IFAC MMM 95

The 8th IFAC International Symposium on Automation in Mining, Mineral and Metal Processing

PREPRINTS

Edited by I.J. Barker

Sun City, South Africa
29 to 31 August, 1995

IFAC – International Federation of Automatic Control

Sponsored by
The IFAC Technical Committee on Automation in Mining, Mineral and Metal Processing

Organised by
South African Council for Automation and Computation

In association with
South African Institute of Measurement and Control
South African Institute of Electrical Engineers
South African Institute of Mining and Metallurgy

Topical Challenges of Control Engineering

BRIAN D O ANDERSON[1]
Research School of Information Sciences and Engineering
Australian National University, Canberra ACT 0200, Australia

ABSTRACT
Classical Control Engineering techniques can only go so far. Applications requirements have posed a series of challenges to the old methods, some of which are discussed in this paper. These include the challenge of complexity, as illustrated by flight control, the challenge of robustness, as illustrated by guided vehicle control and the challenge of achieving adaptivity, as illustrated in a sugar mill control problem. Sometimes more than one challenge may be present, as in an alumina calciner problem and a reheat furnace problem. Controller implementation is a further challenge.

1 INTRODUCTION

Classical Control has found extraordinarily wide application and despite the complexities of a great many plants in processing industries, the humble PID controller, rooted in classical control origins, probably remains the most popular control technology in use today. It is nevertheless the case that classical control has very substantial limitations, which prevent it from addressing the full range of control problems presenting themselves today. These limitations stem from the fact that classical control principally is concerned with single input, single output, time invariant plants, and usually plants of sufficiently low complexity that graphical design tools can lead to a successful controller design.

The inability of classical control methods to tackle all problems of application interest has been one of the spurs to lead to further developments in control system methodologies and theory, and the purpose of this talk is to review some of the challenges which were posed to classical control methodologies and to make some very broad brush remarks about how theoretically-based developments to meet these challenges are being translated into practice.

In this paper, the focus will be on four challenges. There are many others, and no reader should believe that this list is in any sense exhaustive. The challenges are those of complexity, of robustness, of adaptation, and of controller implementation.

[1]The author wishes to acknowledge the funding of the activities of the Cooperative Research Centre for Robust and Adaptive Systems by the Australian Commonwealth Government under the Cooperative Research Centre Program

2 THE CHALLENGE OF COMPLEXITY

Control of a modern aircraft provides a good example of a control challenge with significant complexity. For control of the pitch of the aircraft, flaps and ailerons are used, and the relevant output variables are attitude and angular velocity. Because of the presence of more than one input and more than one output, there is an immediate level of complexity. Classical control has enormous difficulty with multiple input, multiple output problems, apart from those which can be approximated by several decoupled single input, single output problems.

But the aircraft problem is complex for yet another reason. The number of internal variables in a mathematical model (for pitch control purposes) is about 50. The rules of thumb of classical control, the handbook solutions, the graphical procedures, and even excellent physical intuition are simply not enough to design a controller when this sort of complexity is present. This statement would probably hold true even if the system in question were single input, single output.

The determination of an acceptable controller has till fairly recently represented a huge task, two hundred person-years being a typical estimate of the time involved. The methods used could be characterised as modern control methods, but not so modern as to allow escape from the use of tedious trial-and-error methods, [1].

So called linear quadratic design methods and robust control H_∞ design methods constitute the two broad theoretical approaches which are now available for designing controllers in situations like this, and indeed available in the sense of there being commercial software (a useful standard for saying what is in practice achievable). This software is used by companies in sectors apart from aerospace, such as process control, power systems, mineral processing, steel mills. The software is based on theories

