'MULTI-PRODUCT, MULTI-LEVEL PRODUCT CONTROL SYSTEM ANALYSIS'

BY

KEITH POPPLEWELL, B.Sc.

Thesis submitted to the University of Nottingham for the Degree of Doctor of Philosophy, October, 1980.

CONTENTS

Contents (cont)

Contents (cont)

Contents (cont)

ABSTRACT

Several techniques are applicable to the modelling of production and inventory control systems. In this thesis discrete linear control theory is examined as a method of modelling multi-product, multi-level systems. These systems are categorised and a general discrete linear control model is used to determine system stability and to predict system responses to specific patterns of input information. The response of the system to random variability in input or other system variable is also shown to be predictable. A library of sub-system models is provided and the method is illustrated by examples and a case study.

Alternative modelling techniques rely upon sequential simulation, either directly or in solving equations representing the system. The need to include forecasting, inventory and production decision-making procedures makes such models large and their sequential nature imposes the need for complete re-modelling for each system modification and for each input pattern. Where random effects are modelled, protracted runs are necessary to achieve statistically acceptable results. In contrast, discrete linear control theory provides a non-sequential model, thereby alleviating these problems. Thus it is possible both to reduce computing expense and increase the range of systems susceptible to manual analysis. The method is limited by the restriction of linearity, but, in many practical situations this restriction poses no insuperable difficulty in the interpretation of results.

ACKNOWLEDGEMENTS

My thanks to my supervisor Maurice Bonney; to T.I. Raleigh Ltd., for their support to this project; to Kay Mell for her patience and typing skill, and to Alison Popplcwell for the hours spent proof-reading.

CHAPTER 1

INTRODUCTION

1.1 Objective

Production and inventory control in a multi-product, multi-level environment is necessarily complex and analysis and design of such systems is greatly assisted by suitable modelling. The techniques frequently used (e.g. simulation) provide adequate but complex models which are usually completely computer orientated. Such models are often cumbersome in use and, because of their computer dependence, give only limited insight into the modelled system. There would therefore be considerable value in a modelling technique which can be applied rapidly and simply, whilst also being capable of deep and detailed analysis where this is appropriate.

Because of its ability to handle a complete time-series as a single entity, discrete linear control theory may provide such a technique. We shall examine its applicability to the problem of multi-product, multi-level production and inventory control. Our objective will be to establish both the range of possible application of such models, and the form (and so usefulness) of their results.

1.2 Synopsis

We shall discuss in some detail the class of production and inventory problems which concern us (1.3). We consider those multi-product, multi-level systems where decisions are taken on a fixed time-cycle. The method restricts us to linear systems but we suggest that this is not so serious a restriction as to invalidate the technique.

Alternative techniques which may be applied to these systems (e.g. simulation, industrial dynamics) are briefly examined (1.4). Discrete linear control theory is fully described in the literature [Bishop, 1975; Muth, 1977] and we include only basic definitions and the results which are needed in practical analysis (Chapter 2). These are developed into routine techniques for system modelling. The models so derived allow determination of system stability, in many cases by inspection. For more complex cases the reader is directed to mathematical tests of stability. The models may also be used to examine transient responses to standard input patterns and to determine performance in response to stochastic inputs.

We demonstrate the use of these techniques in some detail with the aid of a simple, single-level, single-product example (Chapter 3). A general model is then constructed; this is capable of extension to the full range of production and inventory control systems under consideration (Chapter 4).

Our general technique is only widely applicable if it is possible to model, in detail, a broad range of individual system elements (e.g. forecasting mechanisms, scheduling rules). We therefore give a set of such models (Chapter 5), both for direct use where applicable and to demonstrate the ease with which system element models can be built.

Chapter 6 shows an example of the use of both the general model of Chapter 4 and some of the system elements developed in Chapter 5. Besides its use in detailed and comprehensive system analysis, our technique can be applied as a very rapid, simple, "back of an envelope" method. We illustrate this with a description of an actual application of the development of a relatively simple system which was needed quickly and where only incomplete information was available (Chapter 7).

