# DIGITAL CONTROL USING WEIGHTING SEQUENCE MODELS

A Thesis submitted for the Degree of
Doctor of Philosophy

by

George Alan Hepburn

Department of Electrical Engineering,
University of Glasgow.

# CONTENTS


Page

PREFACE .................................................... (iii)

SUMMARY .................................................... (v)

CHAPTER 1: INTRODUCTION AND SURVEY

1.1 INTRODUCTION .................................................... 1

1.2 THE IDENTIFICATION PROBLEM .................................................... 2

1.3 USE OF WEIGHTING SEQUENCE MODELS IN CONTROLLER DESIGN - A SURVEY .................................................... 6

1.4 PRELIMINARY CONSIDERATIONS .................................................... 11

CHAPTER 2: WEIGHTING SEQUENCE IDENTIFICATION

2.1 GENERATION AND PROPERTIES OF PSEUDO-RANDOM BINARY SIGNALS .................................................... 15

2.2 ERRORS PRESENT IN IDENTIFICATION USING PSEUDO-RANDOM BINARY TEST SIGNALS .................................................... 21

2.3 IDENTIFICATION OF SYSTEMS WITH TIME DELAYS ........ 27

2.4 CONVENTIONS USED IN OBTAINING AND PRESENTING RESULTS OF TEST SIMULATIONS .................................................... 29

2.5 EXAMPLES OF SIMULATED IDENTIFICATION .............. 34

CHAPTER 3: THE FEEDFORWARD CONTROLLER

3.1 ADVANTAGES AND DISADVANTAGES OF THE FEEDFORWARD TECHNIQUE .................................................... 48

3.2 DERIVATION OF CONTROLLER EQUATIONS ................. 49

3.3 FEEDFORWARD CONTROL APPLIED TO SYSTEMS WITH TIME DELAYS .................................................... 52

Page

3.4 STABILITY OF THE FEEDFORWARD CONTROLLER ........... 54

3.5 OPERATION OF FEEDFORWARD CONTROLLER WITH LIMITATIONS ON THE CONTROL SIGNAL ................. 58

3.6 EXAMPLES OF SIMULATED FEEDFORWARD CONTROL ......... 62

CHAPTER 4: THE FEEDBACK CONTROLLER

4.1 PRELIMINARY CONSIDERATIONS ........................ 74

4.2 DERIVATION OF CONTROLLER EQUATIONS ................ 76

4.3 STABILITY OF THE FEEDBACK CONTROL LOOP ............ 79

4.4 IDENTIFICATION IN THE PRESENCE OF FEEDBACK CONTROL 82

4.5 EFFECT OF BIASSED MODELS ON CONTROLLER STABILITY .. 87

4.6 EXAMPLES OF SIMULATED FEED-BACK CONTROL ........... 89

CHAPTER 5: PROGRAM DESCRIPTION

5.1 GENERAL CONVENTIONS .............................. 110

5.2 THE MAIN PROGRAM ................................. 113

5.3 THE SUBROUTINES .................................. 120

CHAPTER 6: CONCLUSIONS

6.1 ACHIEVEMENTS AND LIMITATIONS ...................... 131

6.2 SUGGESTIONS FOR FURTHER DEVELOPMENT ............... 136

APPENDIX I: NOMENCLATURE ........................ 138

APPENDIX II: DERIVATION OF z TRANSFORM FOR SIMULATION OF THE SYSTEM ........... 141

APPENDIX III: BIBLIOGRAPHY ....................... 142

# P R E F A C E

This thesis describes research carried out during the period 1966-1970 in the Electrical Engineering Department of the University of Glasgow, under the supervision of Dr. H. A. Barker, Senior Lecturer in Control Engineering, and Prof. J. Lamb, Professor of Electrical Engineering.

No part of this work has been submitted previously for a qualification at Glasgow or any other University. Some of the material was, however, presented as a paper at the 1969 IFAC Congress in Warsaw[(1)].

Originality is claimed for all the work contained in Chapters 3-6, except where the work of other authors is specifically referenced. While the error elimination techniques described in Chapter 2 are not themselves original the investigation of their combined application is novel.

I wish to acknowledge with gratitude the assistance provided by all those with whom I have come in contact during the course of my research, both within the Engineering Department and at the National Engineering Laboratory, East Kilbride.

In particular, I should like to thank Dr. H. A. Barker for useful suggestions, discussion of and interest in my work; Prof. J. Lamb, for making available the facilities and financial support of the Engineering Department; and Mr. R. H. Weir, the Director of the National Engineering Laboratory, for providing as part of the Regional Experiment in Computer Aided Design, the excellent computing facilities without which this research could not have been completed.

I am also grateful to the members of the typing staff at the National Engineering Laboratory for their work in preparing the text of this thesis, and to the Scientific Research Council for financial support.

G. A. HEPBURN
July 1970

## S U M M A R Y

A brief review of identification techniques for linear systems is presented, special emphasis being given to the relatively new statistical techniques which yield an identification in terms of a weighting sequence model. It is noted that the use of such models in controller design has received little attention, the few exceptions to this being discussed, and it is concluded that further development in this field will prove useful.

An outline proposal for a scheme to simultaneously identify and control a general multi-input, multi-output system whose paths may be represented by linearised models, is given. The proposal envisages the implementation of a combination of feedforward and feedback controllers, both of which utilise weighting sequence models of the system.

The method used in carrying out the identification is described, particular attention being given to the elimination of certain errors inherent in the basic technique. The effectiveness of the correction procedures is demonstrated by example.

A feedforward controller is described which effectively eliminates the effect of measurable disturbance inputs in a system with one controlled output, subsequent considerations being limited to this case. The limitations of this controller are investigated by analytical techniques and examples of simulated operation.

A feedback controller, using techniques similar to those devised for the feedforward controller is introduced to minimise the effects of unmeasurable inputs, and errors in the feedforward controller. An analysis of the stability of this system is given and the important aspects of its performance are demonstrated by means of simulation.

The program used to simulate the operation of the system is presented, together with a general description.

The extent to which it has been possible to implement the original proposals and the limitations which it has been found necessary to introduce at various stages, are summarised. In particular, the failure to achieve satisfactory simultaneous identification and control is noted.

Suggestions are made for a logical continuation of the work.

# CHAPTER 1

## INTRODUCTION AND SURVEY

## 1.1 INTRODUCTION

In the problem of devising a system to control a plant in accordance with some desired performance criteria, three major subdivisions may be defined.

(i) Some knowledge of the plant dynamics must be obtained.

(ii) The configuration and dynamics of the desired controller must be derived.

(iii) A physical device possessing these dynamics must be implemented.

It will, of course, not always be possible to make a clear-cut delineation between these phases.

This dissertation describes an attempt to apply the techniques of digital computation to achieve an integrated solution to all three phases, applicable to a wide class of plant dynamics.

## 1.2 THE IDENTIFICATION PROBLEM

In the field of systems engineering, the term identification has numerous connotations. In this dissertation it is restricted to a specific problem - that of determining the parameters of a structure, postulated as a mathematical model of the system, which when fed with the same input x(t) as the system, produces an output zm(t) which resembles the system output zs(t) as closely as possible.

The set of systems for which this structure may be formulated in terms of linear differential equations constitutes a wide class for which the mathematical analysis is much simplified. Much of classical control theory has been derived in terms of this differential equation formulation, or its equivalents, the Laplace Transform and State Space formulations. The structure parameters to be identified in this case are a set of differential equation coefficients.

Alternatively, such systems may be described in integral equation form by the convolution integral:

$$zs(t) = \int_{-\infty}^{\infty} ws(\tau).x(t - \tau).d\tau \qquad (1.1)$$

where ws(τ) is the weighting function of the system. This offers two advantages over the differential equation formulation: pure time delays may be represented with no additional complexity, and the order of the differential equation describing the system is unspecified. Thus, if these advantages are to be carried through to the model, the structure postulated for this must be capable of modelling systems in which $ws(\tau)$ is of arbitrary order. A structure which closely approximates this ideal is the weighting sequence, which is defined as:

$$(ws)_j = ws(\tau).\delta(\tau - jT) \tag{1.2}$$

where j is any integer, and

T is the sampling period.

Identification procedures using this structure have received considerable attention in the literature (vide infra). The weighting sequence has, however, two disadvantages: while the description of a system governed by a differential equation of order n requires the knowledge of a maximum of 2n + 1 coefficients, where n is usually assumed to be less than 10, the description of such a system by weighting sequence would typically require the specification of some 100 ordinates. Having obtained this somewhat cumbersome model, the problem of its utilisation is formidable due to the difficulty of handling the integral equations involved, and the lack of analytical techniques comparable to those which have been so widely developed for the differential equation formulation.

In general, the determination of model parameters is carried out using a combination of mathematical analysis and practical experimentation. The determination of the weighting sequence is primarily carried out experimentally, since analytical methods invariably require formulation in the more convenient differential equation form as an intermediate step. The most straightforward test is the observation of the response of the system to a unit impulse input. In other than noise-free conditions, however, such a technique would yield a poor signal/noise ratio, since the power of any practical approximation to an impulse is necessarily small. This problem may be ameliorated to some extent by using a step as the test signal, and differentiating the observed response. The differentiation process serves to accentuate the noise level once again, however.

In recent years, much attention has been given to the development of cross-correlation techniques of weighting-sequence identification, starting with the work of Lee[2,3] around 1950. Early attempts used stochastic test signals (e.g. white noise) and were characterised by the long integration times needed to approximate the infinite integrals involved and by the difficulty of generating test signals approximating the statistical properties of white noise.

Around 1960 Anderson, Buland and Cooper[4] and Hughes and No~~r~~ton[5] proposed the use of random binary test signals which aleviated the latter problem. The description in 1964 of the use of cyclic binary test perturbations by Briggs, Hammond, Hughes and Plumb[6] which allowed integrations to be made over the period of the test signal removed the other major practical obstacle, and opened the way for detailed development. The application of these pseudorandom signals to weighting sequence identification, and the refinement of the procedures to minimise the errors involved has received considerable attention, particularly in this country, since that time. A complete bibliography, covering the theoretical and practical aspects of these test signals, has recently been published[7]. Recent work in this field, relevant to the present dissertation, will be described in Chapter 2.

## 1.3 USE OF WEIGHTING SEQUENCE MODELS IN CONTROLLER DESIGN - A SURVEY

In view of the considerable interest which has been shown in the identification of weighting sequences, it is surprising that such scant attention has been paid to their utilisation in the design of controllers, despite the difficulty of analytical treatment mentioned above. In this connection it should be noted that the noise-rejecting properties of cross-correlation identification schemes render them suitable for on-line application, provided the inevitable low-level output perturbation can be tolerated. Such is often the case in chemical plants, for example.

On-line use of the technique introduces the possibility of utilising weighting sequence models in an adaptive controller, which will of necessity be restricted to systems with slowly varying parameters, since the system dynamics must be assumed constant over the identification period.

A survey of the literature indicates very few suggestions for applications of weighting sequence models. Anderson, Buland and Cooper[4,8] proposed a self-adaptive controller using these models in 1959. A closed loop system with an adjustable compensator in the forward path was continuously modelled using a white noise test signal.

The system was assumed to be governed by a second order linear differential equation, and it was desired to maintain the damping factor of this equation at a constant optimum value. Evaluation of successive positive and negative areas of the resulting weighting function model enabled the actual damping factor to be determined, and the parameters of the compensator to be automatically adjusted to remove the error. This system sacrifices the primary advantage of the weighting sequence formulation, namely its ability to describe an arbitrary linear system. In cases where the assumption of a simple form of dynamics is admissable, parameter tracking systems will provide a more rapid identification and require less computing time and storage.

In 1965, Dawkins and Briggs[9] outlined a scheme for an optimum terminal point controller using a general weighting sequence model. The techniques of dynamic programming, to which the discrete form of the weighting sequence is ideally suited, were employed to generate the input sequence of the system, the criterion to be optimised at a point n samples prior to the terminal point being:

$$f^*_n = c \sum_{j=1}^{n} x_j^2 + z_o^2$$

where $x_j$ is the control input to be applied j samples before the terminal point,

$z_0$ is the terminal value of the system output, and

c is a constant.

This criterion was optimised stage by stage, based on a prediction, using weighting sequence models of $z_0$ at each stage. An expanded system using prediction of intermediate points was also described. The authors noted that this was essentially a feed-forward system, and proposed the introduction of a continuously updated estimate of both the system weighting sequence and of any random noise present at the system output, these estimates being used in an unspecified manner to vary the coefficients of the optimisation routine in order to introduce feedback to the system.

The paper was only intended as an outline of a possible future line of investigation, and as such has left some major questions unanswered; for example, the manner in which the identification was to be carried out was not described, and no investigation of the stability of either the open- or closed-loop system had been carried out.

In 1969 Beck, Birch, Gough and Plaskowski[10] described a scheme which had been implemented on a pilot-scale water heating process. The water heater had two inputs - water of randomly varying temperature, and electricity for heating, which could be controlled. It was desired to control the outlet water temperature. Time delays were present in both the inlet and outlet water channels. These were known to be proportional to the water flowrate and hence were easily computed. The weighting sequence of the electrical power input/water outlet channel was modelled using a pseudo-random binary test signal. A fixed weighting sequence model was used to predict the effect of inlet water temperature disturbances, and a feedforward control signal was obtained from this by the minimisation of an unspecified cost function, using an iterative search procedure based on the control path model. Long term drift was counteracted by applying low gain proportional plus integral feedback control, the coefficients for which were calculated by an off-line algorithm.

Since the authors omitted to give any indication of the performance of the system it must be assumed that is is still in the early stages of development. They have, however, recognized the advantages of feedforward predictive control, especially in the presence of time delays, and the scheme represents an attempt to make use of weighting sequences in an area in which their advantages may be utilised. A restriction to the system is the inability to identify the control path when the feedback loop is in operation. The authors proposed that the feedback controller would normally be inoperative and that the controlling software would detect when excessive disturbances appeared at the output, terminate the identification, and close the feedback loop. They did not indicate how this was to be achieved, however.

## 1.4 PRELIMINARY CONSIDERATIONS

The preceding survey indicates a need for further investigation in the field of weighting sequence model application to controller design. This dissertation describes a scheme, applicable to a large class of linear systems, which has been devised with the identification and utilisation of weighting sequence models of all measurable input/output paths as a primary design criterion. Since a practicable on-line cross correlation identification invariably involves the use of a digital computer, it is logical that the control signals should also be generated by this machine. The discrete form of the weighting sequences lends itself to this. As no facilities were available for implementation of the scheme on a plant or even on a hybrid computer installation the entire system has had to be simulated on a general purpose digital machine. It has, however, been borne in mind that an application of this scheme would require to be implemented on one of the small direct digital control computers which have recently become available, as these have the virtues of reliability and low cost.

Since the equations presented subsequently require to be implemented as computer algorithms, the discrete formulation has been used throughout. Except in Chapter 5, sampling is assumed to occur synchronously with unit period, thus eliminating a number of cumbersome constants in many cases. In all equations the nomenclature of Appendix I is used. While this is apparently somewhat complex it has been found to be the only way of avoiding confusing ambiguity.

A general linear system with A inputs and B outputs may be described in terms of the sampled inputs $(xa)_i$ and sampled outputs $(zsb)_i$, using the above conventions, by the relationship:

$$(zsb)_i = \sum_{a=1}^{A} \sum_{j=0}^{\infty} (wsab)_j \; (xa)_{i-j} \qquad (1.3)$$

for b = 1 ... B. The summation commences at j = 0 since $(wsab)_j = 0$ for $j < 0$ for a real system. A requirement for the application of cross correlation identification using cyclic test signals is that the weighting function should settle to a negligible value during the cycle length of the test signal. Implementation of the scheme is therefore only feasible on systems where the assumption

$$(wsab)_j = 0 \text{ for } j \geqslant Ka \text{ for all } b \qquad (1.4)$$

is valid, $K_a$ being a constant for each input. This allows equation (1.3) to be rewritten in closed form.

$$(zsb)_i = \sum_{a=1}^{A} \sum_{j=0}^{Ka-1} (wsab)_j . (xa)_{i-j} \qquad (1.5)$$

The problem considered in this dissertation is defined as follows. Given a system whose dynamics may be adequately represented by linearisation about the operating point, with B outputs which are to be controlled, what input sequences $(xa)_i$ $(a = 1 \ldots B)$ must be applied in order that the outputs $(zsb)_i$ will follow the desired trajectories $(csb)_i$, and be unaffected by disturbances at the remaining A-B inputs? The above implies the assumption that $A \geqslant B$. This must be so, otherwise the problem has no general solution. In terms of the general formulation (1.5) it implies that the control of all outputs is desired. This is admissable since, if we are not interested in any remaining outputs, they may be considered as non-existent for the purposes of the problem definition. The required control sequences $(xa)_i$ $(a = 1 \ldots B)$ are to be calculated using model weighting sequences $(wsab)_j$ obtained by a simultaneously operating cross-correlation procedure using pseudo-random binary test signals $(pa)_i$ superimposed upon the A inputs. This procedure requires the existence at the outputs of low-level perturbations given by:

$$(rsb)_i = \sum_{a=1}^{A} \sum_{j=0}^{Ka-1} (wsab)_j \; (pa)_{i-j} \tag{1.6}$$

Thus the desired outputs of the system are $(csb)_i + (rsb)_i$.

In practice, the actual system output will be the sum of $(zsb)_i$, $(rsb)_i$ and a noise term $(nsb)_i$ which may conveniently be regarded as representing the combined effect of internal system noise, unmeasurable inputs, erroneous modelling, etc. Thus the overall system output is:

$$(usb)_i = (zsb)_i + (rsb)_i + (nsb)_i \tag{1.7}$$

and it is desired to make this as near as possible to $(rsb)_i + (csb)_i$. The scheme to be described has been implemented in three parts, each designed so that it operates independently of the others as far as possible.

(i) An on-line identifier using test signals derived from pseudo-random binary codes updates the models of all paths cyclically.

(ii) A feedforward controller insures that the effects of the disturbance inputs $(xa)_i$ $(i = B + 1 \ldots A)$ are counteracted. This controller also implements the changes in set point required by $(csb)_i$.

(iii) A feedback controller eliminates the $(nsb)_i$ term as far as possible. These operations are dealt with in Chapters 2-4 respectively. All procedures make use of convolution-type algorithms which require the repetitive execution of large numbers of simple calculations and are thus ideally suited to implementation on a small digital computer.

# CHAPTER 2

## WEIGHTING SEQUENCE IDENTIFICATION

## 2.1 GENERATION AND PROPERTIES OF PSEUDO-RANDOM BINARY SIGNALS

As this subject has received considerable attention in the literature, no attempt will be made to give a complete exposition here. The following is a review of that material which is directly relevant to this dissertation.

The generation of a class of pseudo-random binary signals requires, as an intermediate step, the generation of a maximum length null sequence (m-sequence). Successive states of an m-sequence of length $k = 2^n - 1$, which take the logical values 1 or 0, are determined by a logical operation on the previous n states. This operation may be represented by the equation:

$$(1 \oplus a_1 z^{-1} \oplus \ldots \oplus a_n z^{-n})X = 0, \qquad (2.1)$$

where X is the present state of the sequence,

$z^{-1}$ is the unit delay operator,

$\oplus$ denotes addition, modulo 2, and

$a_i = 1$ or 0.

For a null sequence to be maximal length, it is required that the polynomial in $z^{-1}$ be primitive and irreducible, i.e. the polynomial is not a factor of $z^{-J} \oplus 1$ for any $J < 2^n - 1$ and has no factors, modulo 2, respectively[5,11]. Tables of such polynomials are given by Peterson[12]. The m-sequence $(m)_i$ is used to control the switching of the pseudo-random binary signal $(p)_i$ between its two levels in synchronism with a clock of period T as follows.

$$(p)_i = +\sigma, \text{ if } (m)_i = 1$$
$$(p)_i = -\sigma, \text{ if } (m)_i = 0. \qquad (2.2)$$

While other means of generating pseudo-random binary signals are available(13) the m-sequence technique has been chosen, as it combines simplicity of generation with reasonably short repetition periods.

The property of pseudo-random binary signals which renders them suitable for use as test signals in weighting sequence identification is their autocorrelation function. In accordance with the conventions set out in Chapter 1, it is assumed that the clock period T = 1. The signal amplitude $\sigma$ is also taken to be 1, and all other amplitudes quoted are normalized with respect to this. Under these conditions, the discrete autocorrelation of a pseudo-random binary signal $(p\alpha)_i$, $i = 0 \ldots k - 1$ is:

$$(\phi.p\alpha,\ p\alpha)_j = \frac{1}{k}\sum_{i=0}^{k-1} (p\alpha)_i \,.\, (p\alpha)_{i+j}$$
$$= 1 \quad j = 0, k, \ldots$$
$$= -1/k \text{ otherwise.} \qquad (2.3)$$

The application of this signal to input $\alpha$ of the general system formulated in Chapter 1 yields the output sequences

$(rsb)_i$ as in equation (1.6). For this system, the input-output cross correlation is given by:

$$(\phi.p\alpha, rsb)_j = \frac{1}{k}\sum_{i=o}^{k-1} (p\alpha)_i.(rsb)_{i+j}. \tag{2.4}$$

Substituting for $(rsb)_{i+j}$ from (1.6) gives:

$$(\phi.p\alpha,\ rsb)_j = \frac{1}{k}\sum_{i=0}^{k-1} (p\alpha)_i \sum_{m=0}^{K\alpha-1} (ws\alpha b)_m.(p\alpha)_{i+j-m}$$

$$= \sum_{m=0}^{K\alpha-1} (ws\alpha b)_m \frac{1}{k}\sum_{i=0}^{k-1} (p\alpha)_i.(p\alpha)_{i+j-m}$$

(assuming k has been chosen so that $K\alpha \leqslant k$)

$$= \sum_{m=0}^{K\alpha-1} (ws\alpha b)_m.(\phi.p\alpha,\ p\alpha)_{j-m} \tag{2.5a}$$

$$= (ws\alpha b)_j - \frac{1}{k}\sum_{m=0}^{K\alpha-1} (ws\alpha b)_m + \frac{1}{k}(ws\alpha b)_j$$

$$= \frac{k+1}{k}(ws\alpha b)_j - \frac{g\alpha b}{k}, \tag{2.5b}$$

where $g\alpha b$ is the steady-state gain of path $\alpha - b$. Thus the determination of $(\phi.p\alpha,\ rsb)_j$ for $j = 0 \ldots K\alpha - 1$ gives a term proportional to $(ws\alpha b)_j$ with a (typically small) bias term due to the non-zero value of the autocorrelation of the pseudo-random binary signal between peaks.

With pseudo-random binary test signals $(pa)_i$ applied to all inputs, the input-output cross correlations involving input $\alpha$ are:

$$(\phi.p\dot{\alpha},\ rsb)_j = \sum_{a=1}^{A} \sum_{m=0}^{K\alpha-1} (wsab)_m.(\phi.p\alpha,\ p\alpha)_{j-m} \qquad (2.6a)$$

corresponding to equation (2.5a). In order to derive a relationship similar to equation (2.5b), one of two conditions must be met.

(i) $(\phi.p\alpha,\ pa)_j = 0$ for $a \neq \alpha$, all j, or

(ii) $(\phi.p\alpha,\ pa)_j = 0$ for all $a \neq \alpha$, only for j in the range $0 \ldots K\alpha - 1$.

The former condition requires the generation of a set of mutually orthogonal test signals, a solution having been given by Briggs and Godfrey(14).

The latter condition is satisfied by the set of test signals defined by:

$$(p\,\overline{n+1})_i = (pn)_{i-Kn} \quad n = 1,2 \ldots A-1, \qquad (2.7)$$

K in this case having to be chosen so that

$$\sum_{a=1}^{A} Ka \leqslant k. \qquad (2.8)$$

The small bias term between the peaks of the correlation if of course still present. The above set of signals is composed of a pseudo-random binary cycle $(p1)_i$ and A - 1 delayed versions of this. This was first pointed out by Barker and Raeside[15]. The cross correlation between (p1) and the delayed versions is:

$$(\phi.\text{p1}, \text{pa})_j = 1 \qquad j = Ka$$
$$= -1/k \qquad \text{otherwise.} \tag{2.9}$$

Substituting in equation (2.6a) the cross correlation between $(\text{p1})_i$ and the system output is:

$$(\phi.\text{p1}, \text{rsb})_j = (\text{ws1b})_j - \frac{1}{k}\sum_{a=1}^{A}\sum_{i=0}^{Ka-1}(\text{wsab})_i + \frac{1}{k}(\text{ws1b})_j$$

$$= \frac{k+1}{k}(\text{ws1b})_j - \frac{1}{k}\sum_{a=1}^{A}\text{gab}$$

$$\text{for } 0 \leqslant j < K1$$

and in general

$$(\phi.\text{p1}, \text{rsb})_{j+\sum_{a=1}^{n-1} K_a} = \frac{k+1}{k}(\text{wsnb})_j - \frac{1}{k}\sum_{a=1}^{A}\text{gab}$$

$$0 \leqslant j < Kn. \tag{2.b}$$

Thus once again the weighting sequences are identified, the bias term in this case being proportional to the sum of the steady state gains of all paths between the A inputs and output b.

This latter method has been adopted in the present case, as the use of a number of different test signals leads to excessive complications both in generation and in the error-correcting procedures described below.

By evaluating equation (2.10) for all b, a biassed estimate

$$(weab)_j = (wsab)_j - \frac{1}{k+1}\sum_{a=1}^{A} gab \qquad (2.11)$$

is obtained, a new estimate of each path being completed once per cycle of the test signal.

## 2.2 ERRORS PRESENT IN IDENTIFICATION USING PSEUDO-RANDOM BINARY TEST SIGNALS

Consider first the error due to the bias term in equation (2.11), which will typically be small, since k is large. The transient performance of a model based on these estimates will closely resemble that of the system it represents, but the steady state gain will be in error by an amount

$$- \frac{Ka}{k+1} \sum_{a=1}^{A} gab. \qquad (2.12)$$

While the purpose of the feedback controller includes the counteraction of modelling errors, it is nevertheless desirable to minimise this term. Two methods are available to achieve this.

Since $(wsab)_j \rightarrow 0$ as $j \rightarrow Ka$, the bias term may be estimated from equation (2.11) as:

$$\lim_{j \rightarrow Ka} (weab)_j = - \frac{1}{k+1} \sum_{a=1}^{A} gab \qquad (2.13)$$

In practice this limit may be evaluated as

$$\frac{1}{A} \sum_{a=1}^{A} \frac{1}{ia} \sum_{j=Ka-ia}^{Ka-1} (weab)_j \qquad (2.14)$$

where ia is typically Ka/10.

The above represents the average of the last ia terms of all (weab) (a = 1 ... A).

A straightforward method of minimising the bias term (2.12) is to insure that the polarity of the test signals at each input is so chosen that the effective $\sum_{a=1}^{A}$ gab is small, based on an a priori knowledge of the approximate values of all gab. The relative merits of these methods in typical applications will be illustrated in examples given later in this Chapter.

In all subsequent equations, it will be assumed that the model weighting sequences $(\text{wmab})_j$ have been derived from $(\text{weab})_j$ by removal of the bias term using a combination of the methods described above.

So far only the cross correlation between the pseudo-random binary signal and its response term $(\text{rsb})_i$ have been considered. The presence of other terms in the output (equation (1.7)) gives rise to spurious correlations, and steps have been taken to minimise these.

During an identification cycle, the output term due to measurable inputs, $(\text{zsb})_i$, may be estimated using the model obtained during the previous cycle. This estimate is given by:

$$(\text{zmb})_i = \sum_{a-1}^{A} \sum_{j=0}^{Ka-1} (\text{wmab})_j . (\text{xa})_{i-j} . \qquad (2.15)$$

The sequence $(\text{reb})_i = (\text{usb})_i - (\text{zmb})_i$ is used in the cross correlation, since it is not possible to estimate $(\text{nsb})_i$.

Thus spurious correlations are still to be expected unless the system is noise-free and the models exact.

The remaining corrupting terms may be considered to be represented by a power series. In situations in which the application of the scheme to be described is envisaged, e.g. chemical plants, low order terms (drift) will often predominate. Consider a low order corrupting disturbance of the form:

$$(a)_i = a_o + a_1.i. \qquad (2.16)$$

Since correlation is carried out using a fixed cycle of the pseudo-random binary signal and shifted versions of the output sequence, such a disturbance would introduce a bias to each weighting sequence ordinate given by:

$$(\phi.p\alpha,\ a)_j = \frac{1}{k}\sum_{i=0}^{k-1} (p\alpha)_i.(a_o + a_1.i)$$

$$= \frac{1}{k}\left\{a_o \sum_{i=0}^{k-1} (p\alpha)_i + a_1 \sum_{i=0}^{k-1} (p\alpha)_i.i\right\} \qquad (2.17)$$

The terms are seen to be proportional to the average and first moment of the pseudo-random binary signal respectively. Barker[16] has described a method for eliminating correlations due to corruption of the form (2.16) which involves little additional computation, and is consequently considered worthwhile. Equation (2.17) may be rewritten:

$$(\phi.p\alpha,\ a)_j = \frac{1}{k}\left\{a_o + a_1 \sum_{i=0}^{k-1} (p\alpha)_i.i\right\}$$

$$= \frac{1}{k}\left\{a_o + a_1 \frac{k-1}{2} - a_1 \left[\frac{k-1}{2} - \sum_{i=0}^{k-1} (p\alpha)_i.i\right]\right\}$$

$$= \frac{1}{k}\left\{(\bar{a}) - a_1 \left[\frac{k-1}{2} - \sum_{i=0}^{k-1} (p\alpha)_i.i\right]\right\} \qquad (2.18)$$

where $(\bar{a})$ is the average of $(a)_i$ over the identification cycle. Barker postulated that, for every m-sequence derived pseudo-random binary signal, there exists a phase (i.e. a shifted version), termed the reference phase, for which

$$\sum_{i=0}^{k-1} (p\alpha)_i.i = \frac{k-1}{2}. \qquad (2.19)$$

Numerical evaluation of equation (2.19) for various pseudo-random binary signals revealed no contradicting cases. The postulate has since been verified analytically by Ream[17]. When using this reference phase in cross correlation measurements equation (2.18) indicates that, if a term $1/k.(\bar{a})$ is removed from the estimate of each weighting sequence term, the resulting model will be free from corruption due to drift of the form (2.14).

A method of eliminating correlations of a pseudo-random binary signal with disturbances of any assumed order has been proposed by Davies and Douce[18]. This procedure has not been implemented as it involves a large amount of computation both off- and on-line, and yields a poor signal : noise ratio when the disturbance is assumed to contain higher order terms.

It should be noted that all correlations are carried out using a fixed cycle of the test signal, and shifted versions of the output sequence. This avoids the difficulty, experienced by early workers, of weighting sequence corruption due to the non-constant moments of the pseudo-random binary signal when a different starting point is used for the determination of each ordinate. A complete evaluation of $(wmab)_j$ for all a, requires the storage of 2k samples of $(reb)_i$. However, if one weighting sequence ordinate is evaluated at every sampling period, then on the completion of a test signal cycle, a new model will be available, only k samples of $(reb)_i$ having been stored at any one time. This method distributes the computation uniformly in real time, which is also advantageous.

