Plenary Lecture

Digital Control: What do we want and what can we now achieve?

by

Brian D. O. Anderson

International Journal of The Society
of Materials Engineering
for Resources
Vol. 3 No. 1 1995

Physical entity	Excitation	How excitation is achieved
Electric heater	Electric power	Turning switch on or off depending whether you are cold or not
Car engine	Fuel frow	Linking throttle to a foot control (accelerator) and then operating accelerator
Aircraft	Hydraulic power to control surface	Linking control surfaces via actuators to control stick

These are all examples of open-loop control. That is, the system (as opposed to the human operating it) does *not* sense what the response is and therefore cannot take corrective action by way of adjusting the excitation.

Closed-loop control is also possible.

In closed-loop control, the system is provided with information as to what its output or response should be (the 'desired respones') and then it adjusts the excitation so that the actual response approaches or even equals the desired response; very frequently the excitation is determined from the difference between the desired and actual response.

Closed-loop control is summed up in Figure 1.

The following table gives more details for three examples.

Physical entity	Excitation	Closed-loop control approach
Electric heater	Electric power	Thermostat compares desired and actual temperature and switches power accordingly
Car engine	Fuel flow	Cruise control compares desired and actual speed, and adjusts fuel flow and even braking
Aircraft	Hydraulic power to control surface	Automatic pilot compares desired flight trajectory (level, landing eta) with actual, and adjusts control surfaces.

In a great many control design problems, one or both of two key problems have to be addressed: *securing dynamic stability, and securing zero steady state error.* Securing dynamic stability means that the corrective action taken by the controller should not over-compensate and drive the system into oscillation or some catastrophe. Securing steady state error given constant disturbance