A simple interactive computer program designed to carry out the algebraic manipulation of system modelling is presented (Appendix I). Proofs of the results of Chapter 4 are given (Appendix II).

Our conclusion (Chapter 8) is that the z-transform technique has, despite the constraints of discreteness and linearity, a significant area of application. It should be considered as an alternative to simulation or industrial dynamics wherever these constraints are satisfied. Further work based on the theory of z-transforms may relax these constraints. We believe that use of the techniques described here may also ease further research into the general performance of production and inventory systems.

1.3 The Class of Systems Considered

Discreteness

We consider only those production control systems where monitoring and decision occurs at fixed, discrete points in time. Thus we include "periodic review" systems but exclude the "reorder point" systems, where action is triggered by some continuous variable reaching a critical level.

Both reorder point and periodic review systems are in common use. The latter are particularly suited to use in computer based systems where a regular processing timetable is often imposed. We are therefore addressing a significant range of systems.

It is possible that in some areas real-time computer production control systems may cause greater use of reorder point systems. It is, however, unlikely that more than a few periodic review systems will be replaced.

Linearity

A system is said to be *linear* if all relationships between variables are linear. This is a significant limitation on the use of the method since almost all systems in the real world are at best only linear within certain limits. For example stocks cannot actually become negative. It is however often possible to interpret behaviour beyond the limits of linearity without invalidating the use of a linear model - negative stocks may be interpreted as a backlog of orders. Even where such interpretation is not valid a linear model may still be of value for many systems are intended to remain in the linear region and are considered to have failed when they become non-linear (e.g. when the order backlog becomes so great as to affect incoming orders). In systems such as these a linear model may be sufficient in predicting the circumstances in which an undesired departure from the region of linearity will occur.

The value of linear modelling techniques is also increased when we consider the findings of Schneeweiss [1975] and Inderfurth. They suggest that, in many cases, it is possible to derive a linear decision rule which gives equivalent or only slightly poorer results than an optimal non-linear process. In the light of this it is advisable to always consider the possibility of using a linear system before adopting a more complex non-linear system as the performance advantage of the latter may well be insignificant.

Multi-level, multi-product systems

A multi-level, multi-product system controls the flow of materials, components, assemblies and finished products through manufacturing facilities. The products may share common parts at any level of manufacture. Many such systems exist, particularly in the consumer durable industries.

Much theoretical work has been devoted to single-level; single-product environments, or to the control of individual parts within a multi-level, multi-product system. In particular, none of the earlier work using linear control theory has gone beyond the single-level, single-product case. Simon [1952] and Campbell [1953] both developed continuous inventory control models using Laplace transforms, whilst Vassian [1955] uses a discrete model employing geometric transforms and Elmaghraby [1959] uses z-transforms in a highly theoretical way. Pinkham [1958] extends Vassian's work to produce optimal balance between production smoothing and inventory smoothing costs. All of these essentially present an example illustrating the use of control theory techniques and point the way to a deeper investigation of the applicability of these techniques.

Sargent [1966] follows this direction and analyses in detail a more complex inventory control system. He illustrates the practical use of a discrete linear control model and emphasises the analysis of a sophisticated forecasting procedure and its effect upon the whole system. It remains, however, a single-level model.

The general model we present in Chapter 4 embodies a broad range multi-product, multi-level systems. It can handle systems where each level is controlled independently to meet demands from the level above, integrated systems where control at each level uses information about final product demand and mixed systems where some parts are controlled one way and some the other. The model can represent single-level cases and indeed we use single-level examples to illustrate the application of some of the techniques, as well as presenting a multi-product, multi-level example.

