PRBS cross-correlation measurements by hybrid computational techniques

P. A. Jackson

Lanchester College of Technology, Coventry

This paper discusses briefly the identification of system dynamics by cross-correlation. This technique depends upon the well-known result that, if the system-input signal has an impulsive auto-correlation function, the cross-correlation of the input with the output gives the system impulse response.

The method described uses a pseudo-random binary sequence (p.r.b.s.) as the system input with the computation being performed on a general purpose semi-hybrid machine. This machine comprises an analogue computer together with a comprehensive sequential-mode programme control unit. The results are automatically printed out or plotted; typical results are shown.

(Received December 1969)

Introduction

The traditional methods of determining experimentally the dynamic characteristics of a system employ either transient or sinusoidal signals as forcing functions. Transient response tests are usually carried out to determine either the impulse response or the step response of the system. In either case the main practical difficulty is that the magnitude of the impulse or the step must be considerably larger than the system noise, for confident measurements to be made. This means that considerable disturbance of the system must occur, with the attendant possibility of non-linear operation due to saturation effects.

Sinusoidal response measurements are made in the steady state over a range of frequencies and consequently the process tends to be rather lengthy. Although in this case the measurement problems are somewhat easier, the sinusoidal excitation must again be much larger than the system noise for confident estimates. The results obtained are usually expressed graphically in terms of Nyquist, Bode, or other types of plot, from which an approximate system transfer function can be determined, in terms of a series of low order transfer functions.

In many practical situations the use of a large amplitude test signal is often not desirable—or may not even be possible. In such cases, therefore, the traditional methods do not offer a very satisfactory solution and a more useful approach is by correlation techniques.

The theory of correlation techniques is well documented (Douce, 1963). In brief, the essence of this method depends upon the principle that the cross-correlation of the input ($x(t)$) with the output ($y(t)$) of a system, yields the system's impulse response, provided that $x(t)$ has an impulsive auto-correlation function. This is shown in (1) where $g(\tau)$ is the system impulse response and $Rxy(\tau)$ is the cross-correlation at time $t = \tau$:

$$g(\tau) = R_{xy}(\tau) = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t).y(t+\tau).dt.$$

$$= \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t-\tau).y(t).dt. \qquad (1)$$

The best example of a completely random signal is white noise having ideally a flat power spectrum of infinite bandwidth. The auto-correlation function of such a signal, $R_{xx}(\tau)$, is simply an impulse function at $\tau = 0$. Thus, although a white noise test signal satisfies the theoretical requirements, its application in practice is fraught with difficulties and does not give very good results. These problems are discussed in more detail in the following section.

Basic requirements of a cross-correlator

The essential constituents of a cross-correlator can be deduced from (1). The system input $x(t)$ is required to be multiplied by a time advanced version of the output $y(t + \tau)$, or alternatively, $y(t)$ is multiplied by a time delayed version of the input $x(t - \tau)$; the result of this multiplication must then be time averaged over a period