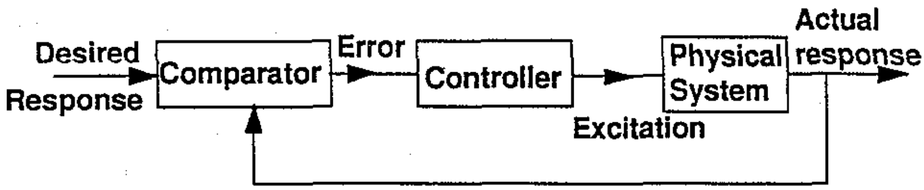

Figure 1 : Closed-loop control

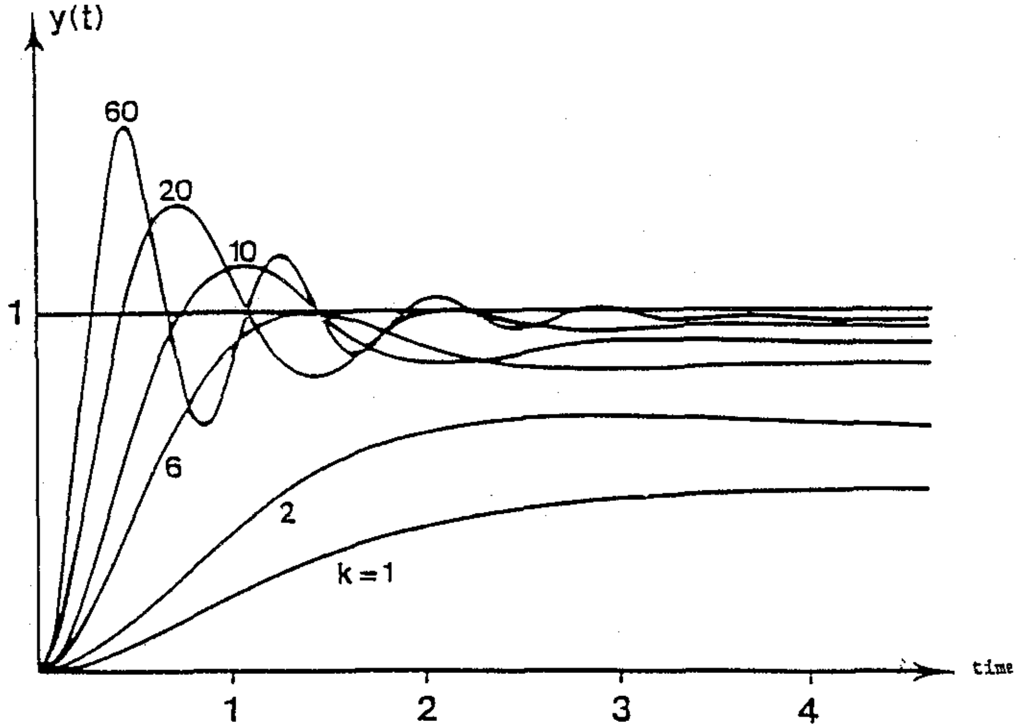

Figure 2 : Step-Response

means that if the desired response is constant, the actual response should (after some transient) *exactly match the desired response even with constant disturbance,* such as constant heat loss through windows (room heating problem) or constant head-wind (cruise control or aircraft control).

It is very common to conider the response of a closed-loop system to a step change (thermostat dial is adjusted, for example). This is the so-called *step response.* It can be sluggish or fast, exhibit great overshoot before becoming correct or almost no overshoot, exhibit oscillatory behaviour before settlimg, or just a fluctuation or two. It can achieve zero steady stste error, or non-zero steady state error.

Figure 2 illustrates a collection of possible step-responses. The curves are indexed by values of

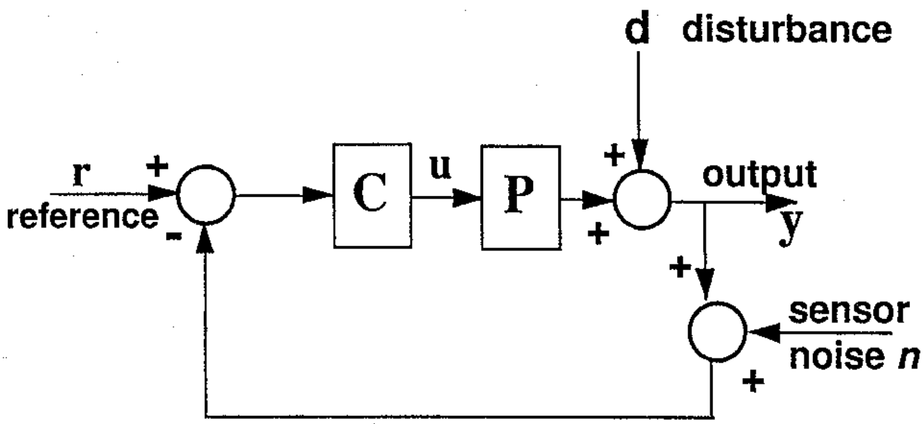

Figure 3 : High loop gain

a parameter used in specifying the controller, and illustrate a frequently occurring situation: there is a *trade-off* between, on the one hand, small steady state error and, on the other hand securing small overshoot. Thus $\kappa = 60$ gives the smallest steady state error, but the largest overshoot. Zero steady state error is achieved only with infinite overshoot.

Much of classical control is concerned with understanding the above issues. Of course, there are scientific underpinnings. As an example let us comment on the use of a *high loop gain.* Electronic amplifier designers were probably the first engineers to realize that if a plant (=vacuum tube) exhibited significant gain variations, the effect of these variations could be greatly ameliorated by including the plant in a high gain loop. Actually, a number of other observations about high gain were also quickly made. Consider Figure 3.

The plant is P and the controller is C. The aim is to have the output track the refernce input r, even in the presence of disturbances d and noise in the feed back signal n.

Evidently, $y = Pu + d$, $u = C\ (r - y - n)$. Putting these together yields (at least fomally)

$$y = \frac{PC}{1+PC} r - \frac{PC}{1+PC} n + \frac{1}{1+PC} d$$

Suppose C is very large. Then P could vary by 30% but for fixed reference r, the output y would change but a little. Also d will affect y just a little. And the output y will track the reference input r closely, at least if the sensor noise n is small. Thus high loop gain:

- suppresses the effect of plant gain variation
- reduces the effect of additive disturbances at the output
- promotes good tracking by the output of a reference input

