

https://dspace.org/

A self-adaptive multivariable computer control system

Ferreira, Adriano Manuel Povoa

2015-03-11
http://hdl.handle.net/10539/17253

Downloaded from a DSpace repository. https://dspace.org/

A SELF-ADAPTIVE MULTIVARIABLE COMPUTER CONTROL SYSTEM

Adriano Manuel Póvoa Ferreira

A Dissertation submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg in part fulfilment of the requirements for the Degree of Master of Science in Engineering.

Johannesburg 1984.

ABSTRACT

This project report summarizes and expands previous work on the design of self-tuning regulators. The objective of the theory developed is to implement a multivariable self-adaptive error-actuated controller on a minicomputer. The various facets of the new algorithm, ie. convergence and stability are then tested experimentally on a non-linear physical system with time-varying dynamics.

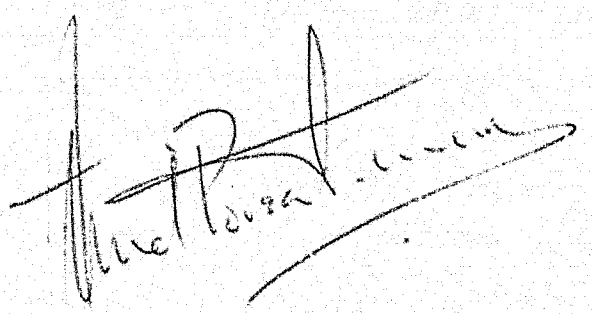
A discrete-time model representing a local linearisation of the non-linear process is developed and identified via a recursive least-squares estimator. This estimator is characterised by a variable weighting of past data. This novel technique ensures that, at each sample, there exists a balance in the information content of the estimator.

The 'reference-tracking' controller is based on the classical approach of pole-assignment. This approach, although non-optimal, is fundamentally more robust and consequently can be more generally applied.

Keywords : Multivariable digital control; parameter adaptability; non-linear, time-varying system; error-actuated controller.

DECLARATION

I hereby declare that this dissertation is my own work and that no part of it has previously been submitted for any degree or examination at any other University.



A M P FERREIRA

January 1984.

DEDICATION

"Em cada pégada de amor
nascerá uma flor de gratidão."

To my mother
Maria da Luz

ACKNOWLEDGEMENTS

I would like to thank the Council for Mineral Technology (MINTEK) for providing the research studentship during which this work was performed.

I wish to express my sincere gratitude to Dr I J Barker for his valuable support.

It has been my good fortune to have worked with Dr I M MacLeod and Mr S.G. McCandlish, the project supervisors. Their interest and guidance is greatly appreciated.

I am also indebted to Mr E Jackwitch for his efficient installation of all necessary modifications to the plant.

My thanks to Mrs G E MacLachlan for deciphering the original and typing this manuscript.

TABLE OF CONTENTS

	Page
ABSTRACT	i
DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF SYMBOLS	vi
LIST OF ILLUSTRATIONS	ix
CHAPTER 1 : INTRODUCTION	1
1.1 : Motivation and Background	2
1.2 : Approaches to Adaptive Control - A Summary	5
1.3 : Problem formulation and Summary of Thesis	9
CHAPTER 2 : SYSTEM MODEL	12
2.1 : Introduction	13
2.2 : Mathematical Model	14
2.3 : State-Space Model	17
2.4 : Survey of Multivariable Design Methods	20
2.5 : Time-domain Computer Simulation	26
2.6 : Discrete-time Model	28
2.7 : Discussion	33
CHAPTER 3 : SYSTEM IDENTIFICATION	34
3.1 : Introduction	35
3.2 : Recursive Least-squares algorithm - Analysis	37
3.3 : Parameter - Adaptability Characteristics	46
3.4 : Simulation Results	55
3.5 : Discussion	64

CHAPTER 4 : SYSTEM CONTROL 68
4.1 : Introduction 69
4.2 : Digital Control Laws - A Survey 71
4.3 : Multivariable Pole-Assignment Controller 81
4.4 : Simulation Results 89
4.5 : Discussion 99

CHAPTER 5 : SOFTWARE ENGINEERING OF SYSTEM 102
5.1 : Introduction 103
5.2 : Structure of the Suite 105
5.3 : Discussion 110

CHAPTER 6 : SELF-TUNING AS COMBINED IDENTIFICATION AND CONTROL 111
6.1 : Introduction 112
6.2 : Self-Tuner Characteristics - Analysis 114
6.3 : Experimental Results 119
6.4 : Discussion 126

CHAPTER 7 : CONCLUSION 129

REFERENCES 134
APPENDICES : 1 - Discrete-time models 139
2 - Software Explanatory listing 141
3 - User Initial Dialogue 157
4 - Format of Data Printouts 159
5 - On-Line User Interactive Dialogue 162

LIST OF SYMBOLS

Q_{ij}	Flow rate from tank 'i' to tank 'j'
Q_i	Flow rate out of tank 'i'
H_i	Height of fluid in tank 'i'
A_i	Cross-sectional area of tank 'i'
K_{ij}	Tank 'i'-to-tank 'j' discharge co-efficient
H_{io}	Level operating point of tank 'i'
z^{-1}	Backward shift operator
U_k	Sequence of input variables to the plant
Y_k	Sequence of output variables from the plant
e_k	Zero-mean white noise sequence
$A(z^{-1})$	Polynomial matrix corresponding to system output
$B(z^{-1})$	Polynomial matric corresponding to system input
$C(z^{-1})$	Polynomial matrix corresponding to system noise
ρ_c	Closed-loop characteristic polynomial
ρ_o	Open-loop characteristic polynomial
T_s	Sampling interval
$\alpha(z^{-1})$	On-line estimated system model parameters
$\beta(z^{-1})$	
$F(z^{-1})$	On-line estimated syste controller parameters (x = 0,1)
$G_x(z^{-1})$	
R	Uncertainty matrix of measurement variables
ϕ	Least-squares estimator gain matrix
P	Least-squares estimator confidence matrix
ψ	Least-squares estimator measurement vector
$\hat{\theta}$	Least-squares estimator parameter vector
λ	Least-squares estimator forgetting factor

LIST OF SYMBOLS (Cont)

ζ Sequence of fitting errors

σ_x Standard deviation of variable x

$E\{\cdot\}$ Expectation operator

k_d System time delay

$G(s)$ General transfer function in Laplace domain

The polynomial matrices in the z^{-1} operator are of order n and of the general form,

$$X(z^{-1}) = X_1 z^{-1} + \ldots + X_n z^{-n}$$

where

X_i, $i = 1 \ldots n$

are pxp matrices and p is the number of system inputs and outputs

LIST OF ILLUSTRATIONS

(i) FIGURES

		Page
1.1	Block diagram of gain-scheduling adaptive scheme	6
1.2	Block diagram of model-reference scheme	7
1.3	Structure of a self-tuner	8
2.1	Overall plant configuration	16
2.2	State-space formulation	19
2.3	Plant partitioning into three transfer functions	21
2.4	Block diagram of system model	30
3.1	Schematic of least-squares algorithm	38
4.1	Regulator Scheme	72
4.2	Schematic of self-tuning regulator	79
4.3	Multivariable self-tuning Controller scheme	84
5.1	Block diagram of system configuration	103
5.2	Block diagram of software structure	105
5.3	Block diagram structure of 'flow rig' control algorithm	107

LIST OF ILLUSTRATIONS (Cont.)

Page

(ii) GRAPHS

3 : Identification Results 56

(Time evolution of system model parameters)

4 : Simulation/Controller Results 90

(Time evolution of system responses and controller parameters)

6 : Experimental Results 120
(Time evolution of system responses)

CHAPTER 1

INTRODUCTION

1.1. MOTIVATION AND BACKGROUND

The simple two or three term controller (PI or PID) unquestionably remains the most common in practice due to its inherent robustness, ease of tuning, tolerance of process changes, and the lack of a detailed process knowledge required during initial design. In spite of this, due to the increasing demands for tighter quality control and for energy conservation it may well be that some critical loops will require controllers with many parameters whose values have to be chosen in a more systematic way. Further, due to changes in production and wear of the equipment the process dynamics may alter considerably making it necessary to re-tune the controllers. This is a time-consuming process, even if the conventional two or three term laws are used, particularly if the process dynamics are slow and there are many interactive loops.

For these reasons control engineers and theorists have searched, for many decades, for a controller that does not need to be tuned, (i.e. automatic tuning). This type of controller has been given different names, for instance adaptive, self-organizing, self-optimizing and learning controller. An adaptive controller can be viewed as a tuning aid for complex control laws or as a means of efficiently controlling time-varying processes.

Research on adaptive control was very active in the early fifties motivated by the design of autopilots for high performance aeroplanes. These earlier adaptive controllers were implemented using analogue techniques, which naturally limited the possibilities. Further the theory was not yet developed and hence the work was characterized by

a lack of fundamental insight and understanding.

In the sixties there were many contributions to control theory, which were fundamental to the development of adaptive control. State space and stability theory were developed. Dual control theory introduced by Feldbaum in 1965 |24| increased understanding of adaptive control. There were also major developments in system identification and parameter estimation.

By the seventies the theory and minicomputer technology were sufficiently developed to lead to a renewed interest in self-tuning techniques. The seminal work of Astrom |3| has given rise to considerable development in the field. This paper led directly to a series of practical implementations, which themselves triggered new theoretical developments as the various problems involved in the practical application of adaptive control became clear.

The field has since exploded with new algorithms and the papers by Clarke |9| and Isermann |12| present detailed surveys of the better known and used techniques.

The current effort on digital self-tuning/adaptive control coincides with the widespread use of microcomputers in industrial control. A self-tuner can now be built, in line with recent trends in automation, as an out-station in a microprocessor-based distributed control system. In addition the multivariable aspects of the algorithms are also starting to be theoretically developed, and this dissertation deals with such a development.

Earlier projects in the U.K. and Sweden (Wellstead |20| and Borrison |6|) have succeeded in implementing multivariable self-adaptive regulators. The work reported here is an extension of the former project. The areas of research were the theoretical development of a multivariable recursive least-squares estimator with variable weighting of past data, the development of an error-actuated multivariable pole-placement controller and the implementation and testing of these in an industrial context (Wellstead tested his algorithm on a scale model).

The field of adaptive control is immense, it is therefore impossible in a dissertation such as this to do justice to all aspects involved with regards the theory and properties of the algorithms. Attention is therefore restricted to the new theoretical developments and to issues relating to the performance of the algorithm. This dissertation traces the evolution, in a systematic manner, of the various algorithms, clarifying the particular difficulties and possibilities that arise.