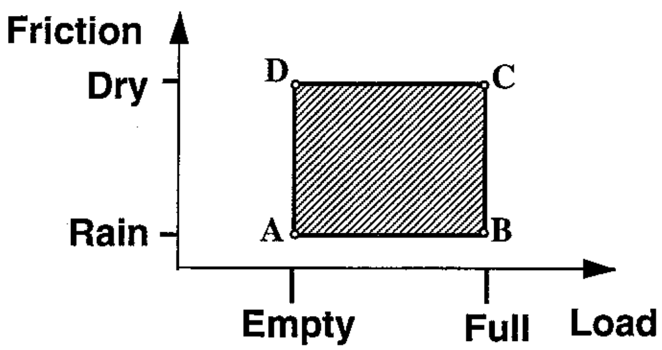

Figure 1: Parametric Robustness: Guided Bus Control

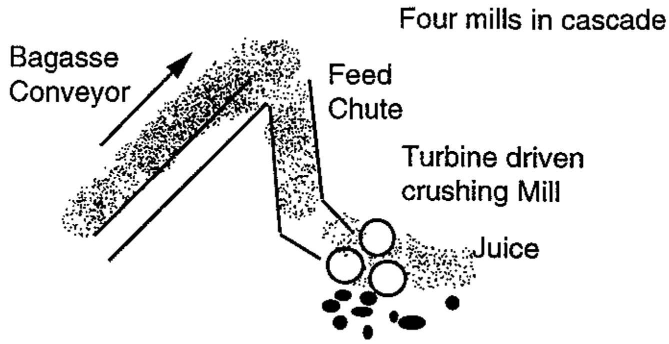

Figure 2: Sugar Mill Control Example

described in books such as [2], [3].

3 THE CHALLENGE OF ROBUSTNESS GIVEN PARAMETER VARIATIONS

In a great many situations, the physical system can be described by a mathematical model which contains parameters, and these parameters are associated with certain physical variables like mass, friction, and the like. It is well known for example that the equations of motion of an aircraft depend on the airspeed, the height, and the load. Another example is drawn from the area of vehicle control using buried wires. Consider Figure 1.

This is an idealisation from a problem that arose in a conceptual design study for a guided bus, [4] see page 58. The mass of the bus varies significantly according to the number of people in it. Also, the friction coefficient, important in considering the tyre-pavement contact, can vary significantly, depending on road conditions. Rain, oil and the like create a very different coefficient than that applying on a dry day. The problem is to design a single controller which will work satisfactorily for all values of the parameters. The parameters are not normally varying rapidly. What is in fact therefore needed is a single controller that will work well for a wide set of fixed plants, each plant in the set being defined by a particular choice of the parameters.

Despite the fact that this problem has been around sometime now, *no design methods are available in commercial software.* Of course there has been theoretical work nibbling at the edges, but a broad scale method is lacking. Even the question "Does there *exist* a single controller which will give satisfactory performance?" is in general not answerable.

One might conjecture that a controller which worked well at parameter settings, A, B, C, and D would work well everywhere. Unfortunately, this conjecture is also lacking general verification, although simpler versions of it can be verified. A recent textbook outlining the state-of-the-art is [5].

The general position therefore is one where control science has not yet provided effective tools for addressing important applications problems involving parameter robustness.

4 THE CHALLENGE OF ADAPTATION

There is a second conceptually different approach to handling problems with parametric uncertainty. The approach is known as adaptation, or adaptive control. The idea is that the controller, besides controlling the plant, contains signal processing software which uses measurements of the plant input and output to infer the values being assumed by the variable parameters in the plant. The controller parameters are then adjusted to suit the values assumed by the plant parameters. Of course, if the plant parameters undergo a step change, it may require some time for the controller to learn the new correct values of the plant parameters, and any noise contaminating the measurements has the potential to cause the controller to make an error in its estimate of the plant parameters. Never-the-less, very broadly speaking, it is possible to learn plant parameters over a time scale significantly longer than the time constants of the controlled process itself, [6], [7], [8].

One example of an application of adaptive control to a two input, two output system is provided by a sugar crushing mill. See Figure 2. The sugar cane is brought on a conveyor and dumped into a feed chute. The overall task of the crushing which occurs at the exit of the feed chute is to maximise the extraction of juice. The variables which are most important in determining this extraction are height of the material in the feed chute, and the chute aperture, or the feed rate to the crushers. The signals which can be adjusted are the turbine governor setting, and the turbine torque. The need for adaptation arises because those physical properties of the feedstock which determine how much juice is extracted vary according to the field from which the feedstock has been harvested.