An alternative method of identification, in which the bias term (2.12) does not arise, is the use of pseudo-random 3 level signals. These signals are generated using m-sequences on the Galois field (-1, 0, 1) and have zero autocorrelation between alternate positive and negative peaks.

Barker[19] has again demonstrated the existence of a reference phase for these signals. It occurs, however, at a point half way between clock pulses of the generating m-sequence. Thus in the digital simulation each sampling period would have to be split into two parts, and the amount of computation increases correspondingly. While these codes might well be utilised in a practical application, it was thought that the validity of the algorithms to be proposed could be investigated equally well using two level signals.

## 2.3 IDENTIFICATION OF SYSTEMS WITH TIME DELAYS

The presence of a time delay does not affect the operation of the identification procedures described above. The first few ordinates of the weighting sequence will merely be close to zero, any discrepancies being due to the corruption effects dealt with in the previous section. However, the operation of the controllers to be described in Chapters 3 and 4 requires that a distinction be made between these ordinates and the initial part of the weighting sequence itself. A test has therefore been introduced to estimate the duration of any time delay present, and set the appropriate elements in the weighting sequence exactly to zero - a condition which may readily be tested for subsequently.

Since an analytical determination of the time delay is not feasible, a somewhat intuitive approach has had to be employed. The condition chosen is as follows.

$$(wmab)_j = 0$$

if $$(weab)_{i+1} - (weab)_i$$

or $$(weab)_i$$

change sign over the range

$$i = j + 1 \ldots j + ia/2 \qquad (2.20)$$

where ia is as defined in equation (2.14). This condition locates the first uniform trend of gradient in the weighting sequence estimate, which in the majority of cases will mark the end of the time delay. In some unusual cases, an alternative test might have to be devised.

## 2.4 CONVENTIONS USED IN OBTAINING AND PRESENTING RESULTS OF TEST SIMULATIONS

A number of features common to many of the test simulations described in this and subsequent chapters are introduced at this point.

The dynamics of the plant under consideration are expressed in terms of transfer functions in the Laplace Transform variable s, since this will probably be most meaningful to the reader. The discrete formulation required for the digital simulation is obtained in terms of z-transforms, the technique for obtaining these from Laplace transforms being commonly known[20]. Note that the required z-transform is that corresponding to the system Laplace Transform W(s) in series with a zero order hold, i.e. of:

$$\frac{1 - e^{-TS}}{s} \cdot W(s). \tag{2.21}$$

The resulting z transform is expressed as a quotient of polynomials in $z^{-1}$.

$$W(z) = \frac{a_o + a_1 z^{-1} + \ldots}{1 + b_1 z^{-1} + \ldots} \tag{2.22}$$

which is a convenient form for implementation by means of a computer algorithm.

The derivation of the equations given throughout this work requires that the analog-digital interfaces are ideal samplers; i.e. all variables are sampled simultaneously in a negligibly short time, and the sampling operation does not affect the values of the variables being sampled.

This may be expressed analytically as:

$$(s)_i = s(t).\delta(iT) \tag{2.23}$$

where $(s)_i$ is the value of the discrete sample produced by sampling a continuous signal $s(t)$ at $t = iT$. The digital-analog interfaces are zero order holds which, on application of a discrete sequence $(s)_i$ produce a continuous signal defined by:

$$s(t) = s_i \quad iT < t \leqslant (i + 1)T \tag{2.24}$$

It is obviously impracticable to tabulate results for a representative set of plant dynamics covering the wide class to which the identifier and controllers are expected to be applicable. The simulations have therefore been carried out using a small number of transfer functions chosen to demonstrate the particular aspects under investigation in each test.

The transfer function most frequently used in the tests is the unity gain second order system.

$$w(s) = \frac{\omega^2}{s^2 + 2s\omega \cos \alpha + \omega^2} \tag{2.25}$$

The derivation of the equivalent z-transform for this system appears in Appendix II. With $\omega = 1$ and $\alpha = 60^{\circ}$ the above reduces to:

$$w(s) = \frac{1}{s^2 + s + 1}. \tag{2.26}$$

Consider the problem of choosing a test signal suitable for the identification of this system. The parameters to be chosen are the sampling interval T, the cycle length k, and the code generating polynomial.

T must be sufficiently short to give good resolution of the weighting sequence estimate. Other authors have determined that sampling periods between $\tau/2$ and $\tau/10$ can be used where $\tau$ is the shortest time constant in the plant dynamics. An analytical determination of the minimum sampling frequency based on signal theory is impractical due to the number of unrealistic assumptions which must be made. A choice of $T = 0.2$ for identification of the system (2.26) gives adequate resolution of the corresponding weighting sequence, as will be seen in the graphs presented subsequently.

This also insures an adequate discrete representation of the continuous variables in the plant since any fluctuations which are sufficiently rapid to be undetected at this sampling frequency will be severely attenuated by the plant itself.

The code length k is now dictated by equations (1.4) and (2.8). For the above system single path identification is carried out with $k = 127$, correspondingly longer cycles being used for multiple path identification.

As stated in Section 2.2, the reference phase of the test signal is always used. Barker[19] has tabulated those generating polynomials which result in a minimum second moment of this phase. The use of these polynomials insures that spurious correlations due to second order disturbances are minimised.

In many simulations, random disturbance or noise signals are required. These are generated by using a series of pseudo-random numbers as the input to a digital filter equivalent to a continuous system of the form (2.25). In any given case the coefficients $\omega$ and $\alpha$ may be chosen so that the characteristics of the resulting signal are similar to those expected of the disturbance signal in a typical practical application. The series of pseudo-random numbers $(s)_i$ is obtained from a pseudo-random binary sequence $(p)_i$ by applying the algorithm:

$$(s)_i = \sum_{j=0}^{n-1} (p)_{i-j} . 2^j \qquad (2.26)$$

where $k = 2^n - 1$, the cycle length of the pseudo-random binary sequence is four times the cycle length of the test signal used in the identification. Thus, for the purposes of these tests, the disturbances generated in this way may be considered random.

The magnitudes of these disturbance signals are given as root mean square values relative to the test signal amplitude of ±1, averaging for the determination of this value being carried out over an integral number of test signal cycles.

Graphs have been plotted with linear interpolation between sample values, as it is felt this aids interpretation of results. All graphs were drawn automatically using a plotting program developed by the author based on Calcomp software. Time axes are calibrated at intervals of ten sampling intervals.

Block diagrams illustrating the system under test are frequently included. In these, the analog-digital and digital-analog interfacing devices have been omitted to avoid complexity unless otherwise stated.

## 2.5 EXAMPLES OF SIMULATED IDENTIFICATION

The following tests were carried out to verify the various aspects discussed previously.

Test 2.1: Identification of a single path under noise-free conditions

Parameters for the simulation were as follows:

Simulated transfer function: $\dfrac{1}{s^2 + s + 1}$

Code generating polynomial: $z^{-7} \oplus z^{-4} \oplus z^{-3} \oplus z^{-2} \oplus 1$

Sampling interval: 0.2.

The system was allowed to run for one cycle of the test signal to allow transients to subside. The identification was commenced at the start of the third cycle, and was complete by the end of this cycle. A non-reference phase of the test signal was used, and no attempt to remove the offset predicted by equation (2.11) was made. Thus the result of the identification, shown in Fig. 2.1(a) is the estimate (well) of equation (2.11), the predicted offset being clearly visible. Evidently in this noise-free case an accurate estimate of this offset would be given by equation (2.14). The two output cycles over which the identification was carried out are shown in Fig. 2.1(b).

Test 2.2: Identification in the presence of low-order drift

An additive drift of the form $a_o + a_1 i$ was introduced at the system output, as shown in Fig. 2.2(a), and the above test repeated. The large offset predicted in equation (2.17) can be seen in the weighting sequence estimate Fig. 2.2(b). Fig. 2.2(c) illustrates the removal of this offset when a reference phase test signal is used, and allowance for the bias predicted by equation (2.18) is made. As expected, the offset is reduced to that obtained in test 2.1, and could be removed by the estimate in equation (2.14).

Test 2.3: Identification in the presence of random noise

An additive noise sequence $(ns1)_i$ was generated as described in Section 2.4. The noise amplitude was adjusted so that over the two output cycles used for the identification, the signal:noise ratio defined as

$$\frac{\sum_{i=0}^{2k-1} \sqrt{(zs1)_i^2}}{\sum_{i=0}^{2k-1} \sqrt{(ns1)_i^2}} \tag{2.27}$$

was unity. Since the noise signal is generated from a random number sequence with zero mean, the drift component of the form $a_o + a_1.i$ over a large number of samples will be small. Thus the results illustrate an extreme case. The corrupted output is shown in Fig. 2.3(a), and the weighting sequence estimate, obtained using a reference phase test signal, appears in Fig. 2.3(b). The corruption evident in the latter would obviously give rise to considerable errors in the computation of control signals. It is also evident that estimates of offset based on equation (2.18) are no longer reliable. The signal:noise ratio for the weighting sequence in Fig. 2.3(b) is approximately 0.95:1. This factor is, however, dependent on the length of the 'settled' portion of the sequence which is included in the evaluation.

The above results further emphasize the need for a feedback controller to minimise the unwanted output components.

Test 2.4: <u>Multiple path identification in the presence of noise</u>

A block diagram of the simulated system for this test appears in Fig. 2.4(a).

Parameters for the simulation were as follows:

Transfer function of path 1: $\dfrac{1}{s^2 + s + 1}$

Transfer function of path 2: $\dfrac{-4}{s^2 + 3.06s + 4}$

Code generating polynomial: $z^{-8} \oplus z^{-6} \oplus z^{-5} \oplus z^{-1} \oplus 1$

Sampling interval: 0.2.

The settling times of the above transfer functions are sufficiently close to allow the identification to be carried out with k1 = k2 = k/2. The gains of the paths were chosen in accordance with the offset minimisation given in Section 2.2.

The signal:noise ratio over the identification period was 5:1 in this case. The resulting compound weighting sequence appears in Fig. 2.4(b). The offset has clearly been removed, and the separation of the individual weighting sequences is straightforward.

Test 2.5: <u>Identification of time delays in the presence of noise</u>

Figs 2.5(a) and (b) illustrate the application of the algorithm given in Section 2.3. These show the initial portion of the weighting sequence concerned only. The system to be identified was the same as the second path in Test 2.4, with a time delay of 7 sampling intervals in series. The algorithm is seen to identify this time delay effectively.

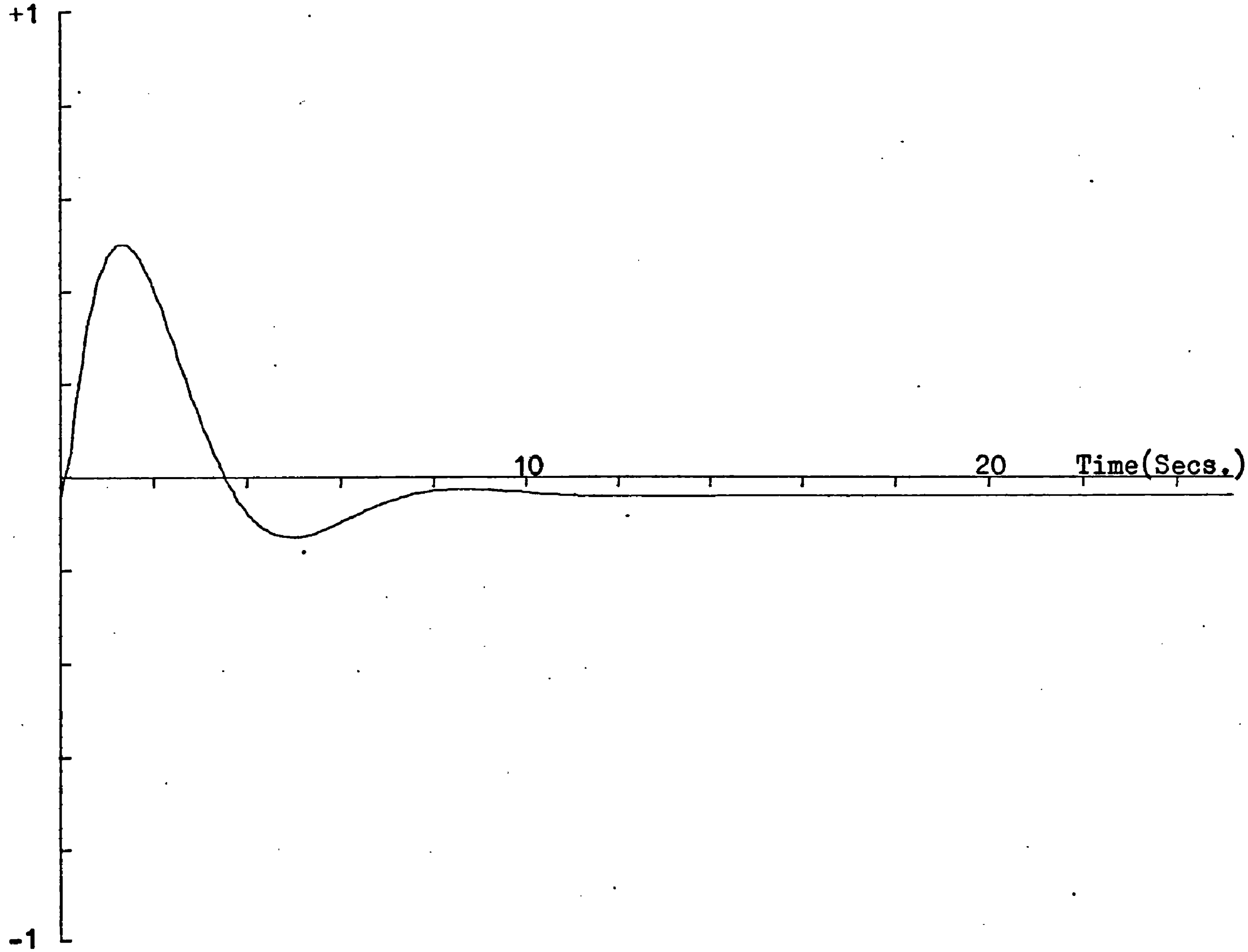



Fig. 2.1(a): Weighting Sequence Identification Obtained in Test 2.1

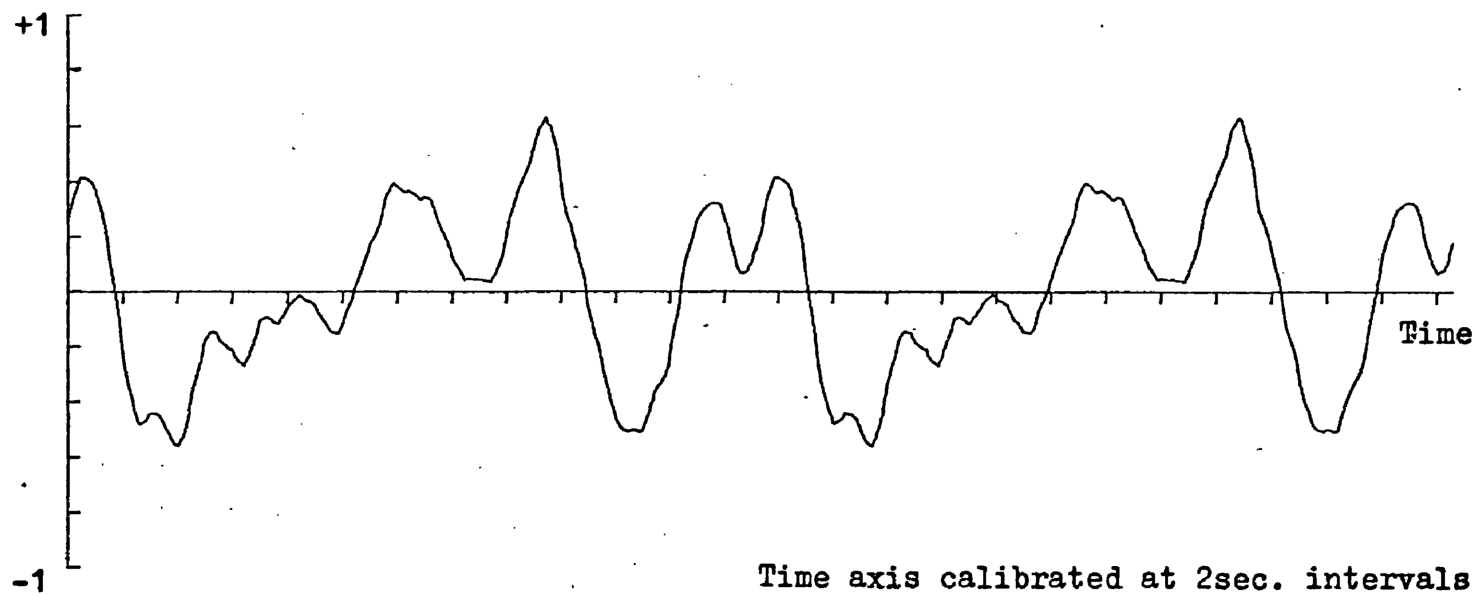



Fig. 2.1(b): System Response Used for Identification in Test 2.1

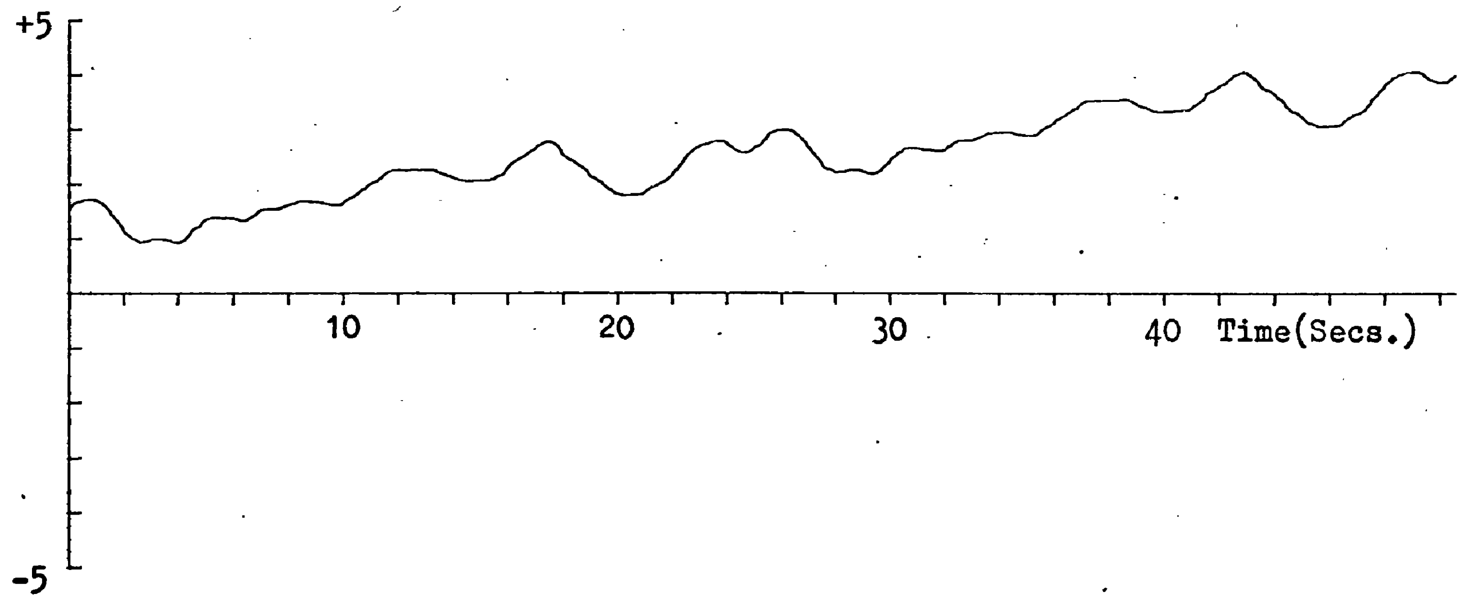



Fig. 2.2(a): System Output Obtained in Test 2.2

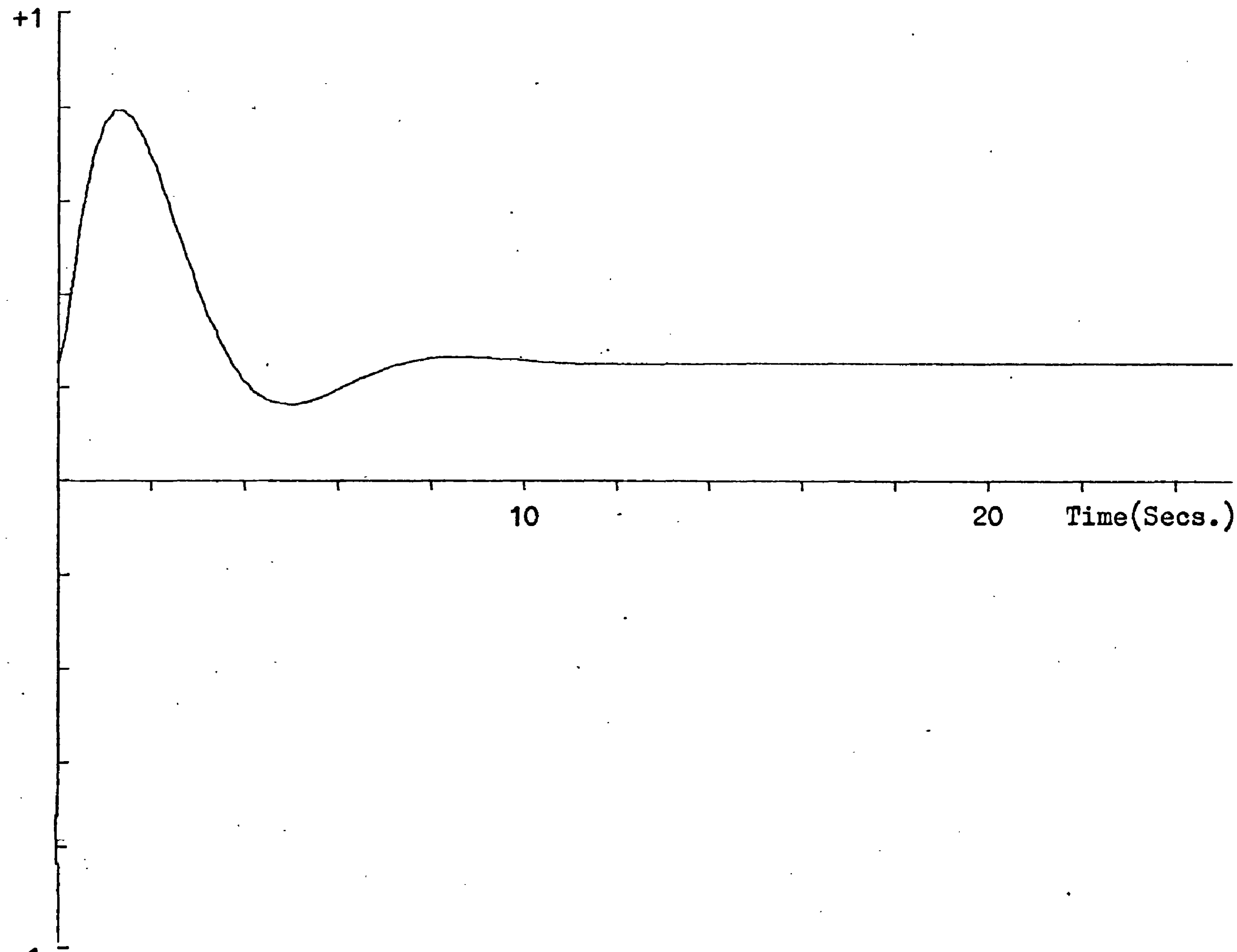



Fig. 2.2(b): Biassed Estimate Produced by Correlation with Non-reference Phase in Test 2.2