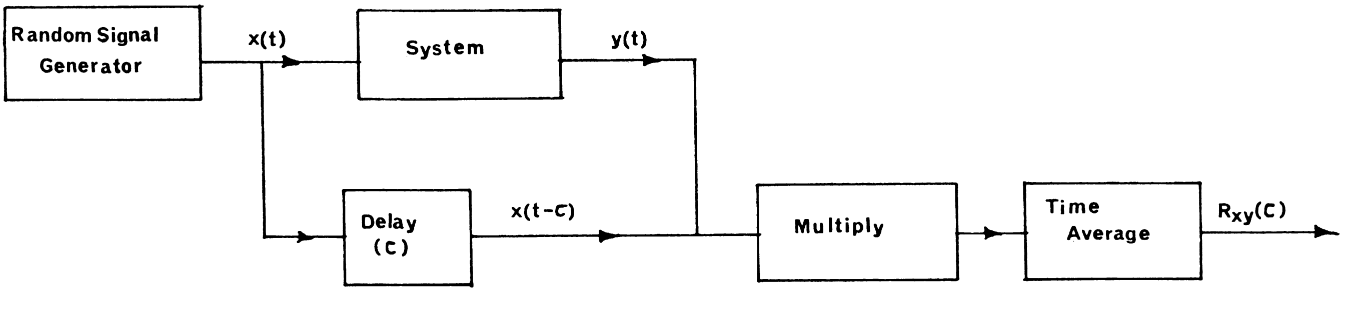


Fig. 1. Basic cross-correlator

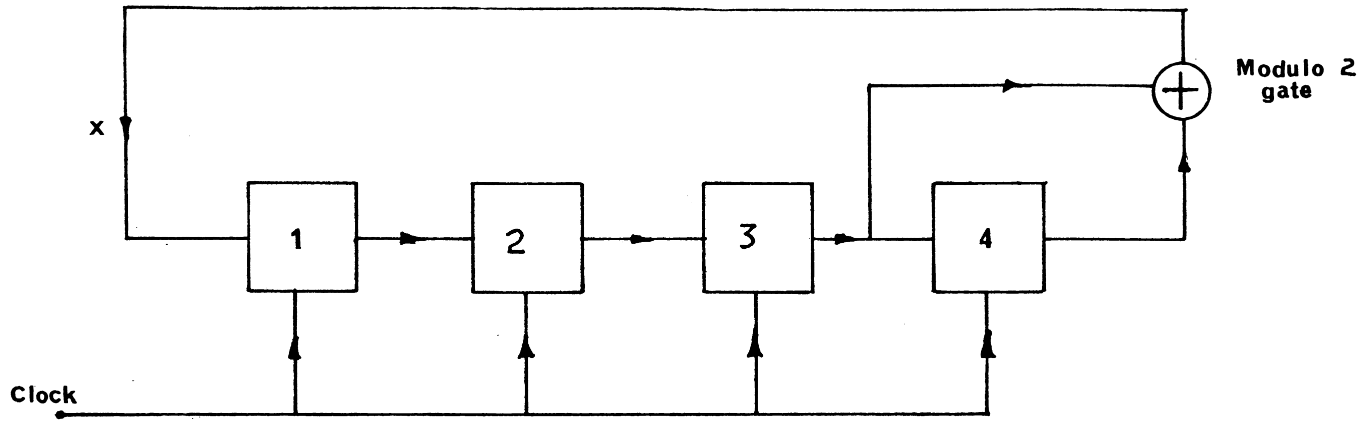


Fig. 2. Four stage PRBS generator of period 15

which ideally should be infinite. **Fig. 1** depicts the basic cross-correlator.

The use of white noise as a test signal in these schemes invokes many practical difficulties, the principal problem being the very long averaging times required to reduce statistical errors to acceptable levels. Other problems involve the generation of a flat power spectrum at low frequencies and the awkwardness of producing delayed versions of the signal. These considerations have led to the use of the periodic pseudo-random binary sequence (p.r.b.s.) as a suitable test signal.

Pseudo-random binary sequences

These sequences may be conveniently generated by means of linear feedback shift registers. Although the description of such registers has been well described (Huffman, 1961) a brief description will prove useful. Consider the four-stage generator shown in Fig. 2. Upon application of a clock pulse the information contained in the register is shifted along by one stage to the right, simultaneously the output (x) of the modulo two gate is transferred to stage one of the register. If the feedback connections satisfy the requirements for the production of a maximum length binary sequence (Petersen, 1961), the sequence will be repetitive with a period of 15 bits (**Fig. 3**); the all zero condition of the register being inadmissible. In general, for a register with N stages, the p.r.b.s. period would be $(2^N - 1)$ bits.

If the binary levels of the p.r.b.s. are assigned convenient positive and negative levels of equal amplitude ($+\alpha$ and $-\alpha$), then the auto-correlation function is of the form shown in **Fig. 4**. As expected the auto-correlation function is repetitive and can be approximated to the 'ideal' impulse function by making N large and Δt small.

Because of the periodicity of the auto-correlation function, time averaging in the cross-correlator is normally only required over one period, and the binary nature of the signal makes the processes of multiplication and delay generation relatively easy.

In practice, the sequence length ($N\Delta t$) must be considerably greater than the significant part of the impulse response and at least several times greater than the largest system time constant. On the other hand if $R_{xy}(\tau)$ is to be a good approximation to the impulse response of the system Δt should be small. However, system noise considerations impose practical limits on the smallness of Δt, and these considerations are discussed in detail by Finnie and Roberts (1965).