1.2. APPROACHES TO ADAPTIVE CONTROL - SUMMARY

There are basically three schemes for the design of an adaptive control system, namely, gain-scheduling, model reference and self-tuning. These are considered in a common framework by Astrom |2|, while Wittenmark |24|, presents a comprehensive survey of adaptive control methods. Subsequently a brief description of the three approaches to adaptive control is included with particular emphasis on the method used in this project namely self-tuning.

An adaptive controller, as the name implies, has the ability to modify its behaviour depending on the performance of the closed-loop system. To this end it performs three basic functions : identification of unknown parameters or measurement of a performance index, decision on the control strategy, and on-line modification of the controller. The key to any adaptive system is therefore to find a way of changing the controller parameters in response to changes in the process and noise dynamics. The various methods discussed subsequently differ only in the way the controller parameters are adjusted.

(i) Gain Scheduling

In this method (see Fig. 1.1.) auxiliary process variables, which correlate well with changes in the process dynamics, are used to influence the controller parameters.

The main advantage is that adjustments to nulify process changes can be made very quickly. This is however an open-loop technique similar to feedforward compensation.

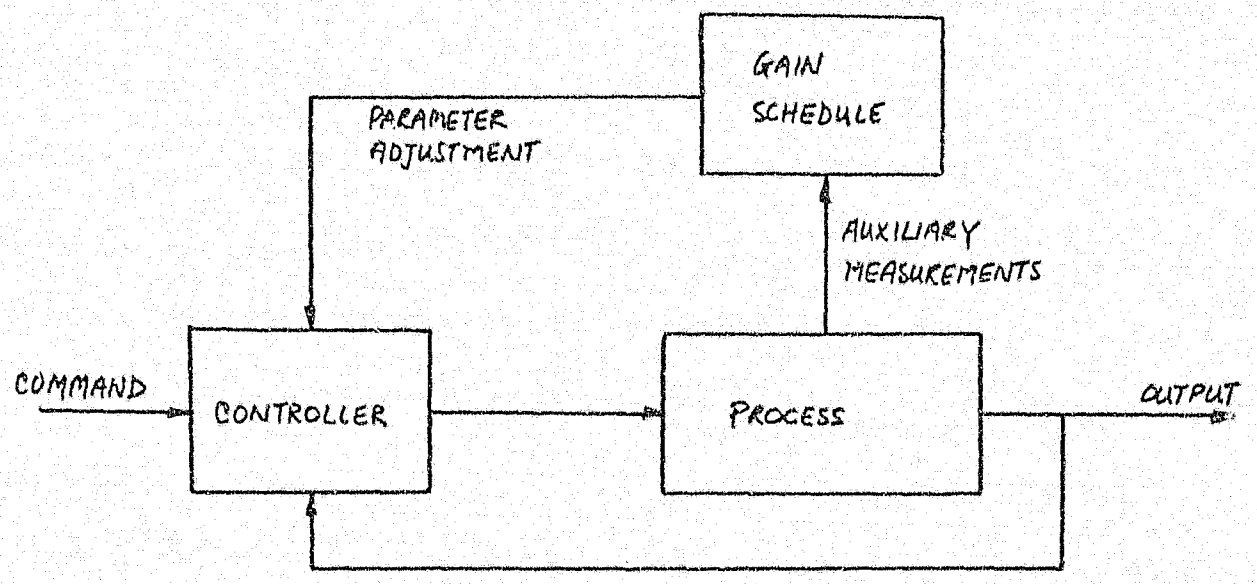


Fig. 1.1 : Block Diagram of Gain Scheduling Adaptive Scheme

(ii) Model reference adaptive systems

This technique is depicted in Fig. 1.2. The reference model yields the ideal behaviour of the process. The key problem is the determination of the adjustment mechanism such that stability and minimal error is obtained. Astrom |2| discusses this method in more detail.

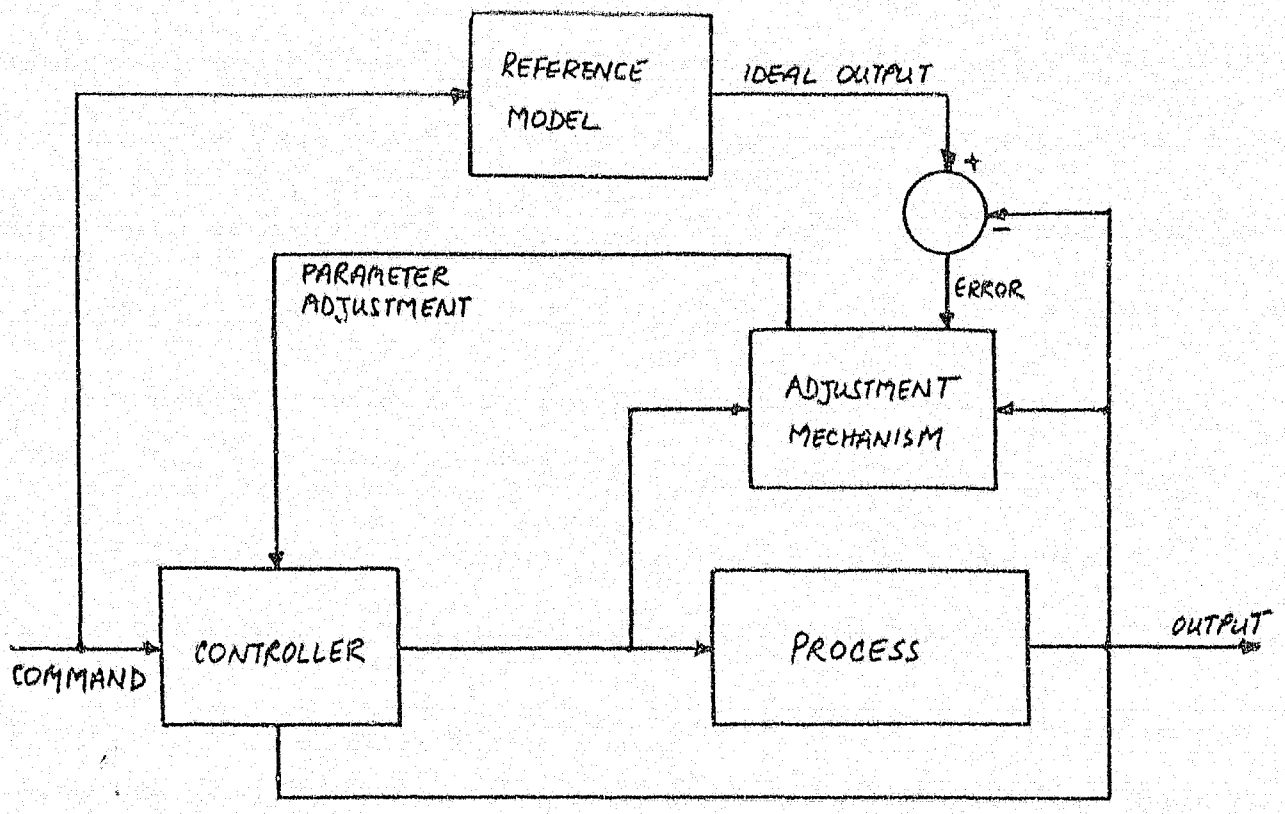


Fig. 1.2 : Block Diagram of Model Reference Scheme

(iii) Self-Tuning systems:

A self-tuning controller has three main elements. There is a standard control law in the form of a difference equation which acts upon a set of values and produces the new control action. A recursive parameter estimator monitors the plants input and output and computes an estimate of the plant dynamics in terms of a set of parameters in a prescribed structural model. These parameter estimates are passed to a control-design algorithm that provides a new set of coefficients for the control law. This mechanism is illustrated in Fig.1.3.

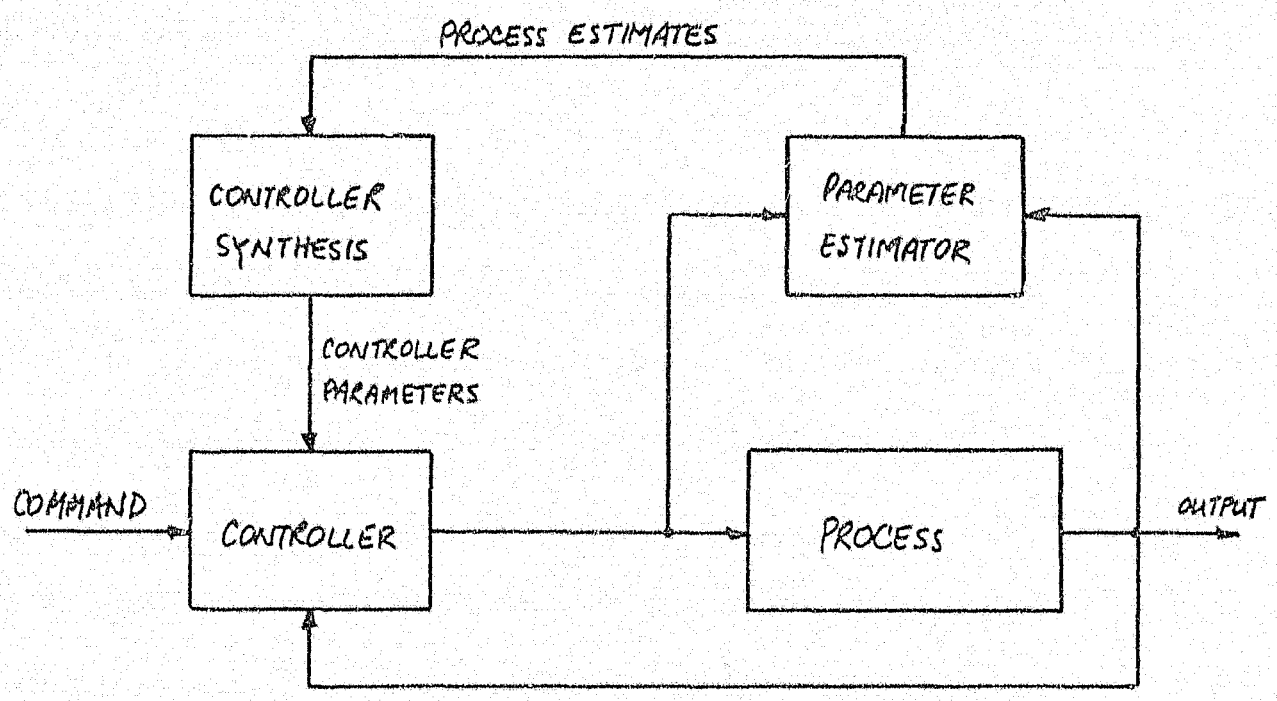


Fig. 1.3 Structure of a self-tuner

The control-design algorithm simply accepts the current estimates and ignores their uncertainties (unlike dual control |24|.) Such a procedure is termed certainty - equivalent control. Self-tuners are very flexible with regards to the design method, and vary according to their performance - objectives. The performance objective of the original algorithm of Astrom |3| is the minimization of the variance of the measured process output. In this project the objective comprises a trade off between control effort and output variances as will be explained in Chapter 4.