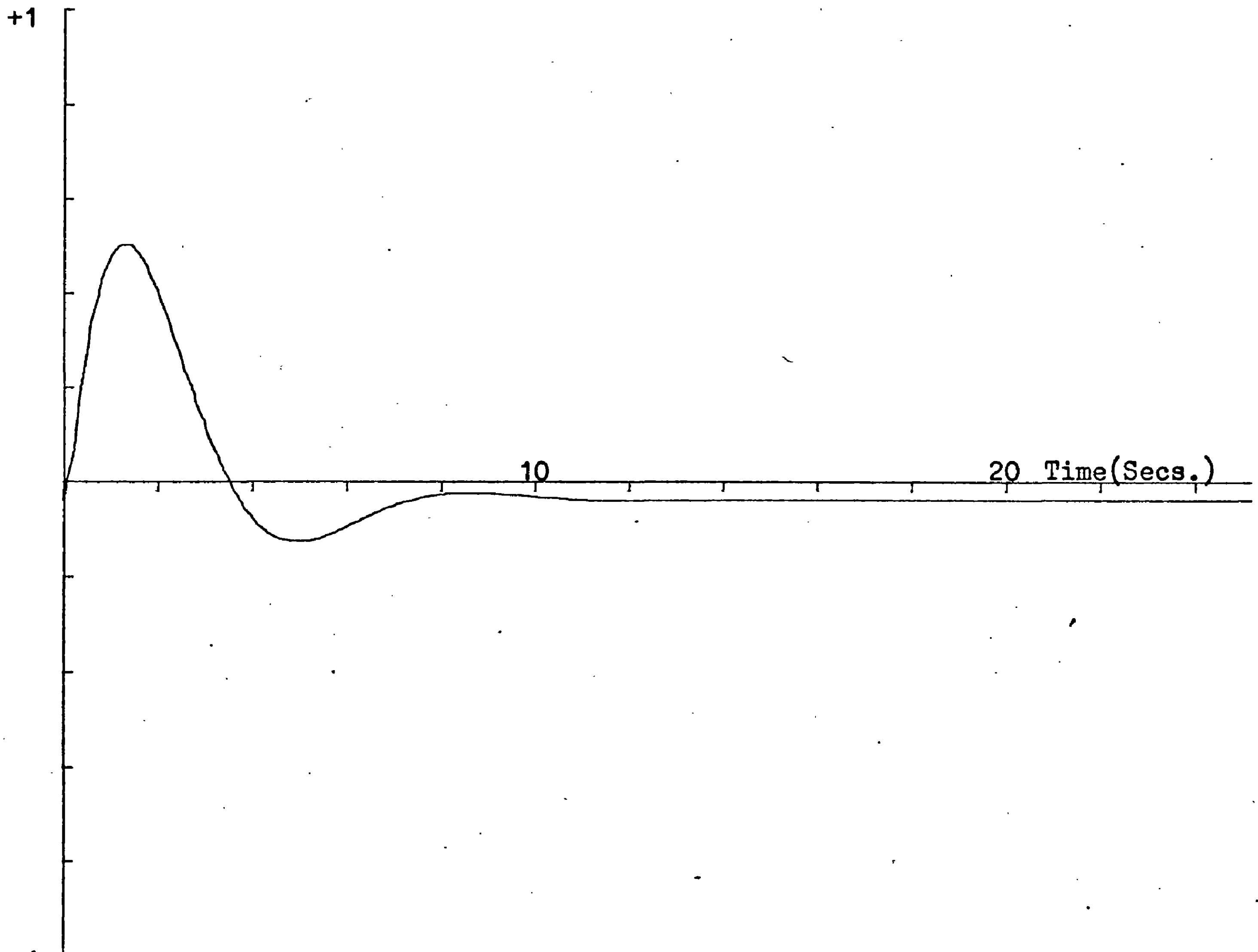



Fig. 2.2(c): Removal of Bias by Use of Reference Phase in Test 2.2

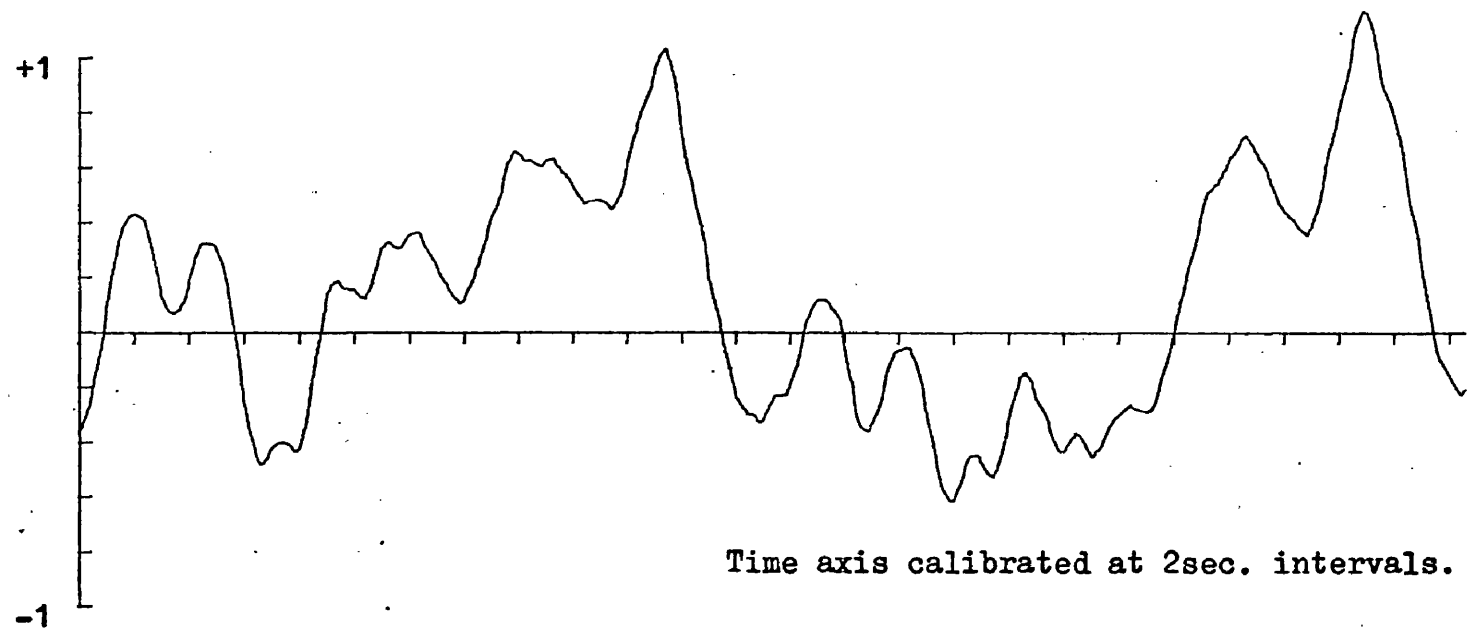



Fig. 2.3(a): Corrupted Output Used in Test 2.3

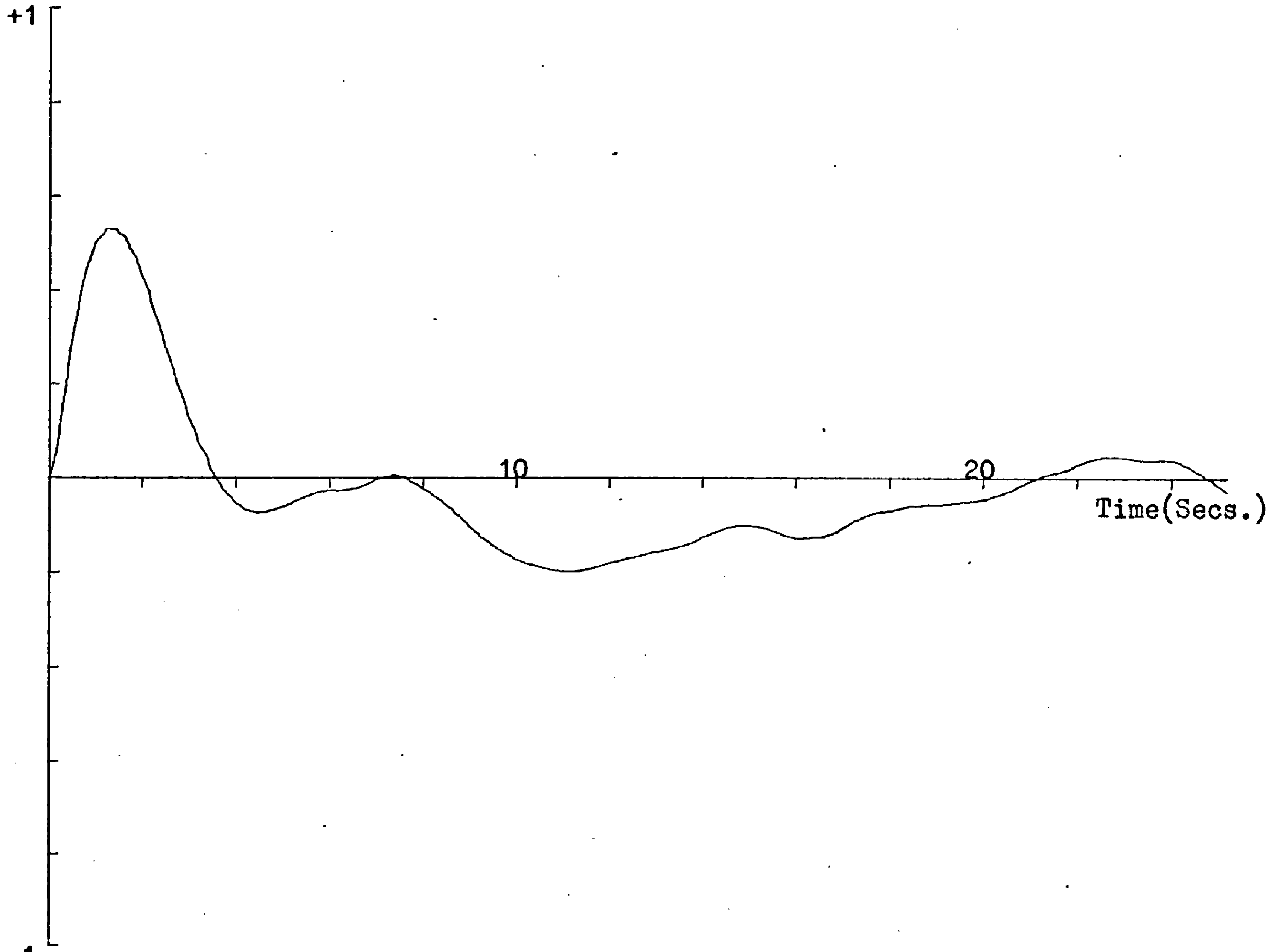



Fig. 2.3(b): Corrupted Weighting Sequence obtained in Test 2.3

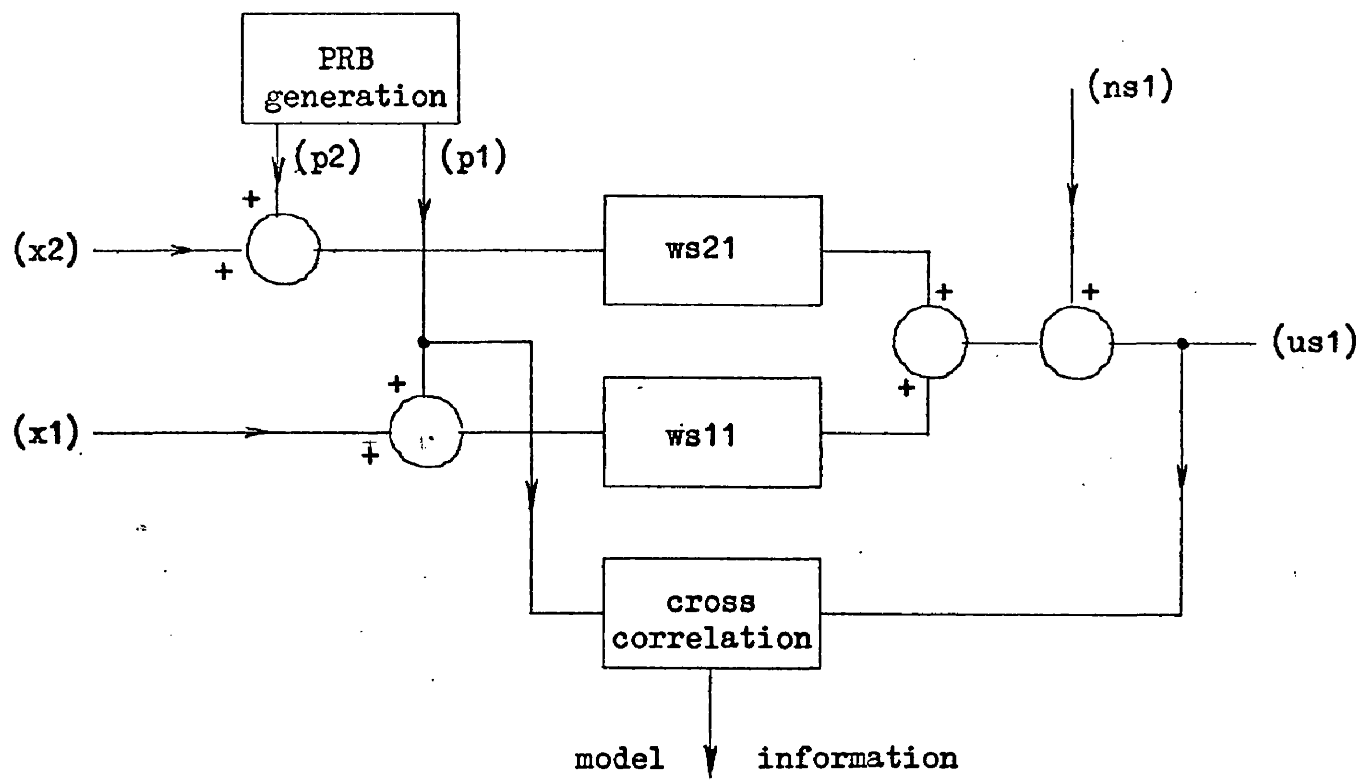



Fig. 2.4(a): Block Diagram for Simulation of Multiple Path Identification

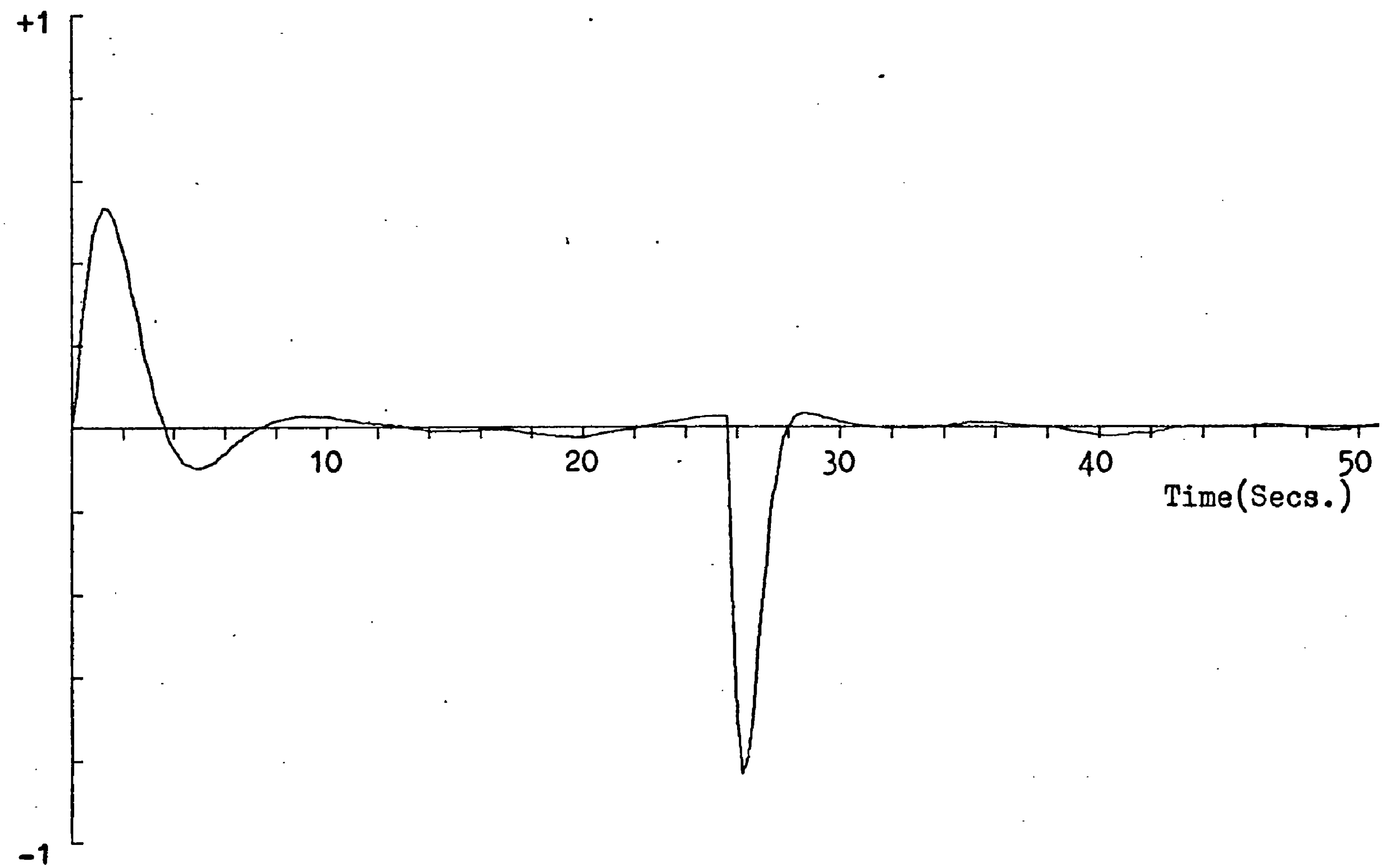



Fig. 2.4(b): Multiple Weighting Sequence Identified in Test 2.4

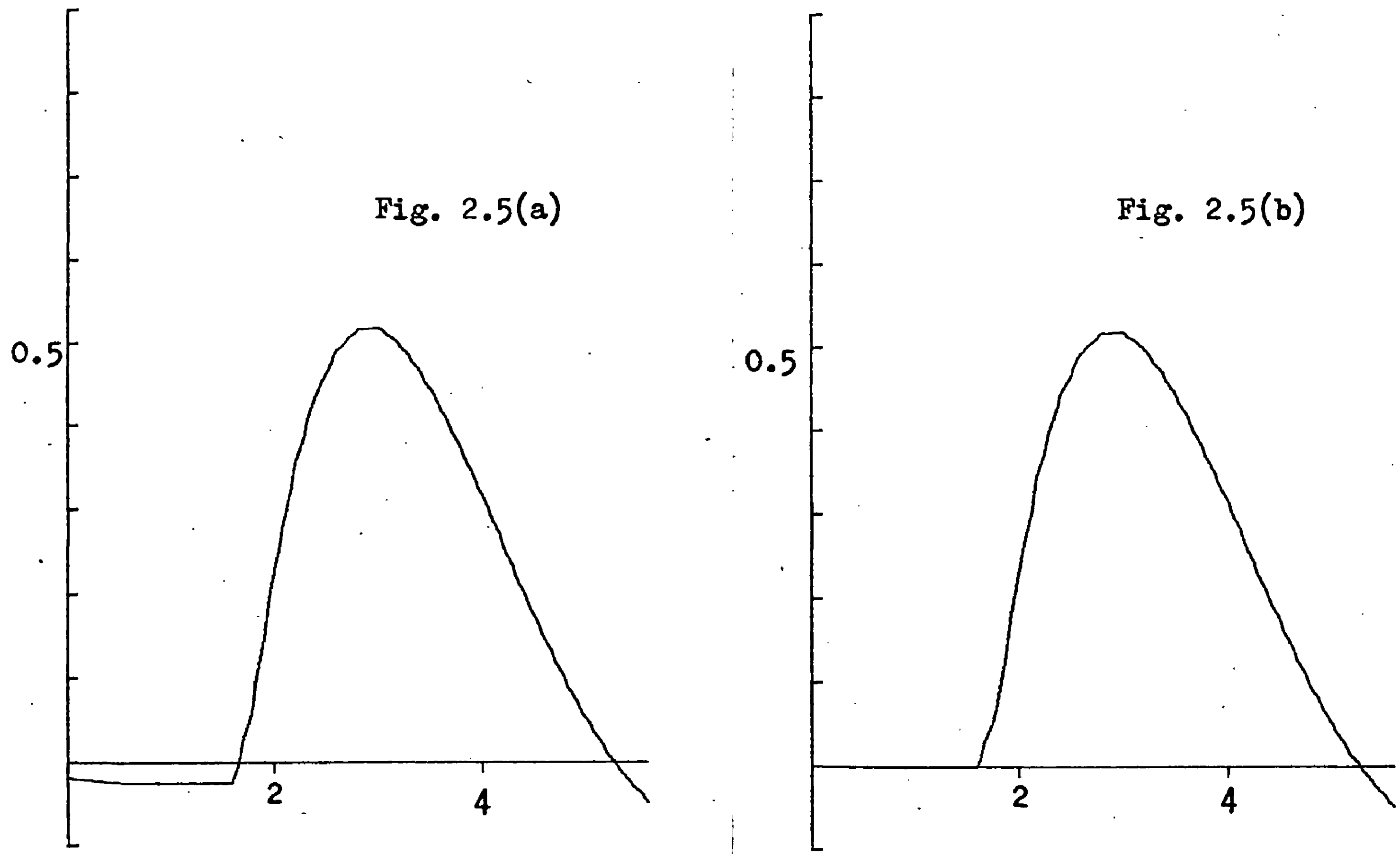



Fig. 2.5: Illustrating Identification of Time Delay - Test 2.5

# CHAPTER 3

# THE FEEDFORWARD CONTROLLER

## 3.1 ADVANTAGES AND DISADVANTAGES OF THE FEEDFORWARD TECHNIQUE

In a feedforward controller, the system outputs are predicted from a knowledge of the inputs, and control signals are generated to cause the outputs to follow the desired trajectories. If perfect prediction and control signal generation can be achieved, then these trajectories will be followed without error. This is in contrast to the feedback controller, where an output error must develop before any corrective action can be initiated. The technique is particularly valuable where the system to be controlled includes time delays, or has a slow initial response. In such cases, with feedback control an output error will remain completely uncorrected for the duration of the time delay. The large phase shifts introduced in the closed loop also introduce a severe limitation on the feedback gain which may be used before instability occurs.

The disadvantage of the feedforward system is its inability to detect any deviation of the outputs from the required values due to errors in the control signal generation, or output disturbances produced by inputs which are inaccessible, or whose effects cannot be predicted.

## 3.2 DERIVATION OF CONTROLLER EQUATIONS

In order to utilise the advantages of both systems, a combination of feedforward and feedback controllers has been investigated, the object of these controllers having been outlined in Chapter 1.

A prediction $(dmb)_i$ of the output disturbances due to the A-B measurable but uncontrollable inputs is given by the convolution summation

$$(dmb)_i = \sum_{a=B+1}^{A} \sum_{j=0}^{Ka-1} (wmab)_j . (xa)_{i-j} \qquad (3.1)$$

where $(wmab)_j$ is the weighting sequence model of the input-output path a-b obtained as described in Chapter 2.

Estimates of the required control signals, $(xa)_i$ a = 1 ... B, are contained implicitly in the relationship

$$\sum_{a=1}^{B} \sum_{j=0}^{Ka-1} (wmab)_j . (xa)_{i-j} = -(dmb)_i \qquad (3.2)$$

Consider first the solution of (3.2) for the common case B = 1, with no time delay in the model (wm11). This yields the recurrence relationship

$$(x1)_i = -\frac{1}{(wm11)_0} \left\{ (dmi)_i + \sum_{j=1}^{K1-1} (wm11)_j . (x1)_{i-j} \right\} \qquad (3.3)$$

The right hand side of this equation may be computed at any time, since previous values of the control sequence will be known.

For the case B = 2, equation (3.2) yields the simultaneous equations

$$\sum_{j=0}^{K1-1} (wm11)_j \cdot (x1)_{i-j} + \sum_{j=0}^{K2-1} (wm21)_j \cdot (x2)_{i-j} = -(dm1)_i \quad (3.4a)$$

$$\sum_{j=0}^{K1-1} (wm12)_j \cdot (x1)_{i-j} + \sum_{j=0}^{K2-1} (wm22)_j \cdot (x2)_{i-j} = -(dm2)_i \quad (3.4b)$$

Solving for $(x1)_i$:

$$(x1)_i = \frac{(wm22)_0\left\{(dm1)_i + \sum_{j=1}^{K1-1}(wm11)_j \cdot (x1)_{i-j} + \sum_{j=1}^{K2-1}(wm21)_j \cdot (x2)_{i-j}\right\} - (wm21)_0\left\{(dm2)_i + \sum_{j=1}^{K1-1}(wm12)_j \cdot (x1)_{i-j} + \sum_{j=1}^{K2-1}(wm22)_j \cdot (x2)_{i-j}\right\}}{(wm21)_0 \cdot (wm12)_0 - (wm11)_0 \cdot (wm22)_0}$$

which on rearrangement gives

$$(x1)_i = \frac{(wm22)_0 \cdot (dm1)_i - (wm21)_0 \cdot (dm2)_i + \sum_{j=1}^{K1-1}\left\{(wm22)_0 \cdot (wm11)_j - (wm21)_0 \cdot (wm12)_j\right\}(x1)_{i-j} + \sum_{j=1}^{K2-1}\left\{(wm22)_0 \cdot (wm21)_j - (wm21)_0 \cdot (wm22)_j\right\}(x2)_{i-j}}{(wm21)_0 \cdot (wm12)_0 - (wm11)_0 \cdot (wm22)_0} \quad (3.5)$$

the expression for $(x2)_i$ being similar. Comparison with equation (3.3) indicates a basic similarity. The weighting sequences in the summations of equation (3.5) are, however, of a compound form. Once again, the algorithm can be implemented in principle. An extension of the above procedure to values of B greater than 2 is evidently possible, but becomes increasingly complex.

Due to limitations on computer availability, it has only been possible to carry out simulations for B = 1, and this case will be assumed throughout the remainder of this work, except where otherwise stated. Despite this limitation, there is still a large number of systems to which the control scheme is applicable.

The algorithms described in this section enable extremely accurate control to be achieved, since, under conditions of accurate modelling, the evaluation of the required control signals is virtually free from error.

## 3.3 FEEDFORWARD CONTROL APPLIED TO SYSTEMS WITH TIME DELAYS

The advantages of using feedforward control in such systems were outlined in Section 3.1. Consideration of equation (3.3) will indicate that some alteration is required to enable the algorithm to operate when the control path model contains a time delay, since, in this case $(wm11)_0$ will be zero. Let each path (wma1) contain a time delay ta such that

$$(wma1)_j = 0 \text{ for } 0 \leqslant j < ta \tag{3.6}$$

this condition having been set precisely by the method described in Section 2.3.

Equation (3.3) may be modified to enable $(x1)_i$ to be evaluated in this case

$$(x1)_i = -\frac{1}{(wm11)_{t1}}\left\{(dm1)_{i+t1} + \sum_{j=t1+1}^{K1-1} (wm11)_j.(x1)_{i-j}\right\} \tag{3.7}$$

which indicates that a prediction of (dm1) is required t1 samples ahead. Equation (3.1) gives this prediction as

$$(dm1)_{i+t1} = \sum_{a=2}^{Ka-1} (wma1)_j.(xa)_{i+t1-j} \tag{3.8}$$

which may be obtained precisely if

$$(wma1)_j = 0 \text{ for } j < t1 \tag{3.9}$$

i.e. if ta $\geqslant$ t1 for all a = 2 ... A.

Thus precise feedforward control is possible provided the time delay in the control path is at least as short as that in any of the disturbance paths. In the event that a control path cannot be chosen to satisfy this criterion, a prediction of $(dm1)_{i+t1}$ using a data extrapolation would have to be used, or the errors introduced by assuming equation (3.9) to be true would have to be tolerated. It is expected that this will seldom be necessary in practice.

## 3.4 STABILITY OF THE FEEDFORWARD CONTROLLER

The open-loop nature of the feedforward controller insures that it is not beset by the instability problems present when feedback is introduced. The three possible sources of instability when using the controller described above are (i) unpredictable divergence of the system outputs due to unmeasurable inputs, long term drift, inaccurate modelling etc., (ii) divergence within the control algorithm (3.3) itself and (iii) interaction between the control and identification procedures. The open-loop nature of the controller allows each factor to be considered in isolation. Note that the restriction on the weighting sequences in equation (1.4) precludes the use of the controller on an inherently unstable system, and hence of instability in the convolution algorithm (3.2).

The first of the above does not constitute instability in the conventional sense, and is inevitable in any pure feedforward system. The factor is represented by the term $(nsb)_i$ in equation (1.7). Since the feedback controller is introduced to minimise this term, a discussion will be deferred until Chapter 4. The second factor is best considered by examining the z-transform wm11(z) equivalent to the weighting sequence model of the control path $(wm11)_j$.

This is

$$wm11(z) = \sum_{j=0}^{K1-1} (wm11)_j . z^{-j} \tag{3.10}$$

and may be written in the above closed form by virtue of the restriction assumed in (1.4). Equation (3.2) may be rewritten in terms of z-transforms

$$wm11(z).x1(z) = -dm1(z) \tag{3.11}$$

Equation (3.3) is the solution of this for (x1), and is therefore equivalent to

$$x1(z) = -dm1(z) \frac{1}{wm11(z)} \tag{3.12}$$

The stability of (3.12), and hence of the control algorithm, requires that wm11(z) have no zeros outside the unit circle in the z-plane. Under conditions of exact modelling, this is equivalent to requiring the control path in the plant to be minimum phase at all times. When a time delay is present, equation (3.7) applies. In terms of z-transforms, this becomes

$$x1(z) = -dm1(z).\frac{z^{-t1}}{wm11(z)} \tag{3.13}$$

which is physically realizable, and will also be stable if the above requirements for the control path, without time delay, are met.

While the occurrence of non-minimum phase dynamics is relatively rare in a single path, they are more frequently encountered in the multi-variable case. The form of the compound weighting sequences in equation (3.5) shows why this is the case. The subtractions within the compound weighting sequences can readily give rise to a non-minimum phase characteristic. This has been observed in the continuous case by Rosenbrock[22]. Thus, applications to multi-output systems are expected to be less general.

In the above it has been assumed that the model weighting sequences used in the convolution and inversion algorithms are ideal. In practice, this will not be the case, the models being corrupted by a random perturbation due to the spurious correlations discussed in Chapter 2. These perturbations will degrade the control signal somewhat, the overall result being an increase in the (nsb) term at the system output. The possibility that this perturbation will lead to instability of the inversion algorithm cannot, however, be dismissed.

Given a particular corrupted weighting sequence, analytical methods (e.g. Jury's Criterion[21]) can be applied to confirm the stability of the inverse system. Such criteria give no information on the degree of stability however, and are therefore of little use in assessing the stability of the algorithm in general when a corrupted model is used. It has therefore been necessary to resort to simulation to demonstrate the stability of the system in a number of practical cases, results being presented in Section 3.6.

## 3.5 OPERATION OF FEEDFORWARD CONTROLLER WITH LIMITATIONS ON THE CONTROL SIGNAL

While the feedforward controller as described so far is an open-loop system, a closed-loop is formed by the introduction of the model updating procedure proposed in Chapter 1. This closed-loop introduces a possible source of long term instability building up over a number of identification cycles. Consider the system operating with a given set of corrupted models. These models will, as pointed out above, contribute to the unwanted $(nsb)_i$ term at the system output. If the result of a subsequent identification is a further increase in model corruption, then the combination of both procedures is clearly unstable. Once again, simulations, the results of which are given in Section 3.6, have been employed to investigate this factor.

The preceding discussion assumes that any control sequence $(xl)_i$ which is generated by the inversion algorithm can be applied to the plant. The plant dynamics are, however, usually linear over a limited range of input signal values only, and if large excursions of the input are demanded this linear region may be exceeded, and the accuracy of the feedforward control will be degraded. The identification procedure will also cease to function correctly in the non-linear region. If, however, a limit is imposed on the control signal amplitude, such that the non-linear region is never used, based on an apriori knowledge of the approximate extent of this non-linear region, then only a temporary deviation from the required output will occur, normal performance being resumed when the computed control signal is once again within the assumed linear region. Demands for large excursions in the control sequence $(xl)_i$ will result from excessively large or rapid variations in the disturbance prediction $(dml)_i$. In most applications, the control path can be expected to have a more rapid rise time than the disturbance paths, and a demand for a control signal exceeding the preset limits would be rare.

In the converse case of a system with a slow control path rise time, it can be expected that the amplitude limits will frequently be encountered, and as a result the transient performance of the controller will be degraded.

In practice however, it is in the use of the feedforward controller to implement changes in the system output that these limits will most frequently be encountered. Suppose it is desired that the system output follow a path defined by the sequence $(csl)_i$. Equation (3.7) may be modified to compute the ideal control sequence $(xl)_i^I$ thus

$$(xl)_i^I = -\frac{1}{(wm11)_{tl}}\left\{(dml)_{i+tl} - (csl)_{i+tl} + \sum_{j=tl+1}^{Kl-1} (wm11)_j.(xl)_{i-j}\right\} \tag{3.14}$$

If, as is typically the case, (cs1) experiences a step change, $(x1)_i^I$ will be extremely large, and the applied signal $(x1)_i$ will be equal to one or other of the preset limits. A temporary deviation of the output from its predicted trajectory, i.e.

$$(rm1)_i + (cs1)_i \tag{3.15}$$

will occur and would, if uncorrected, give rise to corruption of the identification. However a modified prediction can be computed as follows.

The applied input to the inversion algorithm is

$$(dm1)_{i+t1} - (cs1)_{i+t1} \tag{3.16}$$

In the event that limiting action is necessary, the effective input will differ from this; having calculated the applied control signal $(x1)_i$, the effective input $(v1)_i$ is given by

$$(v1)_i = -(x1)_i.(wm11)_{t1} - \sum_{j=t1+1}^{K1-1} (wm11)_j.(x1)_{i-j} \tag{3.17}$$

by rearranging equation (3.14). The term within the summation has already been calculated and hence little extra computation is involved in obtaining $(v1)_i$.

An estimate of the response to the test signals is then given by deducting the effects of all inputs from the system output. Since the combination of inverter and control path may be represented by a delay of t1 and a sign change, a prediction of the effect of this input is: $-(v1)_{i-t1}$.

The effect of the remaining inputs is, of course, $(dm1)_i$, so the response estimate for use in the cross-correlation procedure is obtained as

$$(re1)_i = (us1)_i - (dm1)_i + (v1)_{i-t1} \qquad (3.18)$$

An examination of Fig. 3.1 at this stage will indicate the inter-relationship of the variables in this, the completed feedforward system. The diagram indicates a system with one disturbance path only, since the multi-path system may be reduced to this if superposition is valid.

A simulation of the operation of the system in the presence of an amplitude limit on the control signal is given in the next section.

## 3.6 EXAMPLES OF SIMULATED FEEDFORWARD CONTROL

Fig. 3.1 shows a block diagram of the system used in these simulations. As in the multiple path identification test the parameters chosen for both paths were such as to allow the use of $K1 \doteq K2$.

Test 3.1: Operation of feedforward controller with ideal models

Parameters for the simulation were as follows.

Transfer function of path 1: $\dfrac{1}{s^2 + s + 1}$

Transfer function of path 2: $\dfrac{0.25}{s^2 + 0.194s + 0.25}$

Code generating polynomial: $z^{-8} \oplus z^{-6} \oplus z^{-5} \oplus z^{-1} \oplus 1$

Sampling interval: 0.2.

Noise-free models were obtained by allowing the system to run for 3 cycles of the test signal as in Test 2.1. The disturbance signal shown in Fig. 3.2(a) was then introduced at input (x2), the rms amplitude of this signal being 1, averaged over cycle 4. The resulting deviation of the output from the ideal test signal response is shown on a much enlarged scale in Fig. 3.2(b). The feedforward controller evidently functions extremely accurately in this ideal case, the small error present being due to the approximations inherent in the modelling technique discussed in Chapter 2. The slight discontinuity evident at 25.4 seconds in Fig. 3.2(b) is due to the introduction of the updated (wm21) at this point.

Test 3.2: <u>Operation of the feedforward controller with a noisy disturbance path model</u>

The above test was repeated using the corrupted (wm21) shown in Fig. 3.3(a) obtained with a 5:1 signal:noise ratio during the identification. The disturbance signal (x2) and the computed control signal (x1) are shown in Figs 3.3(b) and (c) respectively. Fig. 3.3(d) shows the output deviation (solid line) along with the predicted deviation in the absence of control (broken line). The use of the controller even with this badly corrupted model achieves a substantial reduction in output error. In particular, there is no evidence of instability in the system.

Test 3.3: <u>Operation of the feedforward controller with noisy control path model</u>

The corrupted (wm11) used in this test is shown in Fig. 3.4(a), the ideal model being indicated by the broken line. As in Test 3.2, the signal:noise ratio was 5:1 during the identification. Figs 3.4(b), (c) and (d) show the signals (x2), (x1) and the predicted and actual output deviations respectively.

Test 3.4: Implementation of changes in output with amplitude limit on control signal

In Section 3.5 it was pointed out that a demand for excessive control signal amplitude is most likely to be encountered when a demand for a rapid change in output level is made. Thus it is convenient to combine an illustration of output trajectory control with one of control signal amplitude limitation. Fig. 3.5(a) shows the unit step demanded output change (dotted) and the change which was implemented by the system when a limit of ±2.5 was imposed on the control signal (x1). The output component due to the signal has been omitted for clarity. System parameters were as in Test 3.1. The control signal generated is seen in Fig. 3.5(b). The method is effective in providing a rapid rise time with very little overshoot.

Test 3.5: Simultaneous identification and control

If the combined identification and control scheme proposed in Chapter 1 is to be practicable it must be demonstrated that:

(i) Starting with a set of erroneous models obtained either with the system in an uncontrolled condition, or by some analytical means, the identification will converge over subsequent cycles of the test signal towards the ideal noise-free model.

(ii) The identification will track gradual changes in system parameters.

Since investigation of these factors requires that the system be run over many cycles of the test signal it was found to be prohibitively expensive to carry out a thorough examination of this problem on the general purpose machine employed for these simulations. The following test indicates that the system presented here requires further development in order to satisfy the above criteria for simultaneous identification and control.

The system of Test 3.1 was run for 3 cycles with a disturbance present at input (x2), resulting in corrupted models of both paths being obtained. These are indicated by the continuous lines in Figs 3.6(a) and (b). The feed-forward controller was operational during the fourth cycle, the updated models obtained over this cycle being indicated by the broken lines in Figs 3.6(a) and (b). These results show a change in the nature of the corruption, the higher frequency components being conspicuous, but no reduction in corruption amplitude. As this problem has also been encountered in the feedback controller a discussion is deferred until Chapter 6.