Computer program

The analogue flow diagram, together with the digital scheme (in simplified block diagram form) are shown in **Fig. 5**. The p.r.b.s. test signal and its delayed version are generated by two separate registers at voltage levels of 0 V. and 10 V. approximately. These levels are converted to the required levels $+\alpha$ V. and $-\alpha$V. by operational amplifiers A1 and B1, which employ a non-linear feedback arrangement. The output from the system $y(t)$ and the delayed version of the test signal $x(t - \tau)$ are then multiplied together and integrated over a period or an integral number of periods of the sequence.

The digital part of the programme is basically required to exercise mode control and to control the clock pulses to the two registers. This involves advancing the state of register 2 by one clock pulse, or possibly more, after a computation of a particular value of $R_{xy}(\tau)$. Other ancillary requirements are the automatic printing and plotting of results and the resetting of the programme. In order to achieve these aims, relatively sophisticated digital mode control facilities are required. The mode control system used is of the sequential type linked directly to an analogue machine and has been described in detail by Bellamy and Hulton (1968).

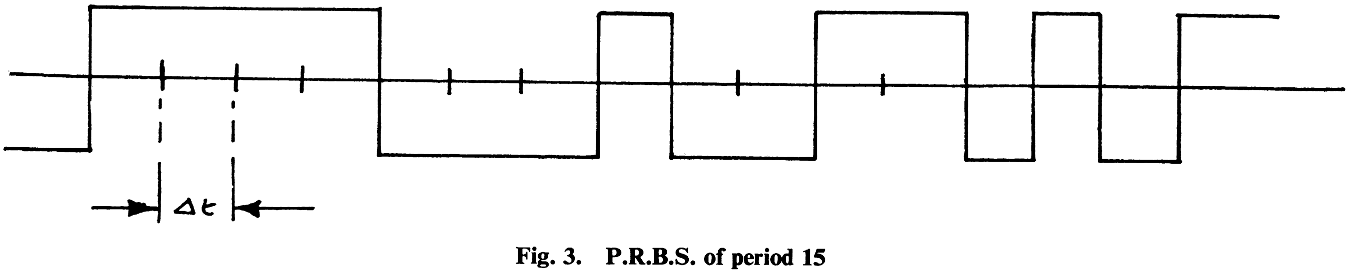


Fig. 3. P.R.B.S. of period 15

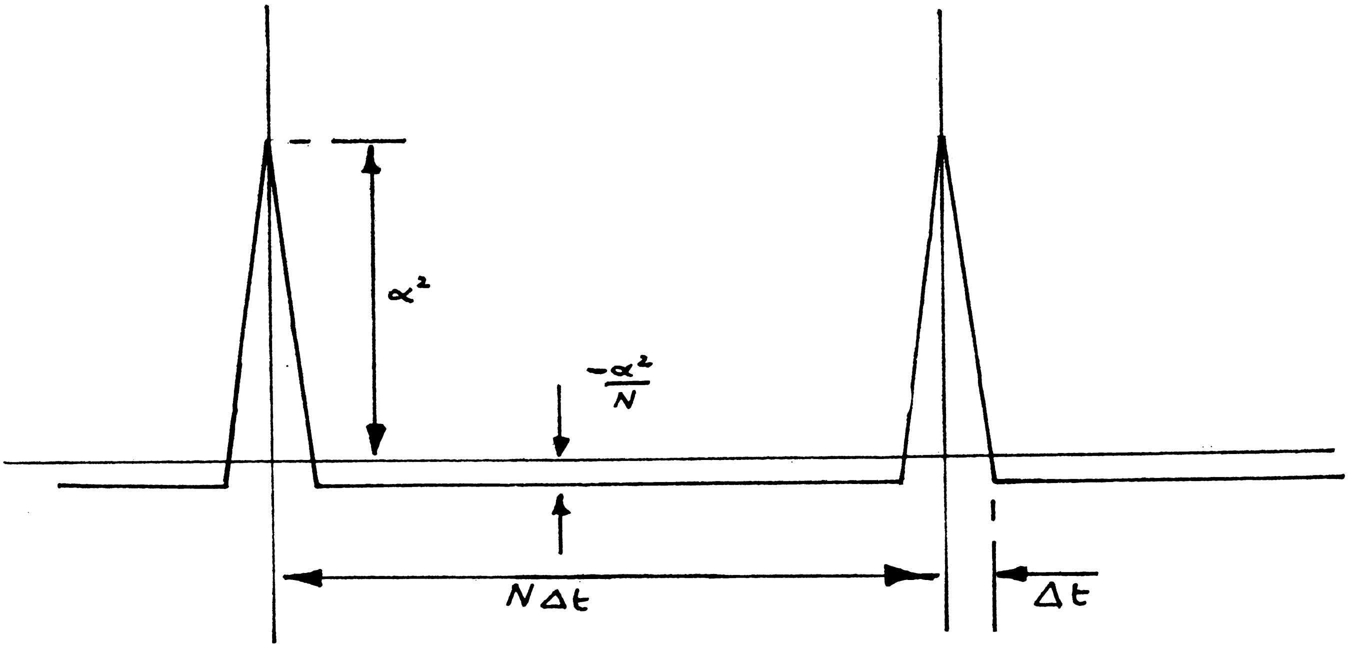


Fig. 4. Autocorrelation function for P.R.B.S.