The above situation is fairly simple to model, in that apart from the unknown parameters, a fairly accurate model of the process can be obtained, [9]. A complete contrast is

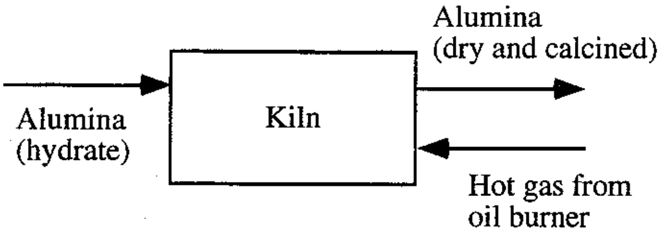

Figure 3: Alumina Calciner

offered by an alumina calciner, see Figure 3.

The variables which one is interested in controlling are:

- discharge alumina temperature (this governs product quality)
- temperature fluctuations (maintenance cost is driven up by fluctuations)
- energy consumption

The variables which can be readily controlled are

- alumina (hydrate, containing free and chemically bound moisture) feed rate
- oil rate feed rate
- air mass feed rate

In addition, measurements can be made of the cold end temperature of the kiln, and of gas composition. Clearly, a mathematical model based on the physics and chemistry of what is happening in the kiln would, even if it could be written down, be immensely complicated. It would not be surprising also to have nonlinear partial differential equations as part of the model.

The function of adaptation here is to learn parameters in a simple model of the process, a model which is readily accepted as being unable to fully describe the process, but a model which can be used for designing a controller for the process. Adaptive control has been achieved for this alumina calciner, despite the crudity of the model, its multiple input multiple output nature, and the presence of a time lag, something which control engineers know well greatly complicates the task of securing effective control. See [10].

Many mineral and metal processes involve heating, chemical reactions, and time delays caused by the slow transport of material through a kiln or furnace, or something of this nature. The control engineer is faced with the process in which there are typically many unknown process parameters, and great difficulties in instrumenting because of the harsh environment.

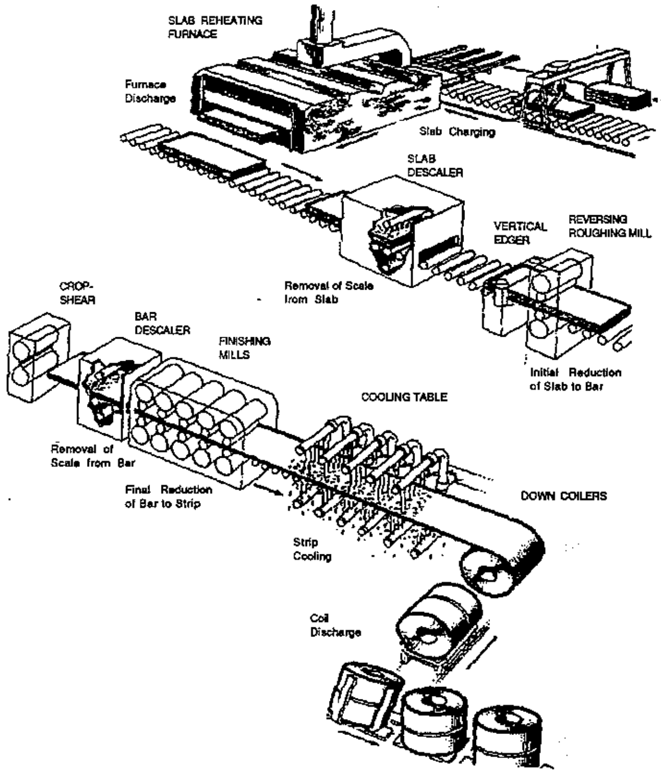

Figure 4: Reheating Furnace and hot strip mill

Another example of such a process is provided by a slab reheating furnace which feeds into a hot strip mill. The input to the furnace consists of slabs which are significantly colder than those of the output of the process, which feed into a hot strip mill. The cold slabs are introduced into a furnace which is some tens of metres long and spend approximately 2.5 hours in that furnace. It is desirable for the slabs to come out with a temperature that is close to the optimum temperature for the hot strip mill. Steel temperature cannot be measured very well, particularly inside the furnace, and the process is subject to many disturbances including production interruptions. Significant improvements on this process have been obtained by a combination of robustness and adaptation together with a clever use of Kalman Filtering to infer temperatures of exiting slabs from the behaviour of those slabs in the subsequent hot strip mill, [11].