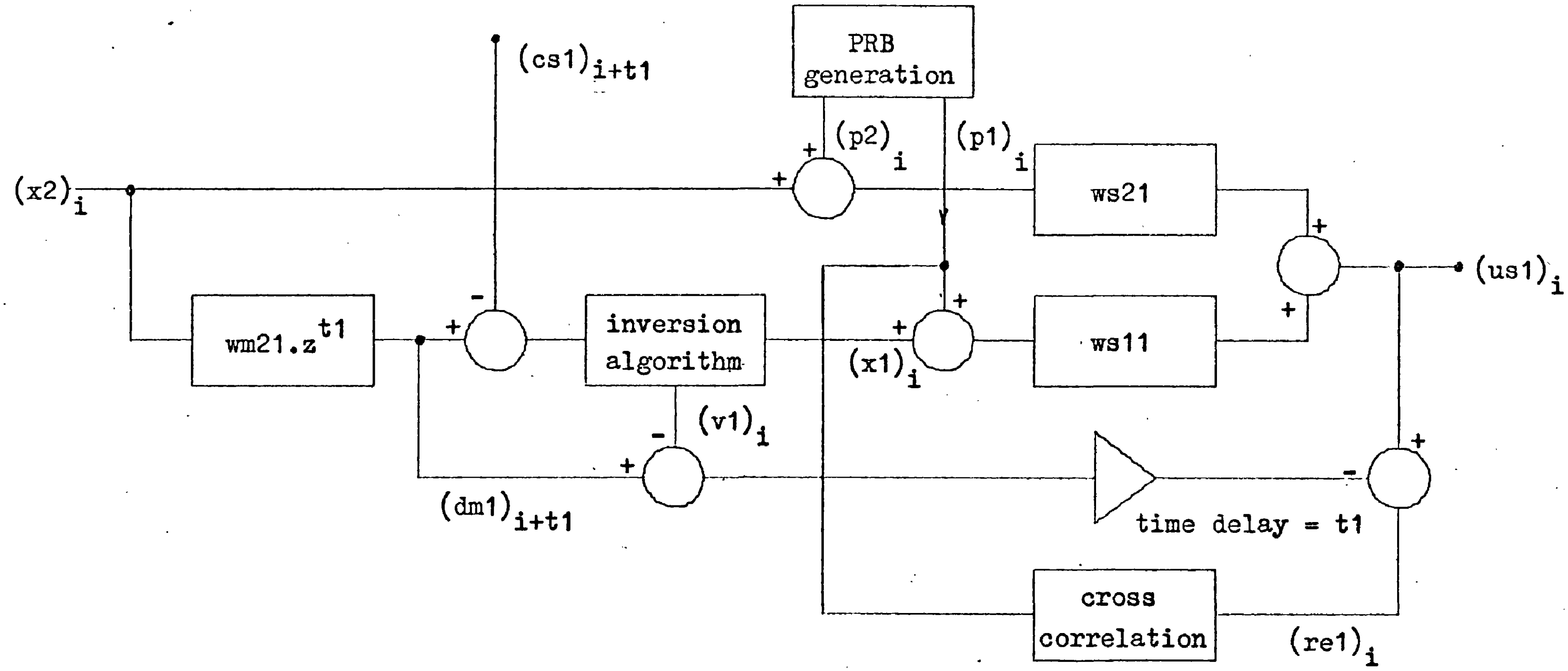



Fig. 3.1: Block Diagram of System Used for Feedforward Control Simulations.

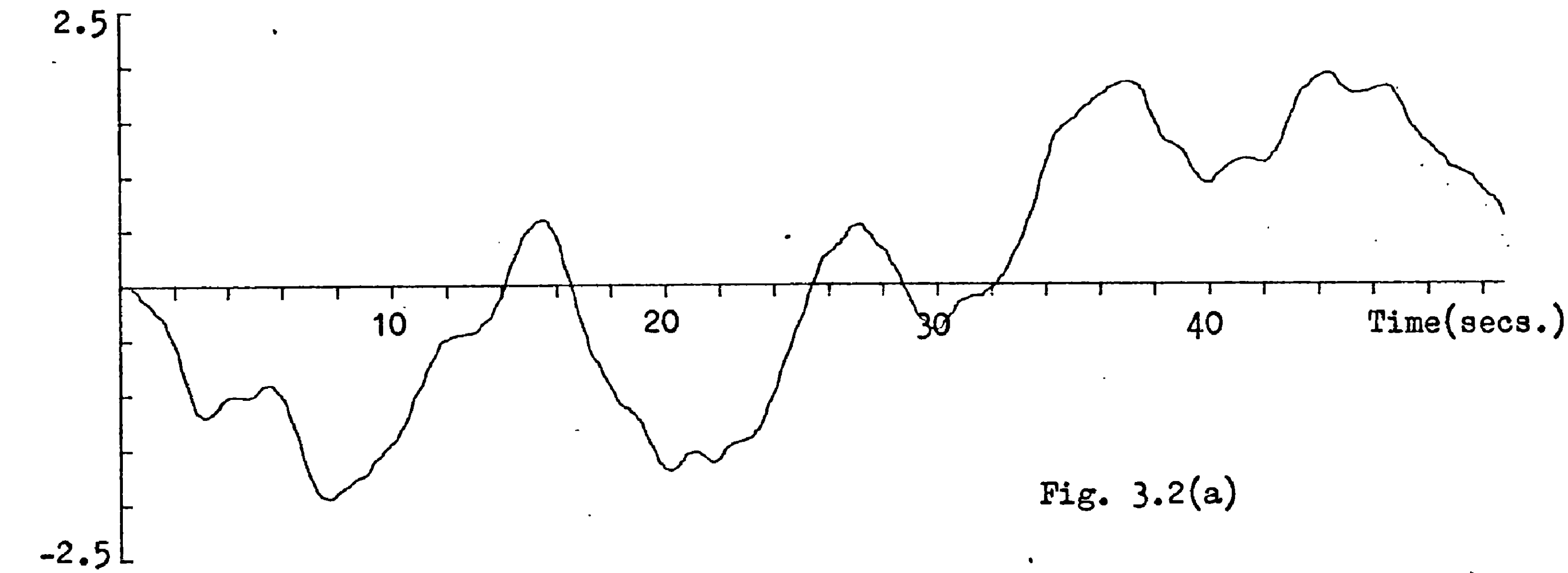



Fig. 3.2(a)

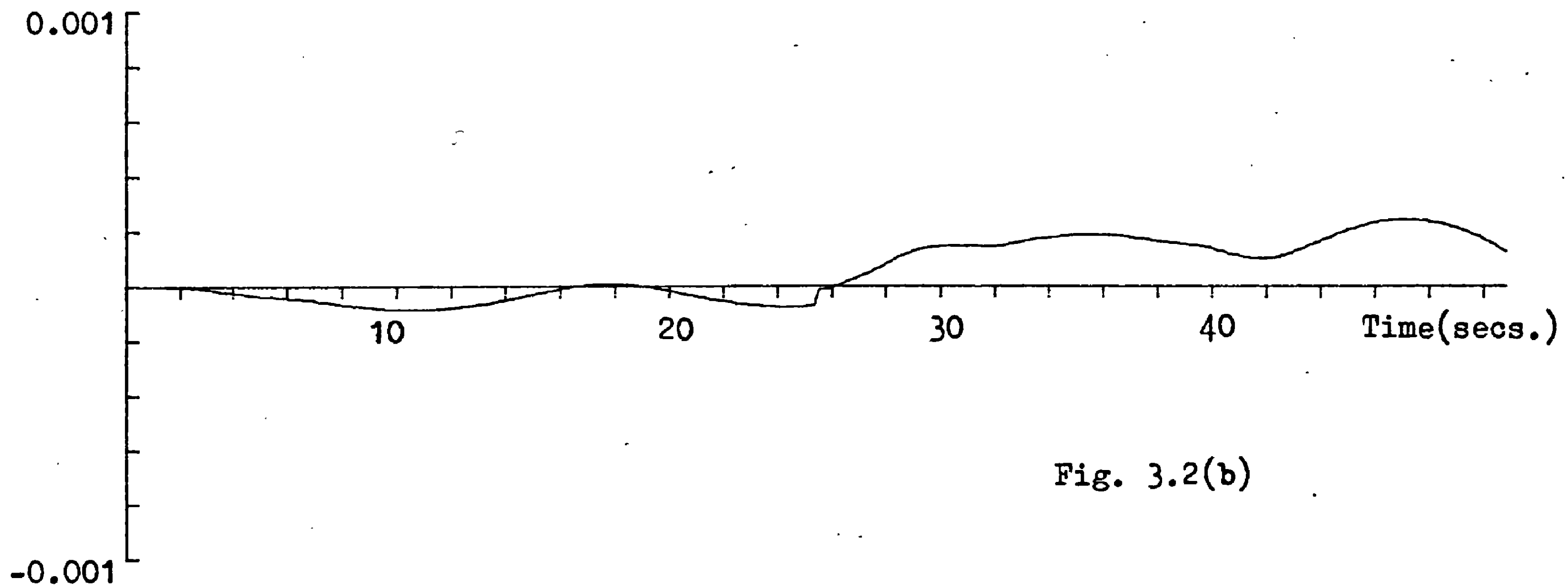



Fig. 3.2(b)

Fig. 3.2: (a) input signal (x2), (b) output deviation - Test 3.1

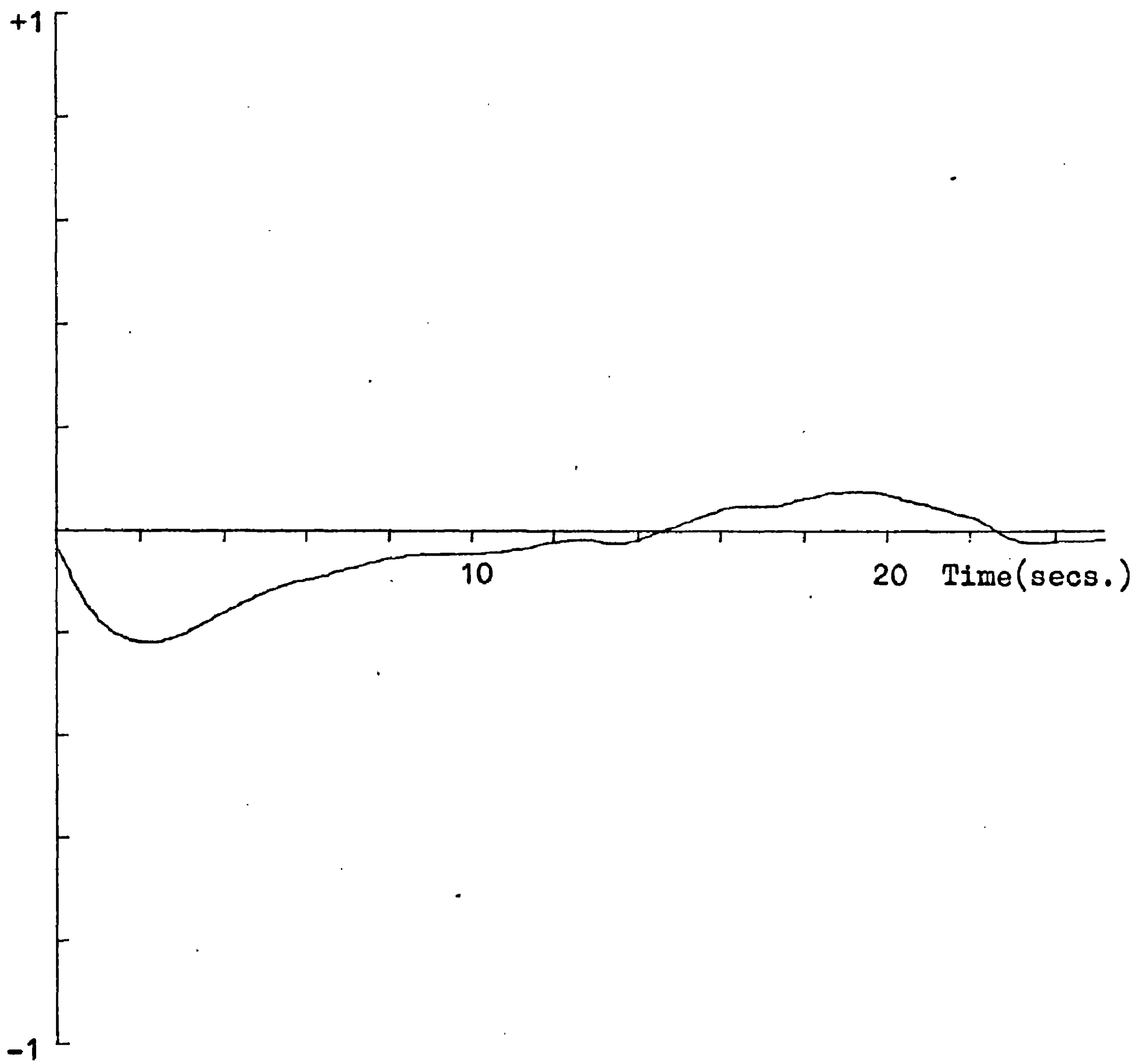
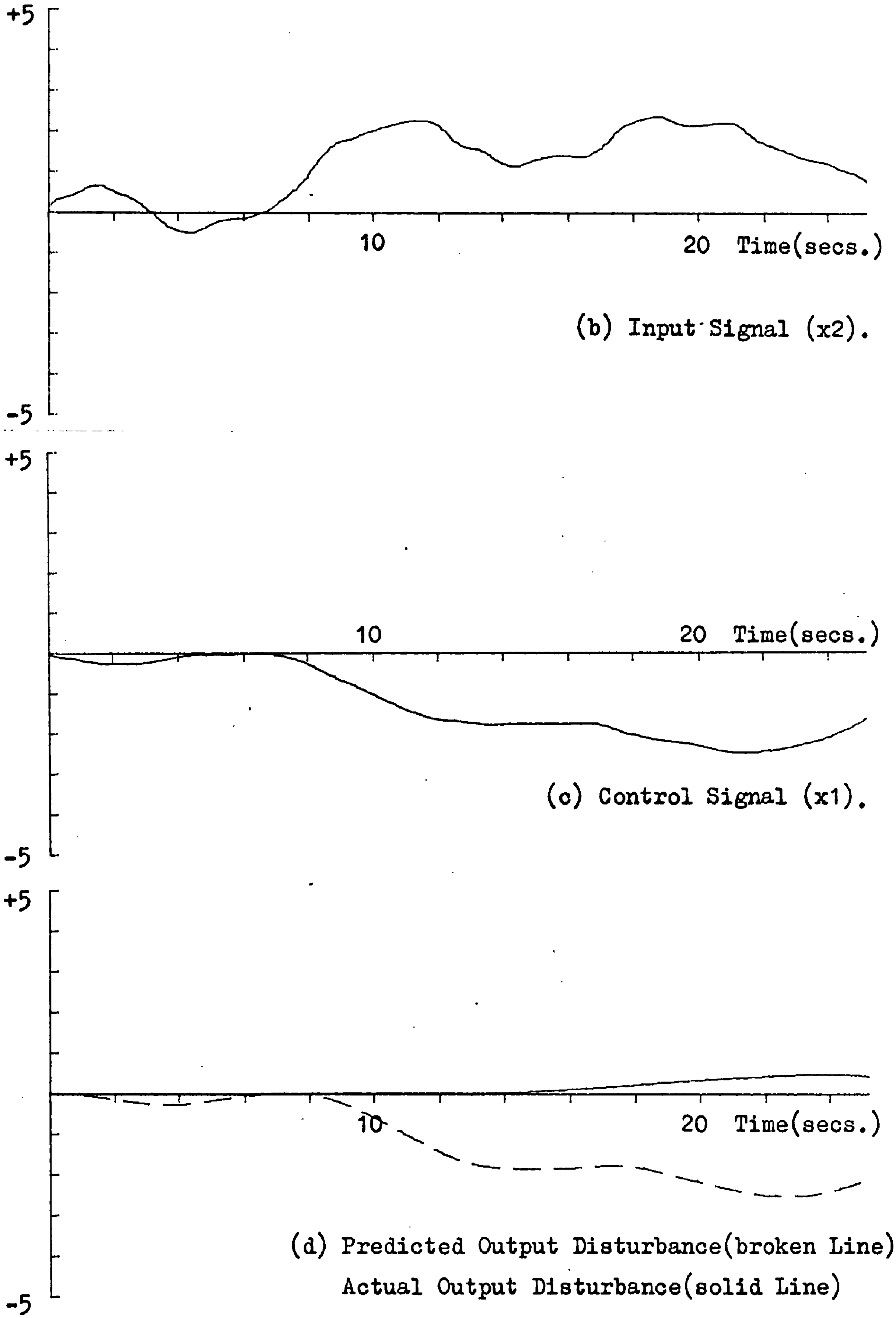



Fig. 3.3(a): Corrupted Model (wm21) used in Test 3.2.




(b) Input Signal (x2).

(c) Control Signal (x1).

(d) Predicted Output Disturbance(broken Line)
Actual Output Disturbance(solid Line)

Fig. 3.3(b),(c),(d): Results of Test 3.2

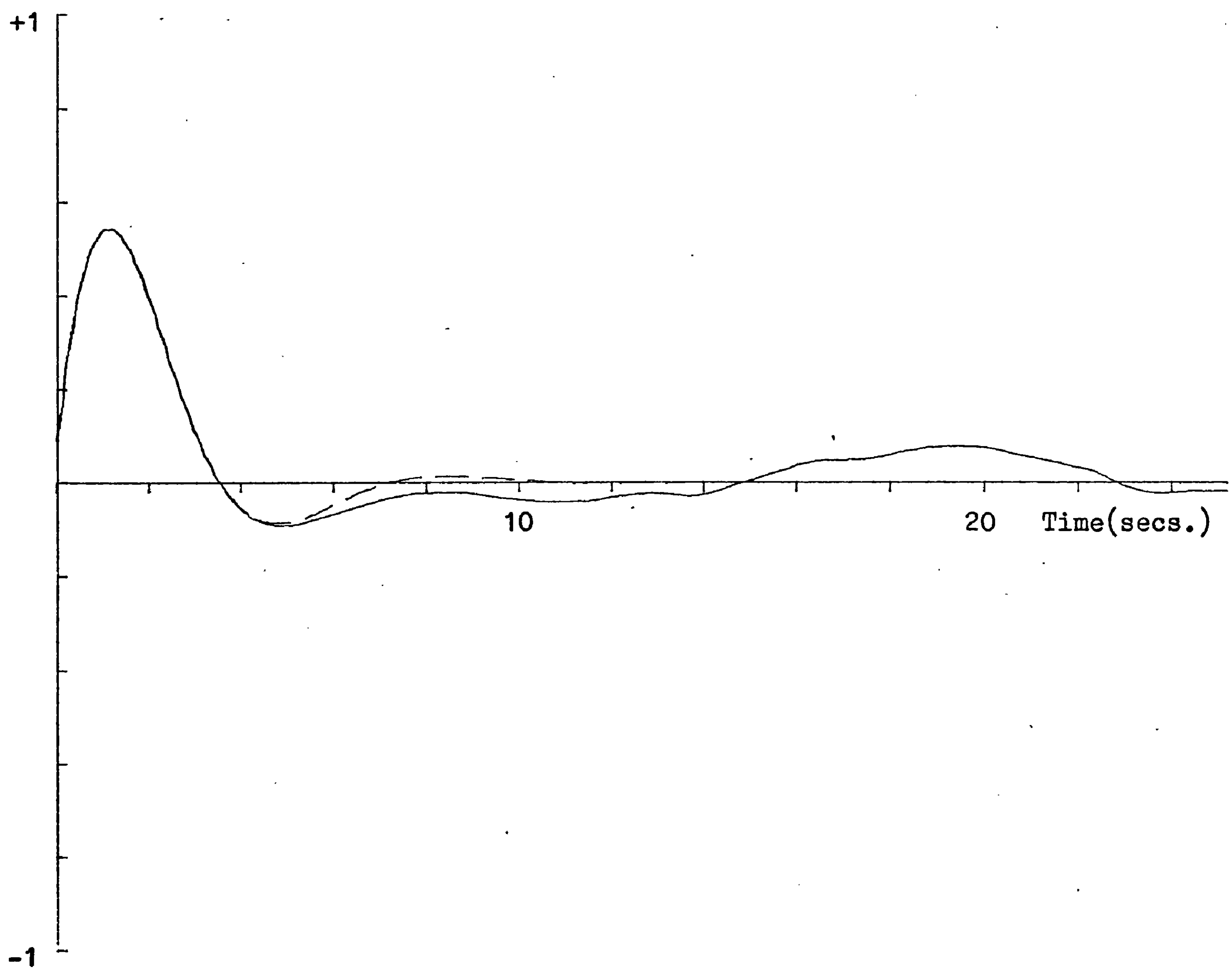



Fig. 3.4(a): Corrupted Model (wm11) used in Test 3.3

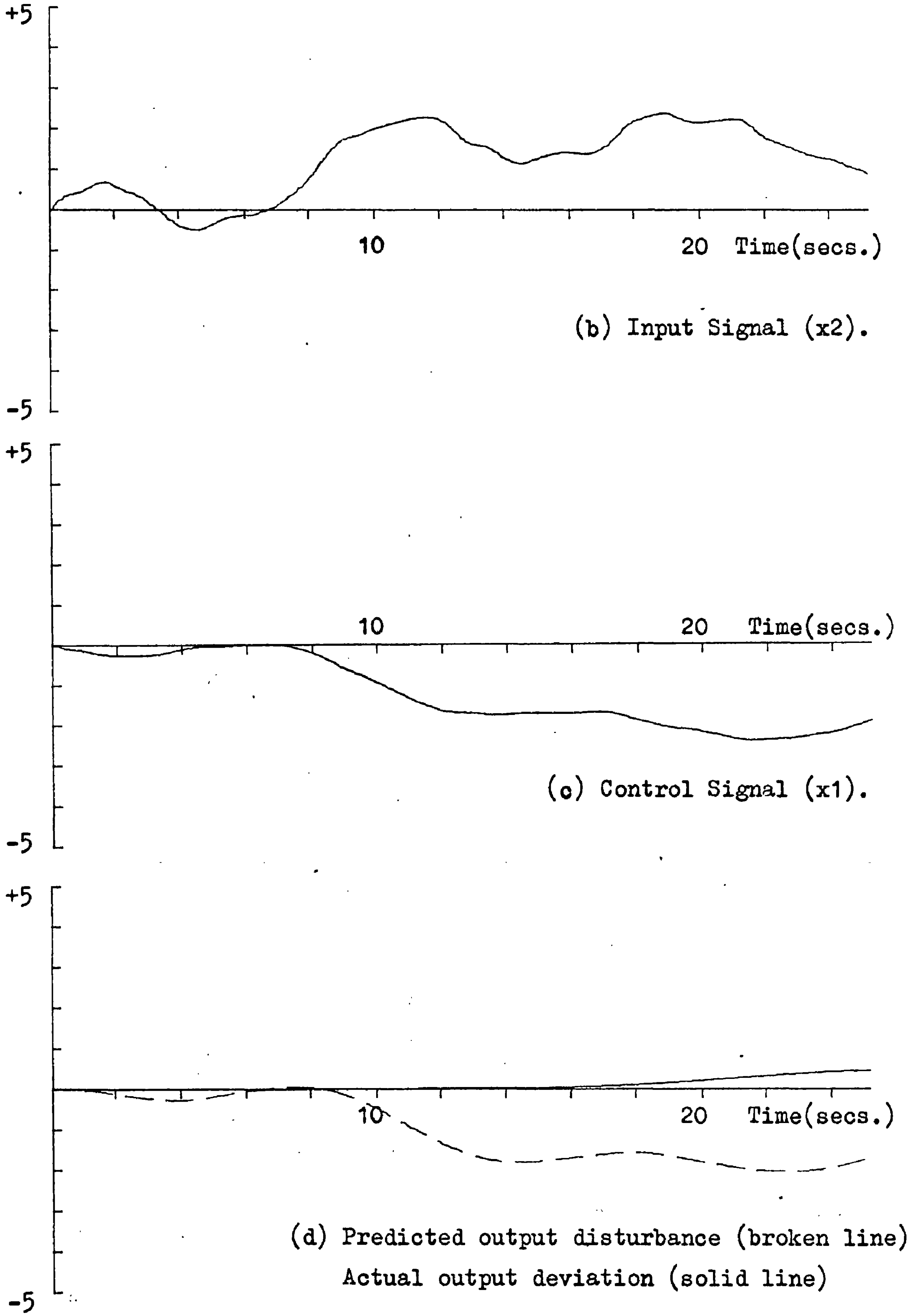



Fig. 3.4(b),(c),(d): Results of Test 3.3.

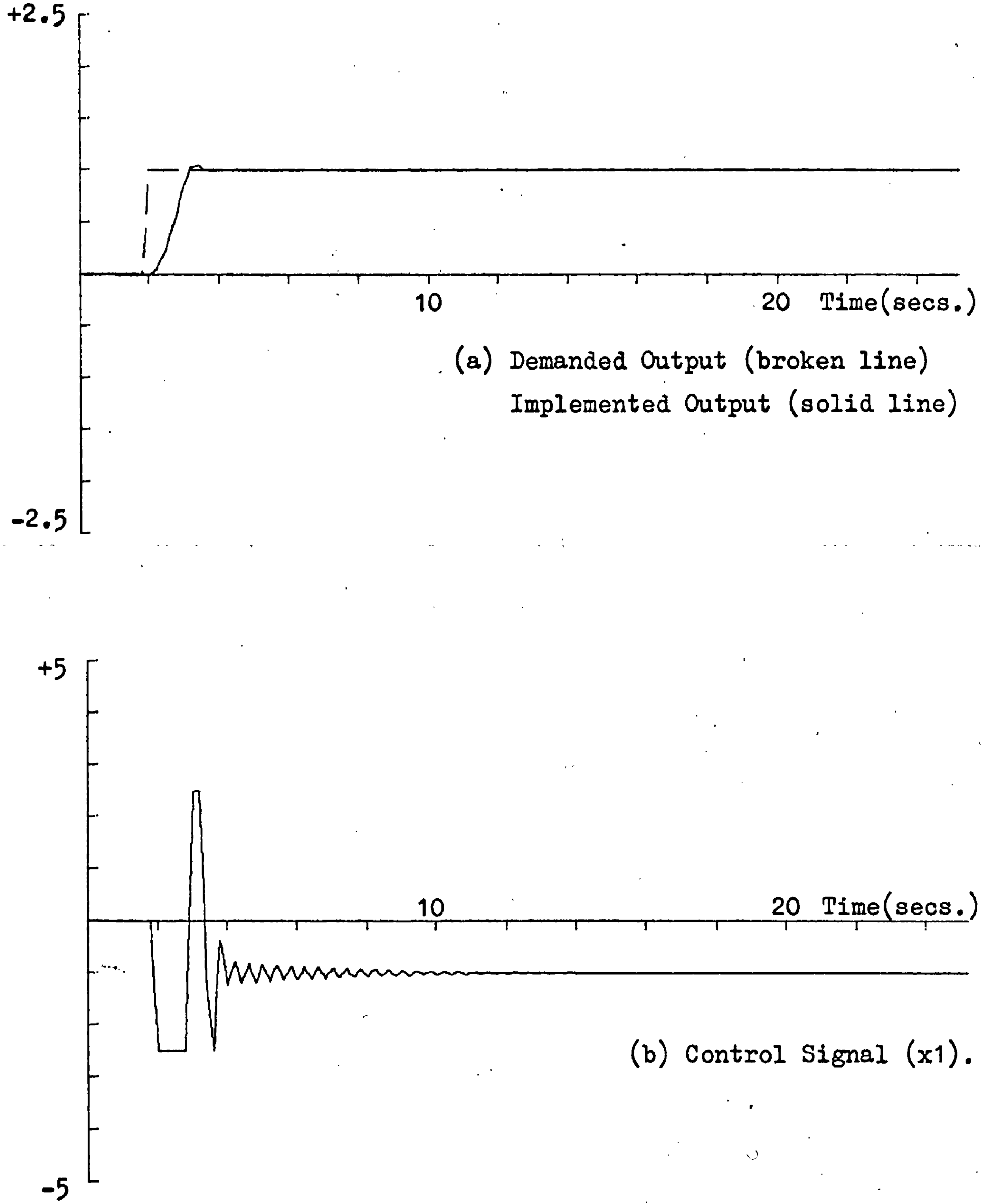
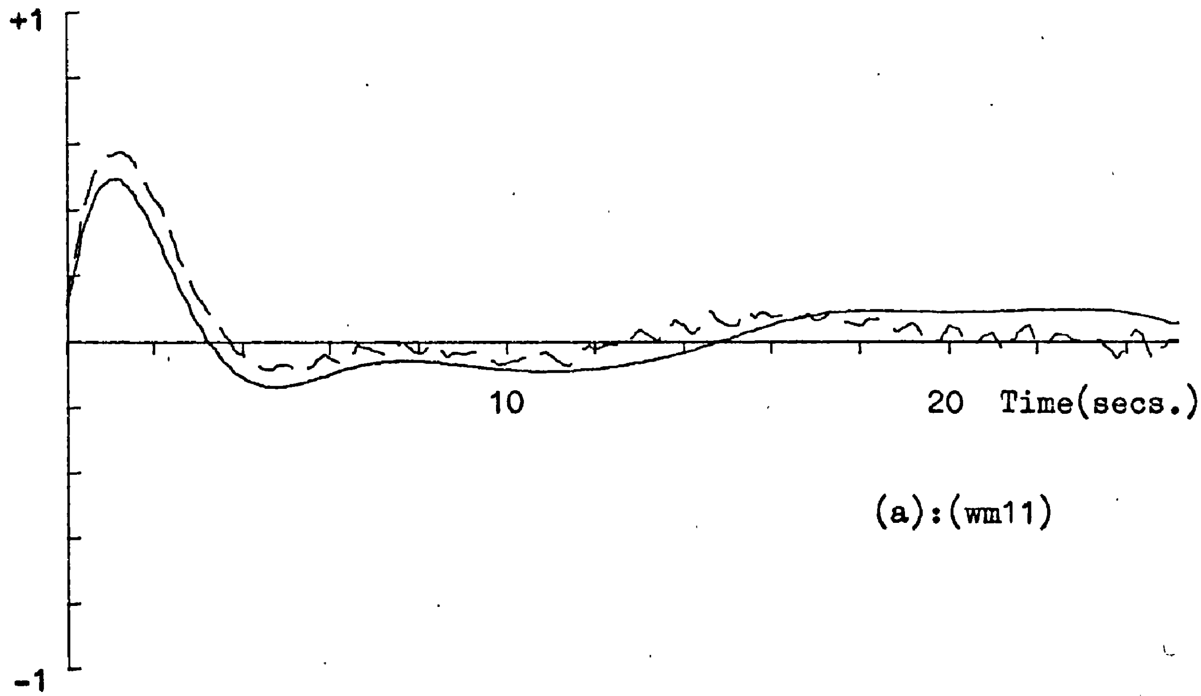



Fig. 3.5: Operation with Amplitude Limit on (x1) - Test 3.4.