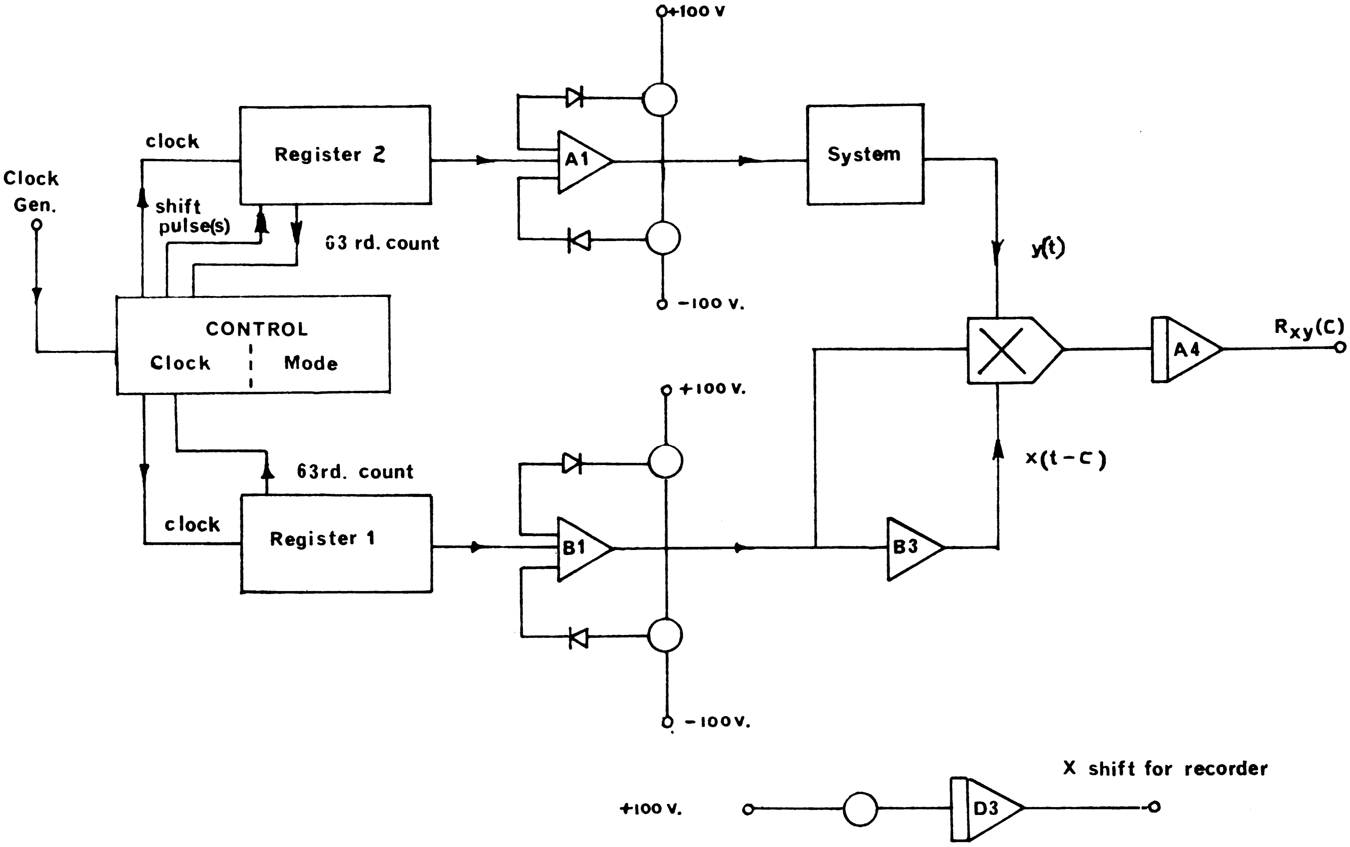
