covariance 'blow-up' (fixed and directional forgetting), parameter drift (fixed and directional forgetting, constant trace) or potential sensitivity to unmeasurable load disturbances (particularly variable forgetting).

In order to overcome these problems two novel estimation algorithms were developed, based upon the extension of the RLS cost-function to include a weighted function of the estimated parameters: 'constrained' least squares (CLS) weights the difference between the estimated parameters and a *nominal* parameter set, and 'damped' least squares (DLS) weights *increments* in the estimated parameters. Both of these were found to have several interesting properties, including the inherent prevention of blow-up; reduced parameter drift; a means of specifying individual parameter adaptivity on-line; and the provision of a meaningful mechanism for incorporating detailed *a priori* process knowledge. In particular it was shown that the variation of the estimated parameters using CLS itself represents a useful fault detection *diagnostic* (i.e. a means of indicating a genuine process change). When combined with a simple fault detection procedure this was demonstrated, in simulation, to enable long-term tracking of time-varying process parameters even in the presence of significant unmeasurable load disturbances. For truly adaptive control it is proposed that this is an more *realistic* estimation philosophy than the 'continuous retention of a small amount of adaptivity' upon which most state-of-the-art algorithms are based, which by its very nature is sensitive to noise and/or load disturbances.

Another important topic considered was the use of various estimator 'jackets', such as dead-bands, normalisation of data, etc. These are *ad hoc* modifications that aim to to enhance estimator robustness in practice. Perhaps most important of these is *parameter constraint* where the most basic *a priori* information (e.g. is the process stable?) should be used to disregard incorrect parameter estimates which could otherwise lead to poor control. Finally the *expert systems* methodology is suggested to offer great potential benefits, either in a supervisory role (i.e. implementing the estimator jackets, fault detection etc.) or as an aid to commissioning.

The main part of the work in this thesis involved the practical application of GPC to actual process control problems, in collaboration with Unilever Research. This consisted of two projects: the SISO control of moisture in an industrial drying plant, and the multivariable control of an experimental spray drying tower. The first series of experiments took place during full-scale soap manufacture and were thus subject to the constraint that

installation, commissioning and evaluation of GPC should not interfere with production. In addition the residence time of the dryer introduced a large time-delay (of magnitude almost twice that of the process rise-time), the on-line measurement of soap moisture content was extremely noisy and subject to unmeasurable load disturbances and drifts, and the process itself was non-linear and often affected by saturation of the steam input.

Nonetheless GPC, in its 'default' specification, was successfully commissioned by using self-tuning, assisted by the introduction of a low-level PRBS probing sequence chosen such that the operational limits upon the soap moisture were not contravened. Within 1 hour of commencing the loop was closed and satisfactory control achieved. Unlike the existing open loop 'manual' control subsequent self-tuned GPC control was consistently offset-free (due to GPC's inherent integral action). One particularly important result was obtained when GPC successfully controlled a *soap base change*, involving the simultaneous change of type of soap entering the dryer, the input feed-rate, and the moisture set point. Using GPC the new set point was achieved far more rapidly and accurately than was previously possible using operator manual control. The fact that GPC was allowed to control the dryer during this critical period of operation is itself important because it indicates operator confidence, an important factor in the practical application of 'new' controllers. It was not possible to compare GPC with alternative conventional control strategies, such as PID, but considering the difficult nature of the process, and the significant time-delay, these trials clearly verified both the robustness of 'default' GPC, and the usefulness of self-tuning.

The second project involved the application of GPC to the multivariable inner loop control of air flow and pressure inside a spray drying tower. This process was non-linear, time-varying, subject to noise and unmeasurable load disturbances, with time-delays in individual channels and a significant *one-way* interaction from the air flow to the pressure loop. Yet after only 20 minutes of PRBS-assisted self-tuning the loop was closed with 'default' MGPC and excellent control of both measurements demonstrated, with offset free regulation; accurate set point following; and satisfactory decoupling of tower pressure. It is important to note that this was achieved with no prior knowledge of the system delay matrix (i.e. the delays in individual channels), the delays being effectively accommodated by overparameterisation of the estimated B parameters in each channel, early B 's tending towards zero as expected. Equally importantly there was no need to choose an appropriate input-output combination because of MGPC's basis upon a 'full' multivariable CARIMA model. The quality of self-tuned MGPC control was far superior to that obtained using

the existing multi-loop, gain-scheduled and manually tuned PID control system.

The absence of production constraints made it possible to demonstrate the potential use of the P , λ and T design parameters on individual loops. These all proved effective and behaved exactly as predicted with, in particular, the choice of T as a low pass filter enabling tailoring of the load disturbance rejection properties *and* dramatically improving the overall quality of control, rendering it less sensitive to noise and the inevitable unmodelled dynamics.

The relative magnitude of the measurements in individual loops (e.g. air flow: 100 kg/min, pressure: -1 mbar) which had been predicted in simulation to affect decoupling proved to be less important in practice, probably because of the presence of noise. Scaling of the data, such that the magnitude of the air flow and pressure measurements were approximately the same, did however improve decoupling as expected. The main benefit of scaling was instead observed for estimation where the 'default' specification of the variable forgetting factor (Fortescue *et al*, 1981) used in both loops failed to sustain adaptivity in the pressure loop until that measurement was scaled up.

The appropriate input-output combination for the tower was evident and easily verified using the well-known Relative Gain Array technique (Bristol, 1966). Therefore the *distributed* GPC control structure (DGPC) was also tested, using one SISO GPC controller for each of the two loops. Once again commissioning via PRBS assisted self-tuning was rapid, straightforward and effective with excellent control routinely achieved. In particular the use of feed-forward dramatically improved decoupling which otherwise depended upon the load disturbance rejection properties of the pressure loop controller. Direct comparison with MGPC is difficult because of the clearly time-varying nature of the process but nonetheless DGPC, with feed-forward, proved to be a viable alternative to the relative complexity of full MGPC. Indeed it was this controller that produced the best results during the trials.

In summary the emphasis of this thesis is on the practical evaluation of self-tuning control. Two successful applications are reported that demonstrate the general applicability of a robust modern control algorithm (GPC) combined with a RLS based estimation procedure. In both cases commissioning was (repeatedly) rapid and effective using a straightforward PRBS assisted self-tuning procedure, with control a dramatic improvement over the existing control systems (i.e. manual and PID). This was achieved without any detailed knowledge of the process. The use of self-tuning is therefore shown to provide

two benefits:

- Rapid, and therefore cheap, expert controller commissioning.
- A means of commissioning a sophisticated controller, such as GPC.

It is the former reason that has to date attracted the most attention from commercial control systems manufacturers (see Chapter 1, Table 1.1), with many now routinely offering 'auto-tuning' PID's as an option in their control systems, or even as stand-alone items. It is to be hoped that the ease with which the results obtained in this thesis were obtained will encourage them to take the next step and use self-tuning for the latter reason.

6.1 Further work

The self-tuning control strategy has much to offer, but for it to gain widespread acceptance then the trials reported here, and elsewhere (see Chapter 1, Table 1.2), are an essential pre-requisite. The unfamiliarity of control engineers with such modern algorithms as GPC is an understandable barrier that can only be overcome if it is clearly, and repeatedly, demonstrated that such an algorithm can provide improved control in practice. Thus practical industrial trials should continue to be an important part of the research effort in self-tuning control. Some additional suggestions are listed below:

- The development of some method for the meaningful specification of GPC's design parameters in engineering terms, e.g. response-time, percentage overshoot, etc.
- A more detailed examination of the effect of unmodelled dynamics upon GPC's closed loop performance and the effect of T upon robustness.
- The development of convergence and stability results for the both 'constrained' and 'damped' least squares.
- The exploitation of the fault detection properties of 'constrained' least squares with a more sophisticated algorithm, supported by practical trials.
- The incorporation of expert systems into the adaptive control environment currently provided by FAUST, and its application to the commissioning and administration of the control and estimation algorithms.

Appendix A

Experimental synopses

The following are tabular synopses of the pole variations obtained using GPCTEST in Chapter 2; and the details of GPC commissioning in the experiments of Chapter 5.