1.3. PROBLEM FORMULATION AND SUMMARY OF THE THESIS

Since the development of the basic theory by Astrom |3| there have been many reported applications of self-tuning. These cover a wide variety of industries : ore-crushing |5|, paper-making |7|, titanium-dioxide kilns |10|, distillation columns |19| and cement-blending |14|. In such processes the main aim is plant output regulation, although a fast stable response to setpoint demand changes and sudden load disturbances are also necessary. MINTEK (Council for Mineral Technology) being active in the design of mineral processing plants is naturally interested in researching the application of adaptive controllers in this area.

The present project investigates self-tuning pole-assignment error-actuated controllers for systems with more than one input and output. Previous work on multivariable self-tuners has focused on minimum-variance |6| or on pole-assignment |20| regulator design. The work described here extends the latter law to accomodate set-point following capabilities. All relevant theoretical developments are discussed in the corresponding sections.

An experiment is designed (Chapters 2 and 5) to test the various theoretical concepts in the context of a real industrial-like environment. The coupled tanks apparatus (Fig. 2.1) is the physical system used to investigate the various developed theoretical concepts in the context of fluid-level control. It exhibits properties typical of this kind of system - non-linear, time-varying parameters coupled with sensor noise, and non-minimum phase behaviour in the discrete-time domain.

Another practical problem addressed in this dissertation (Chapter 5) is related to the digital implementation of the self-tuner on a minicomputer. All this work is, therefore, in line with most recent developments in the process control industry.

Self-tuning is a direct digital control technique which is applied to a continuous-time system. Hence as a prelude to the main body of self-tuning theory, it is necessary to consider the representation of the system by a digital (sampled data) model. This procedure is considered in Chapter 2.

The identification of digital models is covered in Chapter 3, together with details of the recursive implementation and parameter-tracking capabilities. The report then proceeds to discuss (Chapter 4) digital control laws as they occur in self-tuning. The choice of the pole-assignment law is motivated and the novel enhancements are discussed. These two algorithms developed are tested in a plant simulation and the results and analysis are attached in the respective sections.

The discussion then focuses on the application of the self-tuner to the actual plant (Chapters 5 and 6).

The report closes with a section containing guidance upon important factors which the user needs to be aware of during self-tuning. This is an extension of the last sections of Chapters 3 and 4, in particular the significance and limitations of this particular application are considered.

The way in which certain self-tuner parameters are initially set up by the user can have a critical influence on the success or failure of the self-tuning experiment. In fact, as in any controller design method, the user plays a vital role. This statement may seem controversial, however the only step that is automated in self-tuning is the synthesis of the controller parameters from the identification results. The control engineer is then free to attend to more general considerations such as the nature of the control law, the system order, etc. In this vein the discussion sections at the end of each Chapter and Chapter 7, provide additional insight and document the experience acquired in the present system.

The literature, especially in the discussion of the three main streams of self-tuning theory (minimum-variance, weighted minimum-variance and pole-shifting) is vast, especially for the single-input/single-output case and will, therefore, be dealt with in the various Chapters. With regards to the multivariable set-point following error-actuated self-tuners no literature was found, thus motivating the developments presented in Chapters 3 and 4. This perhaps warrants that further theoretical work should be done, especially concerning the following areas :

(i) Convergence properties of the algorithm

(ii) The influence of more general non-linearities upon the algorithm.

CHAPTER 2

SYSTEM MODEL

2.1. INTRODUCTION

In this chapter a detailed analysis of the plant model both in the continuous and discrete time domains is included. A brief description of a simulation package is also appended.

A deeper analysis of the plant configuration in the context of using existing multivariable theory to analyze it is also included in section 2.4. It is seen that in the mulitvariable case the design and implementation of an error-actuated controller is a compromise involving not only stability and accuracy (as in the single variable case) but also interaction, integrity and controller complexity. MacFarlane |15| and Rosenbrock |18| satisfy these design specifications in a manner which permits the extension of the classical scalar feedback theories of Nyquist & Bode to the multivariable vector case.

Another important area of research related to multivariable theory which is beyond the scope of this project is discussed by Owens |29| and Porter |30|. They consider examples of constructing simple, non-adaptive, robust controllers for unknown multivariable systems (unknown in the sense that its model is unknown or of a too complex order to make normal calculations feasable) using only elementary computations based on graphical system open loop step response data.

2.2. MATHEMATICAL MODEL

Any modelling process involves a trade-off between complexity and accuracy. The following plant model was developed in three stages :

(i) system decomposition into subsystems,

(ii) development of functional relationships for each sub-system, and,

(iii) model synthesis.

The system can be decomposed by function into the following subsystems :

(i) Tank

(ii) Outflow pipe and valve

(iii) Motor pump actuators

(iv) Level transducers

The ultimate purpose of the model is the study of system dynamics. Assuming the above system decomposition, it is noticed that the system dynamic properties can be easily confused with the problems of actuation and measurement. To avoid this problem the model developed here incorporates the first subsystem only.

The advantages of this philosophy are twofold. Firstly, a much simpler plant representation is obtained and secondly, the self-tuner can be tested in terms of its ability to cope with operating-point non-linearities (i.e. on-line with model simulation -see ch. 4) and with actuator non-linearities and sensor noise (i.e. on line with plant - ee ch. 6)

The overall system configuration is depicted in Fig. 2.1 and can be seen as a flow integrator to yield level. Analysis can be done using network methods or, as follows, via material balance techniques.

Applying the law of mass conservation to tanks (1) and (2) assuming constant fluid density yields,

$$A_i \frac{dH_i}{dt} = Q_{i(in)} - (Q_{i3} + Q_{i4}), \quad i = 1,2 \qquad - (2.1)$$

Similarly the same principle applied to tanks (3) and (4) yields,

$$A_j \frac{dH_j}{dt} = (Q_{1j} + Q_{2j}) - Q_{j(out)}, \quad j = 3,4 \qquad - (2.2)$$

By applying Bernoulli's equation to a gravity-driven main, it can easily be shown that,

$$Q_{ij} = K_{ij} (H_i)^{\frac{1}{2}} \qquad - (2.3)$$

Substituting into (2.1) & (2.2) and rearranging,

$$\dot{H}_i = \frac{1}{A_i} (Q_{i(in)} - (K_{i3} + K_{i4})(H_i)^{\frac{1}{2}}), \quad i = 1,2$$

$$\dot{H}_j = \frac{1}{A_j} (K_{1j}(H_1)^{\frac{1}{2}} + K_{2j}(H_2)^{\frac{1}{2}} - K_j(H_j)^{\frac{1}{2}}), \quad j = 3,4 \qquad - (2.4)$$

These equations are implemented to obtain a time-domain plant dynamic simulation, as discussed in section 2.5.

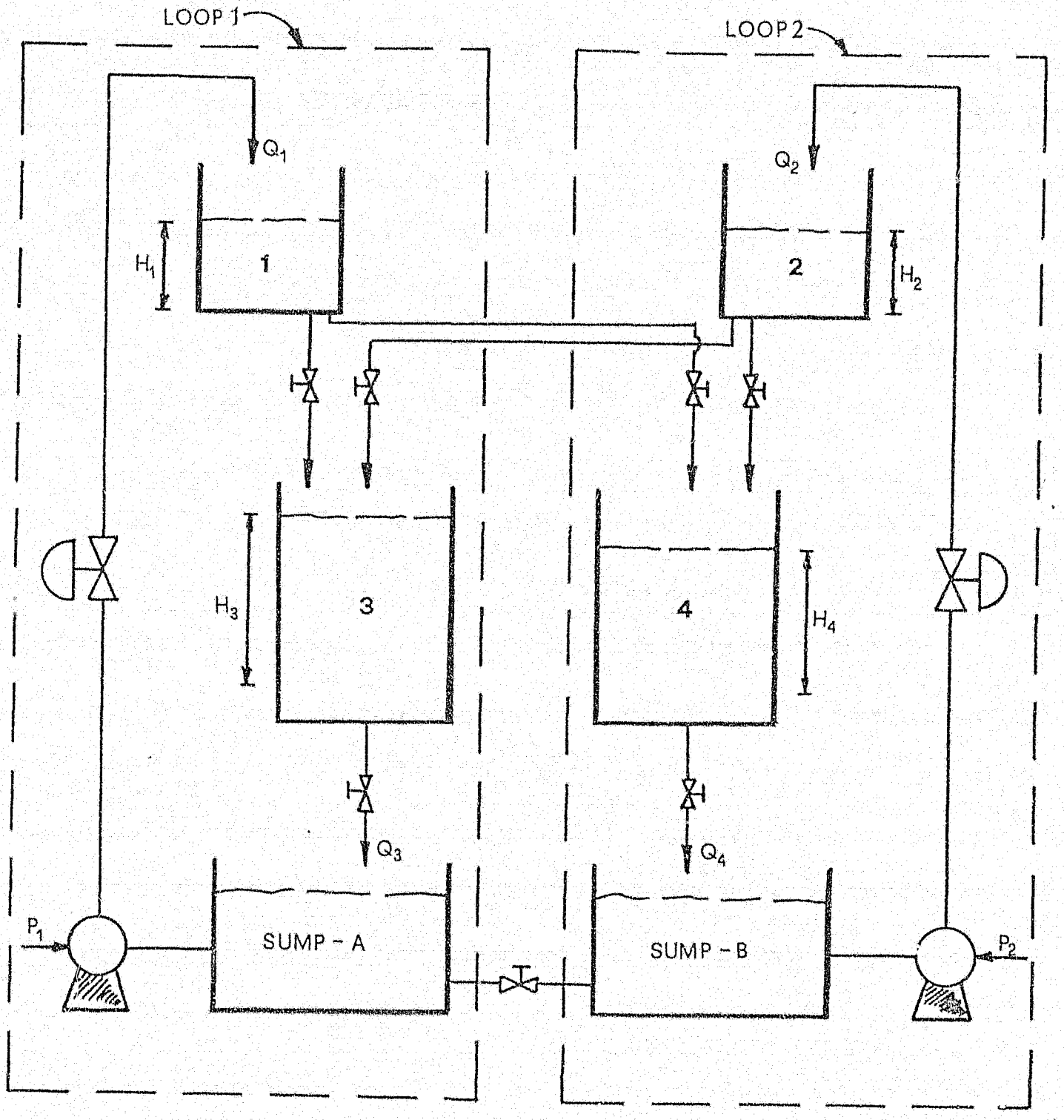


Fig. 2.1 : Overall Plant Configuration

2.3 STATE-SPACE MODEL

In this section a state-space representation of the system is obtained by linearizing equations (2.4). This particular representation is then used to reconfigure the system into its elementary sections. A survey of several multivariable techniques and how each could be used in off-line controller designs depending on the particular system configuration, is then included in section 2.4.