High loop gain has disadvantages also. If C is small, the sensor noise n affects the output y less than if C is high. And if C is high, the plant excitation u may be so big as to over-drive the physical plant P (so that it behaves nonlinealy, explodes or otherwise misbehaves). These points are obvious from the figure and the equations describing it. A further point, not obvious but none the less true, is that high gain can produce oscillatory type of behaviour or even permanent oscillations

(instability). So a high gain:
• can induce instability/oscillatory behaviour
• worsens sensor noise problems
• can cause the plant to be over-driven

The two most important conceptual ideas of classical control may well be high gain and its consequences, and a variety of tools for handling stability questions. Many standard textbooks discuss thess ideas [1], [2], [3].

Nearly all systematic design methods for control systems have been for plants with linear models, and virtually all design is a matter of trade-offs (as reflected in the discussion above of step response and high gain). Classical control essentially provided methods for designing only simple controllers, which are adequate for simple plants, but may not be adequate for complicated plants. Modern control methods yield complicated controllers for complicated plants, and they have to be realized with a computer.

2. Topical Challenges of Control Engineering

2. 1 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 represnted a hugetask, 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, [4].

Two broad theoretical approaches are now available for designing controllers in situations like this, and 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 described in books such as [5], [6].

2. 2 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 4.

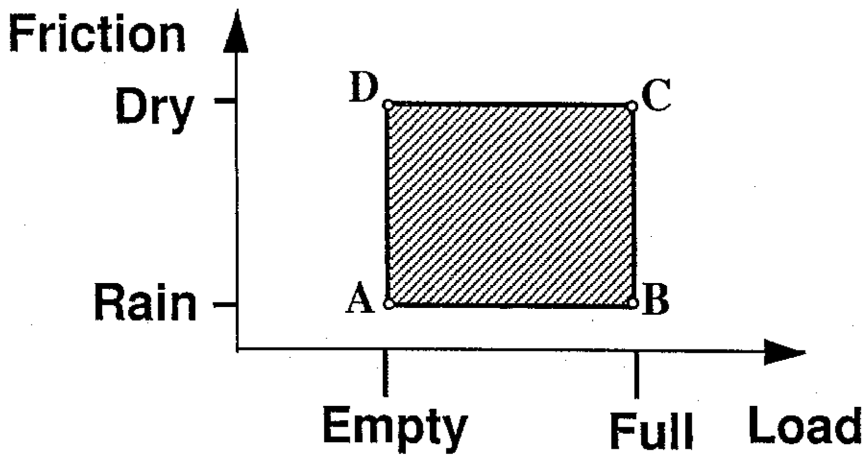

Figure 4 : Parametric Robustness : Guided bus Control

This is an idealisation from a problem that arose in a conceptual design study for a guided bus, [7] 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. 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 [8].

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

2. 3 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 abjusted 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, [9], [10], [11].

One example of an application of adaptive control to a two input, two output system is provided by a sugar crushing mill. See Figure 5. 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 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, [12]. A complete contrast is offered by an alumina calciner, see Figure 6.