A.1 Synopses for Chapter 2

The effect of *NY*:

Pole locations					Fig.2.5
NY	Root	NY	Root	NY	Root
1	-1.49	6	0.641	11	0.740
2	0.103	7	0.673	12	0.749
3	0.393	8	0.697	13	0.756
4	0.522	9	0.715	14	0.763
5	0.594	10	0.729	15	0.768

The effect of *NU*:

Pole locations				Fig.2.6
NU	Root	NU	Root	
1	0.729	6	-0.658	
2	-0.141	7	-0.665	
3	-0.497	8	-0.668	
4	-0.602	9	-0.669	
5	-0.641	10	-1.49	

The effect of *NS*:

Pole locations			Fig.2.7
NU	NS	Root	
1	1	0.729	
1	2	0.732	
2	1	-0.141	
2	2	4.87×10^{-5}	

The effect of $P(q^{-1}) = (1 - 0.5q^{-1})/0.5$ and NY :

Pole locations						Fig.2.8		
NY	Root	Root	NY	Root	Root	NY	Root	Root
1	0.500	-1.49	6	0.697	-1.24×10^{-2}	11	0.757	-3.82×10^{-3}
2	0.539	-1.55×10^{-1}	7	0.715	-9.06×10^{-3}	12	0.763	-3.28×10^{-3}
3	0.597	-5.81×10^{-2}	8	0.729	-6.94×10^{-3}	13	0.769	-2.85×10^{-3}
4	0.641	-3.00×10^{-2}	9	0.740	-5.53×10^{-3}	14	0.773	-2.52×10^{-3}
5	0.673	-1.83×10^{-2}	10	0.749	-4.54×10^{-3}	15	0.777	-2.25×10^{-3}

The effect of λ :

Pole locations Fig.2.9		
λ	Root	Root
0.0001	0.700	-1.49
0.1	0.698	-1.25
0.25	0.695	-0.963
1.0	0.680	-0.164
10.0	$0.791 \pm j0.156$	
100.0	0.841	0.954
1000.0	0.820	1.00

The effect of $T(q^{-1})$, unmodelled dynamics:

Pole locations					Fig.2.12
$\hat{G}(q^{-1})$	$T(q^{-1})$	Root	Root	Root	
Exact	1	0.729	0.591	0.548	
Reduced order	1	$0.948 \pm j0.544$	0.844	0.263	
Reduced order	$1 - 0.95q^{-1}$	$0.850 \pm j0.185$	0.950	0.397	
Reduced order	$(1 - 0.95q^{-1})^2$	$0.921 \pm j0.0196$	$0.526 \pm j0.079$	0.943	

The effect of $T(q^{-1})$, load disturbance rejection:

Pole locations Fig.2.13	
$T(q^{-1})$	Roots
1	0.729
$1 - 0.4q^{-1}$	0.729 0.400
$1 - 0.8q^{-1}$	0.729 0.800
$(1 - 0.8q^{-1})^2$	0.729 0.801 0.799

A.2 Synopses for Chapter 5

Figures	5.5, 5.6	
Sample time	2s	
CONTROLLER	Full multivariable GPC	
Design parameter	Loop 1	Loop 2
$P(q^{-1})$	1	1
$T(q^{-1})$	$1 - 0.8q^{-1}$	$1 - 0.8q^{-1}$
λ	1×10^{-6}	1×10^{-6}
NY	10	10
NU	1	1
Scaling factor	1.0	1.0
ESTIMATOR	RLS with Fortescue <i>et al</i> 's forgetting factor	
	Loop 1	Loop 2
Estimation structure	$\delta B_{11} = 3 \quad \delta A_{11} = 1$ $\delta B_{12} = 3 \quad \delta A_{12} = 1$	$\delta B_{21} = 3 \quad \delta A_{21} = 1$ $\delta B_{22} = 3 \quad \delta A_{22} = 1$
Assumed time-delay	$\hat{k}_{11} = 1 \quad \hat{k}_{12} = 1$	$\hat{k}_{21} = 1 \quad \hat{k}_{22} = 1$
Estimator dead-bands	Lo=0.0 Hi=100.0	Lo=0.0 Hi=100.0
Estimator filter	$T_{est} = 1 - 0.8q^{-1}$	$T_{est} = 1 - 0.8q^{-1}$
Forgetting factor	$\Sigma_0 = 100$ $\beta_{min} = 0.95$	$\Sigma_0 = 100$ $\beta_{min} = 0.95$
Initial covariances	$P(0) = 10I$	$P(0) = 10I$

Figures	5.7, 5.8, 5.11 [†] , 5.12 [†] , 5.13 [†] , 5.14 [†] , 5.18, 5.19	
Sample time	2s	
CONTROLLER	Full multivariable GPC	
Design parameter	Loop 1	Loop 2
$P(q^{-1})$	1	1
$T(q^{-1})$	$1 - 0.8q^{-1}$	$1 - 0.8q^{-1}$
λ	1×10^{-6}	1×10^{-6}
NY	10	10
NU	1	1
Scaling factor	1.0	100.0
ESTIMATOR	RLS with Fortescue <i>et al</i> 's forgetting factor	
	Loop 1	Loop 2
Estimation structure	$\delta B_{11} = 3 \quad \delta A_{11} = 1$ $\delta B_{12} = 3 \quad \delta A_{12} = 1$	$\delta B_{21} = 3 \quad \delta A_{21} = 1$ $\delta B_{22} = 3 \quad \delta A_{22} = 1$
Assumed time-delay	$\hat{k}_{11} = 1 \quad \hat{k}_{12} = 1$	$\hat{k}_{21} = 1 \quad \hat{k}_{22} = 1$
Estimator dead-bands	Lo=0.0 Hi=100.0	Lo=0.0 Hi=100.0
Estimator filter	$T_{est} = 1 - 0.8q^{-1}$	$T_{est} = 1 - 0.8q^{-1}$
Forgetting factor	$\Sigma_0 = 100$ $\beta_{min} = 0.95$	$\Sigma_0 = 100$ $\beta_{min} = 0.95$
Initial covariances	$P(0) = 10I$	$P(0) = 10I$

[†]in this early experiment $T_1 = 1$ and $T_2 = 1$.
[‡]in this experiment either P , T or λ was varied, see accompanying text for details.

Figures	5.15	
Sample time	2s	
CONTROLLERS	Distributed GPC: No feed-forward	
Design parameter	Loop 1	Loop 2
$P(q^{-1})$	1	1
$T(q^{-1})$	$1 - 0.8q^{-1}$	$1 - 0.8q^{-1}$
λ	10^{-6}	10^{-6}
NY	10	10
NU	1	1
Feed-forward	No	No
ESTIMATORS	RLS with Fortescue <i>et al</i> 's forgetting factor	
	Loop 1	Loop 2
Estimation structure	$\delta B = 3 \quad \delta A = 1$	$\delta B = 3 \quad \delta A = 1$
Assumed time-delay	$\hat{k} = 1 \quad \hat{k}_v = 1$	$\hat{k} = 1 \quad \hat{k}_v = 1$
Estimator dead-bands	Lo=0.0 Hi=100.0	Lo=0.0 Hi=100.0
Estimator filter	$T_{est} = 1 - 0.8q^{-1}$	$T_{est} = 1 - 0.8q^{-1}$
Forgetting factor	$\Sigma_0 = 100$ $\beta_{min} = 0.95$	$\Sigma_0 = 100$ $\beta_{min} = 0.95$
Initial covariances	$P(0) = 10I$	$P(0) = 10I$

Figures	5.16, 5.17 [†] , 5.21, 5.22	
Sample time	2s	
CONTROLLERS	Distributed GPC: with feed-forward	
Design parameter	Loop 1	Loop 2
$P(q^{-1})$	1	1
$T(q^{-1})$	$1 - 0.8q^{-1}$	$1 - 0.8q^{-1}$
λ	10^{-6}	10^{-6}
NY	10	10
NU	1	1
Feed-forward	No	Yes
ESTIMATORS	RLS with Fortescue <i>et al</i> 's forgetting factor	
	Loop 1	Loop 2
Estimation structure	$\delta B = 3 \quad \delta A = 1$	$\delta B = 3 \quad \delta A = 2 \quad \delta D = 2$
Assumed time-delay	$\hat{k} = 1 \quad \hat{k}_v = 1$	$\hat{k} = 1 \quad \hat{k}_v = 1$
Estimator dead-bands	Lo=0.0 Hi=100.0	Lo=0.0 Hi=100.0
Estimator filter	$T_{est} = 1 - 0.8q^{-1}$	$T_{est} = 1 - 0.8q^{-1}$
Forgetting factor	$\Sigma_0 = 100$ $\beta_{min} = 0.95$	$\Sigma_0 = 100$ $\beta_{min} = 0.95$
Initial covariances	$P(0) = 10I$	$P(0) = 10I$

[†]This experiment used 'enhanced' feed-forward.