Assuming small signal conditions about an operating point and using a truncated Taylor series, equations (2.4) are linearized,

$$\dot{H}_i = \frac{1}{A_i}\left(Q_{j_{(in)}} - \frac{1}{2}(K_{i3} + K_{i4})(H_{i0})^{-\frac{1}{2}}H_i\right) \quad , i = 1,2 \qquad -(2.5)$$

$$\dot{H}_j = \frac{1}{2A_j}\left(K_{1j}(H_{10})^{-\frac{1}{2}}H_1 + K_{2j}(H_{20})^{-\frac{1}{2}}H_2 - K_j(H_{j0})^{-\frac{1}{2}}H_j\right) \quad ,j = 3,4$$

Note that the system transfer function changes with operating conditions. Equations (2.5) can now be written in their state space format;

$$\dot{h} = Ah + Bq$$
$$\ell = Ch \qquad -(2.6).$$

where,

$$A = \begin{bmatrix} -(B_{13} + B_{14}) & \emptyset & \emptyset & \emptyset \\ \emptyset & -(B_{23} + B_{24}) & \emptyset & \emptyset \\ D_{13} & D_{23} & -C_3 & \emptyset \\ D_{14} & D_{24} & \emptyset & -C_4 \end{bmatrix} \qquad -(2.6.1)$$

$$B = \begin{bmatrix} \frac{1}{A_1} & \emptyset & \emptyset & \emptyset \\ \emptyset & \frac{1}{A_2} & \emptyset & \emptyset \end{bmatrix}^T \quad ; \text{ and} \qquad -(2.6.2)$$

$$C = \begin{bmatrix} \emptyset & \emptyset & 1 & \emptyset \\ \emptyset & \emptyset & \emptyset & 1 \end{bmatrix} \qquad -(2.6.3)$$

Also

$$B_{ij} = \frac{1}{2} \frac{K_{ij} (H_{jo})^{-\frac{1}{2}}}{A_i} \quad ; \; i = 1,2, \; j = 3,4$$

$$C_j = \frac{1}{2} \frac{K_j (H_{jo})^{-\frac{1}{2}}}{A_j} \quad ; \; j = 3,4 \qquad -(2.6.4)$$

$$D_{ij} = \frac{1}{2} \frac{K_{ij} (H_{jo})^{-\frac{1}{2}}}{A_j} \quad ; \; i = 1,2 \; , \; j = 3,4$$

Further

$$\begin{aligned} h &= \text{vector of states} = [H_1 \; H_2 \; H_3 \; H_4]^T \\ q &= \text{vector of inputs} = [Q_1(in) \; Q_2(in)]^T \\ \ell &= \text{vector of outputs} = [H_3 \; H_4]^T \end{aligned} \qquad -(2.6.5)$$

A better understanding of the above formulation can be achieved by analysing it in terms of the model presented in Fig. 2.2. This also defines a clear boundary between instrumentation and control problems, as briefly mentioned at the beginning of section 2.2.

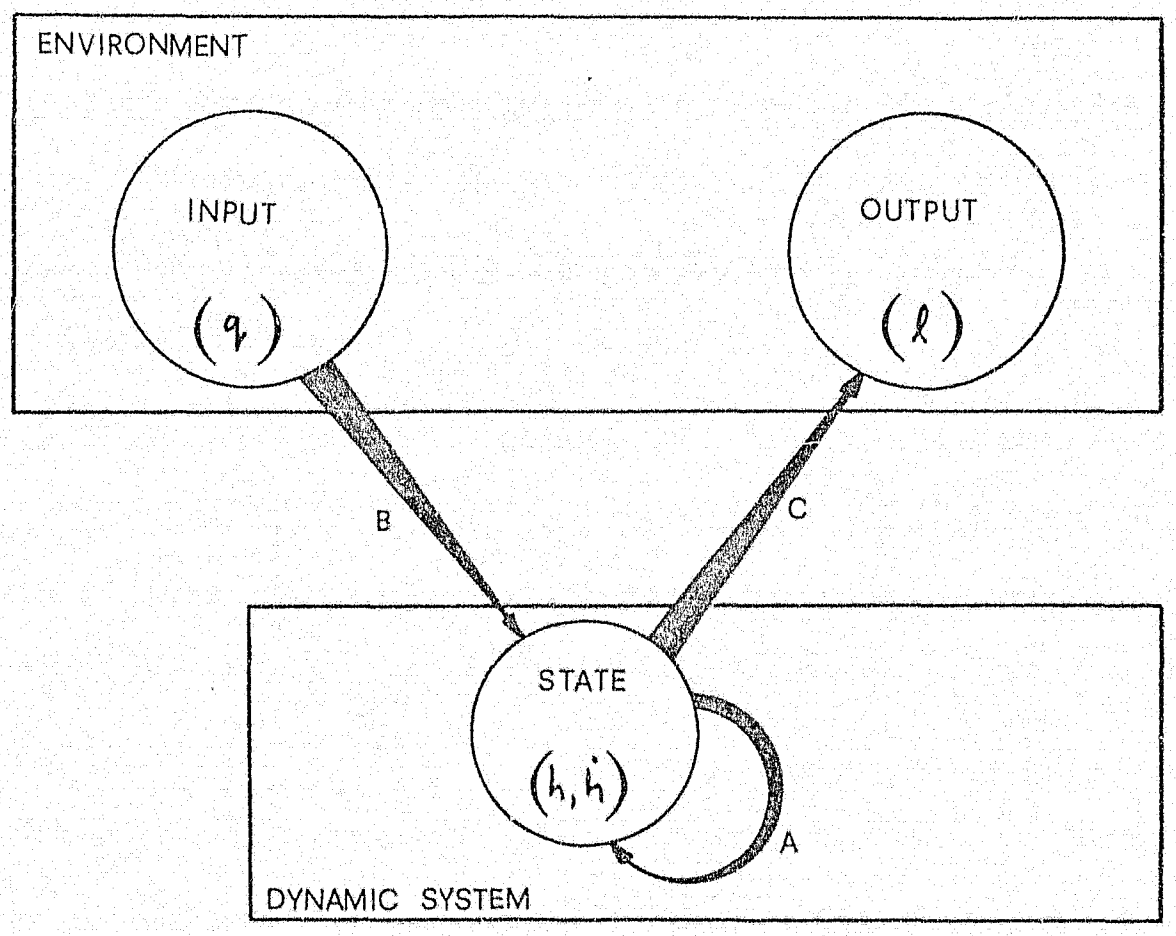


Fig. 2.2 : State-space Formulation

It is therefore concluded that the implemented time domain simulation deals only with the dynamical map, ie. the evolution of the states. The justification for such a simplification has already been mentioned in section 2.2. At this stage it is important to note that such a simplified 'reduced-order' model deepens insight into system structure, ie. the behaviour of the dominant modes. This also enables the use of direct-analytic techniques in the choice of controller structure as is done in the next section.

2.4. SURVEY OF MULTIVARIABLE DESIGN METHODS

This section presents a survey of available methods for design of multivariable controllers as they are applied to the plant under consideration. The techniques discussed are based on linear theory and therefore, applicable on the real system for small pertubations only.

The following simple analysis develop suitable multivariable control laws for the system under study. The ideas used are similar to those of the Inverse Nyquist Assay |18| and Characteristic Loci |15| design methods, however the full generality of these methods is not necessary here.

It must be stressed that the controllers obtained using these methods are only suitable in the case where plant parameters are known to remain essentially constant. In this case, it is believed that there is a vital need to retain the simplicity, familiarity and known robustness of fixed parameter, non-adaptive control methodologies.

The overall system transfer function can be written, referring to Fig. 2.3, in the following Laplace domain format,

$$G(s)_{(system)} = G(s)_{(middle)} \cdot G_0 \cdot G(s)_{(top)} \qquad -(2.7)$$

$$= \begin{bmatrix} \frac{1/A_3}{s+C_3} & \emptyset \\ \emptyset & \frac{1/A_4}{s+C_4} \end{bmatrix} \begin{bmatrix} A_1B_{13} & A_2B_{23} \\ A_1B_{14} & A_2B_{24} \end{bmatrix} \begin{bmatrix} \frac{1/A_1}{s+(B_{13}+B_{14})} & \emptyset \\ \emptyset & \frac{1/A_2}{s+(B_{23}+B_{24})} \end{bmatrix} \qquad -(2.8)$$