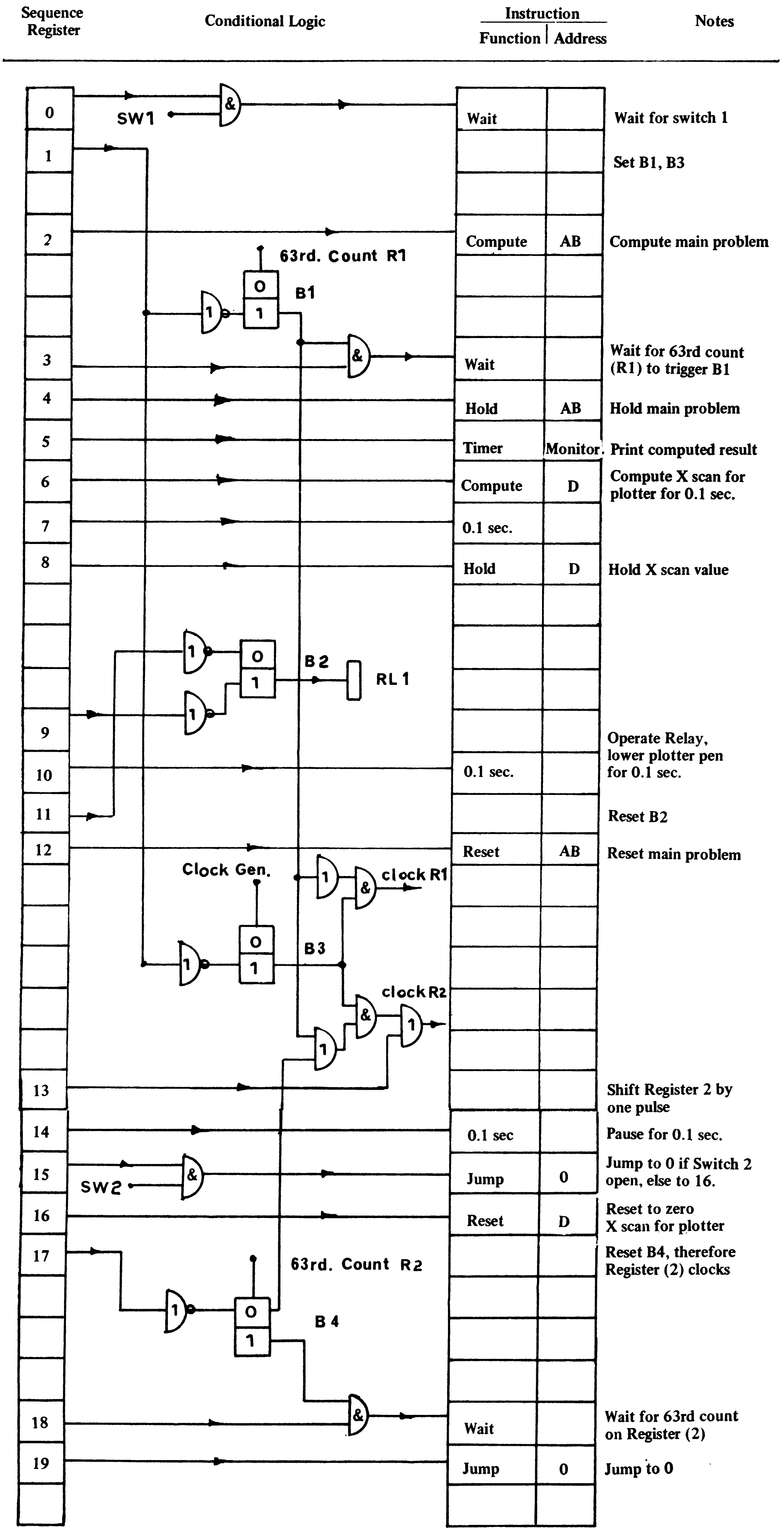