Appendix B

CLS and DLS:

Derivation and Implementation

B.1 Derivation of ‘constrained’ least squares

The ‘constrained’ least-squares algorithm extends the standard criterion function of least-squares with a term weighting differences between the estimated parameter vector, $\hat{\theta}$, and a ‘nominal’ parameter vector, θ_0 . Thus the modified RLS criterion function is given by:

$$J_c(\theta) = \sum_{k=t-N}^t \beta^{t-k} [\bar{y}(k) - \phi(k-1)^T \theta]^2 + \lambda_c(t) \|\theta(t) - \theta_0\|^2 \quad (\text{B.1})$$

where

$\bar{y}(t)$ denotes the differenced system output at time t

$\phi(t-1)$ denotes a vector of differenced data up to time t

$\theta(t)$ denotes an unknown parameter vector to be estimated

$\lambda_c(t)$ denotes the weighting upon parameter increments

β denotes the fixed exponential forgetting factor, $0 < \beta < 1$

This criterion function (B.1) may be rewritten in a matrix formulation:

$$J_c = [\bar{Y}(t) - \Phi(t-1)^T \theta(t)]^T \Omega(t) [\bar{Y}(t) - \Phi(t-1)^T \theta(t)] + [\theta(t) - \theta_0]^T \lambda_c(t) I [\theta(t) - \theta_0] \quad (\text{B.2})$$

where

$$\bar{Y}(t) \triangleq \begin{bmatrix} \bar{y}(t-N+1) & \bar{y}(t-N+2) & \cdots & \bar{y}(t) \end{bmatrix}$$

$$\Phi(t) \triangleq \begin{bmatrix} \phi(t-N) & \phi(t-N+1) & \cdots & \phi(t-1) \end{bmatrix}$$

$$\Omega(t) \triangleq \begin{bmatrix} \beta^N & & & 0 \\ & \beta^{N-1} & & \\ & & \ddots & \\ & & & \beta^0 \end{bmatrix}$$
(B.3)

and N is the number of data points considered. Minimisation of this function with respect to $\theta(t)$ gives,

$$\nabla[J_c] = -\Phi(t-1)^T \Omega(t) [\bar{Y}(t) - \Phi(t-1)^T \theta(t)] + \lambda_c(t) [\theta(t) - \theta_0] = 0 \quad (\text{B.4})$$

denoting the value of $\theta(t)$ that satisfies this equation as $\hat{\theta}(t)$:

$$\hat{\theta}(t) = [\Phi(t-1)^T \Omega(t) \Phi(t-1) + \lambda_c I]^{-1} [\Phi(t-1) \Omega(t) \bar{Y}(t) + \lambda_c \theta_0] \quad (\text{B.5})$$

The development proceeds by defining the inverse of a matrix P as,

$$P(t-1)^{-1} \triangleq \Phi(t-1)^T \Omega(t) \Phi(t-1) + \lambda_c(t) I \quad (\text{B.6})$$

which may be written recursively as

$$P(t-1)^{-1} = \beta P(t-2)^{-1} + \phi(t-1)^T \phi(t-1) + [\lambda_c(t) - \beta \lambda_c(t-1)] I \quad (\text{B.7})$$

Substituting the definition of $P(t-1)$ given in (B.6) into (B.5),

$$\hat{\theta}(t) = P(t-1) [\Phi(t-1) \Omega(t) \bar{Y}(t) + \lambda_c(t) \theta_0] \quad (\text{B.8})$$

Expanding $\Phi(t-1) \Omega(t) \bar{Y}(t)$,

$$\hat{\theta}(t) = P(t-1) [\beta \Phi(t-2) \Omega(t-1) \bar{Y}(t-1) + \phi(t-1) \bar{y}(t) + \lambda_c(t) \theta_0] \quad (\text{B.9})$$

In order to produce a recursive form of (B.9) consider

$$\hat{\theta}(t-1) = P(t-2) [\beta \Phi(t-2) \Omega(t-1) \bar{Y}(t-1) + \lambda_c(t-1) \theta_0] \quad (\text{B.10})$$

thus substituting for $\Phi(t-2) \Omega(t-1) \bar{Y}(t-1)$,

$$\begin{aligned} \hat{\theta}(t) = P(t-1) & [\beta P(t-2)^{-1} \hat{\theta}(t-1) + \phi(t-1) \bar{y}(t) \\ & + \lambda_c(t) \theta_0 - \beta \lambda_c(t-1) \theta_0] \end{aligned} \quad (\text{B.11})$$

and from (B.7),

$$\begin{aligned}\hat{\theta}(t) = P(t-1) & \left[P(t-1)^{-1} \hat{\theta}(t-1) - \phi(t-1)\phi(t-1)^T \hat{\theta}(t-1) - \lambda_c(t) \hat{\theta}(t-1) \right. \\ & \left. + \lambda_c(t-1) \beta \hat{\theta}(t-1) + \phi(t-1) \bar{y}(t) + \lambda_c(t) \theta_0 - \beta \lambda_c(t-1) \theta_0 \right]\end{aligned}\quad (\text{B.12})$$

Defining the estimation error, $\varepsilon(t)$,

$$\varepsilon(t) \triangleq \bar{y}(t) - \phi(t-1)^T \hat{\theta}(t-1) \quad (\text{B.13})$$

enables simplification,

$$\begin{aligned}\hat{\theta}(t) = \hat{\theta}(t-1) & + P(t-1) \phi(t-1) \varepsilon(t) \\ & + P(t-1) [\lambda_c(t) - \beta \lambda_c(t-1)] [\theta_0 - \hat{\theta}(t-1)]\end{aligned}\quad (\text{B.14})$$

This is the recursive parameter update equation for constrained least squares.

B.2 Implementation of ‘constrained’ least squares

The update of constrained least squares proceeds in two stages, first of all the P matrix is updated and then this is used to calculate the new parameter vector.

B.2.1 The parameter update

Given the parameter update equation of (B.14) then this is implemented simply by a matrix sum of the conventional parameter update of least squares plus an additional term $P(t-1)[\lambda_c(t) - \beta \lambda_c(t-1)][\theta_0 - \hat{\theta}(t-1)]$. Because $P(t-1)$ is stored (and updated) in a UDU^T matrix factorisation (Bierman, 1977) then it must actually be recombined before the update can take place. Storage of $\hat{\theta}(t-1)$ is also required.

B.2.2 The P matrix update

Given that

$$P(t-1)^{-1} = \beta P(t-2)^{-1} + \phi(t-1)\phi(t-1)^T + [\lambda_c - \beta \lambda_c] I$$

then let

$$P'(t-1)^{-1} = P(t-2)^{-1} + \frac{[\lambda_c(t) - \beta\lambda_c(t-1)]}{\beta} \sum_{i=1}^n r_i r_i^T \quad (\text{B.15})$$

where r_i are successive *basis* vectors e.g.

$$r_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T \quad (\text{B.16})$$

Thus the update of (B.15) consists of n rank-1 matrix updates. This is then implemented by calling a routine n times that performs the matrix inversion lemma of Goodwin & Sin (1984), in a UDU^T factorised form. This gives

$$P'(t-1) = P(t-2) + \sum_{i=1}^n \frac{P'_{i-1} r_i r_i^T P'_{i-1}}{\frac{\beta}{\lambda_c(t) - \beta\lambda_c(t-1)} + r_i^T P'_{i-1} r_i} \quad (\text{B.17})$$

The P matrix update is completed by the final stage

$$P(t-1)^{-1} = \beta P'(t-1)^{-1} + \phi(t-1)\phi(t-1)^T \quad (\text{B.18})$$

which once again may be written as a sum of rank-1 matrix updates and, using the matrix inversion lemma, may be written recursively as,

$$P(t-1) = \frac{1}{\beta} \left[P'(t-1) - \frac{P'(t-1)\phi(t-1)\phi(t-1)^T P'(t-1)}{\beta + \phi(t-1)^T P'(t-1)\phi(t-1)} \right] \quad (\text{B.19})$$

Note that by performing the overall P matrix update in this order the 'Kalman gains', i.e. $P(t-1)\phi(t-1)$ may be automatically generated by the last call to the UDU^T matrix inversion routine.