We note that, in particular, "material requirements planning" systems [Wight, 1974] fall within the scope of the general model up to the point of generation of net requirements. A full material requirement planning system may also apply non-linear scheduling rules to meet supplier constraints. However, we believe that a model of the linear part of the system is of value and that the evaluation of the supplier scheduling phase of material requirements planning may be based upon the known performance of net requirements generation.

1.4 Alternative Techniques

A number of techniques are available for use in modelling production control systems. Probably the most commonly used of these is simulation, either manual or computer based. A simulation model can be very powerful, allowing detailed modelling of every known aspect of the control system and its environment and generating quantitative analysis of any variables within the system. There is no difficulty in handling non-linear systems. Manual simulations are especially useful as demonstrations both as a means of model validation involving line management and as a means of training all levels of personel in the operation of a system to be implemented.

Unfortunately manual simulation becomes impractical for large and complex systems : the model is simply too slow to operate. Use of computer simulation models relieves this problem and increases the size and complexity of system which can be analysed. However, the demonstration value of the model is, to a great extent, lost. Many computer simulation packages are available commercially and it is seldom difficult to select a package appropriate to the system to be modelled. Typically such packages compile statistics automatically with a minimum of user effort. Random number generators within the package facilitate modelling of stochastic processes.

Difficulties still devleop, however, as size and complexity increase. Simulation packages make heavy demands on core capacity so requiring use of large and expensive computers. The main source of expense is, nevertheless, not the cost of an individual run but the necessity for repeated runs. Because a simulation is numerical in nature, it is necessary to completely rerun the simulation if any system parameter is changed. Furthermore, each run must be fairly long to be statistically valid. Thus to determine an acceptable level of, say, a smoothing constant the simulation must be rerun for a range of values. The necessary number of runs quickly becomes prohibitive where several such parameters are inter-dependent. Computer simulation also tends to be clumsy in examining deterministic responses to standard input patterns.

An alternative to digital simulation, suggested by Lewis [1963, 1967(2)], is the use of analogue and hybrid-analogue computer models. It is significant that in this method we see the experimental tools originally conceived for analysis of physical, engineering control systems brought to bear upon production control systems. In the use of discrete linear control theory we shall see the equivalent theoretical tools transferred in the same way.

Analogue simulation is also restricted as to the size and complexity of system which can be modelled as analogue computers have limited capacity. Very large machines would be needed to model multi-product, multi-level systems. Furthermore analogue computers operate in a continuous manner. They can simulate discrete systems either by continuous approximation or with the aid of "sample hold" mechanisms. There are limits to the precision of results and although in some cases qualitative results may be sufficient this is a significant problem.

Forrester [1961] presents systems dynamics as an all-embracing modelling system. Here a system is modelled as a set of mathematical relationships between state variables called "levels" and "rates". The model is analysed by simulating the action over time of these relationships using the methods of numerical analysis. Coyle [1977] examines the theory of systems dynamics and the methods used to simulate model responses. Computer packages are readily available to carry out the numerical simulations on a constructed model.

Despite the breadth of scope claimed by Forrester for systems dynamics, its use is not universal in modelling. The dialogue between Ansoff et. al. [1968] and Forrester [1968] suggests some of the limitations. In some contrast Roberts [1977] presents a collection of papers on applications of systems dynamics. Unlike simulation, systems dynamics models do not lend themselves to demonstration as they are difficult to present comprehensibly to a layman. The equations and diagrams describing the relationships within the system tend to be esoteric, and it seems strange to use a discrete simulation to analyse a continuous model representing a discrete system. Foo [1978] presents a direct comparison of systems dynamics and discrete linear control theory using simple examples. Although he reaches no conclusions he lists his perception of the relative merits of the two techniques.

Holt et. al. [1968] and Bensoussan [1972] use the principle of optimality together with mathematical models similar to those of systems dynamics. In particular optimal lot sizes are derived in examples. A great deal of work has followed from this [Bensoussan, 1972(2); Crossley, 1972; Bradshaw, 1973, 1974, 1975, 1976; Axsater, 1978]. Most of this work is however of a highly theoretical nature, and does not directly provide a detailed modelling technique.