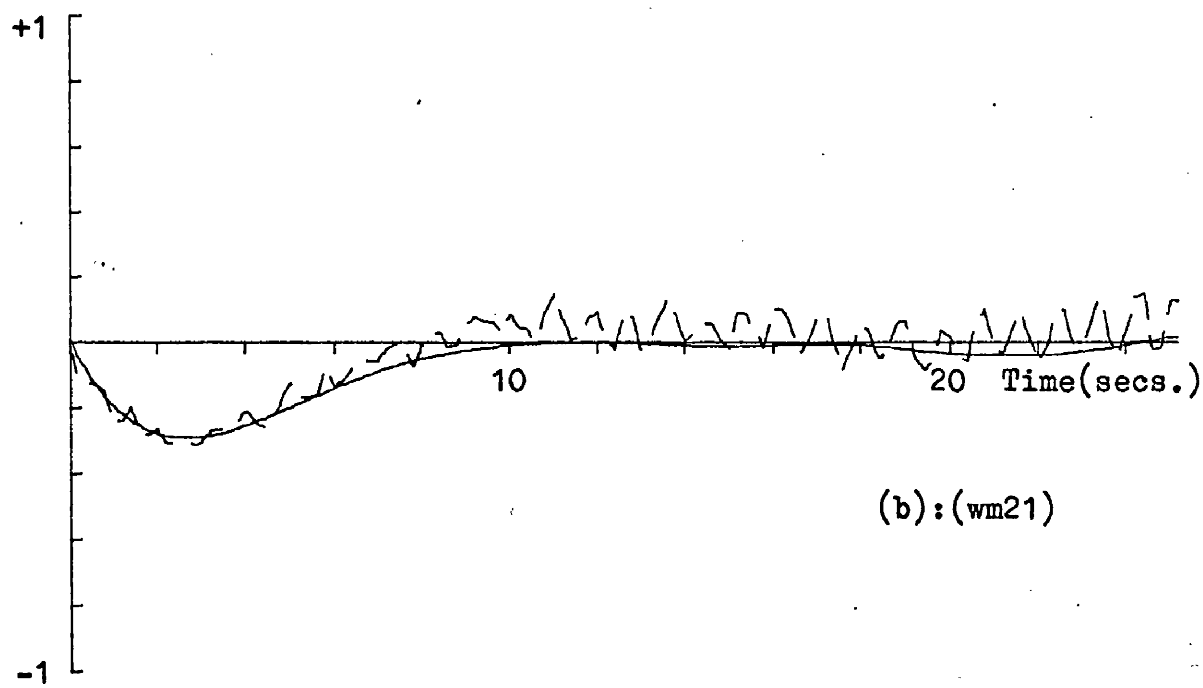




Fig. 3.6: Models Obtained During Simultaneous
Identification and Control - Test 3.5
Solid lines: Initial Identification.
Broken lines: Subsequent Identification.

# C H A P T E R 4

# THE FEEDBACK CONTROLLER

## 4.1 PRELIMINARY CONSIDERATIONS

The necessity for a feedback controller to remove the unwanted output component $(nsb)_i$ in equation (1.7) has been demonstrated in Chapter 1. Conventional feedback control theory could be applied to the design of the feedback path, the resulting transfer function being implemented either as the equivalent discrete algorithm using the control computer, or as a number of inter-connected analog elements. In either case the design procedure would be carried out off line, based on some assumed, or experimentally determined form of system dynamics. The process would involve a considerable amount of manual calculation, and would have to be repeated for each different plant on which the complete control system was being implemented. This approach may well be acceptable in many cases, but lacks the generality inherent in the feedforward system described in the previous chapter.

An alternative approach, based on the weighting sequence model of the control path, and employing a number of the algorithms developed for the feedforward controller, is described here. As with the feedforward controller, the required control sequences are generated by the control computer. The case of a single output system is once again the only one considered.

The introduction of feedback necessitates some modification to the identification procedure, since in the absence of any corrective action the test signal response would be fed back, and control action would attempt to eliminate this term. The methods considered for overcoming this problem as described in Section 4.4.

## 4.2 DERIVATION OF CONTROLLER EQUATIONS

From equation (1.7), the unwanted term $(ns1)_i$ is given by:

$$(ns1)_i = (us1)_i - (rs1)_i - (zs1)_i \tag{4.1}$$

An estimate of this term is:

$$(nm1)_i = (us1)_i - (rm1)_i - (zm1)_i \tag{4.2}$$

where

$$(rm1)_i = \sum_{a=1}^{A} \sum_{j=0}^{Ka-1} (wma1)_j . (pa)_{i-j} \tag{4.3}$$

and

$$(zm1)_i = \sum_{a=1}^{A} \sum_{j=0}^{Ka-1} (wma1)_j . (xa)_{i-j} \tag{4.4}$$

The estimate (4.2) may readily be computed (vide infra), and when added to the inversion algorithm input required by the feedforward controller (equation (3.3)), the ideal input to this algorithm is:

$$(dm1)_i - (cs1)_i + (nm1)_i \qquad (4.5$$

The result, given accurate models, is the complete removal of the unwanted term. This ideal situation is not, however, encountered in practice. Two factors affect the accuracy of the feedback controller: (i) corrupted models lead to inaccuracies in the generated control sequences; (ii) in all real systems, the discrete representation exhibits a time delay of at least one sampling interval, since the continuous system being identified does not have an instantaneous response. Thus equation (3.7) must be applied, and a prediction $(nml)_{i+tl}$ is required. In this case, however, an accurate prediction similar to equation (3.8) is not available, and reliance must be placed on data extrapolators. Since the unwanted term will in many cases be a gradual drift, the predictions provided by zero- and first-order extrapolation, defined respectively by

$$(nmlp)_i = (nml)_i \tag{4.6a}$$

and

$$(nmlp)_i = (nml)_i + tl\{(nml)_i - (nml)_{i-1}\} \tag{4.6b}$$

where $(nmlp)_i$ is the prediction of $(nml)_i$ tl ahead, can be expected to prove satisfactory for time delays of short duration.

Some reduction in the computation of equations (4.3) and (4.4) is possible. Since $(pa)_i = \pm 1$, the multiplications in equation (4.3) may be reduced to addition and subtraction. Accordingly, a special purpose algorithm to perform this specific computation has been introduced in the control program (see Chapter 5).

The overall transfer function of the inversion algorithm and control path, assuming that the model (wml1) is ideal, will in terms of z transforms be:

$$-z^{-t1} \qquad (4.7)$$

Thus the term $(zm1)_i$ of equation (4.4) may be rewritten:

$$(zm1)_i = -(v1)_{i-t1} + (dm1)_i \qquad (4.8)$$

eliminating the need to carry out an additional convolution. To achieve this simplification, it is necessary to know the effective inverter input $(v1)_i$, and it is for this reason that the limiting action described in Section 3.5 was applied as in equation (3.17), and not by simply limiting the amplitude of the control signal $(x1)_i$ subsequent to its computation.

Examination of the block diagram, Fig. 4.1 at this stage will clarify the inter-relationships of the various algorithms and variables so far introduced. In this diagram, the blocks associated with the identification have been omitted to avoid unnecessary complication. The disturbance paths (wsa1) a = 2 ... A have been represented by one path (ws21) as was done in Chapter 3.

## 4.3 STABILITY OF THE FEEDBACK CONTROL LOOP

Provided the limiting action introduced in Section 3.5 is not operational, the closed loop system of Fig. 4.2(a) may be extracted from the block diagram Fig. 4.1, and since the equations describing each block are linear, an analysis using z-transform methods may be performed. The time delay tl may be combined with the predictor to yield the further simplified system of Fig. 4.2(b), in which the transfer function of the feedback element is:

$$G(z) = \frac{1}{1 - z^{-tl}} \tag{4.9a}$$

for the zero order extrapolator defined by equation (4.6a)

or

$$G(z) = \frac{1 + tl - tl.z^{-1}}{1 - (1 + tl)z^{-tl} + tl.z^{-tl-1}} \tag{4.9b}$$

for the first order extrapolator of equation (4.6b).

Assuming the inverter to be based on an exact model, equation (4.7) applies, and the loop transfer function is:

$$\frac{1}{1 - z^{tl}} \tag{4.10a}$$

for the zero order extrapolator, and

$$\frac{tl - (1 + tl)z}{tl - (1 + tl)z + z^{tl+1}} \tag{4.10b}$$

for the first order extrapolator.

Nyquist's Criterion may be applied by plotting the loci of these functions for values of z lying on the unit circle, remembering that, since there is no inversion at the summing junction, it is encirclements of the (+1,0) point which are of interest.

Let the unit circle be defined by

$$z = e^{j\theta} \tag{4.11}$$

then the locus of equation (4.10a) is

$$\tfrac{1}{2}\left\{1 + j\,\frac{\sin\,(t1.\theta)}{1 - \cos\,(t1.\theta)}\right\} \tag{4.12}$$

which is a straight line parallel to the imaginary axis, passing through the ($\tfrac{1}{2}$,0) point, for all t1.

The locus of equation (4.10b) is dependent on t1, and is shown for t1 = 1 and 2 in Figs 4.3(a) and (b) respectively. These graphs were plotted by computer direct from the expression (4.10b).

The stability gain margin using the zero order extrapolator is 2, while in the case of the first order extrapolator it is 1.333 for t1 = 1, and 1.231 for t1 = 2. While these margins do not have the same significance as in a conventional feedback controller, in that the feedback gain is not a design variable, they do indicate that the system stability is not critical. As is to be expected the introduction of the first order extrapolator, with its greater phase shifts; especially with long time delays, has a detrimental effect on the stability margin. The validity of the above analysis was checked by running simulations in which the feedback gain was intentionally increased to lie just above or below the critical figure. These tests confirmed the above results.

The stability of the closed loop in the presence of a corrupted control path model cannot be investigated analytically for the reasons given in Section 3.4. Thus it has once again been necessary to resort to simulation to asses the stability of the system in this case.

## 4.4 IDENTIFICATION IN THE PRESENCE OF FEEDBACK CONTROL

The introduction of the feedback loop will clearly affect the operation of the identification procedures described in previous chapters, since the test signal response term $(rs1)_i$ cannot be distinguished from the unwanted noise $(ns1)_i$ and will therefore be modified by control action. Two methods were considered to achieve identification in the presence of feedback.

(i) Removal from the output signal of an estimate $(rm1)_i$ of the test signal response as described by equation (4.3). It is shown below that, in case of an inaccurate model, the result of the identification is a closer approximation to the required model.

(ii) Extraction from the closed loop system model, obtained by allowing the test signal response to be fed back, of the weighting sequence of the forward path, based on an arbitrarily accurate knowledge of the feedback path, since this is a numerical algorithm.

A simplified block diagram, Fig. 4.4(a) illustrates the relationships pertaining to the analysis of method (i). The following analysis assumes the zero-order extrapolator (equation (4.6a)) to be used.

The z transform formulation is employed, a function s(z) being written as s. From the diagram

$$\text{us1} = \sum_{a=1}^{2} \text{wsa1.pa} + \text{ws11.x1} \tag{4.13a}$$

$$\text{rm1} = \sum_{a=1}^{2} \text{wma1.pa} \tag{4.13b}$$

$$\text{nm1} = \text{us1} - \text{rm1} - \text{zm1} \tag{4.13c}$$

$$\text{v1} = \text{nm1p} = \text{nm1} \tag{4.13d}$$

$$\text{x1} = -\text{v1/wm11.z}^{\text{t1}} \tag{4.13e}$$

$$\text{zm1} = -\text{v1.z}^{-\text{t1}} \tag{4.13f}$$

$$\text{re1} = \text{us1} - \text{zm1} \tag{4.13g}$$

$$= \sum_{a=1}^{2} \text{wsa1.pa} + \text{v1.z}^{-\text{t1}} \left(1 - \frac{\text{ws11}}{\text{wm11}}\right) \tag{4.13h}$$

(by (a), (e), and (f)).

Now $\text{v1} = \text{us1} - \text{rm1} - \text{zm1}$ (by (c) and (d))

$$= \text{re1} - \sum_{a=1}^{2} \text{wma1.pa} \text{ (by (b) and (g))} \tag{4.13j}$$

Hence

$$\text{re1}\left\{1 + \text{z}^{-\text{t1}}\left(\frac{\text{ws11}}{\text{wm11}} - 1\right)\right\} = \sum_{a=1}^{2} \text{wsa1.pa} - \sum_{a=1}^{2} \text{wma1.pa.}\left(1 - \frac{\text{ws11}}{\text{wm11}}\right)\text{z}^{-\text{t1}}$$

(4.13k)

In the event that the control path is an ideal model wm11 = ws11, and by inspection of equation (4.13k), it can be seen that wsa1 will be identified exactly. Otherwise, assume that the models are in error by the small quantities $\Delta 11$ and $\Delta 21$ such that:

$$\begin{aligned} ws11 &= wm11 + \Delta 11 \\ ws21 &= wm21 + \Delta 21 \end{aligned} \tag{4.13m}$$

Equation (4.13k) then gives:

$$\begin{aligned} re1\left(1 + z^{-t1}\frac{\Delta 11}{wm11}\right) &= p1(ws11 + \Delta 11.z^{-t1}) \\ &+ p2\left(ws21 + \Delta 11\,\frac{wm21}{wm11}\,z^{-t1}\right) \end{aligned} \tag{4.13n}$$

Correlation between re1 and p1 will identify the system:

$$\frac{ws11 + \Delta 11.z^{-t1}}{1 + z^{-t1}.\frac{\Delta 11}{wm11}}$$

which reduces to

$$wm11 + \frac{\Delta 11}{1 + z^{-t1}.\frac{\Delta 11}{wm11}}$$

$$\simeq wm11 + \Delta 11 \text{ since } \Delta 11 \text{ is small.}$$

Thus an approximate identification of the control path is achieved.

Similarly, correlation between re1 and p2 yields the identification of:

$$\frac{ws21 + \Delta 11 \frac{wm21}{wm11} z^{-t1}}{1 + z^{-t1} \cdot \frac{\Delta 11}{wm11}} = wm21 + \frac{\Delta 21}{1 + z^{-t1} \cdot \frac{\Delta 11}{wm11}}$$

$$\simeq wm21 + \Delta 21 \text{ as required.}$$

The second proposed method is analysed with reference to the block diagram, Fig. 4.4(b).

Let the feedback path (below the broken line) be represented by G(z). Then the overall transfer function is:

$$W = \frac{ws11}{1 + ws11.G} \tag{4.14a}$$

ws11 may then be extracted by:

$$ws11 = \frac{W}{1 - W.G} \tag{4.14b}$$

which is a positive feedback system having the same feedback path as shown in Fig. 4.4(b). Application of a unit pulse to the simulated system, (4.14b), yields the required weighting sequence $(ws11)_j$. It can be shown that, having determined $(ws11)_j$ in this way, the weighting sequences $(wsa1)_j$ may be determined in a similar fashion.

The method was however found to be impractical when it was implemented in simulations. The attenuation of the response terms was such that the model of the system defined by equation (4.14a) exhibited a very poor signal:noise ratio, and, using this model the implementation of equation (4.14b) invariably proved unstable.

The method adopted in the computer program was therefore the first of those discussed in this section.

## 4.5 EFFECT OF BIASSED MODELS ON CONTROLLER STABILITY

Prior to this point, it has been assumed that the methods of Section 2.2 have been effective in removing the unavoidable model bias discussed in that section. As was seen in Test 2.3, it will not always be possible to estimate this bias accurately in the presence of noise, and therefore if the path gains are not known, or are liable to change, a bias term will exist. That this can lead to instability may be shown as follows:

Consider an attempt to realize the inverse of a model based on a system w(s) where the model has a bias $\Delta$, but is otherwise ideal. Let $w(s) = n(s)/d(s)$ where n(s), d(s) are polynomials in s. Then the inverse of the biassed model will be:

$$\frac{1}{\frac{n(s)}{d(s)} + \Delta}$$

$$= \frac{d(s)}{n(s) + \Delta d(s)} \tag{4.15}$$

It can be shown (23) that for the roots of a polynomial to lie in the left-half plane, all its coefficients must have the same sign. For a positive gain system with left-half plane zeros the signs of d(s) and n(s) will be the same, and in the event that $\Delta$ is negative, the possibility clearly exists that some poles of the inverse system (4.15) will lie in the right-half plane, especially since, in many cases, the order of d(s) will be greater than that of n(s). A simulation demonstrating this factor is given in the next section. The bias term will be proportional to the error in estimation of the term $\sum_{a=1}^{A}$ gal in equation (2.12), and accordingly a term has been introduced in the identification program to allow for the deduction of the bias which would arise in the presence of the maximum expected error in this term, based on apriori knowledge of the system. The removal of excessive bias has little effect on system performance. The alternative would be to utilise the 3-level test signal discussed in Section 2.2.

## 4.6 EXAMPLES OF SIMULATED FEED-BACK CONTROL

Fig. 4.1 shows a block diagram of the system used in these simulations.

Test 4.1: Operation of fecdback controller with ideal models

Parameters for the simulation were as follows.

Transfer function of path 1: $\dfrac{1}{s^2 + s + 1}$

Transfer function of path 2: $\dfrac{0.25}{s^2 + 0.194s + 0.25}$

Code generating polynomial: $z^{-8} \oplus z^{-6} \oplus z^{-5} \oplus z^{-1} \oplus 1$

Sampling interval: 0.2.

The system was allowed to run for 3 cycles of the test signal without control to obtain noise-free models. The signal shown in Fig. 4.5(a) was then introduced as $(nsl)_i$, the r.m.s. amplitude of this signal being approximately 1.2, averaged over the first half of cycle 4. Fig. 4.5(a) also shows the resulting deviation from the ideal output using a zero order data extrapolator (equation (4.6a)) in the feedback loop. The r.m.s. value of this deviation is 0.09. The simulation was repeated using a first-order extrapolator (equation (4.6b)), the resulting deviation, shown in Fig. 4.5(b), having an r.m.s. amplitude of 0.025. The control signals $(xl)_i$ generated in these simulations are shown in Fig. 4.5(b) and (d) for the zero- and first-order extrapolators respectively. The obvious improvement obtained using the latter has led to its adoption for all subsequent tests, unless otherwise stated.

Test 4.2: Operation of feedback controller with noisy model

Test 4.1 was repeated, using the noisy model (wm11) shown in Fig. 3.4(a). The noise signal (ns1) superimposed on the output, and the resulting output deviation are shown by the broken and solid lines respectively in Fig. 4.6(a), the r.m.s. of the deviation in this case being 0.06, compared with an r.m.s. value of 0.25 for the noise signal. It is therefore evident that the noisy model has an extremely detrimental effect on the system's performance. The r.m.s. noise attenuation of only x4.17 being much inferior to the figure of x48 obtained in Test 4.1 with a noise-free model. The effect of the noisy model on the control signal (x1) is also very marked, this signal now assuming a relatively large amplitude alternating character.

Test 4.3: Effect of model bias on inversion algorithm stability

The analysis of Section 4.5 showed that, in the event of incorrect model bias removal, the possibility of instability in the inversion algorithm exists. A simulation with a control path transfer function of $2/(s^2 + s + 1)$, but with other parameters identical to those of Test 4.1 was run. This results in a negative bias of the positive gain control path model. As predicted in Section 4.5, the results of this test show that the system is unstable in this case. The control signal, shown in Fig. 4.7(b), rapidly diverges to the preset limits of ±500, causing a temporary large oscillation in the output signal deviation (Fig. 4.7(a)). Having reached these preset limits, the controller becomes ineffective, the output deviation oscillating about the superimposed noise signal, which is not attenuated.

The test was repeated with a control path transfer function of $0.5/(s^2 + s + 1)$ resulting in a positive bias on the control path model. In this case, as seen in Fig. 4.7(c), the system is both stable and effective.

Test 4.4: <u>Effect of time delay in control path</u>

A time delay of five sampling intervals was introduced in the control path wsll. Test 4.1 was repeated, using the first order data extrapolator in the feedback loop. The superimposed output noise, and output deviation are shown in Fig. 4.8(a), and the applied control signal in Fig. 4.8(b). The system does not show any sign of instability, but is much less effective than the system without time delay simulated in Test 4.1.

Test 4.5: Identification in the presence of feedback control

Starting with ideal models (solid lines, Fig. 4.10) as in Test 4.1, the system was allowed to run for two cycles of the test signal, wm11 being updated at the end of each cycle, and wm21 being updated at the mid-point of each cycle. The results, shown in Fig. 4.9, clearly indicate the progressive degradation of the system as the corruption of both models obtained during the cycle 0-51 seconds, superimposed on the ideal models. Those obtained during the cycle 51-102 seconds were so corrupted that graphing them was pointless. As with the feedforward controller, the system is clearly deficient in this area. This deficiency will be discussed further in Chapter 6.

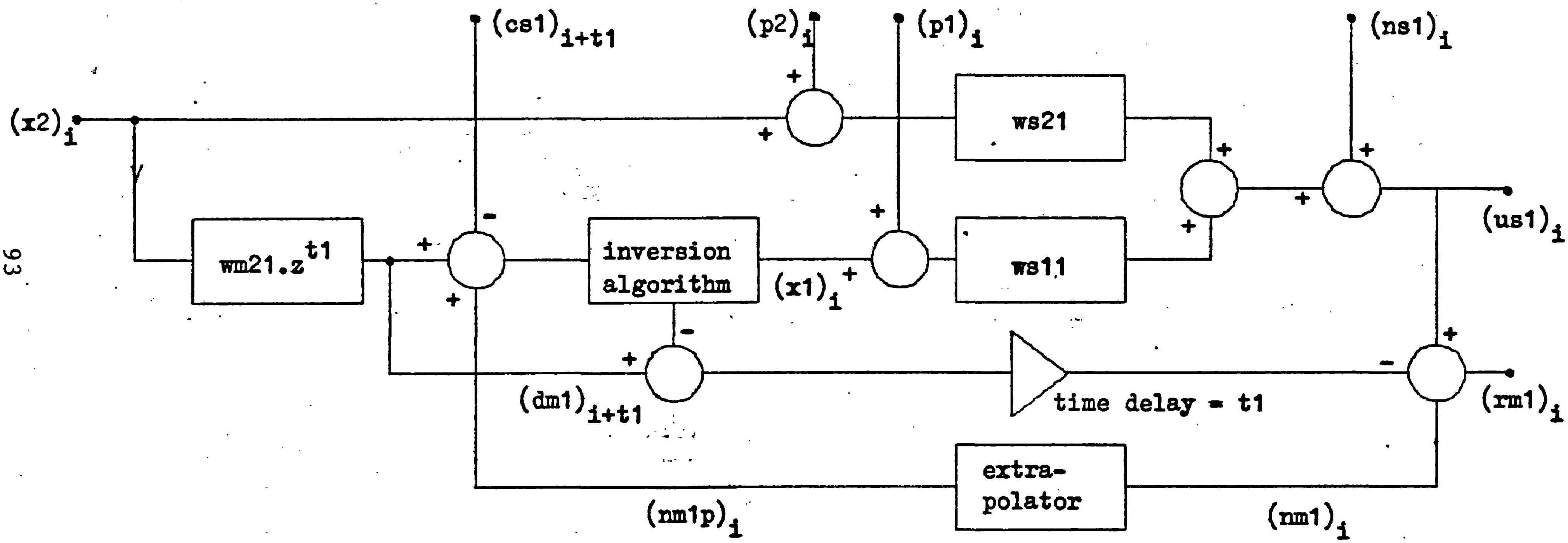



Fig. 4.1: Block Diagram of the combined Feedforward - Feedback System.

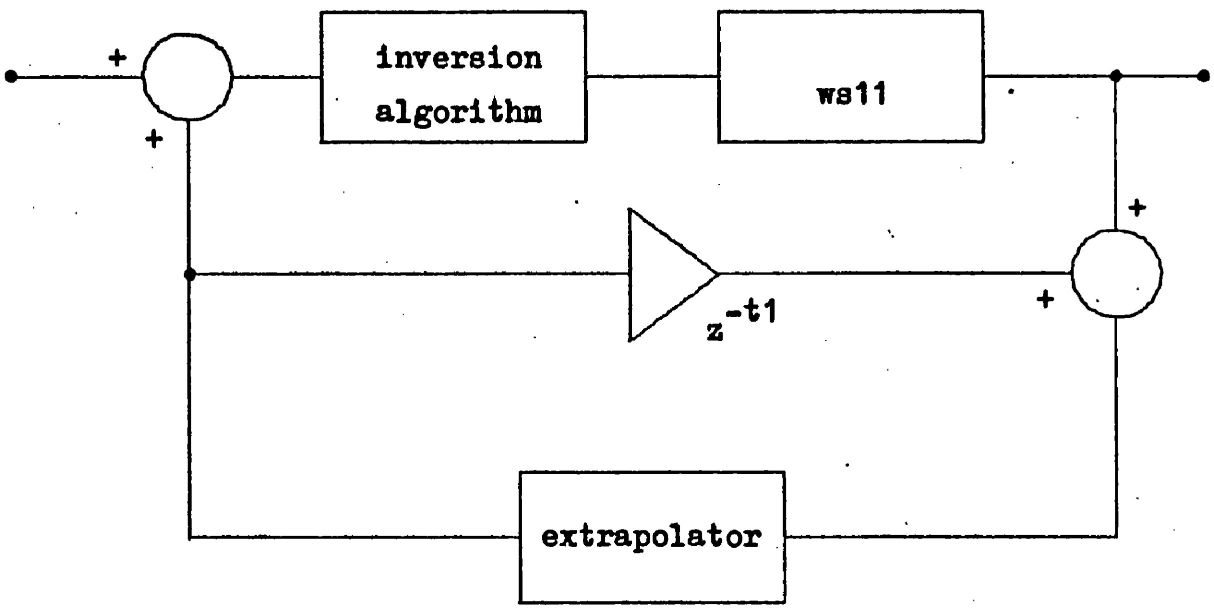



Fig. 4.2(a): Essential Elements in Consideration of Closed-Loop System Stability.

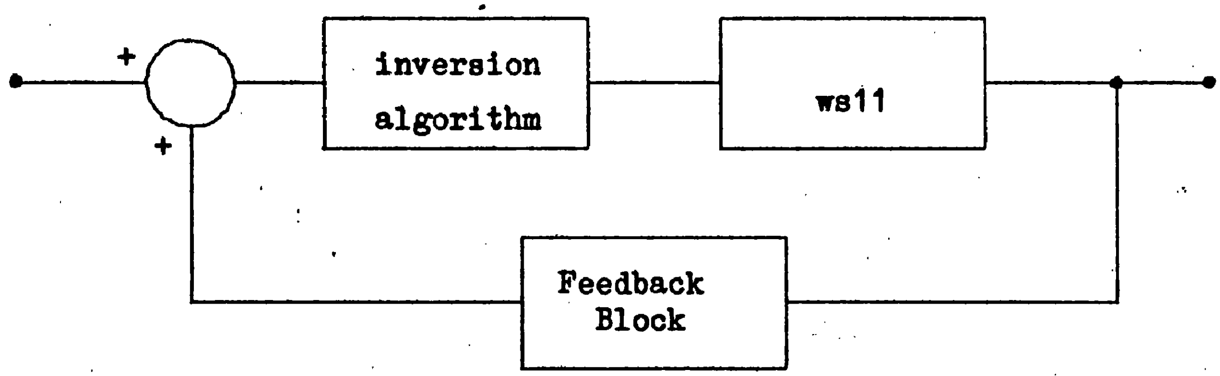



Fig. 4.2(b): Reduced Form of Fig. 4.2(a).

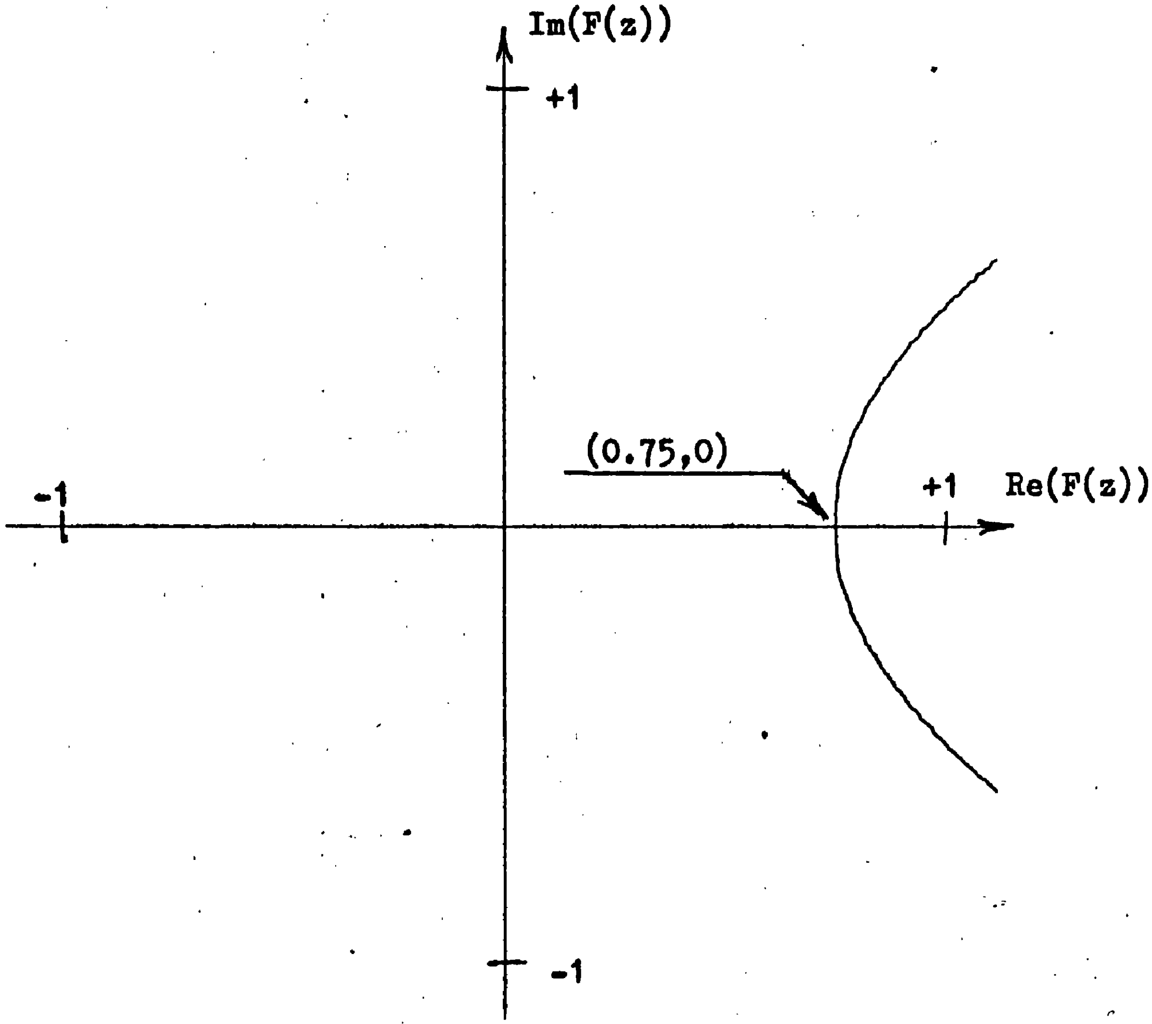



Fig. 4.3(a): Nyquist Plot for Equation 4.10 - t1 = 1.