B.3 Derivation of 'damped' least squares

The 'Damped' least-squares algorithm extends the standard criterion function of the least-squares algorithm with a term weighting increments in the estimated parameter vector, $\hat{\theta}$. Thus the modified criterion function is given by:

$$J_d(\theta) = \sum_{k=t-N}^t \beta^{t-k} [\bar{y}(k) - \phi(k-1)^T \theta]^2 + \lambda_d(t) \|\theta(t) - \theta(t-1)\|^2 \quad (\text{B.20})$$

where the nomenclature is exactly the same as that used in Appendix B.1. This criterion function (B.20) may be rewritten in a matrix formulation:

$$J_d = \left[\bar{Y}(t) - \Phi(t-1)^T \theta(t) \right]^T \Omega(t) \left[\bar{Y}(t) - \Phi(t-1)^T \theta(t) \right] + [\theta(t) - \theta(t-1)]^T \lambda_d(t) I [\theta(t) - \theta(t-1)] \quad (\text{B.21})$$

Minimisation of this function with respect to $\theta(t)$ gives,

$$\nabla[J_d] = -\Phi(t-1)^T \Omega(t) [\bar{Y}(t) - \Phi(t-1)^T \theta(t)] + \lambda_d(t) [\theta(t) - \theta(t-1)] = 0 \quad (\text{B.22})$$

denoting the value of $\theta(t)$ that satisfies this equation as $\hat{\theta}(t)$:

$$\hat{\theta}(t) = [\Phi(t-1)^T \Omega(t) \Phi(t-1) + \lambda_d I]^{-1} [\Phi(t-1) \Omega(t) \bar{Y}(t) + \lambda_d \hat{\theta}(t-1)] \quad (\text{B.23})$$

Given a P matrix defined as in (B.6) then,

$$\hat{\theta}(t) = P(t-1) [\Phi(t-1) \Omega(t) \bar{Y}(t) + \lambda_d(t) \hat{\theta}(t-1)] \quad (\text{B.24})$$

Expanding $\Phi(t-1) \Omega(t) \bar{Y}(t)$,

$$\hat{\theta}(t) = P(t-1) [\beta \Phi(t-2) \Omega(t-1) \bar{Y}(t-1) + \phi(t-1) \bar{y}(t) + \lambda_d(t) \hat{\theta}(t-1)] \quad (\text{B.25})$$

In order to produce a recursive form of (B.25) consider

$$\hat{\theta}(t-1) = P(t-2) [\beta \Phi(t-2) \Omega(t-1) \bar{Y}(t-1) + \lambda_d(t-1) \hat{\theta}(t-2)] \quad (\text{B.26})$$

subtracting (B.26) from (B.25) and substituting for $\Phi(t-2) \Omega(t-1) \bar{Y}(t-1)$,

$$\begin{aligned} \hat{\theta}(t) = P(t-1) & [\beta P(t-2)^{-1} \hat{\theta}(t-1) + \phi(t-1) \bar{y}(t) \\ & + \lambda_d(t) \hat{\theta}(t-1) - \beta \lambda_d(t-1) \hat{\theta}(t-2)] \end{aligned} \quad (\text{B.27})$$

and from (B.7),

$$\begin{aligned} \hat{\theta}(t) = P(t-1) & [P(t-1)^{-1} \hat{\theta}(t-1) - \phi(t-1) \phi(t-1)^T \hat{\theta}(t-1) \\ & - \lambda_d(t) \hat{\theta}(t-1) + \lambda_d(t-1) \beta \hat{\theta}(t-1) + \phi(t-1) \bar{y}(t) \\ & + \lambda_d(t) \hat{\theta}(t-1) - \beta \lambda_d(t-1) \hat{\theta}(t-2)] \end{aligned} \quad (\text{B.28})$$

If $\varepsilon(t)$ is the estimation error (B.13) then the above may be simplified to,

$$\begin{aligned} \hat{\theta}(t) = \hat{\theta}(t-1) & + P(t-1) \phi(t-1) \varepsilon(t) \\ & + \beta P(t-1) \lambda_d(t-1) [\hat{\theta}(t-1) - \hat{\theta}(t-2)] \end{aligned} \quad (\text{B.29})$$

This is the recursive parameter update equation for damped least squares.

B.4 Implementation of 'damped' least squares

The update of damped least squares proceeds in two stages, first of all the P matrix is updated and then this is used to calculate the new parameter vector.

B.4.1 The parameter update

Given the parameter update equation of (B.29) then this is implemented simply by a matrix sum of the conventional parameter update of least squares plus an additional term $\beta P(t-1)\lambda_d(t-1)[\hat{\theta}(t-1) - \hat{\theta}(t-2)]$. Because $P(t-1)$ is stored (and updated) in a UDU^T matrix factorisation (Bierman, 1977) then it must actually be recombined before the update can take place. Storage of both $\hat{\theta}(t-1)$ and $\hat{\theta}(t-2)$ is also required.

B.4.2 The P matrix update

The update of the P matrix proceeds exactly as described for the constrained algorithm in (B.2.2).

Appendix C

Nomenclature and Symbols

C.1 General nomenclature

$A(.)$	Polynomial in terms of an argument ‘.’
$\hat{A}(.)$	Estimated polynomial
$\mathcal{A}(.)$	Polynomial matrix version of $A(.)$
δA	Order of a polynomial, the highest power of its argument
a_j	j^{th} coefficient of a polynomial $A(.)$
a_{ij}	i, j^{th} element of a matrix $\mathcal{A}(.)$
$\hat{a}_j$	j^{th} coefficient of an estimated polynomial $\hat{A}(.)$
$B(.)/A(.)$	Transfer-function as a ratio of polynomials
$A(1)$	Steady-state gain of $A(.)$
$A(0)$	Initial value of $A(.)$
$a(t)$	Value of a continuous <i>or</i> sampled signal at time t
$\{a(t)\}$	Sequence of sampled values
$a(t - j)$	Value of $a(t)$ at the j^{th} previous sample
$\bar{a}$	Mean value of a signal $a(t)$
$a'(t)$	Filtered version of $a(t)$
$a_n(t)$	Normalised version of $a(t)$
$\hat{a}(t + j t)$	j -step ahead prediction of $a(t)$ given data available at time t
$\mathbf{a}$	General vector
$\text{Cov}[A]$	Covariance of A
$E\{A t\}$	Expected value of A given data up to time t

$\ A\ $	Norm of A (square root of sum of squares)
$\nabla[A]$	Grad of A
σ_A^2	Variance of A

C.2 Principal symbols

$A(q^{-1}), B(q^{-1})$	Polynomials of the CARMA and CARIMA process model
$C(q^{-1}), D(q^{-1})$	associated with $y(t)$, $u(t)$, $\xi(t)$ and $v(t)$
$\mathcal{A}(q^{-1}), \mathcal{B}(q^{-1})$	Polynomial matrices of the multivariable CARMA and CARIMA models
$\mathcal{C}(q^{-1})$	associated with $\mathbf{y}(t)$, $\mathbf{u}(t)$, and $\mathbf{e}(t)$
C_0	Desired value for the constant trace algorithm
d	D.C. offset term
D	Diagonal factor of $P(t)$
$\mathcal{D}(q)$	Diagonal matrix, factor of $\mathcal{T}_d$
DCY	Additive disturbance on the process output
$e(t)$	A disturbance term of unspecified character
$E_j(q^{-1})$	Polynomial in the j -step-ahead Diophantine equation
$F_j(q^{-1})$	Polynomial in the j -step-ahead Diophantine equation
$f(t+j)$	The component of $y(t+j)$ that is known at time t
$\mathbf{f}$	Vector of future predicted outputs
$G_j(q^{-1})$	Polynomial, $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$
g_i	Ordinates of the unit step response of B/A or PB/A
$G(s)$	Laplace transform transfer-function
$G(q^{-1}), \hat{G}(q^{-1})$	Discrete process transfer function actual and estimated
$\mathbf{G}$	Matrix whose elements are g_i
$\tilde{G}_j$	Polynomial whose elements are g_i
$\bar{\mathbf{g}}^T$	Top row of the matrix $(\mathbf{G}^T \mathbf{G} + \lambda I)^{-1} \mathbf{G}^T$
$\bar{g}_i$	Element of $\bar{\mathbf{g}}$
$\mathcal{H}(q)$	Lower triangular unimodular matrix, factor of $\mathcal{T}_d$
I	Identity matrix
J	Value of a quadratic cost function
k	Process control signal delay in samples, not including ZOH

$\hat{k}$	Assumed process control signal delay in samples, not including ZOH
k_{min}, k_{max}	Lower and upper bounds upon the time-delay k
k_v	Process feed-forward delay in samples, not including ZOH
$K(t)$	'Kalman' gain in the recursive least squares estimator
n	Number of loops, number of estimated parameters
N	Number of samples, dimension of a vector or matrix
NU	GPC control horizon
NY	GPC final prediction horizon
NS	GPC initial prediction horizon
$P(t)$	'Covariance' matrix in the recursive least squares estimator
$P(q^{-1}), P_n/P_d$	Model following design transfer function
$\mathcal{P}, \mathcal{T}, \Lambda$	Diagonal matrix versions of P, T and λ
q^{-1}	Backward-shift operator, $q^{-1}a(t) = a(t-1)$
$q(t), r(t), s(t)$	Fault detection diagnostics
$Q(q^{-1})$	Control weighting design polynomial
r_j	j^{th} basis vector
r_T	Fault detection threshold
$R(q^{-1}), S(q^{-1})$	Polynomials of equivalent GPC c.l.t.f.
s	Laplace transform operator, $s = d/dt$
$T(q^{-1}), T_c(q^{-1})$	Controller signal conditioning design-filter
$T_{est}(q^{-1})$	Estimator signal conditioning design-filter
$T_d(q)$	Delay matrix, $T_d = \mathcal{H}(q)\mathcal{D}(q)$
T_s	Sample rate of a digital controller
$u(t)$	Control signal
$\mathbf{u}, \{u\}$	Vector or sequence of future controls
U	Upper triangular factor of $P(t)$
$v(t)$	Feed-forward signal
$w(t)$	Set-point for the process measured variable, directional forgetting factor
$\mathbf{w}$	Vector of future set-points
$y(t)$	Measured variable of the process
$\bar{y}(t)$	Differenced measured variable of the process
$\hat{y}(t+j t)$	j -step ahead prediction of y