CHAPTER 2

DISCRETE LINEAR CONTROL THEORY

2.1 Preamble

In this chapter we present the definitions and properties of z-transforms and z-transfer functions relevant to the remainder of the thesis. No attempt is made to prove properties as adequate proofs appear elsewhere [Muth, 1977; Jury, 1964]. We also show the technique of system reduction demonstrated in full by Howard [1963].

2.2 The z-Transform

If g(t) is a time-series having value g(t) at time t, then the z-transform of g(t) is defined as:-

$$g(z) = \sum_{t=0}^{\infty} g(t) z^{-t}$$

where z is a complex number.

We shall frequently abbreviate g(z) to g. The time-series in the t-domain will always be distinguished by the presence of the argument.

Thus a time-series z-transform is a power series in z^{-1}. Tables of z-transforms of most standard time-series are published [Beightler et. al, 1961]. These frequently show simplifications of the power series to shorter expressions.

2.3 Properties of the z-Transform

Linearity

Let $g(t) = h(t) + k(t)$ be time-series. Then:-

$$g(z) = h(z) + k(z)$$

Let a be a real scalar, and $g(t) = ah(t)$. Then:-

$$g(z) = ah(z)$$

Translation

Let δ be an integer, and let $g(t) = h(t-\delta)$ be time-series.

Then:-

$$g(z) = z^{-\delta}h(z)$$

2.4 The z-Transfer Function

Let T be some linear system operating upon an input time-series $h(t)$ to generate the transformed output time-series $g(t)$. This is illustrated in Figure 2.4.a.

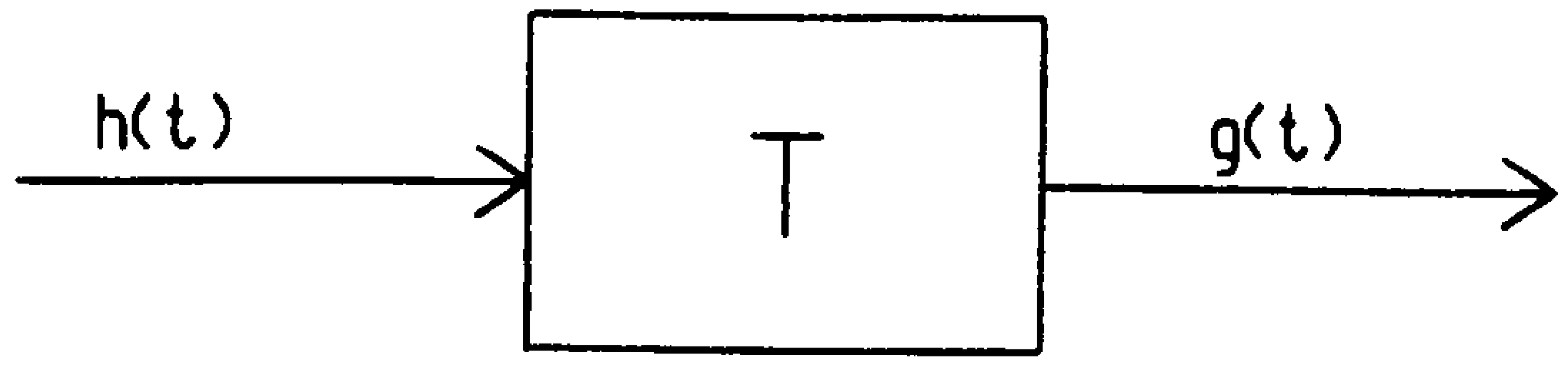

Figure 2.4.a : A Linear System.

Then we define $T(z)$, the z-transfer function of T, to be the z-transform of $g(t)$ when $h(t)$ is the unit impulse at time $t=0$ (i.e. $h(t) = 1$ when $t = 0$, and $h(t) = 0$ otherwise).
We shall frequently abbreviate $T(z)$ to T.