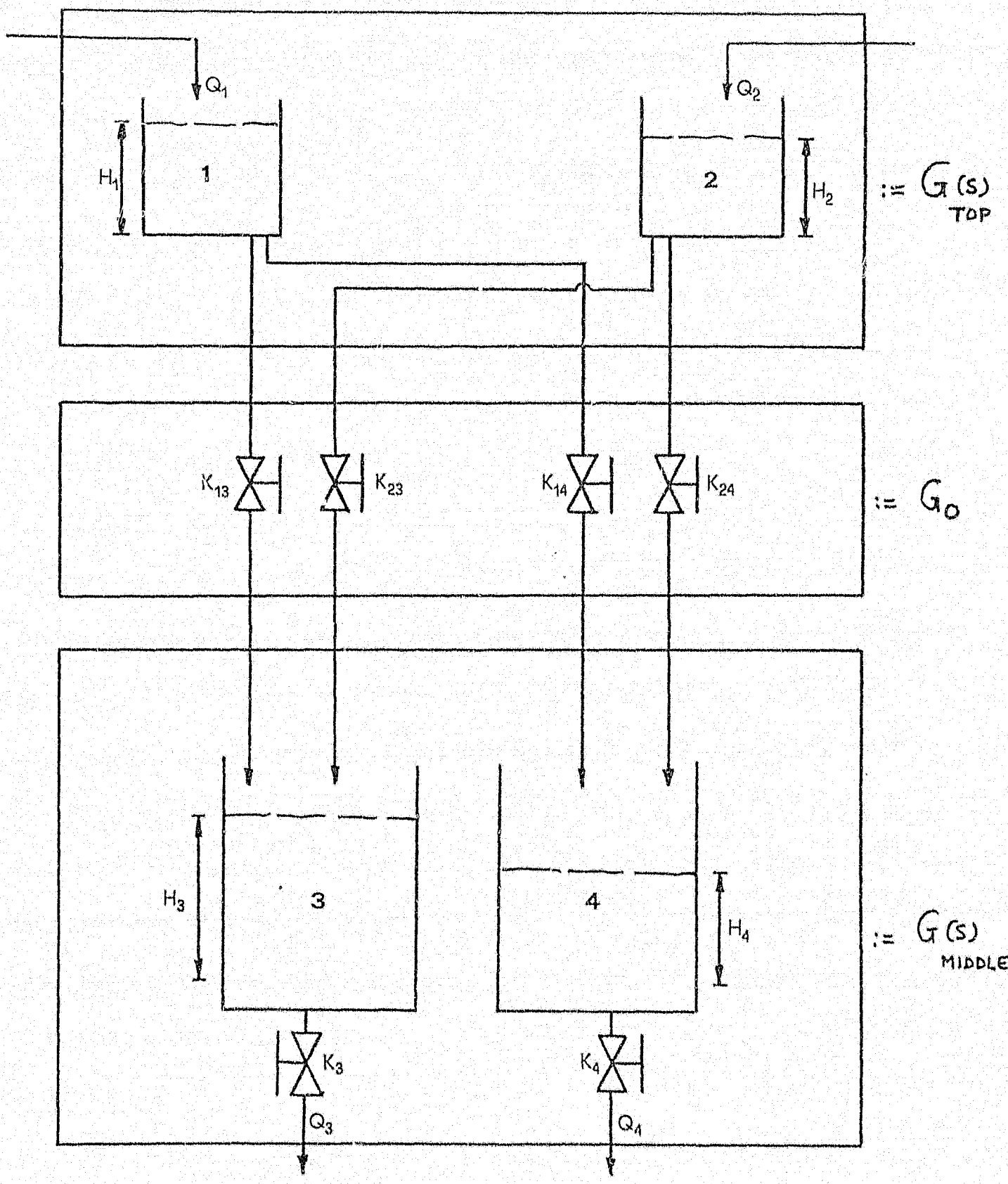


Fig: 2.3 : Plant partitioning into three constituent transfer functions

Depending on the G_o matrix, and hence on the relative interaction discharge coefficients (K_{i3}, K_{i4}, i=1,2) the following conditions can evidently arise

2.4.1: Case (1) $K_{23} = \emptyset = K_{14}$ (Non-interaction)

In this case the system reduces to two separate single-input single-output systems. With the above condition the overall system transfer function becomes,

$$G(s)_{(system)} = G(s)_{(middle)} \begin{bmatrix} A_1B_{13} & \emptyset \\ \emptyset & A_2B_{24} \end{bmatrix} G(s)_{(top)} \qquad -(2\text{-}9)$$

This is obviously equivalent to making $K_{13} = \emptyset = K_{24}$, the inputs have been swapped but system can still be controlled using classical PID techniques.

2.4.2: Case (2): $K_{ij} \neq \emptyset$ (Interaction)

The interacting case is more complex than the above, and three further situations can be conceived, as analyzed subsequently

2A : $K_{13} = K_{24}$; $K_{23} = K_{14}$

The system transfer function becomes,

$$G(s)_{(system)} = G(s)_{(middle)} \begin{bmatrix} G_{01} & G_{02} \\ G_{02} & G_{01} \end{bmatrix} G(s)_{(top)} \qquad -(2.10)$$

Assuming the middle tanks to have identical operating conditions, then (2.10) can be written as,

$$G_{(system)}(s) = \begin{bmatrix} G_{middle} & \emptyset \\ \emptyset & G_{middle} \end{bmatrix} \begin{bmatrix} G_{top} & \emptyset \\ \emptyset & G_{top} \end{bmatrix} \begin{bmatrix} G_{01} & G_{02} \\ G_{02} & G_{01} \end{bmatrix} \quad - (2.11)$$

Therefore by choosing a forward path pre-controller with the following form,

$$K_p = (G_o)^{-1} \quad - (2.12)$$

The system can once again be controlled using non-interacting classical techniques.

2B: $\underline{K_{13} = K_{23} = K_{14} = K_{24}}$

Under these conditions the system transfer function becomes

$$G_{(system}(s) = G_{(middle)}(s) \begin{bmatrix} G_o & G_o \\ G_o & G_o \end{bmatrix} G_{top}(s) \quad - (2.13)$$

$$= \begin{bmatrix} G_1 & G_1 \\ G_1 & G_1 \end{bmatrix} \quad \text{under same assumptions}$$

$$= \begin{bmatrix} G_1 \\ G_1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \end{bmatrix} \quad -(2.14)$$

Hence the ratio,

$$\frac{\rho_C}{\rho_O} = [\, I_2 + GK]$$

$$= \quad I_2 + \begin{bmatrix} G_1 \\ G_1 \end{bmatrix} [1 \quad 1][K]$$

$$= 1 + [1 \quad 1][K]\begin{bmatrix} G_1 \\ G_1 \end{bmatrix} . \qquad -(2.15)$$

Therefore by choosing

$$K = \tfrac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} [K_1 \; K_1], \qquad -(2.16)$$

the above ratio becomes

$$\frac{\rho_C}{\rho_O} = 1 + 2\,(G_1 \; K_1) \qquad -(2.17)$$

The Nyquist plot of the second term can then yield stability information and lead to a reasonable choice of K .

2C: $K_{13} \neq K_{23} \neq K_{14} \neq K_{24}$

This situation is represented by the following transfer function,

$$G_{(system)}(s) = G_{(middle)}(s) \begin{bmatrix} G_{01} & G_{02} \\ G_{03} & G_{04} \end{bmatrix} G_{(top)}(s) \qquad -(2.18)$$

Again

$$\frac{\rho_C}{\rho_O} = [I_2 + GK]$$

$$= \left[I_2 + G_{(middle)}(s) \cdot G_O \cdot G_{(top)}(s) \cdot K(s)\right]$$

$$= \left[I_2 + G_{(top)}(s) \cdot K(s) \cdot G_{(middle)}(s) \cdot G_O\right]$$

By choosing K(s) diagonal this yields,

$$\frac{p_c}{p_o} = I_2 + \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix}\begin{bmatrix} G_{01} & G_{02} \\ G_{03} & G_{04} \end{bmatrix}$$

$$= I_2 + \begin{bmatrix} g_1 G_{01} & g_1 G_{02} \\ g_2 G_{03} & g_2 G_{04} \end{bmatrix}$$

$$= g_1 g_2 \begin{bmatrix} g_1^{-1} + G_{01} & G_{02} \\ G_{03} & g_2^{-1} + G_{04} \end{bmatrix} \qquad -(2.19)$$

Analysis can once again proceed, in the Nyquist plane, of the much simplified second term of equation (2.19), yielding information about stability conditions.

In this section it was shown how classical multivariable controller design techniques in line with Rosenbrock and MacFarlane can be applied to a plant after appropriate manipulation of its transfer function. These methods although robust in the sense of being insensitive to the unknown plant dynamics (17), only cope with relatively constant parameter plants. Further disadvantages are that the pre-controllers tend to be complicated and unstable for non-minimum phase systems.

2.5. TIME-DOMAIN SIMULATION

This computer simulation package has been developed for the purpose of modelling systems described by time-dependent, non-linear differential equations. The basic structure of the program suite follows the specification established by ACSL - Advanced Continous Simulation Language |32|. In this way a simple method of representing mathematical models on the digital computer is achieved.

The time-domain simulation package is a direct implementation of equations (2.4). The integration method chosen was the fourth-order Runge-Kutta. Higher order Runge-Kutta formulae are also available, however the additional accuracy obtained is offset by the additional computational effort required. Hence, there is no significant improvement in efficiency over the fourth-order scheme.

This digital time-domain non-linear simulation can be used in two ways:

(i) to test self-tuning algorithms before trying to control the actual plant, thus enabling the user to acquire deeper knowledge of the algorithms and their parameters.

(ii) to enable the user to obtain better knowledge of the plant by running extensive step tests on the simulation, with varying plant parameters. Graph 2.1 depicts one such run.

In Chapter 5 an overall analysis of the entire software suite is given. The simulation package consists, basically, of two modules, namely the master program and the intergration routines.

The master program contains the user initialization block and the plant-parameter modification block which is optional. It also enables the user to run the simulation, or the plant (i.e. manual mode) or both.

The intergration routines, integrate flow over one sample interval. A fourth-order Runge-Kutta routine performs this in 10 steps. As can be seen from the descriptive software listings in Appendix 2 it was tried to minimize the complexity of each block thus enhancing its readability and testability capabilities. The main advantage of this format is that this particular simulation can be used with other plant models, by altering the plant-dynamic model block.

2.6. DISCRETE-TIME MODEL

It is well known in classical feedback control that many high-order continuous linear, time-variant systems can be approximated, for the purpose of feedback design, by a low-order state-space model. This is due to the presence of approximately cancelling pole-zero pairs in the system transfer function. Furthermore, the validity of such approximations can improve in the closed-loop situation owing to the attraction of closed-loop poles to the system zeros. As might be expected, intuitively, these ideas can be generalised to the multi-variable case.

Self-tuning is a direct digital control technique which is normally applied to continuous time systems. Hence as a prelude to the main body of self-tuning theory (Chapters 3 & 4), it is necessary to consider the representation of systems by digital (Sampled data) models.

The problem is to represent a continuous time system in a form which the digital computer can understand. This means converting ordinary differential equation models into ordinary difference equation models. One way of approachng this is to use existing Z-transform tables to find the equivalent Z-domain formulation of the plant plus zero-order-hold transfer functions.

Two approaches are used. Firstly the coupled tanks second order transfer function with no transportation delays (G(s) is cascaded with a zero-hold yielding the following equivalent pair,

$$\left(\frac{1-e^{-sT_s}}{s}\right) \quad G(s) := \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \qquad -(2.20)$$

Secondly a pure delay approximation of the zero-order hold is used. The validity of this approximation is substantiated by the low pass characteristic of G(s) and fast sampling rates compared to the plant time constants |28|. The equivalent s - z domain pair is

$$e^{-s\frac{T_s}{2}} \qquad := \frac{b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \qquad -(2.21)$$

D.W. Clarke |9| gives a first principles derivation of the discrete time model. In Appendix 1 the formulae for $B(z^{-1})$ and $A(z^{-1})$ polynomials above, in terms of the elements of G(s) are included.

