STRATEGIES FOR MULTIVARIABLE SELF-TUNING CONTROL

Timothy Hesketh
B.Sc. Hons (Electrical Engineering)
M.Sc. (Electrical Engineering)

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy in Industrial Management and Engineering at Massey University.

July, 1981

ABSTRACT

The research reported in this thesis develops strategies for applying self-tuning control theory to multivariable processes. Self-tuning control of scalar processes is now reasonably well established, and attention is being turned to the multivariable problem, with its attendant computational burden to overcome, and additional considerations such as interaction and decoupling. This thesis suggests methods for dealing with these problems, and implements the controllers on one of the readily available desk-top microcomputer systems, providing a cheap yet effective controller.

The research builds on the work of Clarke and Gawthrop [1975, 1979] on implicit controllers. The explicit schemes are derived from controllers developed by Wellstead and others [1979,(a) and (b)], using the work on multivariable controllers by Borisson [1979] and latterly Koivo [1980]. The full multivariable controller relies on standard techniques for pole placement reported by Wolovich [1974].

The proposed controllers remove interactions between loops, and other disturbances, and ensure that each loop output attains the set-point required for that loop. In particular, the explicit pole-placing controller requires a pole-placing calculation for each loop, and removes the interactions with a minimum-variance-like action, the resulting controller being modest in computational needs.

A solution is also proposed for the full multivariable pole-placing problem, which allows a comparison to be made between the full multivariable solution and the simpler controllers proposed. It is found that the full multivariable controller demands more computation, but that the resulting control may be no different from that achieved with the simpler controller.

The programs which are described are written in Fortran, in the form of subroutines which are designed for incorporation into existing control program suites. Alternatively, they may be run with a dedicated supervisory program to handle timing, input and output, display and storage of information. The subroutines are general in their application, and the information they need for any particular process may be established interactively using an offline program.

Simulation results are presented which confirm the robustness of the pole placing controller, and indicate that the proposed techniques may be used to stabilise and control a wide variety of processes; those which are non-minimum phase, certain non-linear or unstable processes. On-line control of a commercial heat exchanger process is reported, the process being similar to others which have been reported, thus providing a point of comparison of self-tuning control with other techniques. The process is multivariable. Good control is achieved, particularly in the face of perturbations to the process which result in changes to the parameters of the model describing the process behaviour, conditions under which some other controllers may not be suitable.

This research has contributed techniques which may be applied successfully to multivariable self-tuning control. Efficient programs have been written which implement the controllers on a microcomputer. Suggestions for future work include the development of a program generator which will allow more compact code to be developed, dedicated to a particular process, and which will execute more quickly. Strategies which better enable self-tuning controllers to deal with non-linear processes are also of interest.

ACKNOWLEDGEMENTS

I wish to thank Professor J. K. Scott for his support, and the facilities offered by the Department of Industrial Management and Engineering; the New Zealand Dairy Research Institute, for access to their plant during experimental phases of this project; Dr R.I. Chaplin, who made possible the use of ISIS and provided supervision during the latter stages of this research; and most of all, my wife, Beryl.

TABLE OF CONTENTS

Page

CHAPTER 2 SELF-TUNING ALGORITHMS

CHAPTER 3 METHOD

NOTATION

Boldface letters are used to represent polynomials, polynomial matrices or matrices. A polynomial is an expression in the backward shift operator q^{-1}. Thus:

$$\mathbf{A}(q^{-1}) = a_0 + a_1q^{-1} + a_2q^{-2} + \ldots + a_{nA}q^{-nA}$$

n_A is defined to be the order of the polynomial **A**. Polynomials operate on process variables so that:

$$\mathbf{A}(q^{-1})y(t) = a_0y(t) + a_1q^{-1}y(t) + \ldots + a_{nA}q^{-nA}y(t)$$

$$= a_0y(t) + a_1y(t-1) + \ldots + a_{nA}y(t-n_A)$$

y is only defined at discrete time instants, y(t) being the (sampled) value of y at the present time. y(t-1), y(t-2) .. represent values of y at previous sampling instants. The sampling interval is not explicitly stated. The dependence of a polynomial on q^{-1} will be omitted for brevity unless ambiguity results.

A polynomial matrix may be represented equivalently by:

$$\mathbf{B} = \mathbf{B}_0 + \mathbf{B}_1q^{-1} + \ldots + \mathbf{B}_{nB}q^{-nB}$$

where the B_i are m x p matrices, or by:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \ldots & \mathbf{B}_{1p} \\ \ldots & \ldots & \ldots & \ldots \\ \mathbf{B}_{m1} & \ldots & \ldots & \mathbf{B}_{mp} \end{bmatrix}$$

where the $\mathbf{B}_{ij}$ are scalar polynomials of order n_B.

The scalar quantity A(1) is the value of polynomial **A** evaluated with $q^{-1}=1$.

Processes may be represented in Auto Regressive Moving Average (ARMA) form by:

$$\mathbf{A}y(t) = q^{-k}\mathbf{B}u(t) + q^{-k}\mathbf{D}v(t) + \mathbf{C}e(t) + d$$

A, **B**, **C** and **D** are polynomials (or polynomial matrices) representing a scalar (or a multivariable) process. The process time delay k is expressed as a whole number of sampling intervals. y(t) is the process output, u(t) the controlled input, v(t) a measured disturbance and e(t) a white noise sequence. d is an offset.

$y(t|t-k)$ is the predicted value of y at time t, given information up to and including time t-k.

Principal symbols used in this thesis

k	time delay
i,j,l,m,n	indices
A, B, C, D	Process polynomials/polynomial matrices
A1, A2, B1, C1	
A', B', D', A", B"	
y(t)	Process output
w(t)	Process set-point
u(t), u'(t)	Controlled input
v(t)	Measured disturbance
e(t), e'(t), e"(t)	White noise sequences
d, d'	Offset values
H, G, E	Controller polynomials $Hu(t) + Gy(T) + Ew(t) = 0$
$\mathbf{F}_c$	Multivariable pole placing polynomial matrix
S1, S2	Polynomials for servo control
P, P', Q, Q', R, S	General polynomials
L, M, B1, B2	
$\emptyset(t)$, $\emptyset_y(t)$, K	Scalar quantities
X, **θ**, **x**(t)	General vectors
$\mathbf{A}_m$, **T**, **T"**	Polynomials specifying poles for pole placement
$\mathbf{A}_0$	Observer polynomial