$\mathbf{y}(t)$	Vector of measured variables
$\bar{\mathbf{Y}}(t)$	Vector of past outputs
$\alpha_1, \alpha_2, \alpha_3$	Fault detection parameters
β	Fixed forgetting factor
$\beta(t)$	Variable forgetting factor
β_{min}	Minimum value for forgetting factor
$\Gamma_j(q^{-1})$	Polynomial, the last δB elements of $G_j(q^{-1})$
$\Delta, \Delta(q^{-1})$	Differencing operator $\Delta = 1 - q^{-1}$
Δ_b	Estimation dead-band
$\varepsilon(t)$	<i>A priori</i> estimation error
$\lambda,$	Absolute control weighting for $Q(q^{-1})$, initial value for $P(0)^{-1}$
λ_{rel}	Relative control weighting for $Q(q^{-1})$
λ_c	Parameter weighting for 'constrained' least squares
λ_d	Parameter weighting for 'damped' least squares
λ_{max}	Maximum eigenvalue of the covariance matrix, $P(t)$
$\theta, \hat{\theta}(t)$	Unknown process parameter vector, and its estimate
θ_0	Nominal parameter set
$\Theta(t)$	Vector of past regression vectors
τ	Trace of the covariance matrix $P(t)$
$\eta(t)$	Normalisation factor
$\phi(t)$	Regressor vector in the RLS estimator
$\psi(t)$	Pseudo-output of the process, $\psi(t) = P(q^{-1})y(t)$
Ψ	Vector of ψ 's for individual loops
σ_n^2	Variance of the noise
Σ_0	Tuning-knob for Fortescue's and Ydstie's variable forgetting factors
$\xi(t)$	Zero mean uncorrelated random variable
$\Omega(t)$	Diagonal weighting matrix for RLS

Bibliography

- [1] ASEA (1983)
Adaptive control - a NEW age in process control: ASEA innovation
- [2] Al-Assaf, Y. (1987)
Self-tuning control: Theory and applications: D.Phil Thesis, University of Oxford, in preparation
- [3] Allidina, A.Y., Hughes, F.M. and Tye, C. (1981)
Self-tuning control for systems employing feedforward: Proc.IEE, Vol.128, Pt.D, No.6, pp.283-291
- [4] Anderson, B.D.O. (1982)
Exponential convergence and persistent excitation: 21st IEEE Conference on Decision & Control
- [5] Årzén, K-E. (1986)
Use of expert systems in closed loop feedback control: ACC
- [6] Åström, K.J. (1970)
Introduction to stochastic control theory: Academic Press
- [7] Åström, K.J. (1975)
Stochastic control theory and some of its industrial applications: Report 7501(C), Division of Automatic Control, Lund Institute of Technology
- [8] Åström, K.J. (1980)
Self-tuning regulators - design principles and applications: in 'Applications of Adaptive Control', ed. Narendra and Monopoli
- [9] Åström, K.J. (1983)
Theory and applications of adaptive control - a survey: Automatica, Vol.19, No.5, pp.471-486
- [10] Åström, K.J. (1984)
Interaction between excitation and unmodeled dynamics in adaptive control: Proc. 23rd IEEE CDC, pp.1276-1281.
- [11] Åström, K.J. and Anton, J.J. (1984)
Expert control: IFAC 9th World Congress, Budapest, Hungary
- [12] Åström, K.J. and Eykhoff, P. (1970)
System identification, a survey: Report 7006, Lund Institute of Technology

- [13] Åström, K.J. and Hägglund, T. (1984)
Automatic tuning of simple regulators with specifications on phase and amplitude margins: Automatica, Vol.20, No.5, pp.645-651
- [14] Åström, K.J., Borisson, U., Ljung, L. and Wittenmark, B. (1977)
Theory and applications of self-tuning regulators: Automatica, Vol.13, No.5, pp.457-476
- [15] Åström, K.J., Hagander, P., and Sternby, J. (1984)
Zeros of sampled systems: Automatica, Vol.20, No.1, pp.31-38
- [16] Åström, K.J. and Wittenmark, B. (1973)
On self-tuning regulators: Automatica, Vol.9, No.2, pp.185-199
- [17] Åström, K.J. and Wittenmark, B. (1980)
Self-tuning controllers based on pole-zero placement: Proc.IEE, Vol.127, Pt.D, No.3, pp.120-130
- [18] Åström, K.J. and Wittenmark, B. (1984)
Computer controlled systems: Prentice-Hall, NJ.
- [19] Bayoumi, M.M. and El-Bagoury, M.A. (1979)
Comments on self-tuning adaptive control of cement raw material: Automatica, Vol.15, pp.693
- [20] Bayoumi, M.M., Wong, K.Y. and El-Bagoury, M.A. (1981)
A self-tuning regulator for multivariable systems: Automatica, Vol.17, No.4, pp.575-592
- [21] Belanger, P.R. (1983)
On type 1 systems and the Clarke-Gawthrop regulator: Automatica, Vol.19, No.1, pp.91-94
- [22] Bierman, G.J. (1977)
Factorization methods for discrete system estimation: Academic Press, New York
- [23] Borison, U. (1975)
Self-tuning regulators - industrial applications and multivariable theory: Report 7513, Division of Automatic Control, Lund Institute of Technology
- [24] Borison, U. (1979)
Self-tuning regulators for a class of multivariable systems: Automatica, Vol.15, No.2, pp.209-215
- [25] Borison, U. and Syding, R. (1976)
Self-tuning control of an ore-crusher: Automatica, Vol.12, No.1, pp.1-7
- [26] Box, G.E.P. and Jenkins, G.M. (1976)
Time series analysis and control (Revised Edition): Holden-Day, San Francisco
- [27] Bristol, E.H. (1966)
On a new measure of interaction for multivariable process control: IEEE Trans. Autom. Control., Vol.AC-11, No.1, pp.133

- [28] Bruijn, P.M. and Verbruggen, H.B. (1984)
Model algorithmic control using impulse response models: Journal A, Vol.25, No.2, pp.69-74
- [29] Cegrell, T. and Hedqvist, T. (1975)
Successful adaptive control of paper machines: Automatica, Vol.11, No.1, pp.53-59
- [30] Chesna, S.A. and Ydstie, B.E. (1985)
Self-tuning and adaptive control of chemical processes: 4th Yale Workshop on Adaptive Control
- [31] Chien, I.L., Mellichamp, D.A. and Seborg, D.E. (1983)
Adaptive control strategies for distillation columns: ACC, San Francisco
- [32] Clarke, D.W. (1981)
Implementation of adaptive controllers: in 'Self-tuning and adaptive control' (ed. Harris and Billings) Peter Perigrinus
- [33] Clarke, D.W. (1981)
Introduction to self-tuning controllers: in 'Self-tuning and adaptive control' (ed. Harris and Billings) Peter Perigrinus
- [34] Clarke, D.W. (1982)
Model following and pole-placement self-tuners: Opt.Control App. and Methods, Vol.3, pp.323-335
- [35] Clarke, D.W. (1984)
Self-tuning control of nonminimum-phase systems: Automatica, Vol.20, No.5, pp.501-517
- [36] Clarke, D.W. and Gawthrop, P.J. (1975)
Self-tuning controller: Proc.IEE, Vol.122, No.9, pp.929-934
- [37] Clarke, D.W. and Gawthrop, P.J. (1979)
Self-tuning control: Proc.IEE, Vol.126, No.6, pp.633-640
- [38] Clarke, D.W. and Gawthrop, P.J. (1981)
Implementation and application of microprocessor-based self-tuners: Automatica, Vol.17, No.1, pp.233-244
- [39] Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987a)
Generalised Predictive Control. Part1: The basic algorithm: Automatica, Vol.23, No.2, pp.137-148 (Also OUEL Report No.1555/84)
- [40] Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987b)
Generalised Predictive Control. Part2: Extensions and interpretations: Automatica, Vol.23, No.2, pp.149-160 (Also OUEL Report No.1557/84)
- [41] Clarke, D.W., Tuffs, P.S. and Mohtadi, C. (1985)
Self-tuning control of a difficult process: 7th IFAC Symposium of Identification and System Parameter Estimation, York
- [42] Clarke, D.W. and Zhang, L. (1984)
Long-range predictive control using weighting-sequence models: OUEL report 1542/84