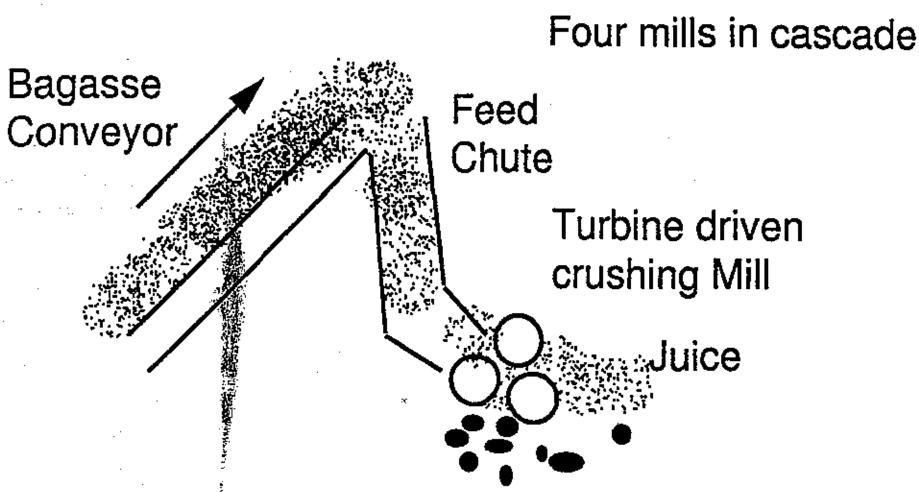

Figure 5 : Sugar Mill Control Example

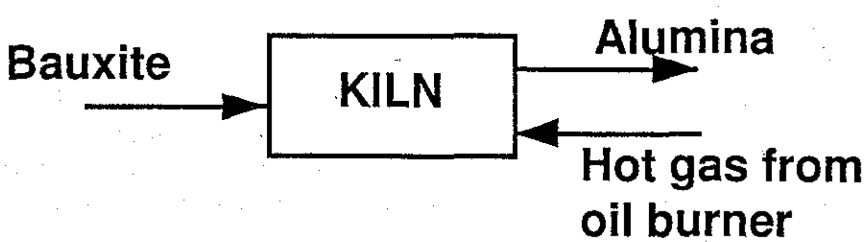

Figure 6 : Alumina Calciner

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

- bauxite 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 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 crusity 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 [13].

3. 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 the other hand, if there is a requirement to implement a controller with a computer, it will have to be a discrete type 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 desigh package by a discrete-time low complexity controller that is known to be numerically reliable? This is the implementation challenge.

3. 1 Controller Complexity Reduction

Consider Figure 7.

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 $T(s) = \frac{P(s)C(s)}{1+P(s)C(s)}$.

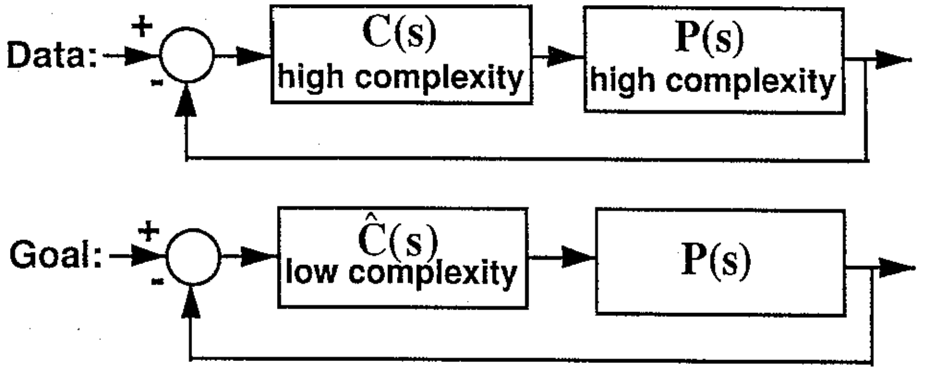

Figure 7 : Controller Reduction

Requiring the two closed loop performances to be similar amounts (more or less) o 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 $s = j\omega$, ω 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, [14], [15]. 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 to be used.
An example of the effectiveness of these methods is provided by the curves shown in Figure 8.

The plant 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 nonminimum 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 abherence 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,