5 THE CHALLENGE OF CONTROLLER IMPLEMENTATION

Mathematical models of physical processes usually involve differential equations, so the underlying independent variable is time, and it is assumed to vary continuously. Modern controller design methods, i.e. those based on commercial software, yield continuous time controllers when the model is continuous time, and also yield controllers which have similar complexity to that of the model. Thus controllers designed by modern methods will normally be continuous time, and often be of high complexity. On

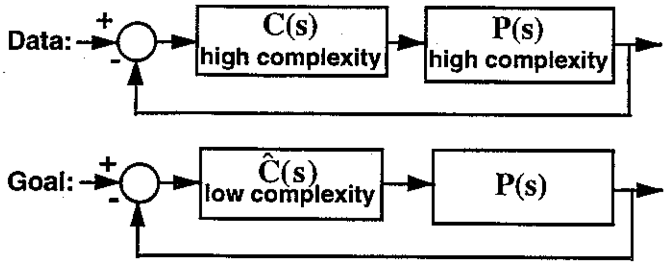

Figure 5: Controller Reduction

the other hand, if there is a requirement to implement a controller with a computer, it will have to be a discrete-time controller; there is frequently also a requirement to have a low complexity controller; and of course the implementation has to be numerically reliable. These observations generate the question: How can one replace a continuous-time high complexity controller resulting from a commercial software design package by a discrete-time low complexity controller that is known to be numerically reliable? This is the implementation challenge.

5.1 Controller Complexity Reduction

Consider Figure 5.

The task is to find a controller $\hat{C}(s)$, of low complexity, which causes the closed loop performance of the plant with $C(s)$ to be like the closed loop performance of the plant with $\hat{C}(s)$. (The symbol s denotes a Laplace transform variable; why it arises is not discussed here.) The term closed loop performance is a fuzzy phrase which connotes many particular aspects of performance. One useful tool for analysing the performance of a closed loop system is the *closed loop transfer function* (or transfer function matrix in the multi-input or multi-output case). In formal terms, this is the quantity $T(s) = \frac{P(s)C(s)}{1+P(s)C(s)}$.

Requiring the two closed loop performances to be similar amounts (more or less) to requiring the difference between the two closed loop transfer functions i.e. $\frac{P(s)C(s)}{1+P(s)C(s)} - \frac{P(s)\hat{C}(s)}{1+P(s)\hat{C}(s)}$ to be small. Of course in what sense this should be small is not altogether clear (transfer functions are most usually studied by examining their values for $s = j\omega, \omega$ real, and then smallness amounts to the above difference having a small value of its maximum magnitude along the $j\omega$ axis). No matter how the precise mathematical statement of the controller reduction objective is set up, what results is a problem that is most unfamiliar to anybody with standard mathematical training in optimization or approximation theory. Never-the-less, there now exist easy-to-use and effective solutions in commercial software for finding a reduced order controller, [12], [13]. They are not quite perfect, in the sense that the approximation error is not made optimally small. The sacrifice of optimality however allows quick and insightful algorithms

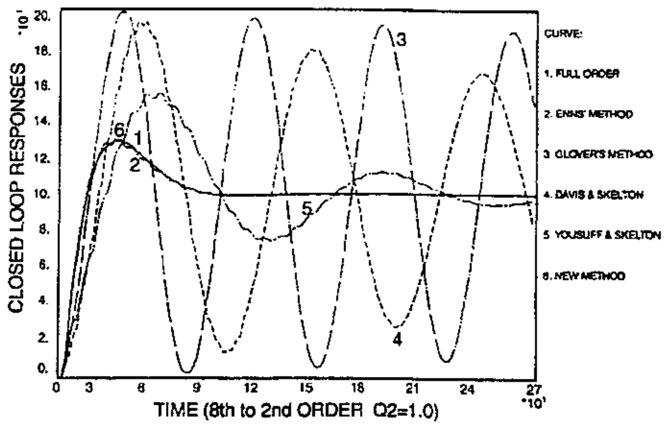

Figure 6: Step Response Comparison

to be used.