The standard digital model of a continuous system produced in this way is therefore given by;

$$y_k = \frac{z^{-k_{dB}}\,(z^{-1})}{1 + A(z^{-1})}\, u_k \qquad -(2.22)$$

In general the system output sequence will be subject to a disturbance n(t) which is an unobservable nuisance one wishes to remove. The above model is thus extended to include this disturbance,

$$y_k = -A(z^{-1})y_k + z^{-k_{dB}}\,(z^{-1})u_k + n_k \qquad -(2.23)$$

One of the purposes of applying feedback control is to reduce the influence of n(t) upon the output y(t). The normal procedure is to assume that n(t) can be represented by the model,

$$n_k = C(z^{-1})\, e_k \qquad -(2.24)$$

where,

$$C(z^{-1}) = 1 + C_1 z^{-1} + \ldots + C_{n_c} z^{-n_c}$$

$e(t)$ $(t = 0, \pm 1, \pm 2 \ldots)$

is a white (uncorrelated) noise sequence.

The final discrete time model can then be written as,

$$y_k = \underbrace{\frac{z^{-k}B(z^{-1})}{1+A(z^{-1})}}_{V(z^{-1})} u_k + \underbrace{\frac{C(z^{-1})}{1 + A(z^{-1})}}_{W(z^{-1})} e_k \qquad -(2.25)$$

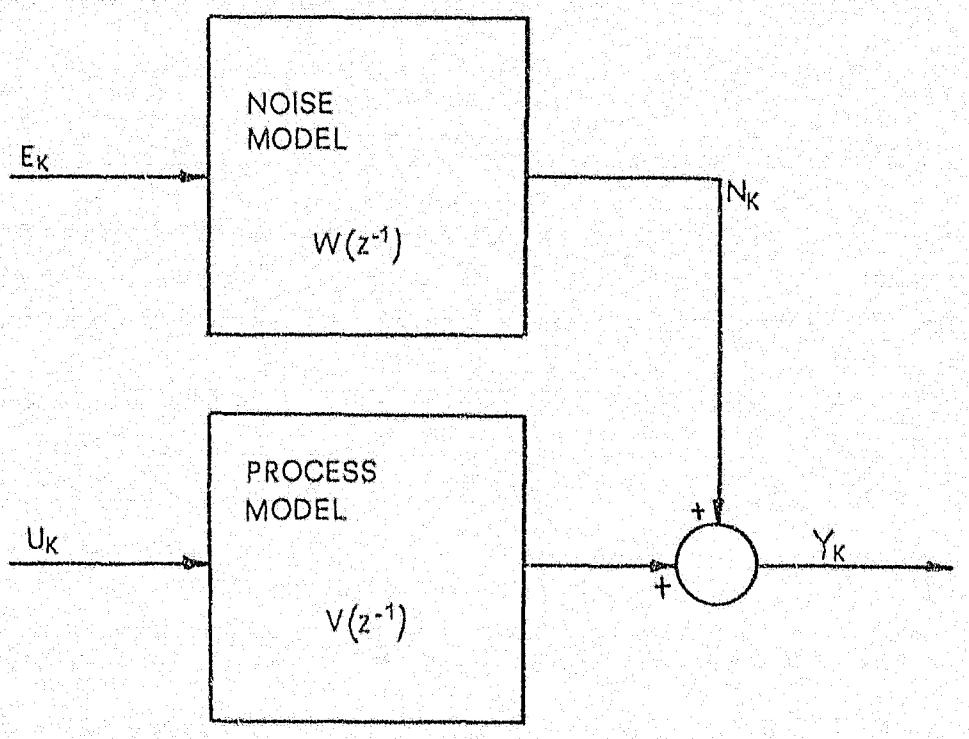


Fig. 2.4. Block diagram of system model

The above theory can be extended to the multivariable case by realizing (from Fig. 2.1) that <u>each</u> input-output relationship can be represented by the above model. If this is done and using the model in 2.21), then (2.26) is obtained. This on-line model of the system is formed under the assumption that $C(z^{-1}) = \emptyset$.

$$\begin{bmatrix} y_k^1 \\ y_k^2 \end{bmatrix} = \begin{bmatrix} -\alpha_1 & \emptyset \\ \emptyset & -\alpha_2 \end{bmatrix} \begin{bmatrix} y_{k-1}^1 \\ y_{k-1}^2 \end{bmatrix} + \begin{bmatrix} -\alpha_3 & \emptyset \\ \emptyset & -\alpha_4 \end{bmatrix} \begin{bmatrix} y_{k-2}^1 \\ y_{k-2}^2 \end{bmatrix} + \begin{bmatrix} \beta_1 & \beta_3 \\ \beta_7 & \beta_5 \end{bmatrix} \begin{bmatrix} u_{k-1}^1 \\ u_{k-1}^2 \end{bmatrix} + \begin{bmatrix} \beta_2 & \beta_4 \\ \beta_8 & \beta_6 \end{bmatrix} \begin{bmatrix} u_{k-2}^1 \\ u_{k-2}^2 \end{bmatrix} \qquad -(2.26)$$

It is important to note that the above model is linear and therefore, applicable on the actual non-linear system for perturbations only. Hence y_k^i and u_k^i in (2.26) must be replaced by their variations from the steady-state values,

$$u_k^i = U_k^i - U_o^i$$

and $$-(2.27)$$

$$y_k^i = Y_k^i - Y_o^i \qquad\qquad i = 1,2$$

where Y_k^i and U_k^i are the measured process input and output signals respectively, and Y_o^i and U_o^i are the corresponding D.C. values.

Substituting (2.27) into (2.26) a similar equation is obtained but with two added terms,

$$\begin{bmatrix} C_{dc}^1 \\ C_{dc}^2 \end{bmatrix} = \begin{bmatrix} (1+\alpha_1+\alpha_3) & Y_o^1 \\ (1+\alpha_2+\alpha_4) & Y_o^2 \end{bmatrix} + \begin{bmatrix} -(\beta_1+\beta_2) & -(\beta_3+\beta_4) \\ -(\beta_7+\beta_8) & -(\beta_5+\beta_6) \end{bmatrix} \begin{bmatrix} U_o^1 \\ U_o^2 \end{bmatrix} \qquad -(2.28)$$

In chapter 5 a more detailed analysis is given on how to estimate these d.c. values. Isermann [13] also discusses this problem.

If the model of equation (2.20) is used then the last term in equation (2.26) is not considered.

2.7. DISCUSSION

The aim of this chapter was to formulate a discrete-time plant model, to obtain a computer simulation of the plant, and to develop some multi-variable controller designs illustrating the lack of design flexibility of the classical MIMO design methods.

The system exhibits features which are typical of process level control loops. In particular, it embodies severe operating-point dependent non-linearities, as is illustrated by considering the linearized system model about two mean operating levels. Further the actuators and sensors have also a non-linear characteristic, these have not, however been considered in the development of the model because their dynamics are much faster than the system, also in this way a clear distinction is made between system control problems and system actuation problems.

Since the system's dominant time constants are of the order of minutes, no problems are envisaged with the finite time delay existent due to the time required to calculate the 'next' control. Otherwise a Z transform table involving fractional time delays given by Wellstead [21] could be of use.

CHAPTER 3

SYSTEM IDENTIFICATION

3.1. INTRODUCTION

One conventional method of determining the dynamic characteristics of an unknown plant or process is to apply a step or sinusoidal test input to the process and analyze the corresponding output. Another method is to introduce a pseudo-random noise input signal and then cross correlate this with the output. The transfer characteristics in either case are therefore described as the step response, frequency response or impulse response, respectively. The main disadvantages of these methods are as follows :

(i) The transfer characteristics if obtained by isolated tests, may only be true for that particular type of input, and may differ from those obtained under actual working conditions. This being especially true if the process has some non-linearity.

(ii) The test input applied to the system may disturb it too much, eg. it could break into oscillation. It often proves too expensive to remove the plant from its normal duties to allow the test input to be applied.

(iii) If the transfer characteristics vary with time the results obtained will, eventually, become irrelevant.

Hence a method is required which enables the determination of the system transfer function model from records of system operation. To this end a recursive multivariable least-squares estimator is developed and implemented in a digital computer. The former methods are justified

mainly when large disturbances can be tolerated, or when interest is centered on system behaviour at low frequencies.

In this chapter the identification of digital models (considered in the previous chapter) is covered, together with details of the recursive implementation as it is used in self-tuning.

The chapter does not include detailed derivations of the standard least-squares algorithm, which are extensively reported in the literature, ie. Eykoff |27| and Davies |26|. Backround material, such as aspects of statistics important for the development of prediction analysis, has also been omitted since it is also well documented in the literature, ie. Wolberg |31|.

The recursive least-squares estimator was chosen as the basic identification scheme because it was the most easily understood. However, there exists a whole class of recursive estimators which have the same basic algorithmic structure. These are formulated by Isermann |12| & |13| and well documented by Astrom and Eykoff |4|.

3.2. RECURSIVE LEAST-SQUARES ALGORITHM : ANALYSIS

In regression analysis (Clarke |9|) an observation (output) is assumed to be a linear combination of explanatory variables (inputs), and a set of observations is used to estimate the weighting on each variable such that some fitting criterion is optimised. The chosen criterion estimates the model parameters such that the sum-of-squares of the errors between the model outputs and the observations is minimised (least-squares).

The fundamental principles of least-squares theory were first proposed by Karl Gauss in the early part of the nineteenth century, for carrying out his work in astronomy. Prior to the availability of digital computers, however, it was rarely used due to the complex computations required. Widespread availability of digital computers has completely altered the [illegible]e theory has become a major tool for parameter estimation.

The reasons for its popularity over several other estimation methods available are twofold. Firstly, the method is easier to understand and hence apply than others, as it does not require a deep knowledge of mathematical statistics. Secondly, the estimates obtained are consistent and unbiased hence possessing optimal statistical properties.

A further characteristic is that a recursive algorithm can easily be derived from the basic least-squares formulation. The need for a recursive solution arises when fresh experimental data continuously become available and the new information has to be used to improve the parameter estimates. A step-by-step updating of the estimates is, in this way, achieved.

A complete derivation of how the recursive formulae can be derived from the original least-squares estimator is included in Davies |26|

The unified form of the recursive least-squares algorithm is summarized by the following five formulae,

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \phi_k \, e_{k+1} \qquad -(3.1a)$$

$$\phi_k = P_k \, \psi_{k+1} \, \gamma_{k+1} \qquad -(3.1b)$$

$$e_{k+1} = Y_{k+1} - \psi_{k+1}^T \, \hat{\Theta}_k \qquad -(3.1c)$$

$$\gamma_{k+1} = [R + \psi_{k+1}^T \, P_k \, \psi_{k+1}]^{-1} \qquad -(3.1d)$$

$$P_{k+1} = \frac{[I - \phi_k \, \psi_{k+1}^T] P_k}{\lambda_{k+1}} \qquad -(3.1e)$$

This set of formulae can be seen as an information processing 'black box', as depicted in Fig. 3.1.