2.5 Properties of the z-Transfer Function

Transformation

For any input time-series h(t) to system T the z-transform of the output time-series g(t) is given by:-

$g(z) = h(z)T(z)$

Linearity

If T generates output time-series g(t) in response to the input of the sum of time-series h(t) + k(t) then:-

$g(z) = (h(z) + k(z))T(z) = h(z)T(z) + k(z)T(z)$

If T generates output time-series g(t) in response to input time-series ah(t) for some scalar a then:-

$g(z) = (ah(z))T(z) = a(h(z)T(z))$

Serial Transformation

Let S and T act serially on input time-series h(t) to produce output g(t) (Figure 2.5.a), and let (ST) represent their action.

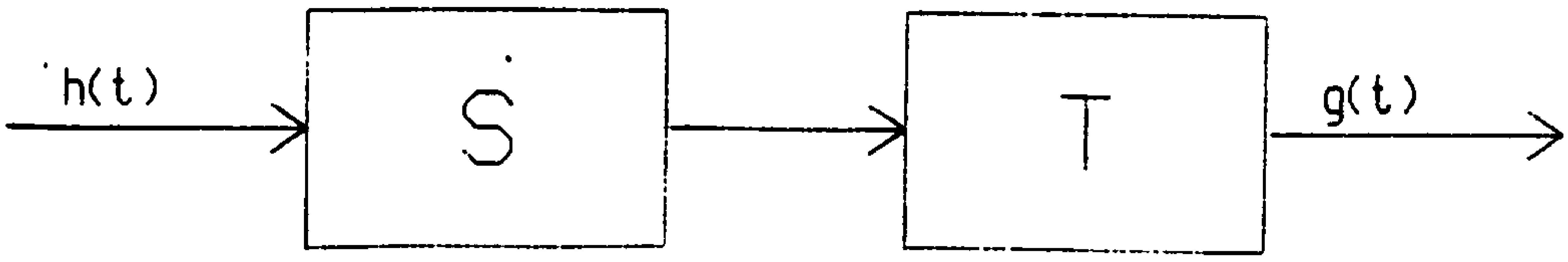

Figure 2.5.a : Serial Transformation

Then we have:-

$g(z) = h(z)(ST)(z) = h(z)S(z)T(z)$

Thus the z-transfer function representing the successive action of two transformations is the product of the z-transfer functions of the two transformations.

Commutation

If S(z) and T(z) are z-transfer functions and g(t) a time-series then:-

$$g(z)S(z) = S(z)g(z)$$

and:-

$$g(z)S(z)T(z) = g(z)T(z)S(z)$$

Derivation

If g(t) is known to be the output of transformation T when h(t) is input then we can derive the z-transfer function T(z) as:-

$$T(z) = \frac{g(z)}{h(z)}$$

2.6 System Reduction

We shall represent systems as block diagrams in which boxes represent transformations and the flows between boxes represent time-series. Addition or subtraction of time-series shall be represented thus:-