An example of the effectiveness of these methods is provided by the curves shown in Figure 6.

The plant (physically a disk-drive system) for which a controller is required is single input, single output, and described by an eighth order differential equation. The plant is open loop unstable, and non-minimum phase (a technical term in the control systems sense, which invariably implies a greater difficulty in finding a controller). Using modern design methods implemented in commercial software, a controller is found which achieves certain specified constraints on bandwidth and disturbance rejection. (The plant is sufficiently difficult to control that the determination of a controller by classical methods might be very difficult.) Now the controller found using modern methods has, not unsurprisingly, order or complexity just like the plant. This means that there are sixteen parameters in the controller. It is desired to reduce the number to five, corresponding to a second order controller, and this in fact can be done. The figure illustrates the result of applying a number of controller reduction methods and then simulating the closed loop step response resulting with each controller. Methods 3,4, and 5 are older, probably now outmoded, schemes for controller complexity reduction. Methods 2 and 6, which give very close adherence of the closed loop response with the reduced order controller with that of the original system, are based on newer ideas, particularly the idea that the key thing which the reduced order controller must do is ensure that the closed loop responses mimic one another, rather than say the open loop responses.

A much more striking example is provided in [13], where the design of a low order multi-variable controller for pitch control of an aircraft is described. A high order controller fulfilling the specifications is obtained by LQG design, and the order is about fifty. Trial and error methods of an enormously time consuming nature had been used to obtain a tenth order controller fulfilling the specifications. The systematic controller reduction methods as described in [12] were able to obtain controllers of dimensions 4 and

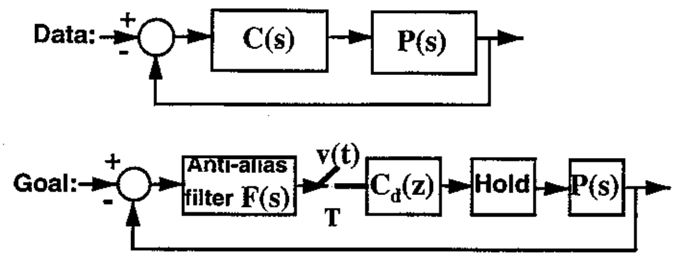

Figure 7: Controller Discretization

5 which were effective.

5.2 Controller Discretization

Consider Figure 7.

The idea is that $C(s)$ is a continuous time controller that has been determined by whatever method. Because implementation of a continuous controller would require analog elements or analog circuitry, and in comparison with digital devices and circuits this is difficult to implement unless the controller is very simple, it is often desired to implement a discrete time controller. The idea of a discrete time controller is that it takes a sequence of sampled values of a signal, spaced apart in time by some chosen duration T. It produces a sequence of outputs, again spaced apart by T, and this sequence of output values is passed through a digital to analog hold circuit, so that the input to the plant is a piecewise constant signal, with the various values of the input signal changing at intervals of T seconds and following the output of the discrete time controller.

The discrete time controller itself implements a difference equation, for example $w(kT) = 0.5w[(k-1)T]+v[(k-1)T]$ (as compared with the continuous time controller which effectively implements a differential equation).

The core question is: "How knowing $C(s)$ can one find $C_d(z)$?" This question has been treated in many textbooks, eg. [14], [15], and you can take your pick from about 12 formulas. Unfortunately, there are examples in which none of them work. This is in part because the wrong question has been posed. The correct question is: "How should one choose $C_d(z)$ to make the two closed loops as similar as possible?" The naturalness of this question is obvious once it has been asked. Since $P(s)$ is an inherent part of the two closed loops, it then becomes likely, if not virtually certain, that the best choice of $C_d(z)$ can not depend on $C(s)$ alone but must also involve $P(s)$. This is a fundamental change of view from that which has applied in generating the textbook answers, and also constitutes the key reason behind the fact that the textbook answers are sometimes ineffective.