- [43] Cluett, W.R. (1986)
Stable adaptive control in the presence of unmodelled dynamics: Ph.D. Thesis, University of Alberta, Canada
- [44] Costin, M.H. and Buchner, M.R. (1983)
Minimum variance control for multivariable systems with different deadtimes in individual loops: IFAC Workshop on Adaptive Systems in Control and Signal Processing, San Francisco
- [45] Cutler, C.R. and Hawkins, R.B. (1987)
Constrained multivariable control of a hydrocracker reactor: ACC, Minneapolis
- [46] Cutler, C.R. and Ramaker, B.L. (1980)
Dynamic matrix control - a computer control algorithm: JACC, San Francisco
- [47] Cutler, C.R., Haydel, J.J. and Morshedi, A.M. (1983)
An industrial perspective on advanced control: AIChE Diamond Jubilee Meeting, Washington, DC
- [48] Davison, E.J. (1976)
Multivariable tuning regulators: the feedforward and robust control of a general servomechanism problem: IEEE Trans. Autom. Control, Vol.AC-21, No.1, pp.35-47
- [49] Dawkins, J. and Briggs, P.A.N. (1965)
A method for using weighting functions as system description in optimal control: IFAC Symposium, Teddington
- [50] De Keyser, R.M.C. (1985)
Adaptive microcomputer control of residence heating: CIB-W'79, Trondheim, Norway
- [51] De Keyser, R.M.C. (1986)
Adaptive dead-time estimation: 2nd IFAC workshop on Adaptive Control and Signal Processing, Lund
- [52] De Keyser, R.M.C. and Van Cauwenberghe, A.R. (1981)
A self-tuning multistep predictor application: Automatica, Vol.17, No.1, pp.167-174
- [53] De Keyser, R.M.C. and Van Cauwenberghe, A.R. (1981)
Self-tuning predictive control: Journal A, Vol.22, No.4, pp.167-174
- [54] De Keyser, R.M.C. and Van Cauwenberghe, A.R. (1982)
Simple self-tuning multistep predictors: IFAC Symposium on Identification and System Parameter Estimation, Washington
- [55] De Keyser, R.M.C. and Van Cauwenberghe, A.R. (1985)
Extended prediction self-adaptive control: 7th IFAC symposium on Identification and System Parameter Estimation, York
- [56] De Keyser, R.M.C., Van de Velde, Ph.G.A. and Dumortier, F.A.G. (1985)
A comparative study of self-adaptive long-range predictive control: 7th IFAC symposium on Identification and System Parameter Estimation,
- [57] De Larminat, P. (1985)
Adaptive control based upon instrumental variables: 7th IFAC symposium on identification and system parameter estimation,

- [58] Dexter, A.L. (1983)
Self-tuning control algorithm for single-chip microcomputer implementation: Proc.IEE, Vol.130, Pt.D, No.5, pp.255-260
- [59] Dion, J.M. and Dugard, L. (1983)
Parametrizations for multivariable adaptive systems.: IFAC workshop on Adaptive Control, San Francisco
- [60] Dion, L. and Dugard, L. (1982)
Direct adaptive control of discrete time multivariable systems.: IEEE CDC, Orlando
- [61] Dugard, L. and Dion, J.M. (1983)
Direct multivariable adaptive control for linear multivariable systems.: University of Newcastle, NSW, Report EE8319
- [62] Dugard, L., Goodwin, G.C. and Xianya, X. (1984)
The role of the interactor matrix in multivariable stochastic adaptive control.: Automatica, Vol.20, No.5, pp.701-709
- [63] Egardt, B. (1979)
Unification of some continuous-time adaptive control schemes: IEEE Trans. Autom. Control, Vol.AC-24, No.4, pp.588-592
- [64] El-Bagoury, M.A. and Bayoumi, M.M. (1980)
Multivariable self-tuning augmented regulators.: Symposium on Adaptive Systems, Bochum
- [65] Elliott, H. and Wolovich, W.A. (1982)
A parameter adaptive control structure for linear multivariable systems: IEEE Trans. Autom. Control., AC-27, pp.340
- [66] Elliott, H., Wolovich, W.A. and Das, M. (1984)
Arbitrary adaptive pole placement for linear multivariable systems.: IEEE Trans. Autom. Control, Vol.AC-29, No.3, pp.221-228
- [67] Ergrand, J.C. (1980)
Applications of multivariable control in a refinery and implementation : JACC, San Fransisco
- [68] Everleigh, V.W. (1967)
Adaptive control and optimization techniques: McGraw-Hill, New York
- [69] Eykhoff, P. (1974)
System identification: Wiley
- [70] Falb, P.L. and Wolovich, W.A. (1967)
Decoupling in the design and synthesis of multivariable control systems.: IEEE Trans. Autom. Control, Vol.AC-12, No.6, pp.651-659
- [71] Favier, G. (1987)
Self-tuning long-range predictive controllers: 10th IFAC World Congress on automatic control
- [72] Favier, G. and Hassani, M. (1982)
Multivariable self-tuning controllers based on the generalized minimum variance strategy.: 21st IEEE Conference on Decision & Control

- [73] Fortescue, T.R., Kershenbaum, L.S. and Ydstie, B.E. (1981)
Implementation of self-tuning regulators with variable forgetting factors: Automatica, Vol.17, No.6, pp.831-835
- [74] Francis, B.A. and Wonham, W.M. (1976)
The internal model principle of control theory: Automatica, Vol.12, pp.457-465
- [75] Friedlander, B. (1982)
A modified prefilter for some recursive parameter estimation algorithms: IEEE Trans. Autom. Control, Vol.AC-27, No.1, pp.232-235
- [76] Gawthrop, P.J. (1977)
Some interpretations of the self-tuning controller: Proc.IEE, Vol.124, No.10, pp.889-894
- [77] Gawthrop, P.J. (1980)
Hybrid self-tuning control: Proc.IEE, Vol.127, Pt.D, No.5, pp.229-236
- [78] Gawthrop, P.J. (1982)
Self-tuning PI and PID controllers: IEEE Conference on Applications of Adaptive and Multivariable Control, Hull
- [79] Gawthrop, P.J. (1986)
Self-tuning PID controllers: Algorithms and implementation: IEEE Trans. Autom. Control., Vol.AC-21, No.3
- [80] Gawthrop, P.J. (1987)
Continuous-time self-tuning control. Vol.1 - Design: Wiley
- [81] Goodwin, G.C. and Long, R.S. (1980)
Generalisation of results on multivariable adaptive control: IEEE Trans. Autom. Control, Vol.AC-25, No.6
- [82] Goodwin, G.C., and Payne, R.L. (1977)
Dynamic system identification: Experiment design: Academic Press, New York
- [83] Goodwin, G.C. and Sin, K.S. (1984)
Adaptive filtering, prediction and control: Prentice-Hall
- [84] Goodwin, G.C., Hill, D.J. and Palaniswami, M. (1985)
Towards an adaptive robust controller: IFAC symposium on Identification and System Parameter Estimation
- [85] Goodwin, G.C., McInnis, B. and Long, R.S. (1982)
Adaptive control algorithms for waste water treatment and pH neutralization: Optimal Control App. and Methods, Vol.3, No.4, pp.443-459
- [86] Goodwin, G.C., Ramadge, P.J. and Caines, P.E. (1980)
Discrete-time multivariable adaptive control: IEEE Trans. Autom. Control, Vol.AC-25, No.3, pp.449-455
- [87] Grimble, M.J. (1983)
LQG multivariable controllers: minimum variance interpretations for use in adaptive schemes: Report ICU/32/Nov 1983