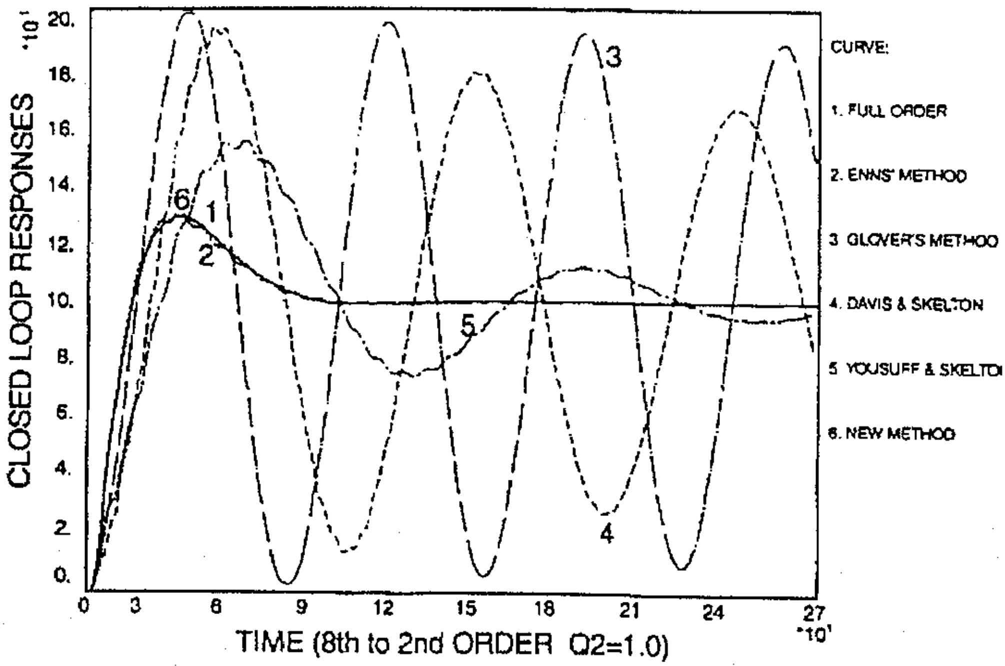

Figure 8 : Step Response Comparison

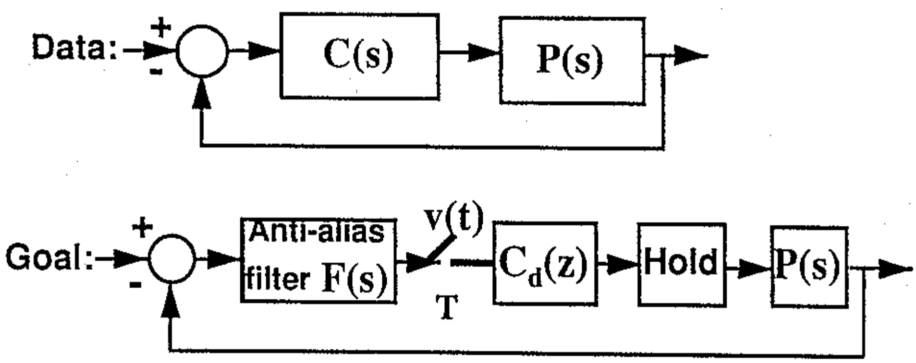

Figure 9 : Controller Discretization

rather than say the open loop responses.

3. 2 Controller Discretization

Consider Figure 9.

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 precewise 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 $\omega(\kappa T) = 0.5\omega[(\kappa\text{-}1)T] + \upsilon[(\kappa\text{-}1)T]$ (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. [16], [17], 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 texbook 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]

We will spare the reader any mathematical details, and simply note thaht this problem has recently been solved, [18], [19], [20].

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, [21]. The methods were applied for plant and controller transfer functions

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

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 10 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, [18]. That method is restricted to plants with a single input single output. The other method is described in references [19], [20].