Fig. 5. Analogue flow diagram and digital scheme



Sequence Register	Instruction Function	Instruction Address	Notes
0	Wait		Wait for switch 1
1			Set B1, B3
2	Compute	AB	Compute main problem
3	Wait		Wait for 63rd count (R1) to trigger B1
4	Hold	AB	Hold main problem
5	Timer	Monitor.	Print computed result
6	Compute	D	Compute X scan for plotter for 0.1 sec.
7	0.1 sec.		
8	Hold	D	Hold X scan value
9			Operate Relay, lower plotter pen for 0.1 sec.
10	0.1 sec.		
11			Reset B2
12	Reset	AB	Reset main problem
13			Shift Register 2 by one pulse
14	0.1 sec		Pause for 0.1 sec.
15	Jump	0	Jump to 0 if Switch 2 open, else to 16.
16	Reset	D	Reset to zero X scan for plotter
17			Reset B4, therefore Register (2) clocks
18	Wait		Wait for 63rd count on Register (2)
19	Jump	0	Jump to 0

Fig. 6. Sequential-mode control program

A typical sequential-mode control programme is shown in detail in **Fig. 6.** In this particular programme a six-stage register generating a sequence of period 63 bits is used. The entire programme is controlled by the switch SW1, which starts the computation, and also permits it to be stopped at any stage if required. The closing of switch SW1 causes the AND gate output to fall to zero and the 'Wait' instruction to be released. Sequence (1) sets the initial states of bistables B1 and B3 to the logical (1, 0) condition, and consequently 'opens' the two AND gates connected to B3, thus enabling the system clock pulses to pass to the two shift registers (R1 and R2). Simultaneously Sequence (2) causes computation to commence. Computation is arranged by the resetting procedure (described later) to commence with all stages of Register (1) in the logic '1' state. The computation then proceeds until this state of Register (1) is again reached, when a simple gating procedure produces a control pulse (63rd count) to reset B1. The 'Wait' instruction being held by Sequence (3) is then released, causing the computation to pass to the 'Hold' condition (Sequence (4)). At Sequence (5) the estimate of $R_{xy}(\tau)$ obtained is printed out automatically, while Sequences (6), (7) and (8) cause the integration of a fixed voltage for 0·1 sec. by operational amplifier D1, to provide the X scan for an X-Y plotter. The plotter pen is then lowered (Sequences (9), (10) and (11)) to plot the computed estimate of $R_{xy}(\tau)$. The problem is then reset (Sequence (12)) and an extra shift pulse transmitted to Register (2) (Sequence (13)). With switch SW2 open, the programme returns to Sequence (0). A further computation then takes place with the p.r.b.s. produced by Register (1) lagging the p.r.b.s. produced by Register (2) by one further bit as compared to the previous computation. The computation continues until switch SW1 is opened. The programme may then be reset by closing SW2.

On closing SW1 momentarily, the programme proceeds through Sequences (0) to (14) as before. It then continues on to Sequences (16), (17) and (18), where the X scan for the X-Y plotter is reset, and further clock pulses are transmitted to Register (1) until the 'all ones' state of this register is obtained. Control then returns to Sequence (0) with the entire programme reset to the initial conditions.

Experimental results

Cross-correlation measurements have been made on many different systems, particularly on analogue simulated control systems. Two examples of typical measurements on noise-free simulated systems are shown in **Fig. 7.** In the test of Fig. 7(*a*), $N = 63$ and $\Delta t = 0{\cdot}1$ sec. for the p.r.b.s. and for the test of Fig. 7(*b*) $N = 127$, and $\Delta t = 0{\cdot}1$ sec. These results have not been corrected for the non-zero value of the p.r.b.s. auto-correlation function that exists for values of τ different from zero. The results demonstrate that with a careful choice of the parameters of the p.r.b.s. good estimates of the impulse response of a system are possible.

Conclusions

A hybrid computational technique for the evaluation of the dynamics of a system in terms of its impulse response function has been described. The programme only allows