A very crude mathematical statement of the objective is: choose $C_d(z)$ to make $PC(1+PC)^{-1} - PHC_dSF(1+PHC_dSF)^{-1}$ small. [Here S is the sampling operation, and F a so-called anti-alias filter, introduced for quite technical reasons]

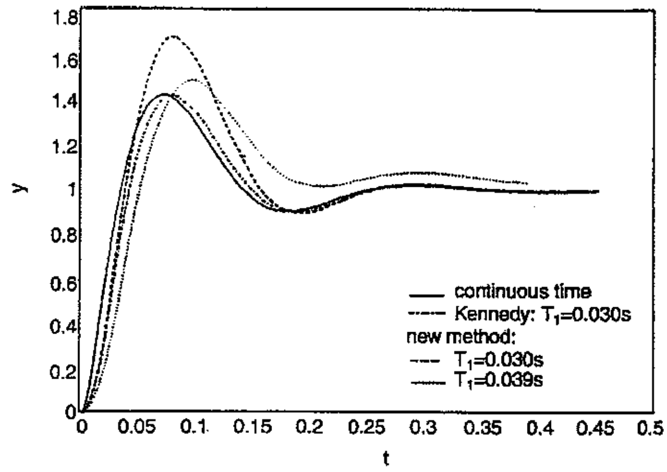

Figure 8: Step Response (Katz Example)

We will spare the reader any mathematical details, and simply note that this problem has recently been solved, [16], [17], [18].

A standard textbook published a few years ago compares a number of the then available discretization methods, none of which generated $C_d(z)$ taking into account the plant, [19]. The methods were applied for plant and controller transfer functions

$$P(s) = \frac{863.3}{s^2} \quad C(s) = \frac{2940(s+29.4)}{(s+294)^2}$$

.

A discretization time of $T = 0.030$ was chosen. Out of eight standard methods, only one resulted in a discrete time controller for which the closed loop remained stable, but the closed loop performance was unacceptable, and in particular, there was an enormous discrepancy between the response to a unit step with the original continuous time controller, and with the stabilizing discrete time controller. Figure 8 shows the results of two newer methods for discretizing a controller. One method developed by Kennedy, was used to design a controller for the Australia Telescope when the standard textbook methods failed, [16]. That method is restricted to plants with a single input single output. The other method is described in references [17], [18].

5.3 Discrete Controller Implementation

A discrete controller is defined in terms of its transfer function, but the actual implementation of the controller will involve arithmetic operations on finite word length quantities. This means that round-off of signals occurs after every arithmetic operation (quantization noise), and coefficients used for multiplying are necessarily quantities which are implemented with a finite word length, and as

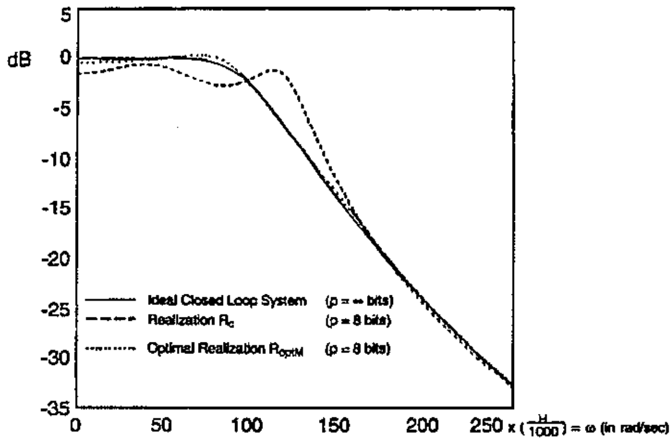

Figure 9: Frequency Response Comparison

a result, there may be an approximation involved at this point.

This could all perhaps be coped with, except for the fact that there is not a unique way to implement a prescribed discrete time controller transfer function, but an infinity of ways, all with different coefficients. Obviously then, the way the controller is implemented can be very important in terms of the effect on the overall closed loop performance of quantization errors and coefficient representation errors. How one might best implement the controller to minimise the deleterious effects is something that has only very recently been determined, [20], despite the problem being flagged many years ago.

Figure 9 shows a frequency response using a discrete time controller with an infinite number of bits, and two discrete time controllers implementing the same nominal transfer function, one optimally chosen and one chosen in a simple but not particularly thoughtful way. The latter shows that very substantial variations from the ideal frequency response can result.