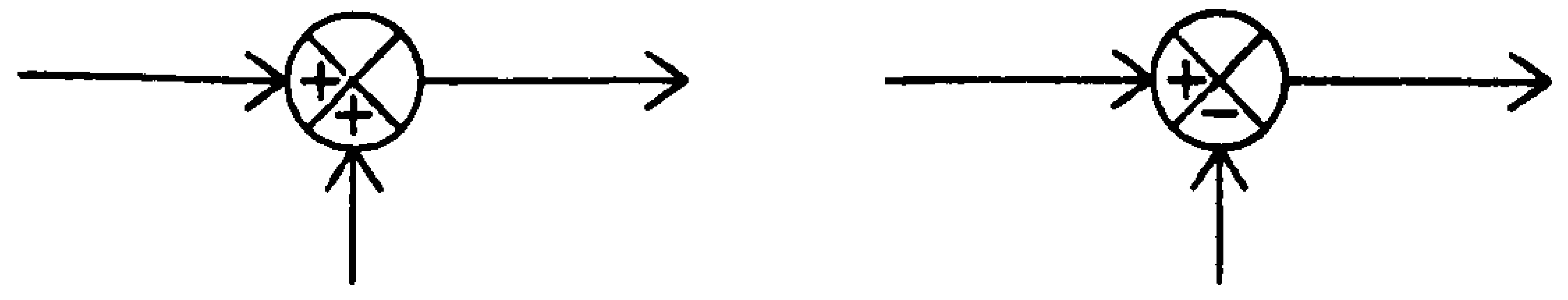

Figure 2.6.a : Time-Series Addition & Subtraction

Where a time-series is input to more than one transformation this is represented by a branch in the flow:-

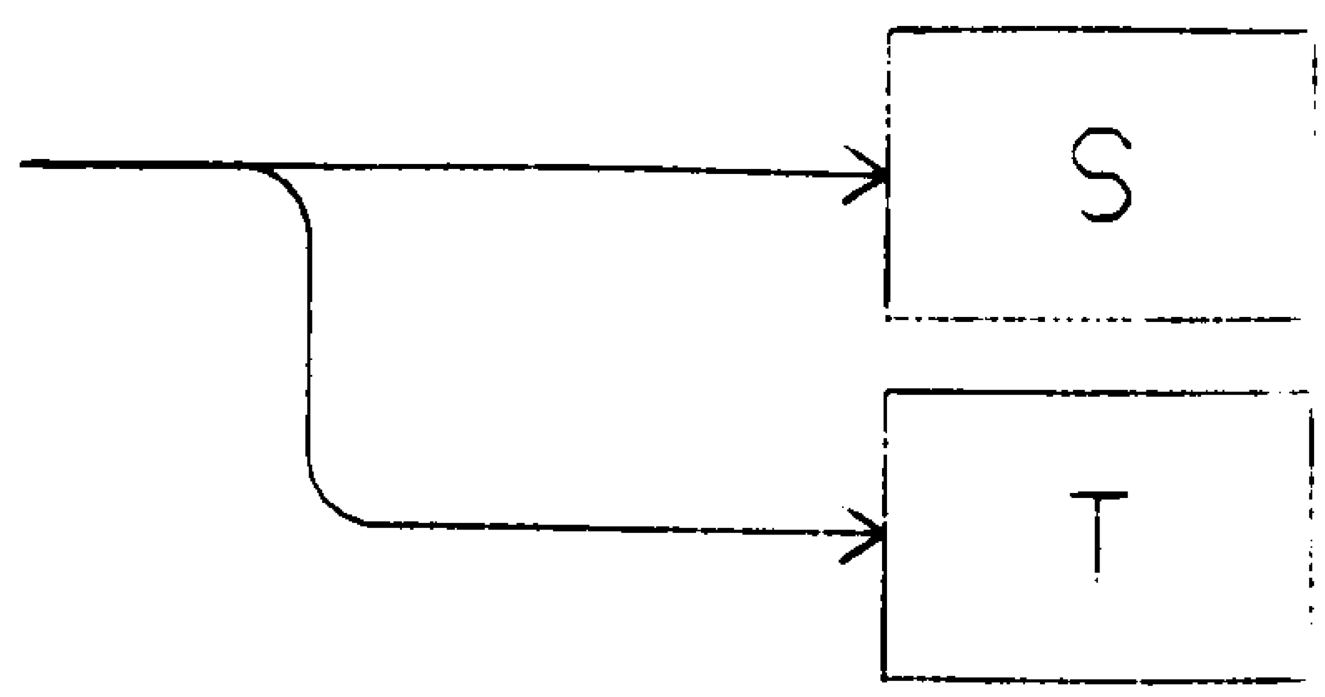

Figure 2.6.b : Time-Series Input To Two Transformations

Using such diagrams we can represent any linear system. Depending upon the complexity of the system there may be transformations acting in series or on parallel branches and loops; the latter may well be nested.