3. 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

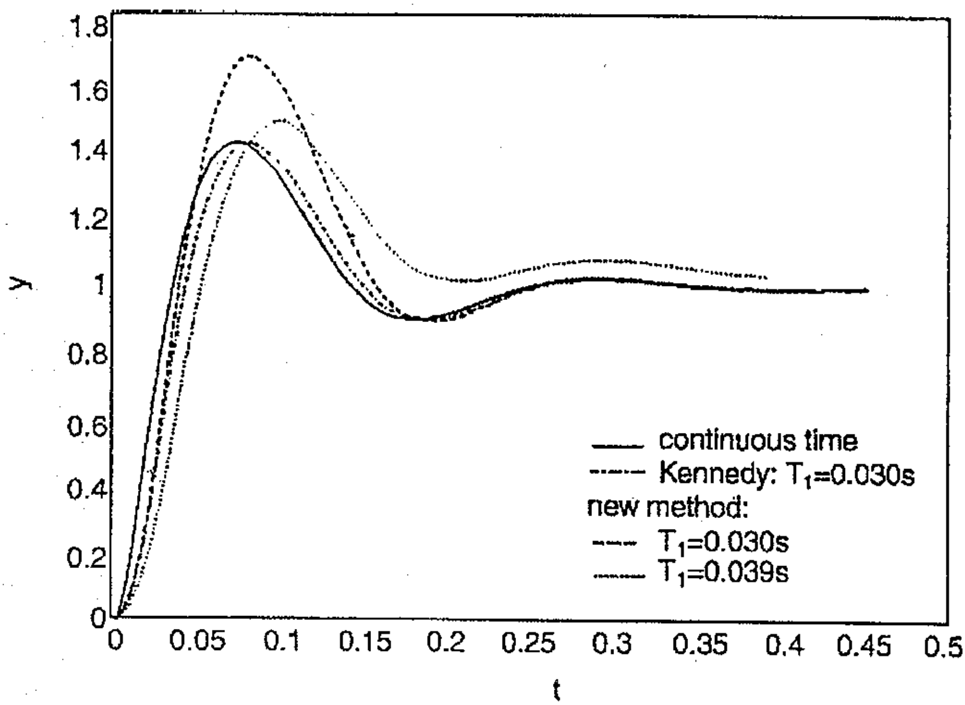

Figure 10 : Step Response (Katz Example)

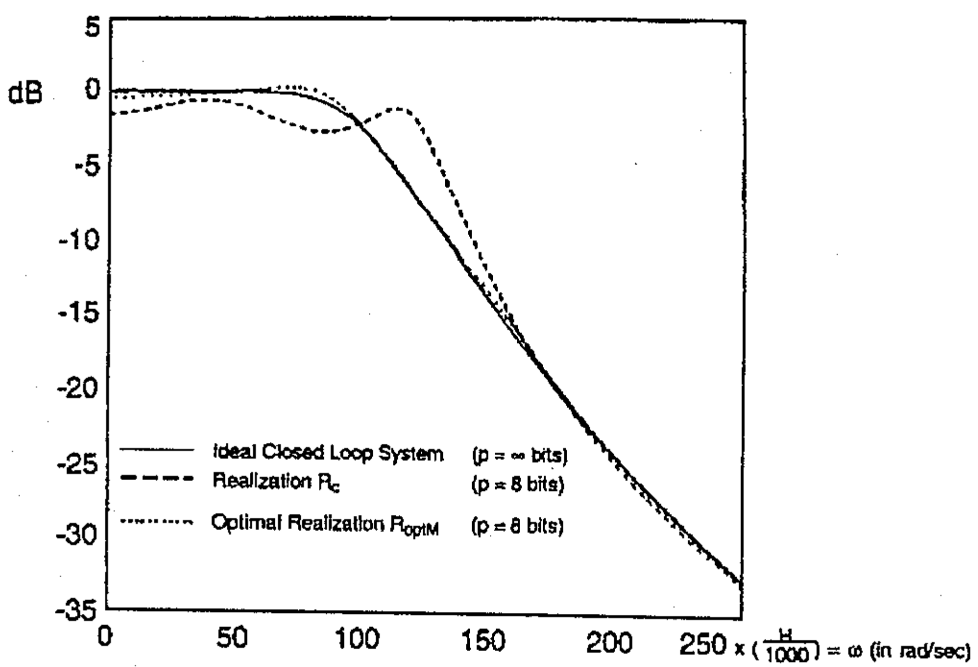

Figure 11 : Frequency Response Comparison

length, and as 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 cofficients. 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, [22], despite the problem being flagged many years ago.

Figure 11 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 praticularly thoughtful way. The latter shows that very substantial variations from the ideal frequency response can result.

4. 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.

References

[1] B.C. Kuo, *Automatic Control Systems*, Prentice Hall, Englewood Cliffs, 1975

[2] T.E. Fortmann and K.L. Hitz, *An Introduction to Linear Control Systems*, Marcel Dekker, New York, 1977

[3] R.N. Clark, *Introduction to Automatic Control Systems*, John Wiley & Sons, New York, 1964

[4] 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

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

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

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

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

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

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

[11] 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

[12] 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

[13] P.M. Mills, P.L. Lee and P. McIntosh, "A practical Study of Adaptive Control of an Aluminer