6 FUTURE CHALLENGES

In the preceding material, one clear mismatch between existing theoretical capabilities and applications demands has been identified already, and this is design for parametric robustness.

Let us however note one other very significant future challenge. There is a sparsity of systematic nonlinear design procedures. Despite decades of work on nonlinear systems, very, very few readily usable general design methods have been forthcoming.

More generally, this paper has left far more important applications problems unaddressed than addressed. We have really said very little about software, and concepts of real time control. Similarly, the whole vast field of hybrid systems and discrete event systems has been left untouched. This will certainly be a very fertile area for practically motivated, easy-to-use theoretical developments.

References

[1] Y. Liu, B.D.O. Anderson and Ly Uy-Loi, "Coprime Factorization controller reductions with bezout identity induced frequency weighting", *Automatica*, vol. 26, pp. 233-249, 1990

[2] B.D.O. Anderson and J.B. Moore, *Optimal Control: Linear Quadratic Methods*, Prentice Hall, Englewood Cliffs, 1989

[3] M. Green and D.J.N. Limebeer, *Linear Robust Control*, Prentice Hall, Englewood Cliffs, 1994

[4] J. Ackermann, *Abtastregelung*, Springer-Verlag, Berlin Heidelberg New York, 1983, pp. 58

[5] B.P. Barmish, *New Tools for Robustness of Linear Systems*, Macmillan, New York, 1994

[6] G.C. Goodwin and K.S. Sin, *Adaptive Filtering, Prediction and Control*, Prentice Hall, Englewood Cliffs, 1984

[7] K.J. Åström and B. Wittenmark, *Adaptive Control*, Addison Wesley, Reading, 1989

[8] B.D.O. Anderson, R.R. Bitmead, C.R. Johnson, P.V. Kokotovic, R.L. Kosut, I.M.Y. Mareels, L. Praly and B.D. Riedle, *Stability of Adaptive Systems: Passivity and Averaging Analysis*, MIT Press, Cambridge, 1986

[9] A.G. Partanen and R.R. Bitmead, "Excitation Versus Control Issues in Closed Loop Identification of Plant Models for a Sugar Cane Crushing Mill", *Proceedings 12th World IFAC Congress*, vol. 9, pp. 49-52, Sydney, 1993

[10] P.M. Mills, P.L. Lee and P. McIntosh, "A practical Study of Adaptive Control of an Aluminer Calciner", *Automatica*, vol. 27, pp. 441-448

[11] A.J. Connolly, R.R. Bitmead, S. Crisafulli, M. Green, P. Stone, "Discharge Slab Temperature Estimation", *MMM95* to be presented

[12] Integrated Systems Inc., MATRIX$_x$ Model Reduction Module, 1991

[13] B.D.O. Anderson and Y. Liu, "Controller Reduction: Concepts and Approaches", *IEEE Trans. Automatic Control*, vol. 34, pp. 802-812, 1989

[14] K.J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, Prentice Hall, Englewood Cliffs, 1990

[15] G.F. Franklin, J.D. Powell and M.L. Workman, *Digital Control of Dynamic Systems,* Addison Wesley, Reading, 1990

[16] R. Kennedy and R. Evans, "Digital Redesign of a Continuous Controller Based on Closed Loop Performance", *Proc. IEEE Conference Decision and Control*, pp. 1898-1901, 1990

[17] J.P. Keller and B.D.O. Anderson, "H_∞ -optimal Controller Discretization", *International Journal of Robust and Nonlinear Control*, vol. 1, pp. 125-137, 1991

[18] J.P. Keller and B.D.O. Anderson, "A New Approach to the Discretization of Continuous-Time Controllers", *IEEE Transactions on Automatic Control*, vol. AC37, pp. 214-223, 1992

[19] P. Katz, *Digital Control using Microprocessors*, Prentice Hall, Englewood Cliffs, 1981

[20] A.G. Madievski, B.D.O. Anderson and M. Gevers, "Optimum Realizations of Sampled-Data Controllers for FWL Sensitivity Minimization", *Automatica*, to appear