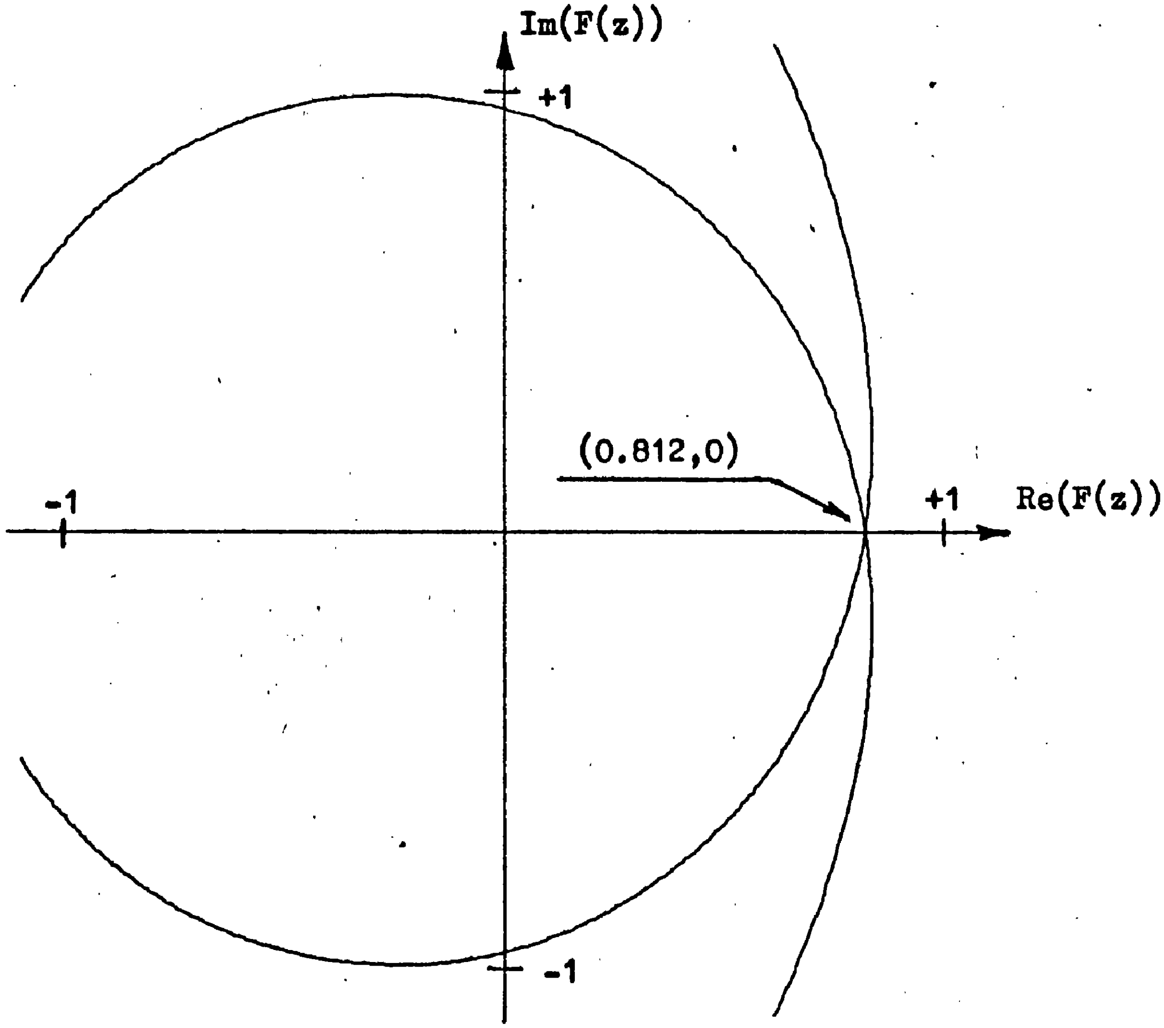



Fig. 4.3(b): Nyquist Plot for Equation 4.10 - t1 = 2.

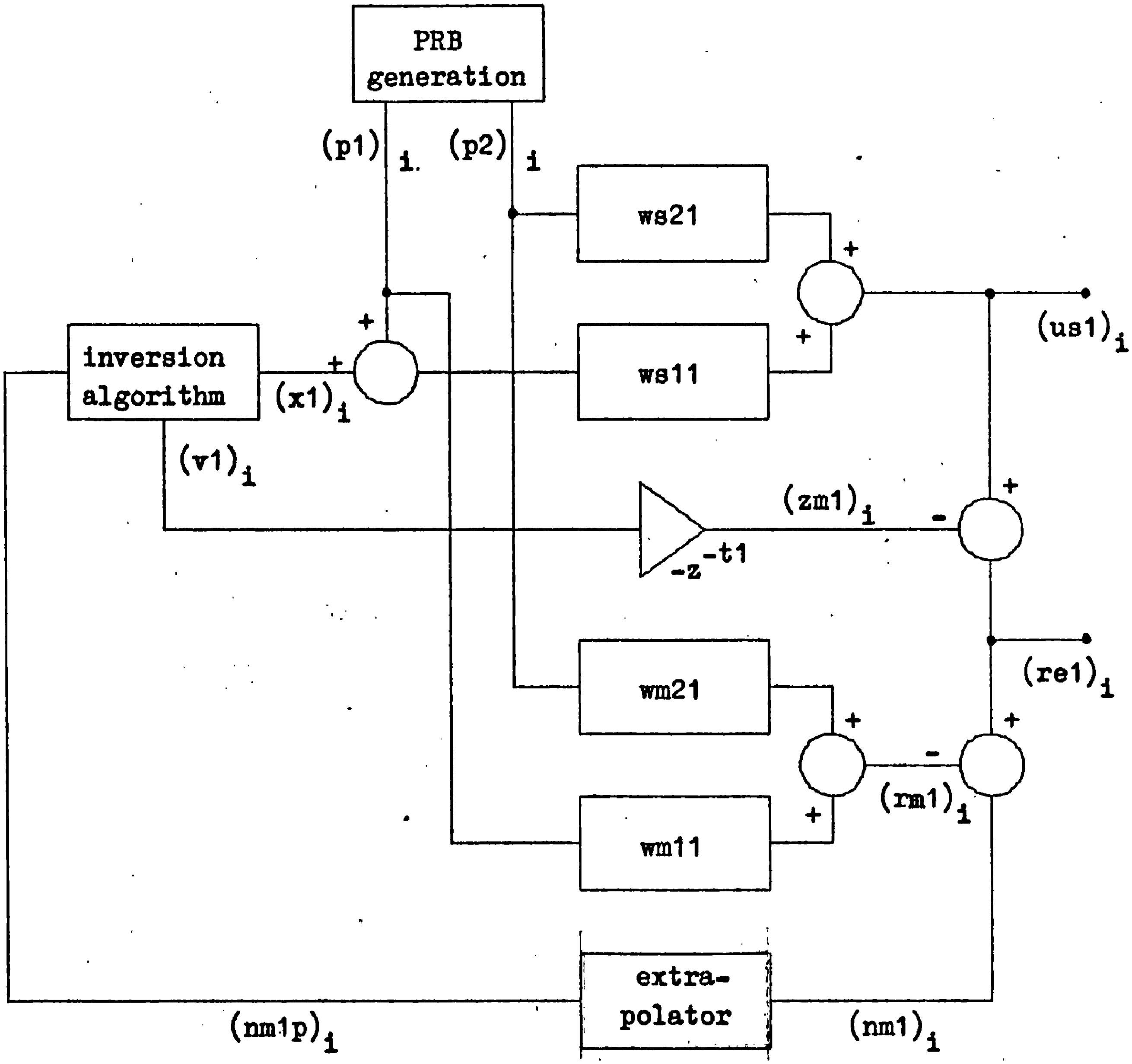



Fig. 4.4(a): Essential Elements in the Combined Identification and Control Scheme.

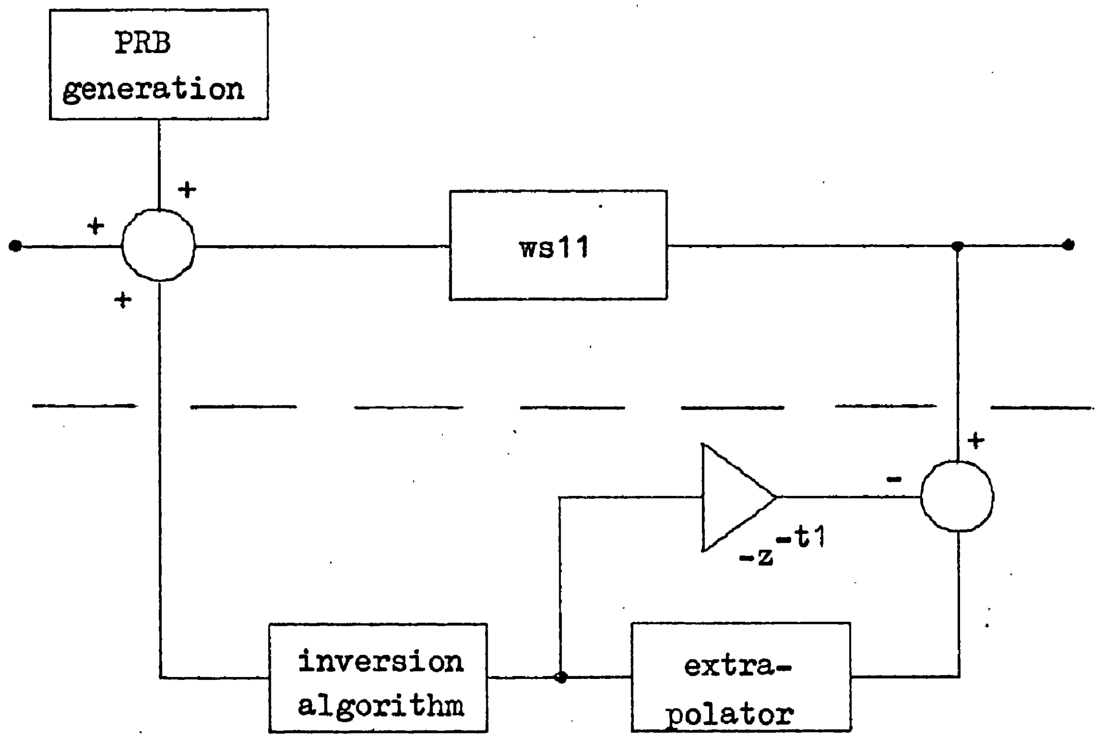



Fig. 4.4(b): Proposed Alternative Method of Identification in Presence of Feedback.

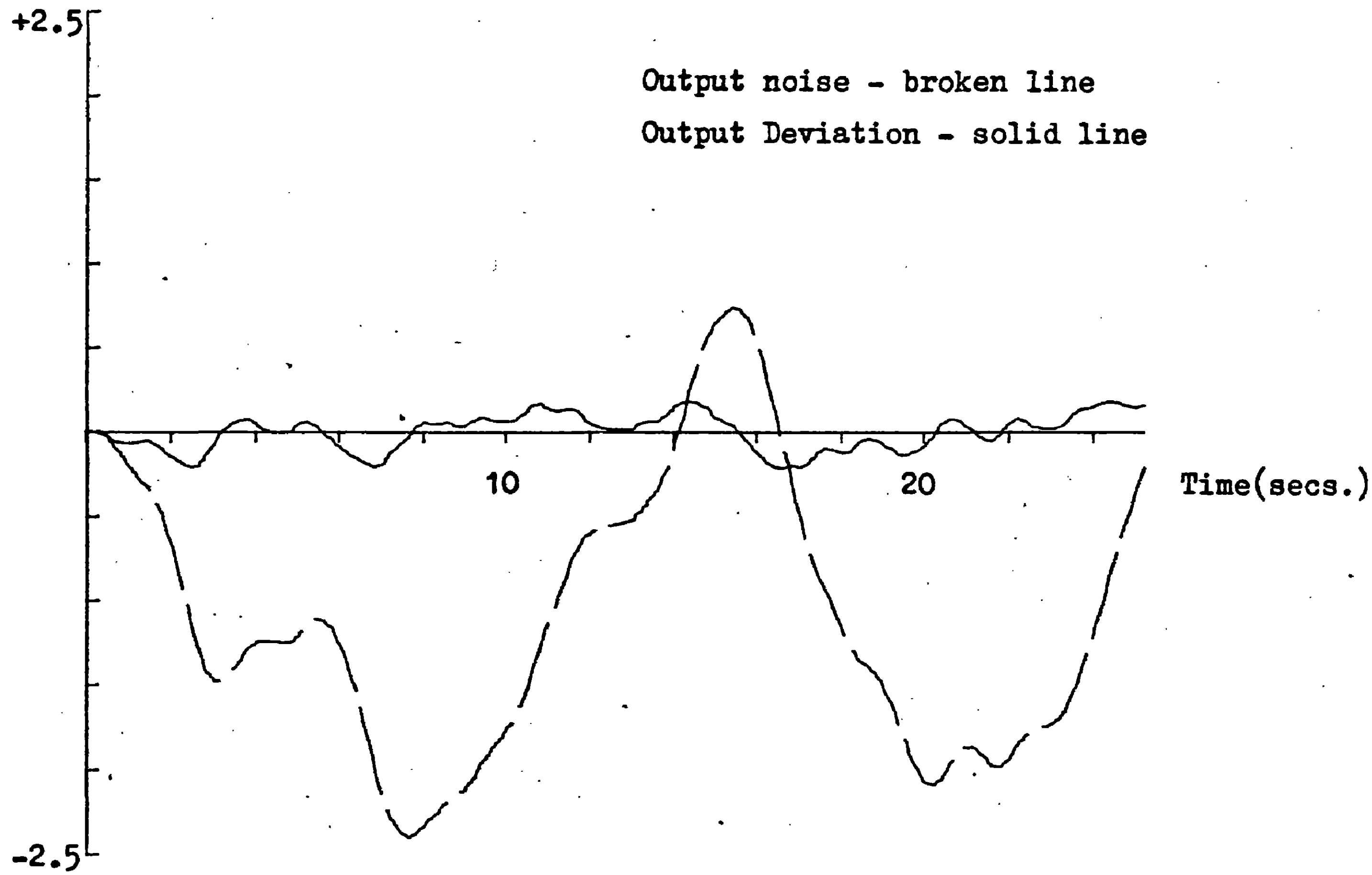



Fig. 4.5(a): Feedback Controller with Ideal Models - Test 4.1
Zero Order Extrapolator