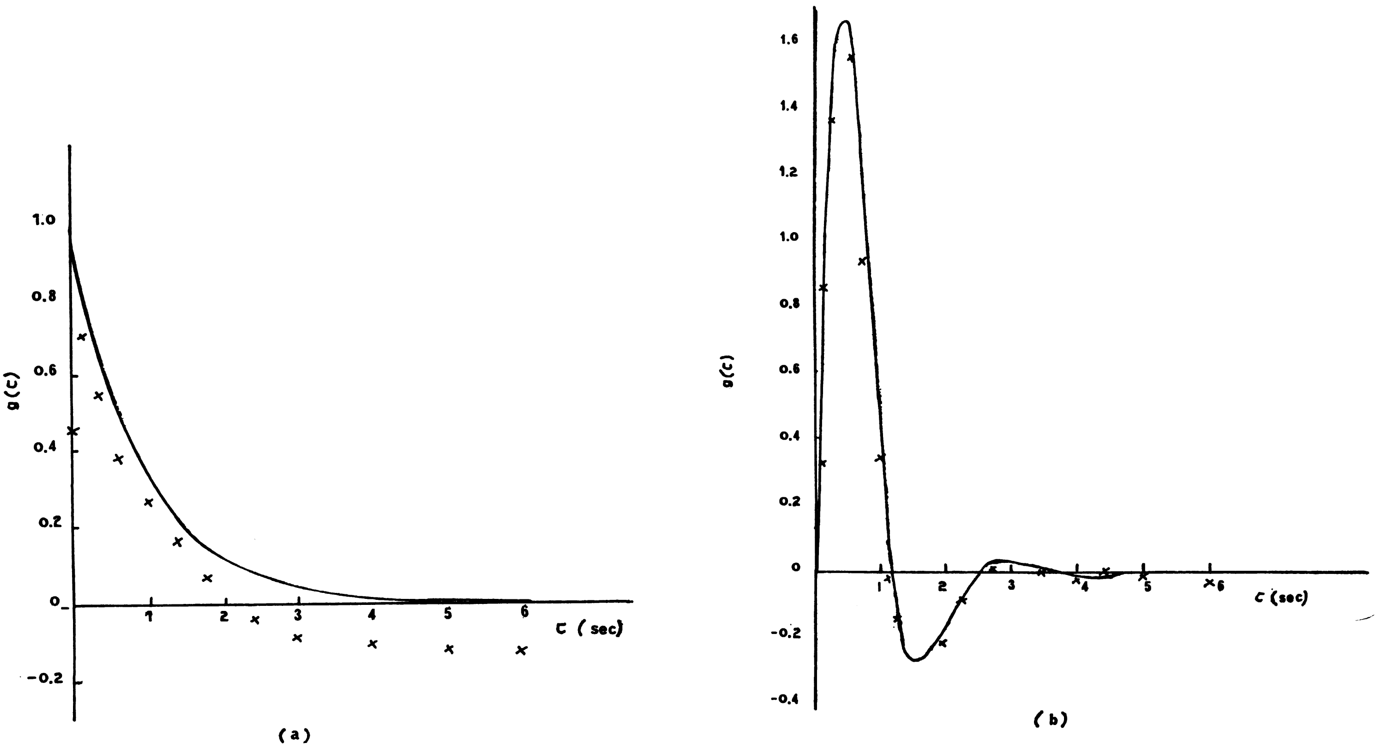


Fig. 7. Measured impulse responses

——— Theoretic plots
× × × × Measured values

(*a*) First order system $\left(\frac{1}{1 + s1.0}\right)$

(*b*) Second order system $\left(\frac{9}{s^2 + 3s + 9}\right)$

the evaluation of one value of the function at a time and is consequently relatively slow. On the other hand the programme is very convenient and easy to use, with results automatically plotted out. It is also very economical in terms of the total hardware requirement.

Acknowledgements

This work was carried out in the Electrical Engineering Department, Lanchester College of Technology, Coventry, and the author is grateful to Dr. N. W. Bellamy for the facilities provided and the encouragement given.

References

Bellamy, N. W., and Hulton, L. J. (1968). A Sequential Mode Control System for Hybrid Computation, *Control*, Vol. 12, No. 117.

Douce, J. L. (1963). *Mathematics of Servomechanisms*. English Universities Press, Chapter 6.

Finnie, B. W., and Roberts, G. T. (1965). Some Practical Aspects of Pseudo Random Signal Testing, I.E.E. Colloquium on Random Signal Testing.

Huffman, D. A. (1961). Synthesis of Linear Sequential Coding Networks, Proc. 3rd. London Symposium on Information Theory.

Petersen, W. W. (1961). *Error Correcting Codes*. M.I.T. Press.

Correspondence

To the Editor
The Computer Journal

Interpretation of limited entry decision table format

Sir,

A number of quite distinct issues were raised by Carrick (1970) in his brief comment on my recent paper (King, 1969). I should like to try and distinguish them and apologise for writing at rather greater length than he. Some of his rather minor points I will ignore.