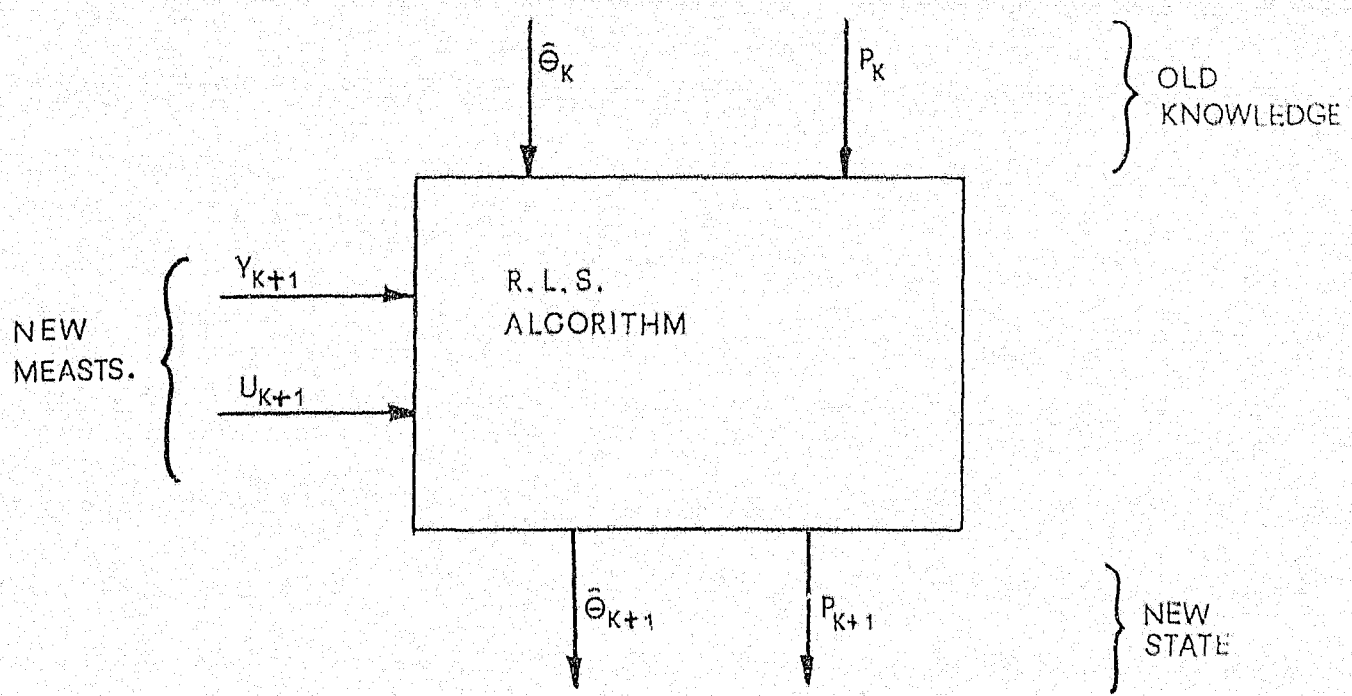


Fig 3.1 : Schematic of l.s. algorithm

Equation (3.1a) is typical of the updating nature of this sequential algorithm. The new parameter vector $\hat{\theta}_{k+1}$ is given by the old estimate $\hat{\theta}_k$ plus a correction term. This term is proportional to the error e_{k+1} of fitting the previous estimate $\hat{\theta}_k$ to the new data or measurement vector ψ_{k+1} as calculated by (3.1c)

The estimator gain ϕ_k is calculated from (3.1b). The confidence matrix P_{k+1} is a measure of the quality of the estimates, the size of its diagonal elements determining the variability of the corresponding element in the estimate vector. The name confidence matrix springs from the fact that equation (3.1a), after substitution of (3.1b) can be written as,

$$\hat{\theta}_{k+1} = \hat{\theta}_k + P_k \psi_{k+1} \gamma_{k+1} e_{k+1} \qquad -(3.1f)$$

This matrix is itself updated every step via (3.1e). If the parameters were essentially constant during the period of estimation, so that most recent data was as good as older data for providing information about the unknown parameter values then (3.1e) would be valid with a forgetting factor λ equal to unity. However, as is explained in more detail, and in the context of multivariable estimation, in the next section, this is not the case. Hence a real-time sequential algorithm has to be developed to closely track time-varying parameters. The improvement is that a weighting scheme is introduced to place heavier emphasis on the more recent data when the situation warrants such action. This is done via the forgetting factor method explained in section 3.3, or the random walk method explained subsequently.

The main problem arises from the fact that as the number of recursions in the least-squares algorithm increases, the parameter estimates converge and become more steady. This convergence is reflected in a decrease in the size of the elements of P. With reference to equation (3.1f), the magnitude of P governs the size of the update to the parameter estimates. When parameters are time-varying it is necessary to prevent the 'confidence matrix' from becoming too small, thus allowing the error e, when the system output γ deviates from the model output, to provide a continuous correction to the parameter estimates.

In practice, a trade-off is sought between the parameter adaptive capabilities (requiring P large) and the noise rejection features of the estimator (requiring P small).

As mentioned, the forgetting factor approach for manipulating the magnitude of the elements P, amounts to building a finite memory into the estimator. By decreasing λ the elements of P are slightly increased at each step, thus allowing the forgetting of old parameter values faster.

The random walk approach increases the size of the P matrix, when adaption is required by adding a constant matrix to it. Owing to the format of the confidence matrix, ie. diagonal elements are the variances of the estimates, whilst the off-diagonal elements represent the covariances between the respective estimates, individual elements of P can be adjusted via the random-walk method. Hence this method is more selective, but also requires more care in its application. No adequate literature was found on the usage of the random walk method. Further

the forgetting factor method, as will be seen in the next section, is an entirely general method, and it is thought that the size of the elements of the random walk correction matrix would be dependent on the particular application.

Forgetting factors are used to maintain an adaptive capability throughout the particular experiment. Random-walks can, perhaps be used when the algorithm is required to returne from an undesirable situation, in a parameter selective way. For the above reasons random walks were disregarded.

The uncertainty matrix R in (3.1.d) is diagonal and represents the expected value of the variance of the observaticns. In other words if the actual level in the tank is x, then it is assumed that the input part of the measurement and control processor reads $x + \Delta x$. In this way the particular diagonal element of R is $E(\Delta x^2)$. The diagonality of R stems from the obvious fact that level measurement in one tank does not affect that of the second tank. Also since the inputs to the system are not dependent on each other, the γ matrix is diagonal, this will be clearly seen subsequently when the form of the measurement and estimate vectors is outlined.

The γ inverse matrix can be seen to represent the confidence or covariance matrix of the error vector e. This is because R is the covariance matrix of the output measurement while $\psi^T P \psi$ is the covariance matrix of the estimated system output using the old parameters.

Further analysis leads to writing (3.1a) as follows,

$$\tilde{\Theta}_{k+1} - \tilde{\Theta}_k = \phi\ (Y_{k+1} - \hat{Y}_{k+1}) \qquad -(3.1g)$$

where $\hat{Y}_{k+1} = \psi^T{}_{k+1}\ \hat{\Theta}_k$ $\qquad -(3.1h)$

Hence according to the earlier analysis of the estimator gain matrix, it is concluded that (3.1g) can be described by the following,

$$\tilde{\theta}_{new} - \hat{\theta}_{old} = \left(\frac{\text{Lack of confidence in } \tilde{\theta}}{\text{Lack of confidence in } \gamma}\right) (Y_{k+1} - \hat{Y}_{k+1}) \qquad -(3.1i)$$

Re-writing (3.1b) and (3.1d) as follows,

$$\phi_k = \frac{P_k\ \psi_{k+1}}{[R + \psi^T{}_{k+1}\ P_k\ \psi_{k+1}]} \qquad -(3.1j)$$

Combining the last two equations with the points made in the previous paragraph it is concluded that the matrix $P_k\ \psi_{k+1}$ must be the covariance matrix of the left hand side of equation (3.1i)

One further point to be considered when dealing with the recursive least-squares estimator is that of initial conditions. It is agreed that there are two ways of starting the estimator, analyzed briefly as follows,

(i) Input the first 'm' data measurements and solve for $\hat{\Theta}_m$ and P_m directly from the original non-recursive least-squares formulation,

$$\hat{\Theta}_m = (X_m^T X_m)^{-1} X_m^T Y_m$$
$$P_m = (X_m^T X_m)^{-1} \qquad -(3.2)$$

Thus allowing the algorithm to iterate from (m+1) points onward. X_m is a matrix comprising of the measurement vectors for the first 'm' samples.

(ii) Set $\hat{\Theta}_o$ to some arbitrary value of $P_o = \alpha I$, where a is a very large positive scalar, reflecting the erroneous nature of $\hat{\Theta}_o$. Starting with these values and iterating 'm' times using equation (3.1) yields,

$$P_m = (P_o^{-1} + X_m^T X_m)^{-1}$$
$$\hat{\Theta}_m = P_m [X_m^T Y_m + P_o^{-1} \hat{\Theta}_o] \qquad -(3\text{-}3a)$$

If α is very large then P_o^{-1} is very small and hence negligible, obtaining

$$P_m = (X_m^T X_m)^{-1}$$

and $\qquad -(3.3b)$

$$\hat{\Theta}_m = P_m X_m^T Y_m$$

which agreed totally with equations (3.2) above.
In section 3.4 some results of the effect of starting with different initial guesses are presented. The further these initial guesses are from their actual values the greater the probability of divergence.

If equation (2.23) is re-written in the following form,

$$y_k = -\sum_{i=1}^{n_a} \alpha_i\, y(t-i) + \sum_{j=1}^{n_b} \beta_j\, u(t-j) + \varsigma_k \qquad -(3.4)$$

and again in vector matrix form,

$$y_k = \psi_k^T \hat{\Theta}_k + \varsigma_k \qquad -(3.5)$$

Now assuming the two system discrete-time models given by (2.20) and (2.21) respectively, the following parameter vectors are formed,

$$(i)\ \hat{\Theta}^T = [\alpha_1;\alpha_2;\alpha_3;\alpha_4;\beta_1;\beta_4;\beta_2;\beta_3;C_{dc}^1;C_{dc}^2] \qquad -(3.6a)$$

$$(ii)\ \hat{\Theta}^T = [\alpha_1;\alpha_2;\alpha_3;\alpha_4;\beta_1;\beta_5;\beta_2;\beta_6;\beta_3;\beta_7;\beta_4;\beta_8;C_{dc}^1;C_{dc}^2] \qquad -(3.6b)$$

where the last two β's ($\beta_2;\beta_3$) in the former and the last four β's ($\beta_3;\beta_7;\beta_4;\beta_8$) in the latter represent the interaction terms that is those parameters for one output which are affected by the other input.