- [88] Grimble, M.J. and Moir, T.J. (1983)
Weighted minimum variance controller for nonsquare multivariable systems: Report ICU/21, University of Strathclyde
- [89] Grimble, M.J., Moir, T.J. and Fung, P.T.K. (1982)
Comparison of WMV and LQG self-tuning controllers: IEEE Conference on Applications of Adaptive and Multivariable Control, Hull
- [90] Hägglund, T. (1983)
New estimation techniques for adaptive control: Report LUTFD2/(TFRT-1025)/1-120((1983), Lund Institute of Technology
- [91] Harris, C.J. and Billings, S.A. (editors) (1981)
Self-tuning and adaptive control: theory and applications: Peter Perigrinus Ltd.
- [92] Hayes-Roth, F., Waterman, D. and Lenat, D. (1983)
Building expert systems: Addison-Wesley, Reading, Mass.
- [93] Hesketh, T. and Sandoz, D.J. (1985)
Application of a multivariable adaptive controller: ACC, Minneapolis, MN
- [94] Hiram, Y. and Kershenbaum, L. (1985)
Overcoming difficulties in the application of self-tuning controllers: American Control Conference, Boston
- [95] Hodgson, A.J.F. (1982)
Problems of integrity in applications of adaptive controllers: D.Phil Thesis, Oxford University and OUEL report 1436/82
- [96] Ioannou, P.A. and Kokotovic, P.V. (1984)
Instability analysis and improvement of robustness of adaptive control: Automatica, Vol.20, No.5, pp.583-594
- [97] Irving, E. (1979)
Improving power network stability and unit stress with adaptive generator control: Automatica, Vol.15, pp.31
- [98] Irving, E., Falinower, C.M. and Fonte, C. (1986)
Adaptive Generalised Predictive Control with multiple reference model: 2nd IFAC workshop on Adaptive Control and Signal Processing, Lund
- [99] Isermann, R. (1980)
Practical aspects of process identification: Automatica, Vol.16, No.5, pp.575-587
- [100] Isermann, R. (1981)
Digital control systems: Springer-Verlag
- [101] Isermann, R. (1982)
Parameter adaptive control algorithms - A tutorial: Automatica, Vol.18, No.5, pp.513-528
- [102] Isermann, R. (1984)
Process fault detection based upon modelling and estimation methods: Automatica, Vol.20, No.4, pp.387-404

- [103] Isermann, R. (1985)
Parameter adaptive control systems - A review on methods and application: IFAC Symposium on Identification and System Parameter Estimation, York
- [104] Johnson, C.R., Anderson, B.D.O. and Bitmead, R.R. (1984)
A robust, locally optimal model reference adaptive controller: Proc. 23rd IEEE CDC, pp.993-997.
- [105] Jota, F.G. (1987)
The application of self-tuning control technique to a multivariable process: D.Phil Thesis, University of Oxford
- [106] Jota, F.G. and Dexter, A.L. (1986)
Self-tuning control of a chilled water cooling coil: simulated and experimental assessment: 2nd International Conf. on System Simulation in Buildings, Liege
- [107] Jota, F.G., Dexter, A.L., Fertig, L.P. and Wooster, W.G. (1986)
Distributed Pascal Plus: A high level language for low-cost distributed: OUEL Report No.1618/86
- [108] Kalman, R.E. (1958)
Design of a self-optimising control system: Trans. ASME, Vol.80, pp.468-478
- [109] Karny, M., Halouskova, A., Bohm, J., Kulhavý, R. and Nedoma, P. (1985)
Design of linear quadratic adaptive control: Theory and algorithms: Kybernetika
- [110] Keviczky, L. and Kumar, K.S.P. (1979)
The multivariable extension of Clarke's self-tuning controller: Report, Center for Control Sciences, University of Minnesota
- [111] Keviczky, L. and Kumar, K.S.P. (1981)
Multivariable self-tuning regulator with generalized cost-function: Int. J. Control, Vol.33, No.5, pp.913-921
- [112] Keviczky, L., Hetthessy, J., Hilger, M. and Kolostori, J. (1978)
Self-tuning adaptive control of cement raw material blending: Automatica, Vol.14, No.6, pp.525-532
- [113] Koivo, H.N. (1980)
A multivariable self-tuning controller: Automatica, Vol.16, No.4, pp.351-366
- [114] Kosut, R.L. and Friedlander, B. (1985)
Robust adaptive control: conditions for global stability: IEEE Trans. Autom. Contr., AC-30, pp.610-624
- [115] Kramer Jr., J.D.R. (1960)
On control of linear systems with time-lags: Information and Control, No.3, pp.299-326
- [116] Kreisselmeier, G. and Anderson, B.D.O. (1986)
Robust model reference adaptive control: IEEE Trans. Autom. Control., AC-31, pp.127-133
- [117] Kučera, V. (1979)
Discrete linear control: Wiley

- [118] Kučera, V. (1980)
Stochastic multivariable control: a polynomial equation approach: IEEE Trans. Automat. Control, Vol.AC-25, No.5, pp.913-919
- [119] Kulhavý, R. and Karny, M. (1984)
Tracking of slowly varying parameters by directional forgetting: IFAC 9th World Congress, Budapest, Hungary
- [120] Kurz, H. and Goedecke, W. (1981)
Digital parameter-adaptive control of processes with unknown dead time: Automatica, Vol.17, No.1, pp.245-252
- [121] Lam, K.P. (1980)
Implicit and explicit self-tuning controllers: D.Phil Thesis, Oxford University and OUEL report 1134/80
- [122] Lam, K.P. (1985)
On a self-tuning controller with retained and changeable memory: 24th CDC Conference, Ft. Lauderdale, Fl., USA
- [123] Lambert, M. (1987)
Adaptive control of flexible systems: D.Phil Thesis, University of Oxford, in preparation
- [124] Lammers, H.C. (1984)
Towards the practical application of self-tuning adaptive controllers: Thesis, Technische Hogeschool, Delft
- [125] Lee, T.H. and Hang, C.C. (1985)
A performance study of parameter estimation schemes for systems with unknown dead time: ACC Boston
- [126] Lelić, M.A. and Zarrop, M.B. (1987)
Generalised pole-placement self-tuning controller. Part 1: Basic algorithm: Int. J. Control., Vol.46, No.2, pp.547-568
- [127] Lelić, M.A. and Wellstead, P.E. (1987)
A generalised pole-placement self-tuning controller: Part II - an application to robot manipulator control: Int. J. Control, Vol.46, No.2, pp.569-602
- [128] Ljung, L. and Söderström, T. (1983)
Theory and practice of recursive estimation: The M.I.T. Press, Cambridge, Massachusetts
- [129] Luecke, R.H., Lewis, J.C., Lin, H.Y. and Yoon, W.K. (1985)
Dynamic matrix control of a batch distillation column: American Control Conference, Boston
- [130] Luenberger, D.G. (1973)
Introduction to linear and non-linear programming: Addison-Wesley, Reading, Mass.
- [131] M'Saad, M., Duque, M. and Landau, I.D. (1986)
On the applicability of adaptive control: 2nd IFAC workshop on Adaptive Control and Signal Processing, Lund

- [132] M'Saad.M., Duque.M, and Landau.I.D. (1987)
Thermal process robust adaptive control: an experimental evaluation: 10th IFAC World Congress on automatic control
- [133] MacFarlane,A.G.J. (1972)
A survey of some recent results in linear multivariable feedback theory.: Automatica, Vol.8, pp.455-492
- [134] Martin,G.D. and Van Horn,L.D. (1982)
IDCOM hierarchical multivariable control application: Conference on Industrial Computer Systems, Hamilton, Ontario
- [135] Martin-Sanchez,J.M. (1984)
A globally stable APCS in the presence of bounded noises and disturbances: IEEE Trans. Autom. Control., AC-29, pp.461-464
- [136] Martin-Sanchez,J.M. and Shah,S.L. (1984)
Multivariable adaptive predictive control of a binary distillation column.: Automatica, Vol.20, No.5, pp.607-620
- [137] Maurath,P.R., Seborg,D.E. and Mellichamp,D.A. (1986)
Achieving decoupling with predictive controllers: ACC
- [138] McDermott,P.E. (1984)
Self-tuning control of a fixed bed autothermal reactor: Ph.D Thesis, University of California.
- [139] McDermott,P.E. and Mellichamp,D.A. (1984)
A decoupling pole placement controller for a class of multivariable processes: IFAC 9th World Congress, Budapest, Hungary
- [140] Mohtadi,C. (1986)
Studies in advanced self-tuning algorithms: D.Phil Thesis, Oxford University and OUEL Report No.1689/87
- [141] Mohtadi,C. and Clarke,D.W. (1986)
Generalised predictive control, LQ, or pole-placement: a unified approach: CDC conference, Athens
- [142] Mohtadi,C., Shah,S.L. and Clarke,D.W. (1986)
Generalized predictive control of multivariable systems.: OUEL report 1640/86
- [143] Montague,G.A. and Tham,M.T. (1986)
A comparison of multivariable long range predictive control with GMV control in a highly nonlinear environment.: Dept. Chem. Eng., University of Newcastle Upon Tyne
- [144] Montague,G.A., Tham,M.T. and Morris,A.J. (1986)
A comparison of multivariable long range predictive control with GMV control in a highly nonlinear enviroment.: ACC
- [145] Morris,A.J., Fenton,T.P. and Nazer,Y. (1977)
Application of self-tuning regulators to the control of chemical processes: IFAC/IFIP Conference on Digital Computer Applications to Process Control, The Hague