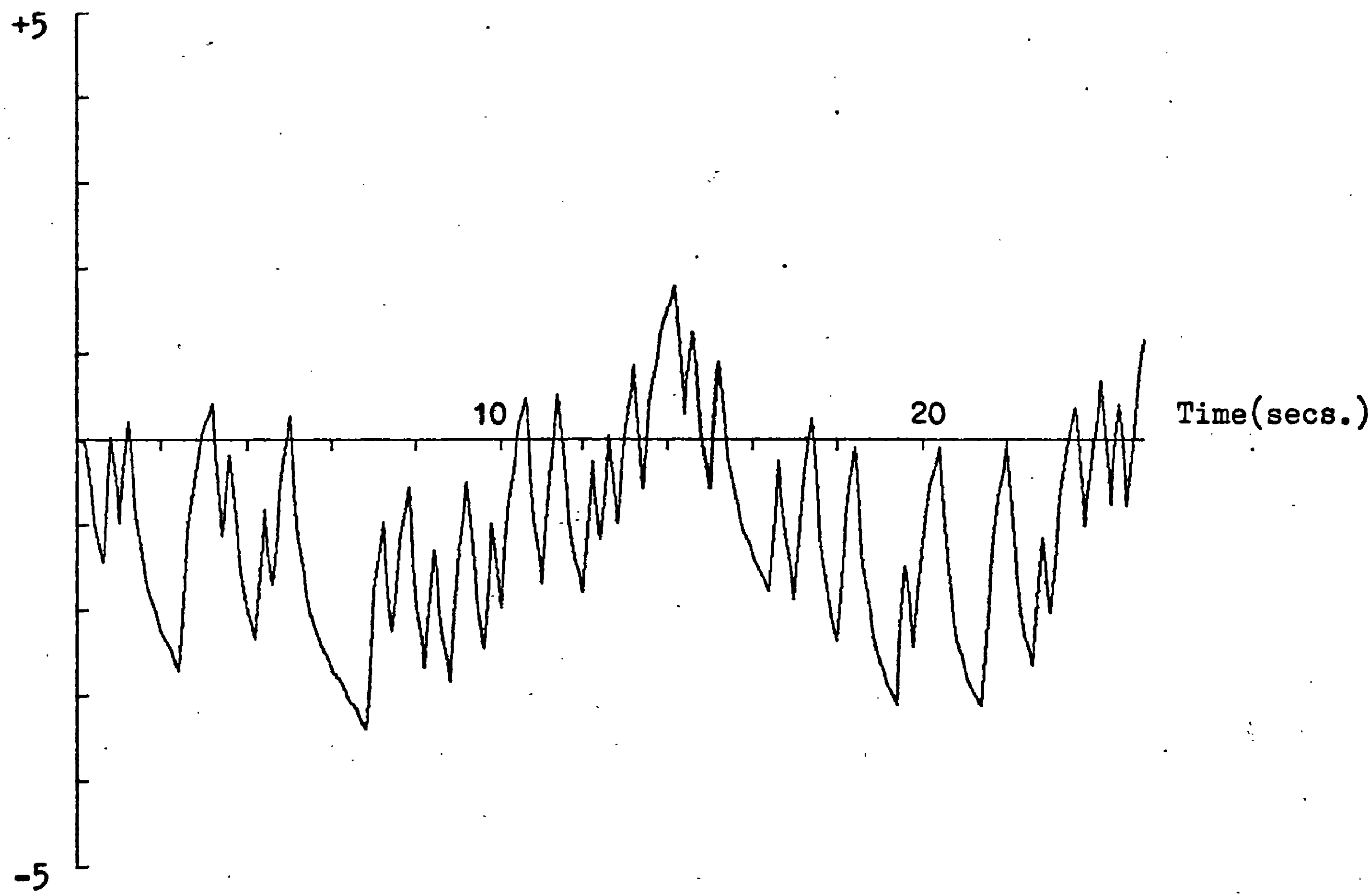



Fig. 4.5(b): Control Signal - Zero Order Extrapolator - Test 4.1

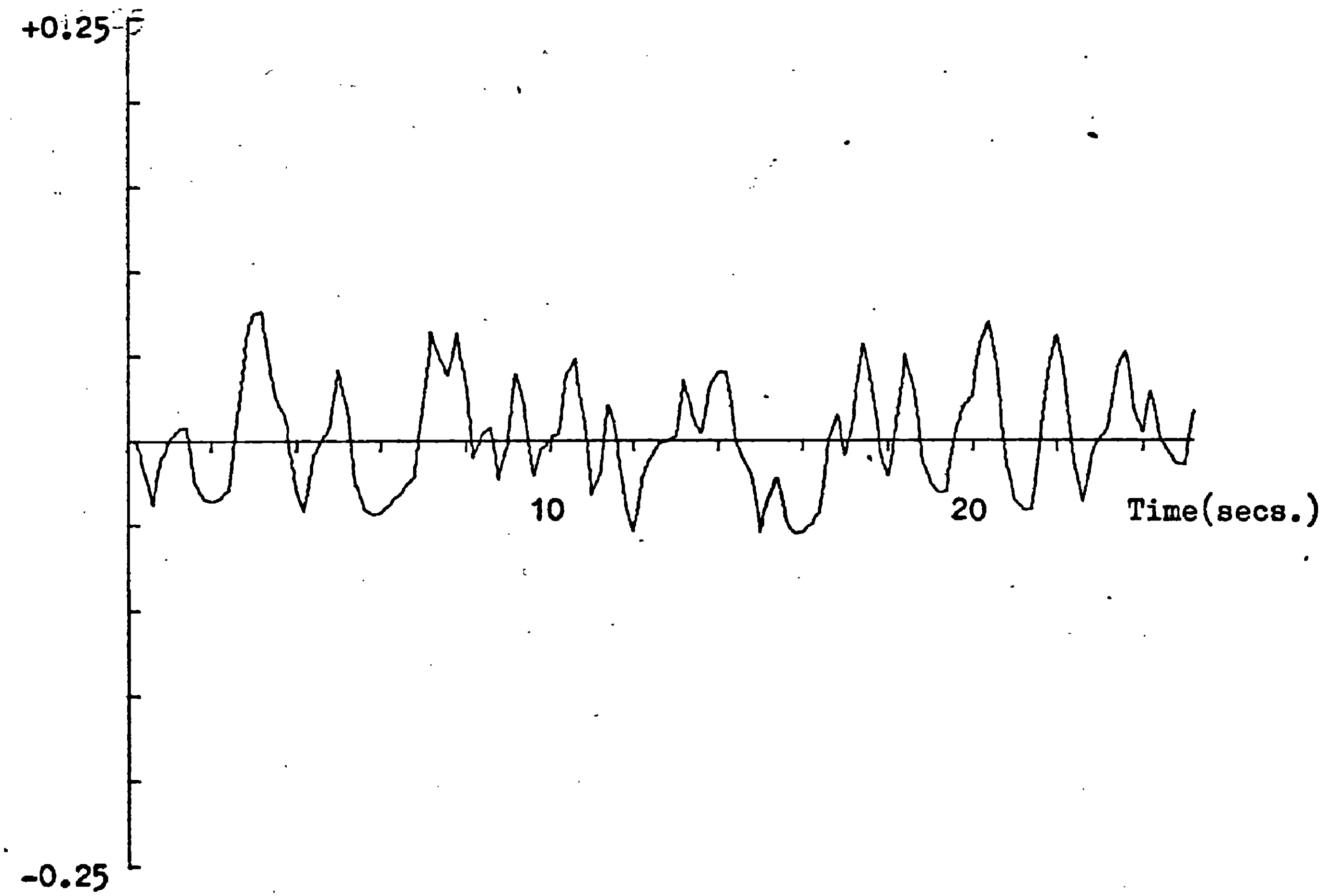



Fig. 4.5(c): Output Deviation - First Order Extrapolator - Test 4.1

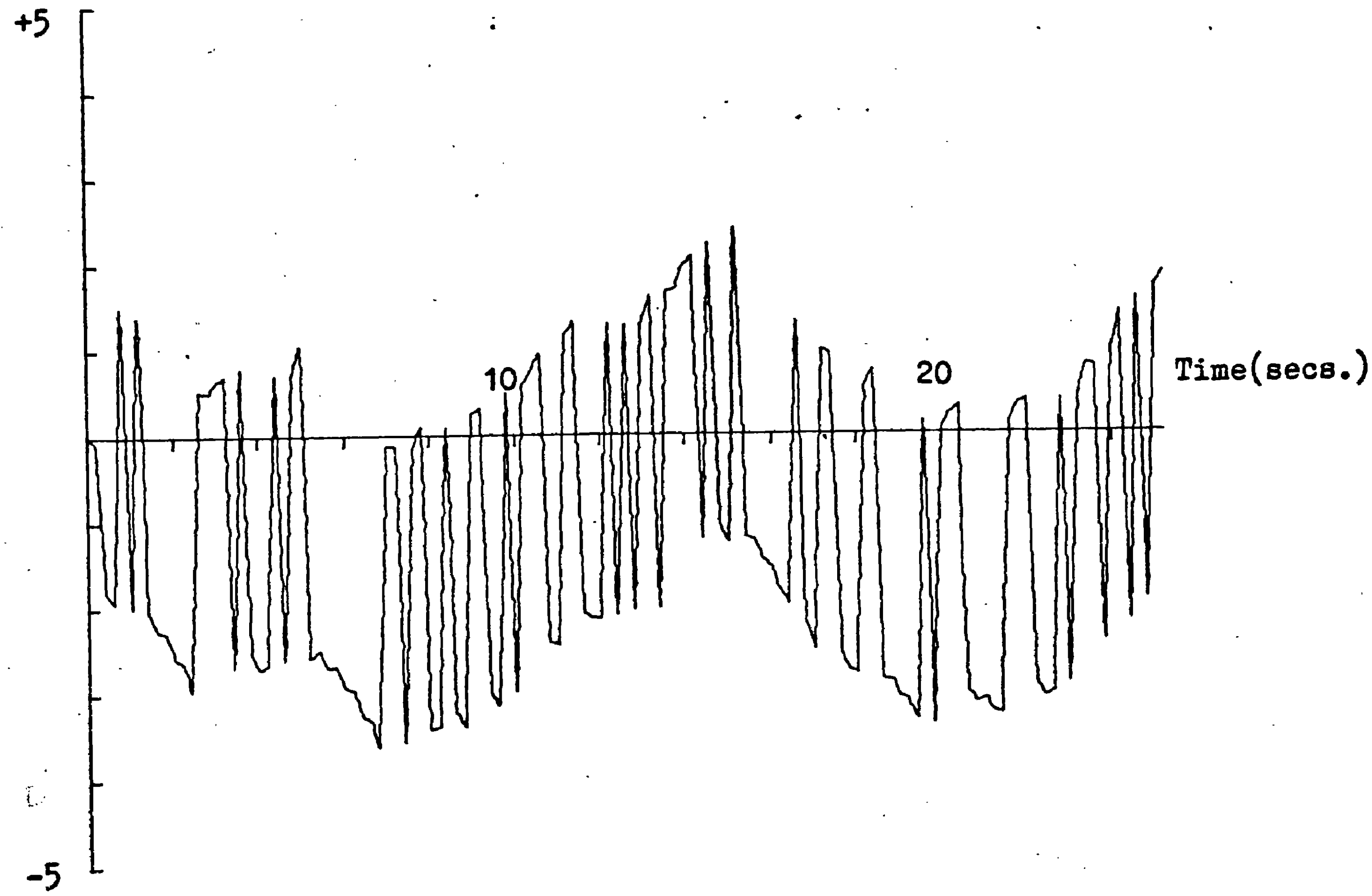



Fig. 4.5(d): Control Signal - First Order Extrapolator - Test 4.1

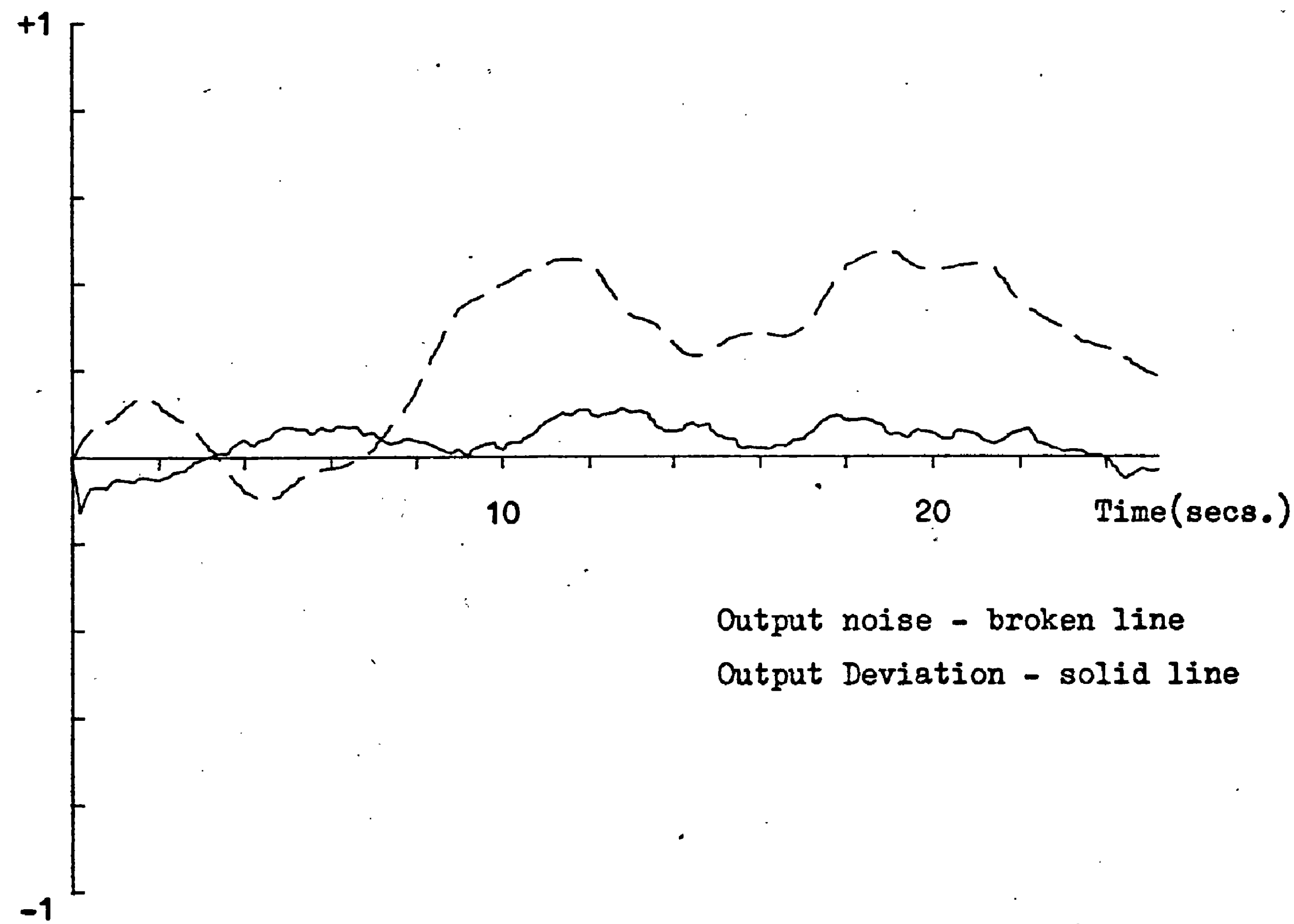



Fig. 4.6(a): Feedback Controller with Noisy Model - Test 4.2

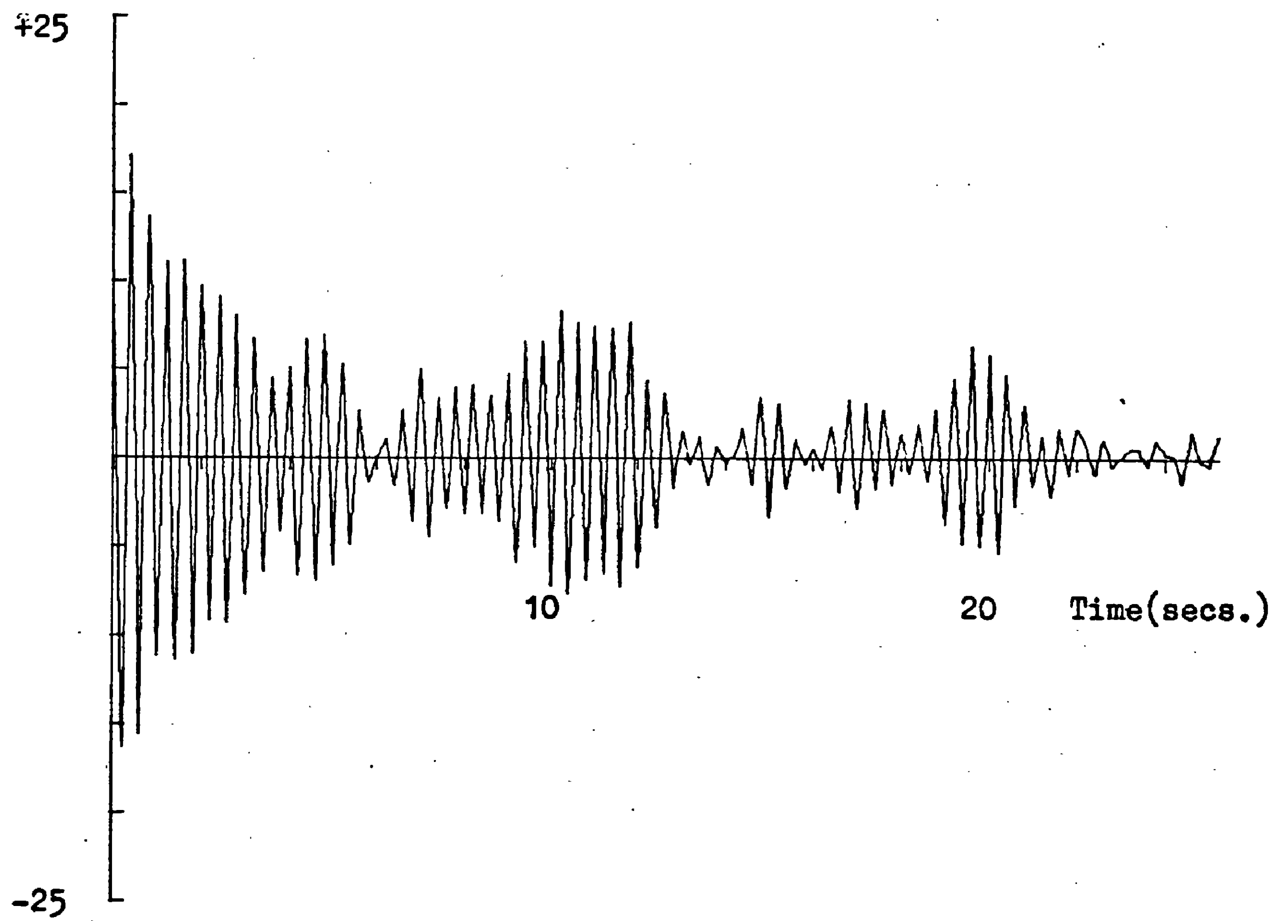



Fig. 4.6(b): Control Signal with Noisy Model - Test 4.2

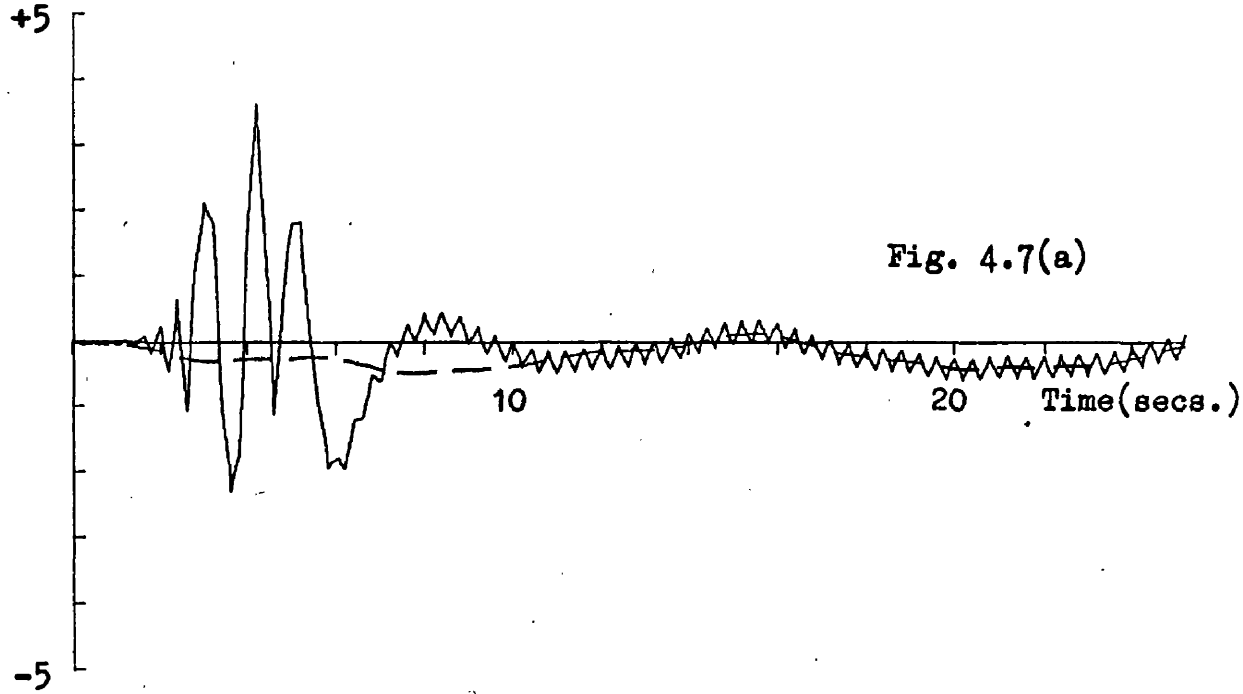



Fig. 4.7(a)

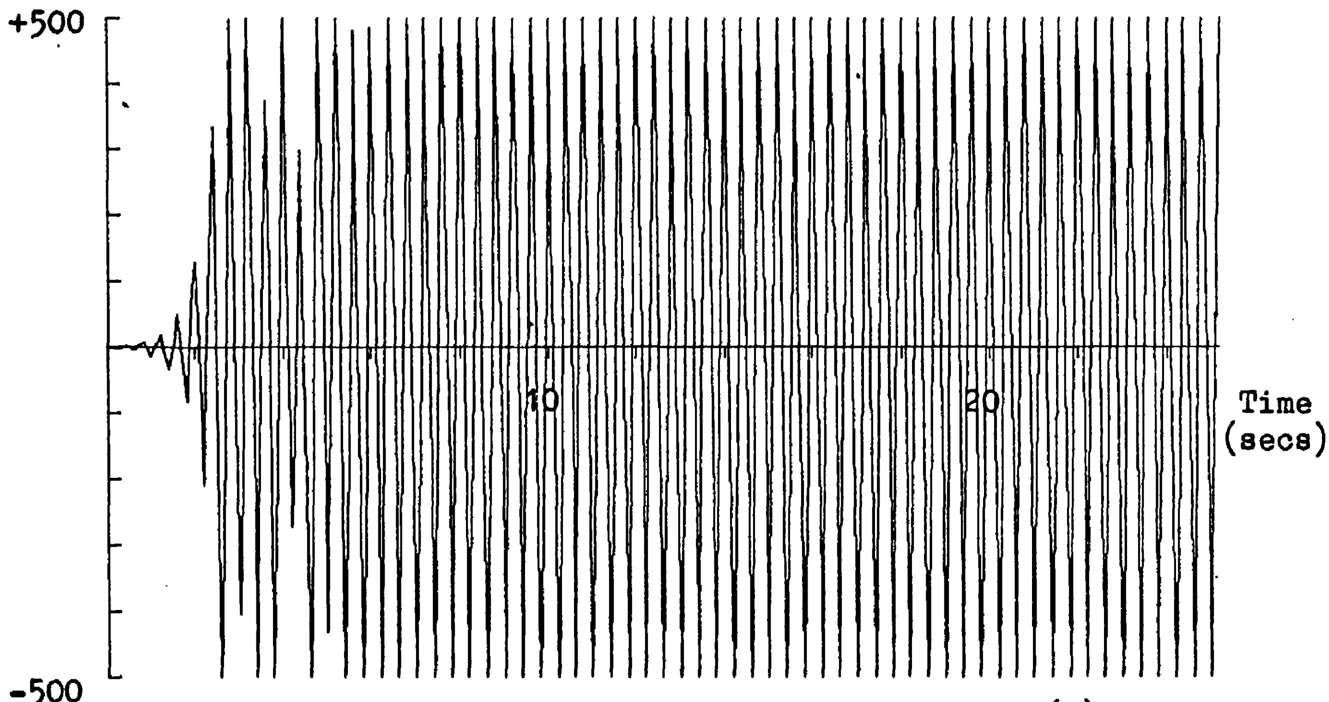
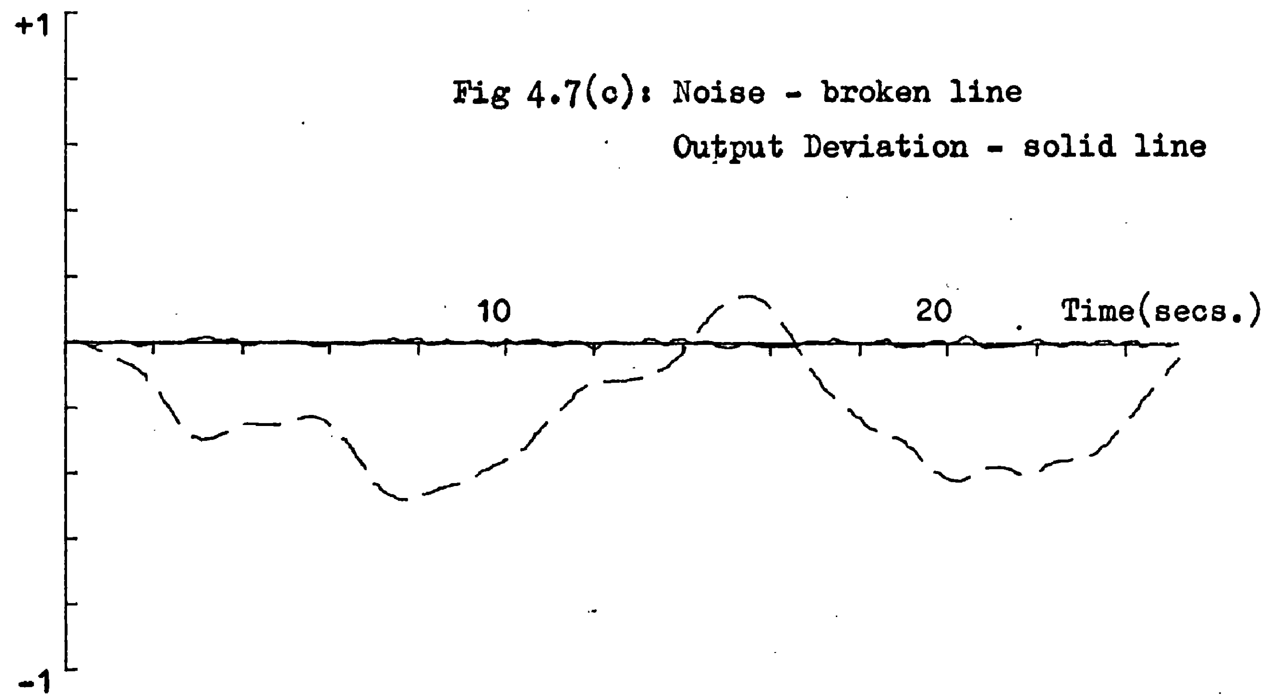



Fig 4.7(b)

Feedback controller with destabilising bias on model.

(a): Noise - broken line, and Output Deviation - solid line.

(b): Control Signal.



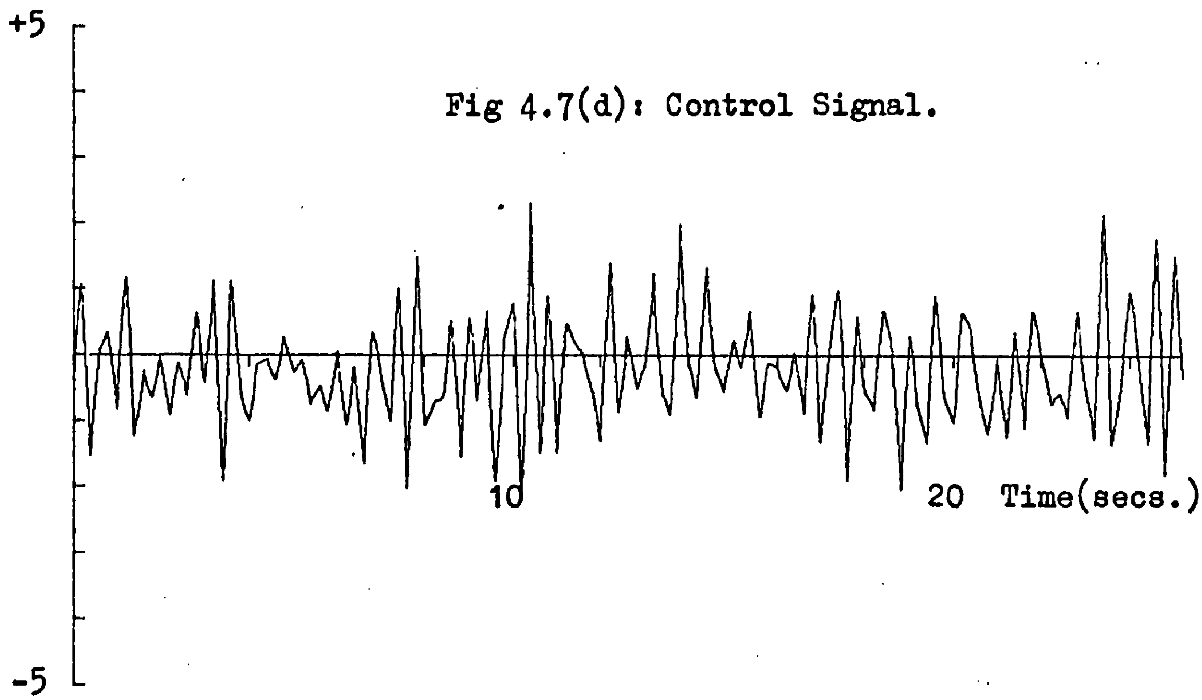
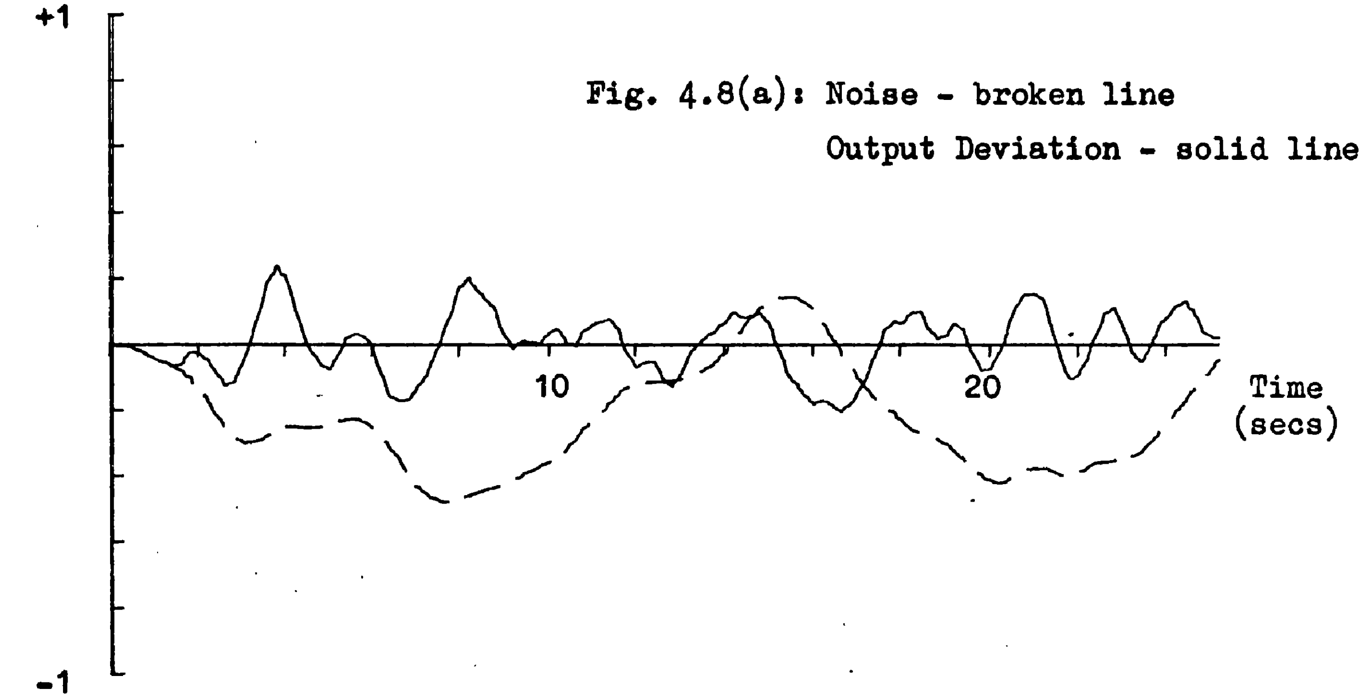




Fig. 4.7 (c) and (d):
Feedback Controller with Stabilising bias on Model.



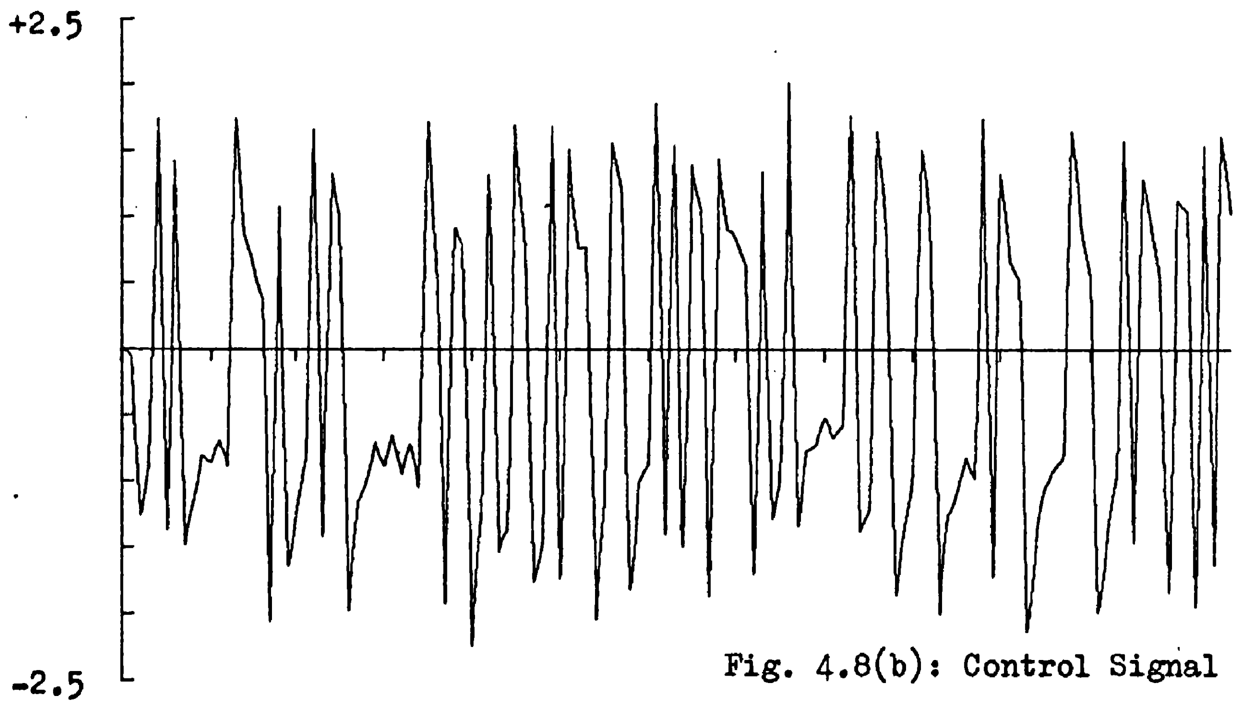




Fig. 4.8: Feedback Controller with Time Delay in Feedback Path - Test 4.4

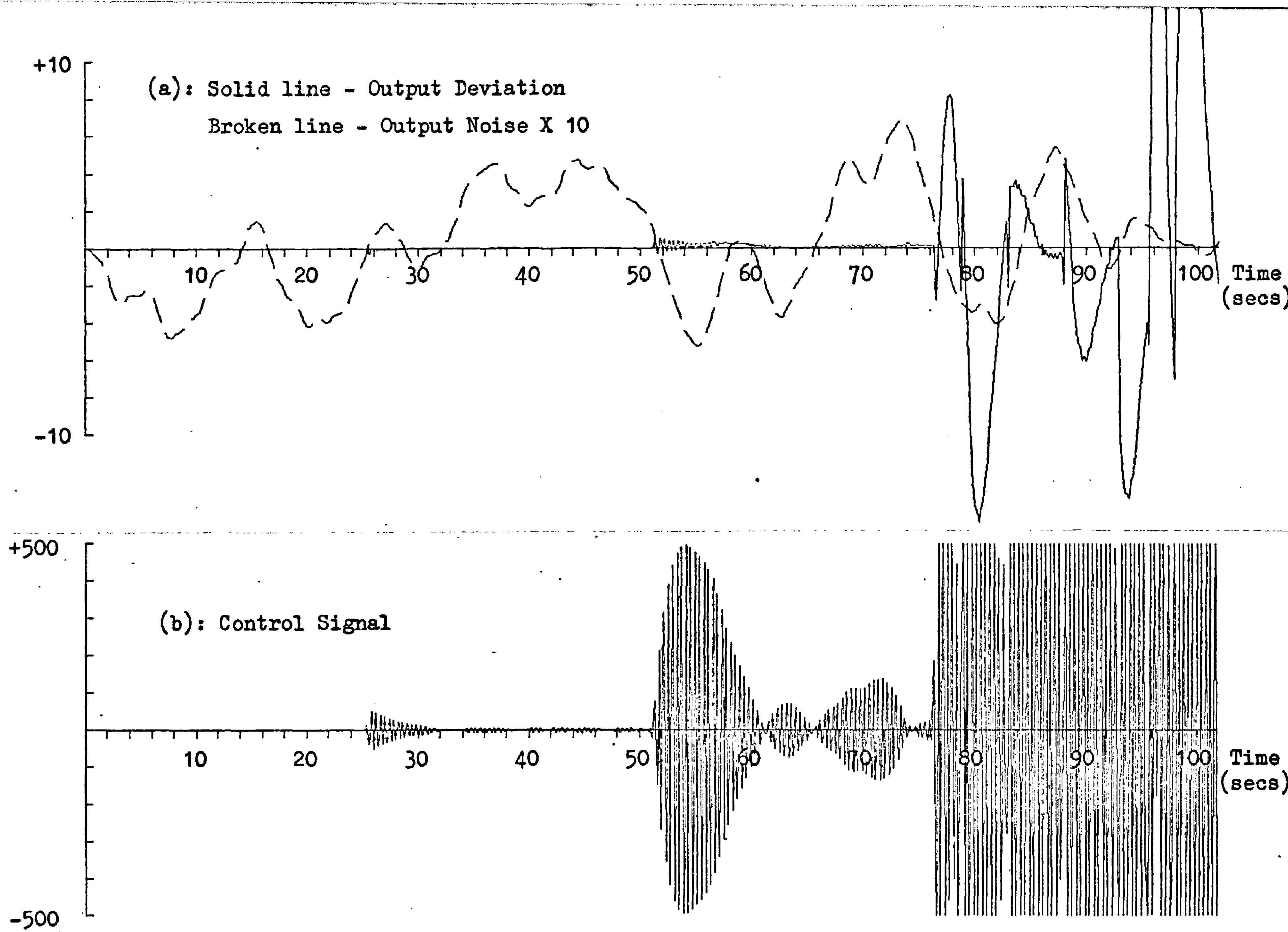



Fig. 4.9: Feedback Control with Simultaneous Identification - Test 4.5

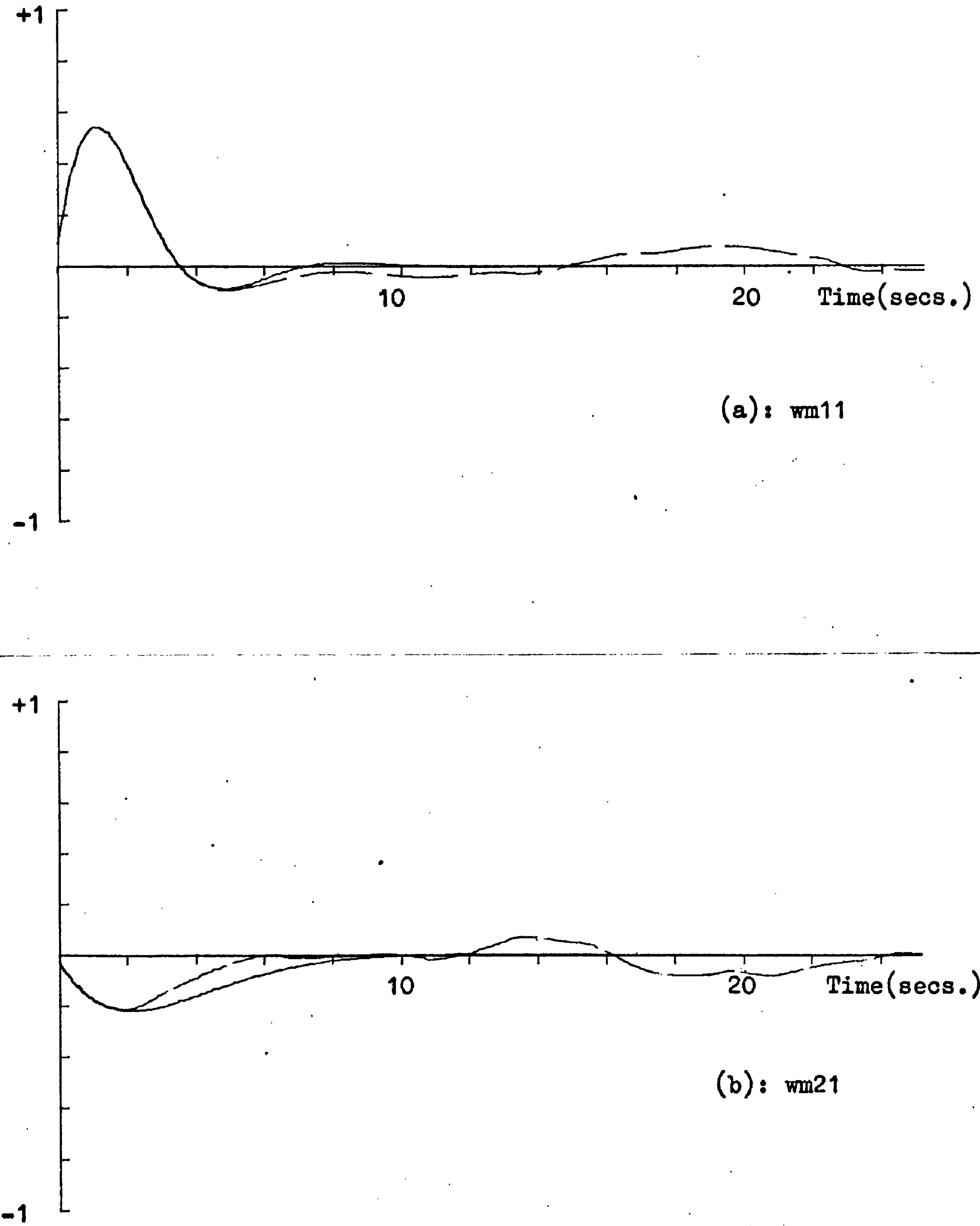



Fig. 4.10: Initial and Subsequent Identifications Under Feedback Control - Test 4.5
Solid lines - Initial Identification
Broken lines - first subsequent Identification

# C H A P T E R 5

# PROGRAM DESCRIPTION

## 5.1 GENERAL CONVENTIONS

The program described in the following pages is that devised for simulation of the combined feedforward, feedback and identification procedures, illustrated by a combination of Figs 4.1 and 4.4(a). In the majority of the tests described previously, modifications of this general program were made to allow the particular factors under investigation to be demonstrated without interference from unwanted areas of the simulation.

The language in which the various elements are written is Univac Fortran V[24]. This language contains as a subset, standard ASA Fortran IV and any reader familiar with the latter should have no difficulty in interpreting the program.

The variable notation follows that employed in previous chapters as closely as possible. In some cases it has been necessary to amend names to insure that they are compatible with the Fortran naming conventions for real and integer variables. The reader is referred to Appendix I, in which the equivalent names from program and text are listed.

Two major differences do exist between the equations derived previously and those implemented in the program: the sampling interval in the program is a parameter which is set in the input data, allowing the investigation of a more general range of systems than would be permissible with a unity sampling period. This has the effect of modifying some of the constants in a number of equations. The requirement of Fortran that a dimension (array) A commence at element A(1) and extend to some preset upper limit has necessitated a departure from previous conventions in regard to variable suffixing.

The most recently calculated values of each variable, which all had the suffix 'i' in the text, have a variety of suffices in the program, since it would otherwise be necessary to reserve storage for the past histories of a number of variables of much greater length than is required for the computations involving these specific variables.

Weighting sequences are stored with the first non-zero ordinate in the first location of the corresponding array, time delays being stored as a separate integer number.

While these changes have been necessary to keep program storage within reasonable limits, no claim is made that the storage use has been optimised. If this had been done, many intermediate variables would have been eliminated and the program would lose its resemblance to the equations within the text, and would be less comprehensible to the reader.

Should the system be implemented on a small control computer, the program would have to be completely rewritten, account being taken of interface scaling factors, interrupt processing and many other factors, so a storage optimisation at this stage would be pointless. Advantage has been taken of the fact that, on the Univac 1108, all variables are initially zero. This enables a large section of variable initialization to be omitted.

## 5.2 THE MAIN PROGRAM

A listing of this element appears in the following pages. The program has been written to make maximum use of subroutines, resulting in a greatly clarified presentation which the reader should have little difficulty in associating with the equations derived in previous chapters, and with the block diagrams, Figs 4.1 and 4.4(a).

The program logic is such that the simulation runs initially for three test signal cycles, during which time the control computation section is by-passed. At the end of this time, models of both paths have been computed, the logic variable RUNUP assumes the value .FALSE., and control action commences, the required random disturbance variables being read from data files (10) and (12). The two main counters within the program are ICYCLE and ICOUNT, which correspond to the number of cycles of the test signal which have elapsed since the start of the simulation, and the sampling interval within the current cycle respectively.

The variable NCODE is an integer obtained from reference 12, which defines the generating polynomial for the m-sequence on which the test signal is based. The sequence length, k, is calculated from this. Since it is assumed that the system allows the text variables k1 and k2 to be equal, these are both represented by the program variable KBY2.