Taking into consideration the model described by equation (2.26) it is possible to form the measurement vector as follows,

$$(1)\ \psi^T = \begin{bmatrix} -y_{k-1}^1 & \emptyset & -y_{k-2}^1 & \emptyset & u_{k-1}^1 & \emptyset & u_{k-1}^2 & \emptyset & 1 & \emptyset \\ \emptyset & -y_{k-1}^2 & \emptyset & -y_{k-2}^2 & \emptyset & u_{k-2}^2 & \emptyset & u_{k-1}^1 & \emptyset & 1 \end{bmatrix} \qquad -(3.7a)$$

$$(ii)\ \psi^T = \begin{bmatrix} -y_{k-1}^1 & \emptyset & -y_{k-2}^1 & \emptyset & u_{k-1}^1 & \emptyset & u_{k-2}^1 & \emptyset & u_{k-1}^2 & \emptyset & u_{k-2}^2 & \emptyset & 1 & \emptyset \\ \emptyset & -y_{k-1}^2 & \emptyset & -y_{k-2}^2 & \emptyset & u_{k-1}^2 & \emptyset & u_{k-2}^2 & \emptyset & u_{k-1}^1 & \emptyset & u_{k-2}^1 & \emptyset & 1 \end{bmatrix}$$

$$-(3\text{-}7\text{-}b)$$

The structure of the program that implements the algorithm in (3.1) is attached in Appendix 2. It basically consists of a main program that executes the required matrix/vector operations and a formation routine that assembles the data in the format given by (3.7) at the start of each calculation or sample interval.

Bearing in mind the above equations, it is realized that parts of P refer to the estimated coefficients of $A(z^{-1})$ and the remainder concerns the estimated coefficients of $B(z^{-1})$ and of the steady-state values (C_{dc}^{i} i = 1,2), one can partition theconfidence matrix as,

$$P_k = \begin{bmatrix} P(A)_k & P(A,B)_k & P(A,C_d)_k \\ . & P(B)_k & P(B,C_{dc})_k \\ . & & \\ . \quad . \quad . & . & P(C_{dc})_k \end{bmatrix} \qquad -(3.8)$$

$P(A)_k$ being a submatrix specifying the variability or degree of confidence in the coefficients of $A(z^{-1})$ whereas $P(A,B)_k$ and its transposed element $P(B,A)_k$ define the co-variability of the coefficients of $A(z^{-1})$ and $B(z^{-1})$.

In the next chapter a brief discussion is given of how this recursive least-squares estimator fits in with the self-tuning property of the control system.

3.3. PARAMETER-ADAPTABILITY CHARACTERISTICS :

One way of rationalizing the use of forgetting factors and random walk is in terms of a mass balance on information. The recursive least-squares identifier accumulates extra information at each step. The increased information content is reflected in the improved parameter estimates and a decrease in the size of the elements of the confidence matrix. If the system being identified changes, then some of the old information will be wrong, as far as the new system is concerned. As mentioned previously, the forgetting factor and random walk methods provide a means of draining off old information at some rate.

The aim is to achieve an information balance such that the amount of forgetting at each step corresponds to the amount of new information in the latest measurement, thereby ensuring that the estimation is always based on the same amount of information. Further, as the results prove, a large factor in the success or otherwise of an identification, and hence self-tuning experiment is the way in which the algorithm can pay-off old information for new.

If this theory is to succeed, two concepts have to be analyzed in the context of multivariable parameter identification. Firstly, one must define a measure of the information content of the estimator, and secondly, a forgetting factor must be chosen at each calculation step such that this information content is kept constant.

Evidently the error between the actual measurement and calculated value using the parameter vector must have a central role to play in the way

this information content is defined and also in the way the forgetting factor is manipulated to achieve the objective of information balance. It is useful to remember at this stage the way in which the parameter adaptability characteristics of the estimator are obtained.: The confidence matrix is not allowed to become very small and hence any discrepancy between the actual and calculated outputs will lead to an adequate change in the particular parameters as given by (3.1a).

There are basically two conditions that can arise with regards to the error magnitude, these are examined below.

(i) Error is small, this can be due to three factors,
   (a) process has not been excited,
   (b) there has been an excitation with a near correct set of parameter values, and,
   (c) the estimator is sensitive enough to significantly reduce parameter errors.

   In all three cases the correct course of action is to retain as much information as possible by choosing a forgetting factor close to unity, ie. by giving the estimator an infinite memory.

(ii) Error is large, this can only mean, in terms of the estimator parameters, that these are incorrect. The estimator sensitivity should then be increased by choosing a lower forgetting factor, hence shortening the effective memory length of the estimator. This evidently gives rise to an increase in the

magnitude of the confidence matrix thus also increasing the size of the update to the parameter estimates.

Fortescue |11| suggests a new single-input/single-output recursive least-squares algorithm with variable weighting of past data, by choosing, at each sample interval, a forgetting factor which maintains constant a scalar measure of the information content of the estimator. This algorithm is subsequently analysed and adequately modified to accommodate the multivariable nature of the estimator under consideration.

The information content of the estimator is defined as the weighted sum of the squares of the error, which as derived by Albert |1| can be expressed recursively as,

$$\Sigma_{k+1} = \lambda_{k+1}\Sigma_k + [1-\psi_k^T \phi_{k+1}][Y_{k+1} - \hat{Y}_{k+1}]^2 \qquad -(3.9)$$

The strategy to choose the forgetting factor is then simply implemented by keeping the information content constant, ie. make

$$\Sigma_{k+1} = \Sigma_k = \ldots = \Sigma_o \qquad -(3.10)$$

Hence from (3.9) and (3.10) one obtains the forgetting factor updating equation,

$$\lambda_{k+1} = 1 - \frac{1}{\Sigma_o}(1 - \psi_k^T \phi_{k+1})\, \epsilon_{k+1}^2 \qquad -(3.11)$$

The second term on the right-hand side of the latter equation is now analysed in detail and a multivariable equivalent to Fortescue's algorithm will then be derived.

It was noted in the previous section that the matrix inverse of γ was the covariance matrix of [illegible] error in the estimates. Hence from (3.1.d).

$$[R + \psi^T P \psi] = \text{covariance matrix of } (Y-\hat{Y}) = E\{e^2\} \qquad -(3.12a)$$

In single-input/single-output analysis it is common practice to re-write the last equation in the following equivalent form.

$$[1 + \psi^T \frac{P}{R} \psi] = \frac{E\{e^2\}}{R} \qquad -(3\text{-}12b)$$

Also from equation (3.1j) it is readily seen that under the same transformation as (3.12b) above,

$$\frac{P}{R}\psi = \phi[1 + \psi^T \frac{P}{R} \psi] \qquad -(3.12c)$$

Substitution of (3.12c) into (3.12b) and some manipulation the following relationship is obtained,

$$1 = \frac{E\{e^2\}}{R} (1 - \psi_k^T \phi_{k+1}) \qquad -(3.13)$$

Therefore equation (3.11) due to this equality can be expected to yield,

$$\lambda_{k+1} = 1 - \frac{1}{\Sigma_{o/R}} \qquad -(3.14)$$

And hence the speed of adaptation is determined by the asymptstic memory length,

$$\frac{\Sigma_o}{R} \quad \frac{1}{1-\lambda_{k+1}} \qquad -(3.15)$$

It was noted in the previous section that the matrix inverse of γ was the covariance matrix of the error in the estimates. Hence from (3.1.d),

$$[R + \psi^T P\psi] = \text{covariance matrix of } (Y-\hat{Y})$$
$$= E\{e^2\} \qquad -(3.12a)$$

In single-input/single-output analysis it is common practice to re-write the last equation in the following equivalent form,

$$[1 + \psi^T \frac{P}{R} \psi] = \frac{E\{e^2\}}{R} \qquad -(3\text{-}12b)$$

Also from equation (3.1j) it is readily seen that under the same transformation as (3.12b) above,

$$\frac{P}{R}\psi = \phi[1 + \psi^T \frac{P}{R} \psi] \qquad -(3.12c)$$

Substitution of (3.12c) into (3.12b) and some manipulation the following relationship is obtained,

$$1 = \frac{E\{e^2\}}{R} (1 - \psi_k{}^T \phi_{k+1}) \qquad -(3.13)$$

Therefore equation (3.11) due to this equality can be expected to yield,

$$\lambda_{k+1} = 1 - \frac{1}{\Sigma_{o/R}} \qquad -(3.14)$$

And hence the speed of adaptation is determined by the asymptstic memory length,

$$\frac{\Sigma_o}{R} = \frac{1}{1-\lambda_{k+1}} \qquad -(3.15)$$

In the multivariable case R is no longer a scalar and hence the division leading to (3.12b) is not valid.

In the multivariable variable forgetting factor formulation one must consider the following matrix product,

$$N = e^T \left[R + \psi^T P \psi\right]^{-1} e \qquad -(3.16a)$$

Now writing the error vector as $e^T = [e_1 , e_2]$, and assuming the expected condition that the errors (e_1 and e_2) are uncorrelated then (3.16a) can also be written as,

$$N = [e_1 \; e_2] \begin{bmatrix} \sigma^2_{e_1} & \emptyset \\ \emptyset & \sigma^2_{e_2} \end{bmatrix}^{-1} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \qquad -(3.16b)$$

Expanding (3.16b) yields,

$$N = \frac{e_1^2}{\sigma^2_{e_1}} + \frac{e_2^2}{\sigma^2_{e_2}} \qquad -(3.16c)$$

$$= \text{dimension of } Y$$

Hence the multivariable counterpart of (3.11) is,

$$\lambda_{k+1} = 1 - \frac{1}{\Sigma_0} \frac{e^T_{k+1}\left[R+\psi^T_{k+1} P_k \psi_{k+1}\right]^{-1} e_{k+1}}{\dim\{Y\}}$$

$$\frac{1}{\Sigma_0 \dim\{Y\}} e^T_{k+1} \gamma_{k+1} e_{k+1} \qquad -(3.17)$$

From the asymtotic memory length can be deduced to be,

$$\Sigma_0 = \frac{1}{1-\alpha_{k+1}} \qquad -(3.18)$$

Author Ferreira A M P
Name of thesis A self-adaptive multivariable computer control system 1984

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.