- [146] Morris,A.J., Nazer,Y. and Wood,R.K. (1981)
Single and multivariable applications of self-tuning controllers.: in 'Self-tuning and adaptive control' (ed. Harris and Billings) Peter Perigrinus
- [147] Morris,A.J., Nazer,Y. and Wood,R.K. (1982)
Multivariate self-tuning process control: Optimal Control App. and Methods, Vol.3, No.4, pp.363-387
- [148] Morris,A.J., Nazer,Y., Wood,R.K. and Lieuson,H. (1980)
Evaluation of self-tuning controllers for distillation column control: IFAC Symposium on Digital Computer Applications to Process Control, Dusseldorf
- [149] Mosca,E., Zappa,G. and Manfredi,C. (1984)
Multistep horizon self-tuning controllers: the MUSMAR approach: IFAC 9th World Congress, Budapest, Hungary
- [150] Nazer,Y. (1981)
Single and multivariable self-tuning controllers.: Thesis, Dept. Chem. Eng., University of Newcastle
- [151] Ortega,R., Praly,L. and Landau,I.D. (1985)
Robustness of discrete-time direct adaptive controllers.: IEEE Trans. Autom. Control., AC-30, pp.1179-1187
- [152] Parks,P.C., Schaufelberger,W., Schmid,Chr. and Umbehauen,H. (1980)
Applications of adaptive control systems: Symposium on Adaptive Systems, Bochum
- [153] Patel,R.V. and Munro,N. (1982)
Multivariable system theory and design: Pergammon Press, Oxford
- [154] Peel,D., Morris,A.J. and Tham,M.T. (1985)
Univariate self-tuning control in a distributed control environment: IEE Conference Control '85, Cambridge
- [155] Peterka,V. (1970)
Adaptive digital regulation of noisy systems: IFAC Symposium on Identification and Process Parameter Estimation, Prague
- [156] Peterka,V. (1984)
Predictor-based self-tuning control: Automatica, Vol.20, No.1, pp.39-50
- [157] Peterka,V. and Åström,K.J. (1973)
Control of multivariable systems with unknown but constant parameters.: IFAC Symposium on Identification and System Parameter Estimation, The Hague, Netherlands, pp.535-544
- [158] Prager,D.L. and Wellstead,P.E. (1980)
Multivariable pole assignment self tuning regulators.: Proc.IEE, Vol.128, Pt.D, No.1, pp.9-18
- [159] Praly,L. (1984)
Robust model reference adaptive controllers, part 1: stability analysis: Proc. 23rd IEEE CDC, pp.1009-1014

- [160] Ragazzini, J.R. and Franklin, G.F. (1958)
Sampled-data control systems: McGraw-Hill
- [161] Richalet, J., Rault, A., Testud, J.L. and Papon, J. (1978)
Model predictive heuristic control: applications to industrial processes: Automatica, Vol.14, No.5, pp.413-428
- [162] Rohrs, C.E. (1982)
Adaptive control in the presence of unmodeled dynamics: Dept. Elec. Eng. Comp. Sci., MIT, LIDS Report No. TH-1254
- [163] Rohrs, C.E., Athans, M., Valavani, L. and Stein, G. (1984)
Some design guidelines for discrete-time adaptive controllers: Automatica, Vol.20, No.5, pp.653-660
- [164] Rosenbrock, H.H. (1974)
Computer-aided control system design: Academic Press Inc., New York.
- [165] Rouhani, R. and Mehra, R.K. (1982)
Model algorithmic control (MAC) ; basic theoretical properties: Automatica, Vol.18, No.4, pp.401-414
- [166] Samson, C. (1983)
Stability analysis of adaptively controlled systems subject to bounded disturbances: Automatica, Vol.19, No.1, pp.81-86
- [167] Sanoff, S.P. and Wellstead, P.E. (1984)
Expert identification and control: Control Systems Centre, UMIST, Report 618
- [168] Scattolini, R. and Clarke, D.W. (1985)
Multivariable model-following self-tuning control with offset rejection.: OUEL report 1572/85
- [169] Schumann, R., Lachmann, K.H. and Isermann, R. (1981)
Towards applicability of parameter adaptive control algorithms: IFAC 8th World Congress, Kyoto, Japan
- [170] Seborg, D.E., Edgar, T.F. and Shah, S.L. (1986)
Adaptive control strategies for process control: a survey: AIChE Journal, Vol.32, No.6, pp.881-913
- [171] Shinsky, F.G. (1979)
Process control systems: McGraw-Hill, New York.
- [172] Singh, R.P. and Narendra, K.S. (1984)
Prior information in the design of multivariable adaptive controllers.: IEEE Trans. Autom. Control, Vol.AC-29, No.12, pp.1108-1111
- [173] Smith, O.J.M. (1959)
A controller to overcome dead-time: Instrument Society of America Journal, Vol.6, No.2, pp.28-33
- [174] Sripada, N.R. and Fisher, D.G. (1987)
Improved least squares identification for adaptive controllers: ACC, Minneapolis.

- [175] Tacey, E. (1987)
The thought processors: The Engineer, May
- [176] Tuffs, P.S. (1984)
Self-tuning control: algorithms and applications: D.Phil Thesis, Oxford University and OUEL report 1567/85
- [177] Tuffs, P.S. (1985)
FAUST: introduction and overview: OUEL report 1568/85
- [178] Tuffs, P.S. and Clarke, D.W. (1985)
FAUST: a software package for self-tuning control: IEE Conference Control '85, Cambridge
- [179] Tuffs, P.S. and Clarke, D.W. (1985)
Self-tuning control of offset: a unified approach: Proc.IEE, Vol.132, Pt.D, No.3, pp.100-110
- [180] Van Cauwenberghe, A.R. and De Keyser, R.M.C. (1985)
Self-adaptive long-range predictive control: ACC, Boston
- [181] Vogel, E.F. (1982)
Adaptive control of chemical processes with variable dead-time: Ph.D. Thesis, University of Texas
- [182] Vogel, E.F. and Edgar, T.F. (1982)
An adaptive multivariable pole-zero placement controller for processes with variable dead-time: AIChE, Los Angeles
- [183] Warwick, K. (Ed) (1987)
Implementation of self-tuning controllers: Peter Peregrinus Ltd.
- [184] Warwick, K. and Rees, D. (Ed) (1986)
Industrial digital control systems: Peter Peregrinus Ltd.
- [185] Waterman, D. (1986)
A guide to expert systems: Addison-Wesley, Reading, Mass.
- [186] Wellstead, P.E. and Prager, D.L. (1981)
Self-tuning multivariable regulators: in 'Self-tuning and adaptive control' (ed. Harris and Billings) Peter Perigrinus
- [187] Wellstead, P.E. and Sanoff, S.P. (1981)
Extended self-tuning algorithm: Int.J.Control, Vol.34, No.3, pp.433-455
- [188] Wellstead, P.E., Edmunds, J.M., Prager, D. and Zanker, P. (1979)
Self-tuning pole/zero assignment regulators: Int.J.Control, Vol.30, No.1, pp.1-26
- [189] Williams-Gardner, A. (1971)
Industrial drying: Leonard Hill, London
- [190] Willsky, A.S. (1976)
A survey of design methods for failure detection in deterministic: Automatica, Vol.12, pp.601-611

- [191] Wittenmark, B. and Åström, K.J. (1984)
Practical issues in the implementation of self-tuning control: Automatica, Vol.20, No. 5, pp.595-605
- [192] Wolovich, W.A. and Elliott, H. (1983)
Discrete models for linear multivariable systems: Int. J. Control, Vol.38, No.2, pp.337-357
- [193] Wolovich, W.A. and Falb, P.L. (1976)
Invariants and canonical forms under dynamic compensation: SIAM J. Control Optim., Vol.14, No.6, pp.996-1008
- [194] Woollatt, E. (1985)
The manufacture of soaps, other detergents and glycerine: Ellis Horwood Series, Industrial Technology, John Wiley & Sons
- [195] Wong, K.Y. and Bayoumi, M.M. (1981)
Multivariable Self-Tuning Regulators: Proc. IEEE Cont. Decision Conf., San Diego, pp.978
- [196] Ydstie, B.E. (1982)
Robust adaptive control of chemical processes: Thesis, Department of Chemical Engineering and Chemical Technology, Imperial College
- [197] Ydstie, B.E. (1984)
Extended horizon adaptive control: IFAC 9th World Congress, Budapest, Hungary
- [198] Ydstie, B.E., Kershenbaum, L.S. and Sargent, R.W.H. (1985)
Theory and application of an extended horizon self-tuning controller: AIChE Journal, Vol.31, pp.1771
- [199] Young, P.C. (1968)
The use of linear regression and related procedures for the identification of dynamic processes: Proc. 7th IEEE, Symposium on Adaptive Processes, UCLA