The output section is a typical one used in many of the tests for Chapter 4.

```
      COMMON T,KBY2,K,I2K,N,NCODE
      LOGICAL RUNUP
      DIMENSION A1(20),A2(20),B1(2),B2(2),OP1(20),OP2(20),
     A          RE1(2048),SIGIN(20),CONTIN(20),ICN(4),
     A          WM11(512),WM21(512),W(1024),IP1(1024),IP2(1024),
     A          X2(512),X1(512),ZM1(20)
C
C
C   COUNTER ASSIGNMENT SECTION
      READ(5,1)NCODE
      ND= NDIG(NCODE)
      I= NCODE/(10**(ND-1))
      N= 1
      IF(I.GT.3)N= N+1
      IF(I.GT.1)N= N+1
      N= 3*(ND-1)+N-1      @ N IS THE ORDER OF THE GENERATING POLYNOMIAL
      K= 2**N-1            @ K IS THE CODE LENGTH
      KBY2= K/2
      I2K= 2*K
C
C
C   TEST SIGNAL GENERATION SECTION
      CALL ORDER(ICN,ND)
      CALL PRB(ICN,IP1)
      DO4 I=1,K
      IF(IP1(I).EQ.0)IP1(I)=  -1
    4 IP2(I)= IP1(I)
      CALL SHIFT(IP2,KBY2)
C
C
```

```
C
C   INITIAL SETUP SECTION
      READ(5,1)IFINAL
      CALL COEFF(A1,B1)
      CALL COEFF(A2,B2)
      READ(5,14)MODE
      IF(MODE.EQ.'FFWD'.OR.MODE.EQ.'BOTH')REWIND 10
      IF(MODE.EQ.'FBAK'.OR.MODE.EQ.'BOTH')REWIND 12
      REWIND 11                             @ 11,13 ARE OUTPUT FILES
      REWIND 13
      CALL INVERT(WM11,X1,V1)               @INITIATES VARIABLES
      CALL BXCORR(W,IP1,RE1,.FALSE.         @ WITHIN THE ROUTINES
      CALL NEWLIN(6,3)
      T= -T   @SETS CONDITION FOR SUBSEQUENT CALLS ON INVERT & BXCORR
      CS1= 0.
      RUNUP= .TRUE.
      ICYCLE= 1
 5    ICOUNT= 1
C
C
```

```
C     MAIN PROGRAM SECTION
        WRITE(6,11)ICYCLE
  6     SIGIN(20)= IP2(ICOUNT)
        CONTIN(20)= IP1(ICOUNT)
        CALL KSYNT(CONTIN,OP1,A1,B1)
        CALL KSYNT(SIGIN,OP2,A2,B2)
        US1= OP1(20) + OP2(20)
        IF(RUNUP)GOTO 7   @ RUNUP IS .TRUE. UNTIL MODELS ESTABLISHED
        IF(MODE.EQ.'FFWD'.OR.MODE.EQ.'BOTH')READ(10,1)X2(KBY2)
        IF(MODE.EQ.'FBAK'.OR.MODE.EQ.'BOTH')READ(12,1)ANS1
        ANS1= ANS1/5.
        US1= US1 + ANS1
        RM1= BCONV(WM11,IP1,ICOUNT,NT1)
       A    + BCONV(WM21,IP2,ICOUNT,NT2)
        DM1P= CONVOL(X2,WM21,NT2-NT1+1)
        ANM1= US1 - ZM1(1) - RM1
        ANM1P= EXTRAP(ANM1,NT1)
        IF(MODE.EQ.'FFWD')ANM1P= 0.
        V1= DM1P + ANM1P - CS1
        CALL INVERT(WM11,X1,V1)
        ZM1(NT1+1)= -V1 + DM1P
        CONTIN(20)= CONTIN(20) - X1(KBY2)
  7     SIGIN(20)= SIGIN(20) + X2(KBY2)
        RE1(I2K)= US1 - ZM1(1)
C
C
C
```

```
C   OUTPUT SECTION
      IF(ICYCLE.GE.3)WRITE(6,13)X2(KBY2),DM1P,ANS1,US1,V1,X1(KBY2)
      IF(ICYCLE.LT.4)GOTO 17
      RIPL= US1 - RE1(K)
      WRITE(11)ANS1,RIPL,X1(KBY2)
C
C
C   CROSS CORRELATION SECTION
 17   IF(ICYCLE.LT.3)GOTO 19
      CALL BXCORR(W,IP1,RE1,ICOUNT.EQ.K)
      IF(ICOUNT.NE.KBY2)GOTO 18
      CALL TXFER(W,1,KBY2,WM11)
      CALL DELAY(WM11,NT1)
      DO171 I=1,NT1
 171  CALL SHIFT L(WM11,KBY2)
      WRITE(6,99)NT1,(WM11(J),J=1,KBY2)
 18   IF(ICOUNT.NE.K)GOTO 19
      CALL TXFER(W,KBY2+2,K,WM21)
      CALL DELAY(WM21,NT2)
      DO181 I=1,NT2
 181  CALL SHIFT L(WM21,KBY2)
      WRITE(6,99)NT2,(WM21(J),J=1,KBY2)
      RUNUP= .FALSE.
 99   FORMAT(' WEIGHTING SEQUENCE',I6,/,(E13.4))
C
C
```

```
C    END OF LOOP HOUSEKEEPING
 19     ICOUNT= ICOUNT+1
        CALL SHIFT L(OP1,20)    @ VECTORS MOVE BACK IN TIME
        CALL SHIFT L(OP2,20)
        CALL SHIFT L(SIGIN,20)
        CALL SHIFT L(CONTIN,20)
        CALL SHIFT L(RE1,I2K)
        CALL SHIFT L(X1,KBY2)
        CALL SHIFT L(X2,KBY2)
        CALL SHIFT L(ZM1,20)
        IF(ICOUNT.LE.K)GOTO 6
        IF(IFINAL-ICYCLE)9,9,20
 20     ICYCLE= ICYCLE+1
        GOTO 5
 9      IF(MODE.EQ.'FFWD'.OR.MODE.EQ.'BOTH')REWIND 10
        IF(MODE.EQ.'FBAK'.OR.MODE.EQ.'BOTH')REWIND 12
        REWIND 11
        REWIND 13
        STOP
C
C
C    FORMAT BLOCK
 1      FORMAT()
 2      FORMAT(' ALAN HEPBURN Z6')
 8      FORMAT(L1)
 11     FORMAT(///,' CYCLE NUMBER',I3,
       A        ///5X,'DISTURB',8X,'EFFECT',9X,'NOISE',
       A        8X,'OUTPUT',8X,'F/BACK',7X,'CONTROL'///)
 13     FORMAT(7E14.4)
 14     FORMAT(A6)
 15     FORMAT(1H+,98X,I6)
        END
```

## 5.3 THE SUBROUTINES

The listings of the subroutines and functions used in the program appear in the following pages. In most cases these use variable names which are the same as, or an abbreviated form of, those used in the main program.

(i) PRB, OPT, SHIFT, ORDER, NDIG

These routines generate the m-sequence on which the test signal is based. An option read within PRB, allows the routine OPT to be called, which shifts the m-sequence to yield the reference phase. If this option is not called, the sequence begins at the point where the previous N elements are all 1.

(ii) COEFF, KSYNT

COEFF generates the z-transform coefficients which are used by KSYNT to simulate the system paths. The expressions used for these coefficients are derived in Appendix II.

(iii) CONVOL, INVERT, EXTRAP, SHIFT L

These routines perform the required control computations. The comments within the listings are self-explanatory. Note that the sign of T is used as a logical variable, and is negative outwith the 'Initial Setup Section' of the Main Program Element.

(iv) BXCORR, BCONV, TXFER, DELAY

BXCORR computes 1 ordinate of the weighting sequence, W, at each call. The variable RUNSUM is used to compute the mean value of the cycle over which correlation is taking place, for the purpose of bias removal. At the end of each half cycle of the test signal, TXFER updates the appropriate model weighting sequence, and DELAY evaluates the time delay portion of the new model. BCONV is a special purpose convolution routine used to compute the response of the modelled system to the test signal.

```
      SUBROUTINE PRB(ICN,IP)
      COMMON T,KBY2,K,I2K,N
C     GENERATES PRB CODE ON G.F.(1,0) IN ARRAY IP.
C     OPTIONAL GENERATION OF REFERENCE PHASE.
      LOGICAL BOOL
      DIMENSION ICN(4),IP(512),ISTATE(11),IFB(11)
 1    FORMAT(' CODE LENGTH',I4)
 2    FORMAT(' FEEDBACK FROM STAGES')
 3    FORMAT(I3)
      WRITE(6,1)K
      WRITE(6,2)
      ICN(1)= ICN(1)/2
      M= 1
      DO5 I=1,N
      IFB(I)=(ICN(M)+1)/2-ICN(M)/2
      IF(IFB(I).EQ.1)WRITE(6,3)I
      ICN(M)= ICN(M)/2
      ISTATE(I)= 1
 5    M= M+(I+1)/3-I/3
      DO8 I=1,K
      IP(I)= ISTATE(1)
      M= 0
      DO6 J=1,N
      IF(IFB(J).EQ.0.OR.ISTATE(J).EQ.0)GOTO6
      M= M+1
 6    CONTINUE
      NM= N-1
      DO7 J=1,NM
 7    ISTATE(N-J+1)= ISTATE(N-J)
 8    ISTATE(1)= (M+1)/2 - M/2
 9    FORMAT(L1)
      READ(5,9)BOOL
      IF(BOOL)CALL OPT(IP)
      RETURN
      END
```

```
      SUBROUTINE OPT(IP)
      COMMON T,KBY2,K
C     REFERENCE PHASE GENERATION.  ERROR EXIT IS TO GUARD
C     AGAINST STICKING IN CORRELATION LOOP.
      DIMENSION IP(1)
      DO4 I=1,K
      IPHI= 0
      DO3 J=1,K
      M= I+J-1
      L= J
      IF(M.GT.K)M=M-K
      IF(IP(M))2,2,3
 2    L=-L
 3    IPHI=IPHI+L
      IF(IPHI.EQ.(K+1)/2)GOTO6
 4    CONTINUE
 5    FORMAT(////,' FAIL IN CODE OPTIMISATION')
      WRITE(6,5)
      STOP
 6    CALL SHIFT(IP,I-1)
 7    FORMAT(////' CODE OPTIMISED')
      WRITE(6,7)
      RETURN
      END
```

```
      SUBROUTINE SHIFT(IA,N)
      COMMON T,KBY2,K
C     SHIFTS INTEGER ARRAY SIZE K N PLACES TO LEFT.
      DIMENSION IA(1),IDUMP(1024)
      DO2 I=1,K
      M= 0
      IF(N+I.GT.K)M=-K
 2    IDUMP(I)= IA(M+N+I)
      DO3 I=1,K
 3    IA(I)= IDUMP(I)
      RETURN
      END


      SUBROUTINE ORDER(ICN,I)
      COMMON T,KBY2,K,I2K,N,NCODE
C     EXTRACTS INDIVIDUAL DIGITS OF PRB CODE NUMBER.
      DIMENSION ICN(I)
 1    FORMAT(' PRB CODE NUMBER',I4)
 2    FORMAT(' CODE ORDER',I4)
      WRITE(6,1)NCODE
      DO4 J=1,I
      ICN(J)= NCODE - NCODE/10*10
 4    NCODE= NCODE/10
      WRITE(6,2)N
      RETURN
      END


      FUNCTION NDIG(I)
C     EVALUATES NUMBER OF DIGITS IN INTEGER NUMBER I.
      ARG=I
      NDIG= ALOG(ARG+0.5)/ALOG(10.)+1
      RETURN
      END
```

```
      SUBROUTINE COEFF(CA,CB)
      COMMON T
C   EVALUATES NUMERATOR COEFFS. CA,  AND DENOMINATOR COEFFS. CB,
C   FOR Z-TRANSFORM SYNTHESIS OF:
C
C                                               2
C                                         G.OMEGA
C                               ----------------------------------
C                                2                               2
C                               S + 2.S.OMEGA.COS(ALPHA) + OMEGA
C
C   WITH SERIES TIME DELAY OF NT SAMPLING INTERVALS.
      DIMENSION CA(20),CB(2)
 1    FORMAT()
 2    FORMAT(' HEADING READ INTO THIS FORMAT')
 3    FORMAT(//12X,' ALPHA = ',F5.2,//12X,' OMEGA = ',F5.2//12X,
     A       '  GAIN = ',F5.2//' SAMPLING INTERVAL = ',F5.2,
     A                      //'         TIME DELAY = ',I4)
      READ(5,2)
      WRITE(6,2)
      READ(5,1),ALPHA,OMEGA,G,T,NT
      WRITE(6,3),ALPHA,OMEGA,G,T,NT
      ALPHA= ALPHA*0.017453
      A= OMEGA*COS(ALPHA)
      B= OMEGA*SIN(ALPHA)
 5    FORMAT(/////6H A1 = ,E10.3,//6H A2 = ,E10.3,//6H B1 = ,E10.3,//
     A       6H B2 = ,E10.3/////)
      AT= A*T
      BT= B*T
      EX= EXP(-AT)
      CA(1+NT)= G*(1-EX*(AT*SIN(BT)/BT+COS(BT)))
      CA(2+NT)= G*EX*(EX+AT*SIN(BT)/BT-COS(BT))
      CB(1)= -2.*EX*COS(BT)
      CB(2)= EX**2
      WRITE(6,5)CA(1+NT),CA(2+NT),CB(1),CB(2)
      RETURN
      END
```

```
      SUBROUTINE KSYNT(X,Y,A,B)
C     Z-TRANSFORM SYNTHESIS.  X IS SYSTEM INPUT,  Y IS SYSTEM
C     OUTPUT,  A NUMERATOR COEFFS,  B DENOMINATOR COEFFS.
      DIMENSION X(20),Y(20),A(20),B(2)
      Y(20)= 0.
      DO 1 I=1,20
 1    Y(20)= Y(20) + X(20-I)*A(I) - Y(20-I)*B(I)
      RETURN
      END


      FUNCTION EXTRAP(X,NT)
C     1ST. ORDER DATA EXTRAPOLATOR.  PREDICTION NT AHEAD.
      EXTRAP= X + NT*(X-XOLD)
      XOLD= X
      RETURN
      END


      SUBROUTINE SHIFT L(A,J)
C     SHIFTS ARRAY A,  SIZE J,  1 PLACE LEFT.
      DIMENSION A(1)
      LIMIT= J-1
      DO1 I=1,LIMIT
 1    A(I)= A(I+1)
      RETURN
      END
```

```
      FUNCTION CONVOL(X,W,NT)
      COMMON T,KBY2
C   X IS THE SYSTEM INPUT VECTOR,  W THE WEIGHTING
C   SEQUENCE,  AND NT THE FORWARD TIME SHIFT.
      DIMENSION W(1),X(1)
      Y= 0.0
      IF(NT.LT.1)NT= 1
      DO1 I=NT,KBY2
 1    Y= Y+W(I)*X(KBY2-I+1)
      CONVOL= -Y*T
      RETURN
      END


      SUBROUTINE INVERT(W,X,Y)
      COMMON T,KBY2
C   INVERSION ALGORITHM.  ON SUBSEQUENT CALLS T IS (-)VE.
C   Y IS MODIFIED ON RETURN IF AMPLITUDE LIMITS ENCOUNTERED.
      DIMENSION W(1),X(1)
      IF(T)2,1,1
 1    DIVT= 1/T
      READ(5,4)SAT
      RETURN
 2    SIGMA= 0.
      DO 3 I=2,KBY2
 3    SIGMA= SIGMA + W(I)*X(KBY2-I+1)
      X(KBY2)= (Y*DIVT-SIGMA)/W(1)
      IF(ABS(X(KBY2)).LE.ABS(SAT))RETURN
      X(KBY2)= SIGN(SAT,X(KBY2))
      Y= (W(1)*X(KBY2)+SIGMA)/DIVT
 4    FORMAT()
      RETURN
      END
```

```
      SUBROUTINE BXCORR(W,IP,OP,LAST)
      COMMON T,KBY2,K,I2K
C     CONTINUOUS X-CORRELATION ROUTINE.  SUBSEQUENT CALLS WITH
C     T (-)VE.  OPTIONAL BIAS CORRECTION WHEN DG READ (+)VE.
      DIMENSION W(1),IP(1),OP(1)
      LOGICAL LAST
      IF(T.LT.0.)GOTO1
      A= 1/T/(K+1)
      IOWN= 1
      LAST= .FALSE.
      RUNSUM= 1.1E37
      READ(5,6)DG
      RETURN
1     SUM= 0.
      IF(RUNSUM.LT.1.0E37)GOTO3
      RUNSUM= 0.
      ITOP= I2K-1
      DO2 I=K,ITOP
2     RUNSUM= RUNSUM+OP(I-1)
      GOTO 31
3     RUNSUM= RUNSUM+OP(I2K-1)-OP(K-1)
31    DO4 I=1,K
4     SUM= SUM+IP(I)*OP(I+K-1)
      W(IOWN)= (SUM-RUNSUM/K)*A
      IOWN= IOWN+1
      IF(DG.LE.0.)GOTO 5
      W(IOWN)= W(IOWN) + DG/(K+1)
5     IF(.NOT.LAST)RETURN
      IOWN= 1
      RETURN
6     FORMAT()
      END
```

```
      FUNCTION BCONV(W,IP,ICOUNT,NT)
      DIMENSION W(1),IP(1)
      COMMON T,KBY2,K
C   EVALUATES RESPONSE OF SYSTEM DEFINED BY WEIGHTING
C   SEQUENCE W TO BINARY SIGNAL IP.
      BCONV= 0.
      DO1 I=1,KBY2
      N= 1-NT+ICOUNT-I
      IF(N.LE.0)N= N+K
      IF(IP(N))2,2,3
 2    BCONV= BCONV - W(I)
      GOTO1
 3    BCONV= BCONV + W(I)
 1    CONTINUE
      BCONV= -BCONV*T
      RETURN
      END


      SUBROUTINE TXFER(A,LO,IHI,B)
C   TRANSFERS ELEMENTS LO TO IHI OF ARRAY A TO ARRAY B.
      DIMENSION A(1),B(1)
      LIMIT= IHI-LO+1
      DO1 I=1,LIMIT
 1    B(I)= A(LO+I-1)
      RETURN
      END
```

```
      SUBROUTINE DELAY(W,N)
      COMMON T,KBY2,K
C     LOCATES AND SETS TO ZERO THE TIME DELAY PORTION AT
C     START OF WEIGHTING SEQUENCE.  DELAY IS N - ASSUMED > 0
      DIMENSION W(1)
      L=KBY2/20
      M=0
      W(1)= 0.
      DO6 I=1,KBY2
      IF(W(I+1)-W(I))1,3,2
 1    IF(M)11,11,3
 11   M=M-1
      IF(L-IABS(M))4,6,6
 2    IF(M)3,21,21
 21   M=M+1
      IF(L-IABS(M))4,4,6
 3    M=0
 6    CONTINUE
 4    LIM=I-L
      N= 1
      DO5 J=1,I
      N= J
      IF(J.LE.I-L)W(J)=0.
      IF(W(J)*W(I))51,5,5
 51   IF(ABS(W(I)).LT.ABS(W(LIM)))RETURN
 5    CONTINUE
      RETURN
      END
```

# CHAPTER 6

## CONCLUSIONS

## 6.1 ACHIEVEMENTS AND LIMITATIONS

The proposed objectives of the present investigation were defined in section 1.4. While all aspects of this proposal have not proved to be successful, a basis has been established for the implementation of a practical digital control scheme. The nature of the problems encountered clearly defines the specific areas in which further investigation is required, and should be of considerable benefit to other workers contemplating the development of similar schemes.

It was shown in section 1.4 that the use of weighting sequence plant models in a control scheme necessarily imposes certain limitations on the generality of plant dynamics which can be handled. All input-output paths within the plant must be approximately linear, and their weighting sequences must settle to a negligible value in finite time. If this time is very long compared with the shortest time constant in the system, the weighting sequence formulation becomes cumbersome due to the large number of ordinates which must be stored.

The identification procedure developed in Chapter 2 showed that the measures introduced to eliminate unwanted correlation with drift signals of up to first order were effective. The amplitudes of spurious correlations due to higher order terms is not, however, negligible, and when correlation is carried out over one cycle of the test signal, useful models can only be obtained with signal:noise ratios of the order of 5:1 or better. The identification procedure requires some a priori knowledge of the system time constants to enable the test signal parameters to be chosen, and a knowledge of the steady state path gains permits model bias to be minimised. It is required that the system dynamics remain substantially constant over the identification period.

In the development of the feedforward controller in Chapter 3, it was found necessary to introduce two further restrictions on the types of system which would be handled. It was shown that consideration of a general multi-output system would be excessively complex both analytically and computationally, and hence further development was confined to systemswith one controlled output. The algorithms developed to implement the feedforward control were only applicable to systems in which the control path ws11(s) was known to be minimum phase. Subject to these limitations the feedforward system functioned as required. Tests 3.2 and 3.3 indicated that corruption of the models used was not a critical factor. As was to be expected, the corruption lead to a reduction in control accuracy, ~~but the controller only became ineffective when the control path well(s) had poles close to the right half s plane.~~ The results obtained with amplitude limits applied to the control signal are particularly relevant to many practical applications.

The feedback controller proved to be less tolerant of corrupted models, as was demonstrated by Test 4.2. While the system remained stable, its performance, relative to that of the system with an ideal model (Test 4.1) was much degraded. The introduction of a time delay in the control path also had an extremely detrimental effect on controller performance.

Examination of the results of the tests in Chapters 3 and 4 reveals a marked difference in the form of control signal generated in the feedforward and feedback modes. The oscillatory nature of the control signals generated with feedback control is due to the derivative terms implicit in the inversion algorithm. Since the slope of the feedback signal is changing randomly, the control signal will alternate between successive approximations to the ideal signal. This effect will be accentuated by corruption of the model, and by the use of a first order extrapolator in the feedback path.

In section 4.3 it was noted that tests had been run with intentional errors in model gain. Similar tests were also run with errors in other model parameters. In all cases, the effect on controller performance was slight. Test 4.3 showed that incorrect bias does not seriously degrade the system, provided it is not in such a sense as to introduce right half plane poles in the inversion algorithm.

It would therefore appear that high order corruption terms play a major role in degrading the performance of the feedback controller.

The inability of the system to provide a continuously updated identification of the plant is perhaps the most serious deviation from the original concept set out in Chapter 1. In its present form, the system could only be applied to plants whose dynamics remain substantially constant. The identification could then be carried out off line and results average over a number of identification cycles, yielding a great reduction in model corruption. The advantages of using the weighting sequence formulation noted in Chapter 1 would still be retained. It is possible that in certain cases, a scheme for alternate identification and control phases, similar to that proposed in reference 10 could be implemented. This treatment would, however, only be applicable to a very small class of practical systems, and is not of sufficient generality to warrant consideration in this context.

## 6.2 SUGGESTIONS FOR FURTHER DEVELOPMENT

The simulations which produced the results given in Chapters 2-4 have reached a stage where further expansion on the general purpose machine used would become excessively expensive. As a first step in the continued development of the system, the transfer of the programs to a small digital machine, where time consumption is less of a problem, should be undertaken. This machine could be part of a hybrid computer installation, the plant itself being simulated on the analog section of the computer, or it could be linked to a pilot plant. The former is preferable, as it would allow investigation of a wide range of possible applications.

This initial implementation should be carried out based on fixed weighting sequence models, particular attention being given to producing a program which will run efficiently and use a minimum of storage. This would yield a system which could be directly applied to a number of processes. The speed and storage requirements of the resulting program would enable specific areas of applicability to be defined.

Consideration should then be given to the development of a viable identification for on-line use. Since identification is being undertaken over a limited period in a noisy environment, it will not be possible using known cross-correlation techniques, to achieve a completely accurate identification on each cycle of the test signal in the presence of changing plant parameters, since this would be tantamount to precise identification of the unknown noise signal. In the development of the identification scheme the necessity of obtaining a good model of the control path should be given particular consideration. It is possible that a learning method of identification, where information from a number of past test signal cycles is utilised to produce the final model, which adapts itself slowly to plant parameter changes, would overcome the problems encountered. Since the feedback controller continues to function with inexact models, the system should not break down during the period of adaptation.

Further work is possible in the development of the multi-output scheme which was terminated in Chapter 3. Practicalities will probably limit investigation in this area to the case of a two-output plant, however.

Development of identification techniques for non-linear systems based on multi-level pseudo-random signals could open the way to applying the principles developed herein to such systems.

# A P P E N D I X   I

## NOMENCLATURE

# APPENDIX I

## NOMENCLATURE

| Text symbol | Program symbol (where appropriate) | Description |
|---|---|---|
| a | | Input subscript |
| b | | Output subscript |
| A | | Number of system inputs |
| B | | Number of controlled system outputs |
| $(csb)_i$ | CS1 | Required trajectory at output b |
| $(dmb)_i$ | | Estimated disturbance at output b |
| | DM1P | $= (dm1)_{i+t1}$ |
| gab | G | Steady-state gain of path a-b |
| i | | Suffix denoting present value of variable |
| j | | Suffix used with weighting sequences |
| k | K | Length of test signal cycle (sampling intervals) |
| Ka | KPY2 | Maximum length of any weighting sequence associated with input a |
| (m) | | m-sequence |
| n | N | Order of test signal generating polynomial |

| Text symbol | Program symbol (where appropriate) | Description |
| --- | --- | --- |
| $(nsb)_i$ | ANS1 | Noise component at output b |
| $(nmb)_i$ | ANM1 | Estimated noise component at output b |
| $(nmbp)_i$ | ANM1P | Predicted value of $(nmb)_i$, t1 ahead |
| $(pa)_i$ | IP1,IP2 | Pseudo-random test signal |
| $(rsb)_i$ | | Response at output b to pseudo-random test signals |
| $(rmb)_i$ | RM1 | Computed response at output b to pseudo-random test signals, based on weighting sequence system model |
| $(reb)_i$ | RE1(I2K) | Estimated response at output b to pseudo-random test signals, for use in cross correlation |
| s | | Laplace transform variable |
| T | T | Sampling interval |
| ta | NT1,NT2 | Series time delay in path a-1 |
| $(usb)_i$ | US1 | System output b |
| $(v1)_i$ | V1 | Effective input to inversion algorithm |
| (wsab) | WS11,WS21 | System weighting sequence, path a-b |
| (wmab) | WM11,WM21 | Model weighting sequence, path a-b |
| (weab) | | Biassed weighting sequence estimate |

| Text symbol | Program symbol (where appropriate) | Description |
|---|---|---|
| $(xa)_i$ | X1,X2 | System input a, before addition of $(pa)_i$ |
| $(zsb)_i$ | | Component at system b due to all (xa) |
| $(zmb)_i$ | ZM1 | Computed component at output b due to all (xa), based on weighting sequence models |
| z | | z transform variable |
| (Ø.a,b) | | Cross correlation of a,b |

# APPENDIX II

## DERIVATION OF z TRANSFORM FOR SIMULATION OF THE SYSTEM

# APPENDIX II

## DERIVATION OF z TRANSFORM FOR SIMULATION OF THE SYSTEM

$$w(s) = \frac{w^2}{s^2 + 2sw\cos(\alpha) + w^2}$$

The z transform of this transfer function in series with a zero-order hold is required, i.e. of:

$$G(s) = \frac{1 - e^{-Ts}}{s}.\ \text{(expression for w(s) above)}$$

$$= (1 - e^{-Ts})\left[\frac{1}{s} - \frac{s + w.\cos(\alpha)}{\{s + w.\cos(\alpha)\}^2 + \{w.\sin(\alpha)\}^2}\right]$$

Let a = w.cos (a), b = w.sin (a).

Taking z transforms (e.g. using tables in reference 20):

$$G(z) = (1 - z^{-1})\left\{\frac{z}{z-1} - \frac{z^2 e^{2aT} - ze^{aT}\cos(bT)}{z^2 e^{2aT} - 2ze^{aT}\cos(bT) + 1}\right.$$

$$\left. - \frac{a}{b}\,\frac{ze^{aT}\sin(bT)}{z^2 e^{2aT} - 2ze^{aT}\cos(bT) + 1}\right\}$$

which reduces to

$$G(z) = \frac{\{1 - e^{-aT}\cos(bT) + \frac{a}{b}\sin(bT)\}z^{-1} + e^{-aT} + \frac{a}{b}\sin(bT) - \cos(bT)\ z^{-2}}{1 - 2e^{-aT}\cos(bT).z^{-1} + e^{-2aT}z^{-2}}$$

from which expression the coefficients computed in the subroutine COEFF (Chapter 5) are derived.

# A P P E N D I X   I I I

## BIBLIOGRAPHY

## APPENDIX III

### BIBLIOGRAPHY


1. BARKER, H. A. and HEPBURN, G. A. On line computer control using weighting function models. IFAC Congress, Warsaw, June 1969.

2. WEISNER, J. B. and LEE, Y. W. Experimental determination of system functions by the method of cross correlation. IRE National Convention, New York, 1950.

3. LEE, Y. W. Applications of statistical methods to communications problems. Report No 181, MIT Research Laboratory of Electronics, 1951.

4. ANDERSON, G. W., COOPER, G. R. and BULAND, R. N. The aeroneutronic self-optimising automatic control system. Proc. NADC Flight Symposium, January 1959.

5. HUGHES, M. T. G. and NORTON, A. R. M. Measurement of control system characteristics by means of a cross correlator. Proc. IEE, 1962, 109 (Part B), 77.

6. BRIGGS, P. A. N., HAMMOND, P. H., HUGHES, M. T. G. and PLUMB, G. O. Correlation analysis of process dynamics using pseudo-random binary test perturbations. Proc. I. Mech. E., 1964-65, 179, Part 3H, p 37.

7. NIKIFORUK, P. N. and GUPTA, M. M. A bibliography on the properties, generation and control system applications of shift-register sequences. Int. J. Control, 1969, Vol. 9, No 2, p 217.

8. ANDERSON, G. W., BULAND, R. N. and COOPER, G. R. Use of cross correlation in an adaptive control system. Proc. National Electronics Conference, 1959, Vo. 15, p 31.

9. DAWKINS, J. and BRIGGS, P. A. N. A method for using weighting functions as system description in optimal control. Proc. 2nd IFAC Symposium on Theory of Self-adaptive Control Systems, September 1965, p 234. Published by Inst. Soc. America.

10. BECK, M. S., BIRCH, P. R., GOUGH, N. E., PLASKOWSKI, A. A predictive control scheme for dead-time processes using a learning method of process identification. IFAC Congress, Warsaw, June 1969.

11. GOLOMB, S. W., BAUMENT, L. D., EASTERLING, M. F., STIFFLER, J. J., VITERBI, A. J. Digital communications with space applications, Prentice-Hall, 1964.

12. PETERSON, W. W. Error correcting codes. MIT Press and J. Wiley & Sons, New York, 1961.

13. EVERETT, D. Periodic digital sequences with pseudo noise properties. GEC Journal, Vol. 33, No 3, p 115, 1966.

14. BRIGGS, P. A. N. and GODFREY, K. R. Pseudo-random signals for the dynamic analysis of multivariable systems. Proc. IEE, 1966, Vol. 113, Part 7, p 1259.

15. BARKER, H. A. and RAESIDE, D. Linear modelling of multivariable systems with pseudo-random binary input signals, Automatica Vol. 4, p 393, 1968.

16. BARKER, H. A. Choice of pseudo-random binary sequences for system identification. Electronics Letters, 1967, 3, p 524.

17. REAM, N. Proof of the drift-resistant property of binary m sequences. Electronics Letters, 1968, 4, p 380.

18. DAVIES, W. D. T., DOUCE, J. L. On line system identification in the presence of drift. FAC Symposium on the Problems of Identification in Automatic Control Systems, Prague, June 1967.

19. BARKER, H. A. The reference phase of pseudorandom signals. Proc. IEE, Vol. 116, No 3, March 1969.

20. RAGAZZINI, J. R. and FRANKLIN, G. F. Sampled data control systems, McGraw-Hill, 1968.

21. JURY, E. I. A simplified stability criterion for linear discrete systems. Proc. IRE, 50, 1493 (1962).

22. ROSENBROCK, H. H. On the design of linear multivariable control systems. Proc. 3rd IFAC Congress, London 1966.

23. THALER, G. J. and BROWN, R. G. Analysis and design of feedback control systems. McGraw-Hill, p 200.

24. Univac 1108 Fortran V Programmers Reference. UP-4060, Rev. 1.