First, let me contradict his statement that my view is that standards in data processing are undesirable because they are unnecessarily restrictive. I have *never* said this and it is most certainly *not* my view. In practical data processing work, standards are important and vital and the wider and more complex the scope of the work, the more important they become. It seems clear that Mr. Carrick and I are in complete agreement on this. The difficulty seems to be that he has not appreciated that the adoption of standards normally occurs in two phases: first, there comes a clear recognition that standards are necessary; secondly, there is the careful selection of the particular standards to be used. The recognition that standards must be adopted does not mean one should immediately grab the first ring binder to hand with the word 'Standards' on the cover and adopt its contents, any more than should the man, deciding that he wishes to get married, rush out into the street and embrace the first long haired person he encounters. It is, for example, particularly useful to have a standard programming language in an installation or over a group of installations. However, to decide *what* this language or languages sholud be is not easy—FORTRAN and COBOL or PL/1? In the conversation to which Mr. Carrick refers, my remarks were about *which* standards one should adopt. There are some doubtful ones. The remarks of Tully (1969)—see particularly his fifth paragraph—deserve a wider reading than they have hitherto received.

On the question of leaving entries in a decision table blank rather than using dash, I agree that this is undesirable. If Mr. Carrick would look again at the sentences to which he referred, he will not find I recommend it but merely comment that it is sometimes done. If he doubts this, I suggest he looks at p. 146 of the recent text based on the NCC systems analysis course (Daniels and Yeates, 1969). I hope that his attempt to remedy this particular matter by prayer will be successful.

To come to the main point—the use of decision tables and the conventions which should apply—Mr. Carrick is merely confusing things with his well intentioned but misguided attempt at oversimplification. He states, '... if we can agree that decision tables are a means of communication', without appreciating that this needs to be qualified by saying what is to be communicated.

One may want to communicate efficient algorithms (specifications of the necessary logical tests to achieve particular ends—'what it is *necessary* to do in order to decide ...'). On the other hand, one may use a table as a tool for problem analysis; for the consideration of all combinations of states of the logical variable to decide what action is required in each circumstance and to ensure that no situation has been overlooked. This type of use leads to descriptive tables ('... what is true when ...') which document the values of the logical variables in the various circumstances. These do not necessarily specify good problem *solutions* and may specify unnecessary activity. The table shown in Fig. 8 (p. 323) of my paper is meant to be a simple illustration of how this occurs.

It may well be that in some contexts the use of descriptive tables for the specification of algorithms is satisfactory since the time inefficiency introduced into the programs will not matter. I find it difficult to accept in general, however, Mr. Carrick's notion that it simplifies things to consider 'impossible situations' and then carefully specify 'impossible actions' in such cases for the sake of 'completeness'. I suspect he has little practical experience of working with tables with more than four or at most five conditions. With seven, eight or nine interrelated conditions, I doubt he will find his suggestions as simple as claimed.

I should make it clear that my view is that the hand checking of tables of any significant size is not a sensible way of proceeding at all since this is a tedious and error-prone activity which can easily be computer-aided. I would gladly demonstrate to him how this activity can be quickly and easily accomplished, using the normal and now widely available time sharing services. I consider it unreasonable that computer professionals concerned with systems analysis and design should not have such facilities at their elbow.

Rather than Mr. Carrick's, my own preference is for the approach recommended in the ICL systems procedures manual (ICL, 1969). The producers of this manual have carried out well the important but difficult task of taking the current state of development of a topic as exemplified by current papers in the literature and the discussions among those interested in the field, and reducing this to a set of recommendations for practical day-to-day use. A tool for the man who must get on with the job on hand and cannot be distracted by, and may not even be interested in, discussions and difficulties arising in the further development of such tools. As methodology for systems analysis and design develops it is clear that this type of activity will become increasingly important.

Yours faithfully,
P. J. H. King

Department of Computer Science
Birkbeck College
University of London
Malet Street, London WC1
30 September 1970

References

Carrick, D. A. (1970). Letter to the Editor, *The Computer Journal*, vol. 13, p. 220

Daniels, A., and Yeates D. (1969). *Basic training in systems analysis*, Pitman, 275 pp.

ICL (1969). Systems procedures manual: a professional approach to systems analysis and design. (See appendix 2).

King, P. H. J. (1969). The interpretation of limited entry decision table format and relationships among conditions, *The Computer Journal*, Vol. 12, pp. 320-326.

Tully, C. J. (1969). DP management and standards. Touche Ross and Co.