Following Howard [1963] we can reduce any such system to a single z-transfer function compounded of the z-transfer functions of the elementary transformations of the system. Such reduction uses the properties of the z-transform and z-transfer function listed above (Sections 2.3 & 2.5). The process is explicitly described by Howard, who also demonstrates an alternative diagrammatic representation using signal flow graphs. Chapter 3 demonstrates in detail the reduction of a system, using these techniques.

2.7 System Stability

We define a system to be stable if its response to any input impulse is bounded in time. That is, given any $\delta > 0$ we can choose a finite t' such that the response $g(t)$ to an input impulse will satisfy $g(t)<\delta$ for all $t > t'$.

The definitions of z-transfer function (2.4) and z-transform (2.2) imply that the z-transfer function is a power series in z^{-1}. Such a power series can in turn be expressed as a quotient of polynomials $\frac{N(z)}{D(z)}$. We define the characteristic equation of a system to be $D(z) = 0$.

It is a necessary and sufficient condition of system stability that the roots of the characteristic equation lie within the unit circle in the z-plane. (For proofs see Jury [1964] and Muth [1977]).

Where it is possible to solve the characteristic equation it is, therefore, a simple matter to determine system stability. It is not always possible however to explicitly solve a polynomial equation of degree greater than 2. When this problem arises it is possible to use algorithms establishing the location of equation roots relative to the unit circle. Such algorithms are described by Truxall [1955].

2.8 Transient Responses

For any standard input time-series we can obtain the z-transform either by definition (2.2) or from z-transform tables [Beightler, 1961]. The z-transform of the output time-series is then the product of input z-transform and system z-transfer function. To determine the response time-series we invert its z-transform either by using tables or, where the z-transform appears as a power series in z^{-1}, by using the z-transform definition. Bishop [1975] demonstrates the application of several transient response criteria and the selection of system parameters.

2.9 Noise Response

Noise input to a system can be described as a time-series of impulses, selected at random, according to some statistical distribution. The response to each individual impulse is described by the system z-transfer function (2.4) and, for a stable system, is bounded in time. Thus for any required precision we can describe, in the z-domain, the response to the noise impulse x(t) at time t as:-

$$a_0x(t) + a_1x(t)z^{-1} + \ldots + a_nx(t)z^{-n}.$$

All earlier and later impulses have similar responses and so the resultant response seen at a given time t is:-

$$a_0x(t) + a_1x(t-1) + \ldots + a_nx(t-n).$$

Let us assume that the noise distribution is time-independent. Then x(t),...,x(t-n) are all selected from the same distribution, so the terms $a_0x(t),\ldots,a_nx(t-n)$ are each selected from a scaled copy of this distribution.

However, the distribution of a random variable derived as the sum of random variables is, by definition, the convolution of the distributions of the summands.

Hence we have a simple method of determining system responses distributions to input random noise;

(i) Express the system z-transfer function as a power series in z^{-1}.

(ii) Limit that power series by setting a limit of significance on the coefficients. This is always possible for a stable system.

(iii) Write down for each term a copy of the input distribution scaled by the coefficient of the power of z.

(iv) Convolve these to give an output distribution.

This process is well-defined and finite, though it may, depending upon the degree of precision required, be tedious. It is, however, susceptible to computer assistance (e.g. Appendix I).

CHAPTER 3

A SIMPLE INVENTORY CONTROL SYSTEM

3.1 Preamble

Here we consider a very simple inventory control system. Our intention in doing so is to illustrate the techniques described above (Chapter 2).

A diagrammatic representation of the system is given (3.1.a). Stock is held to meet customers' demands and is replenished by deliveries against schedules placed upon a supplier. A single product is considered and no interest is taken in the supplier's problems in meeting delivery schedules.

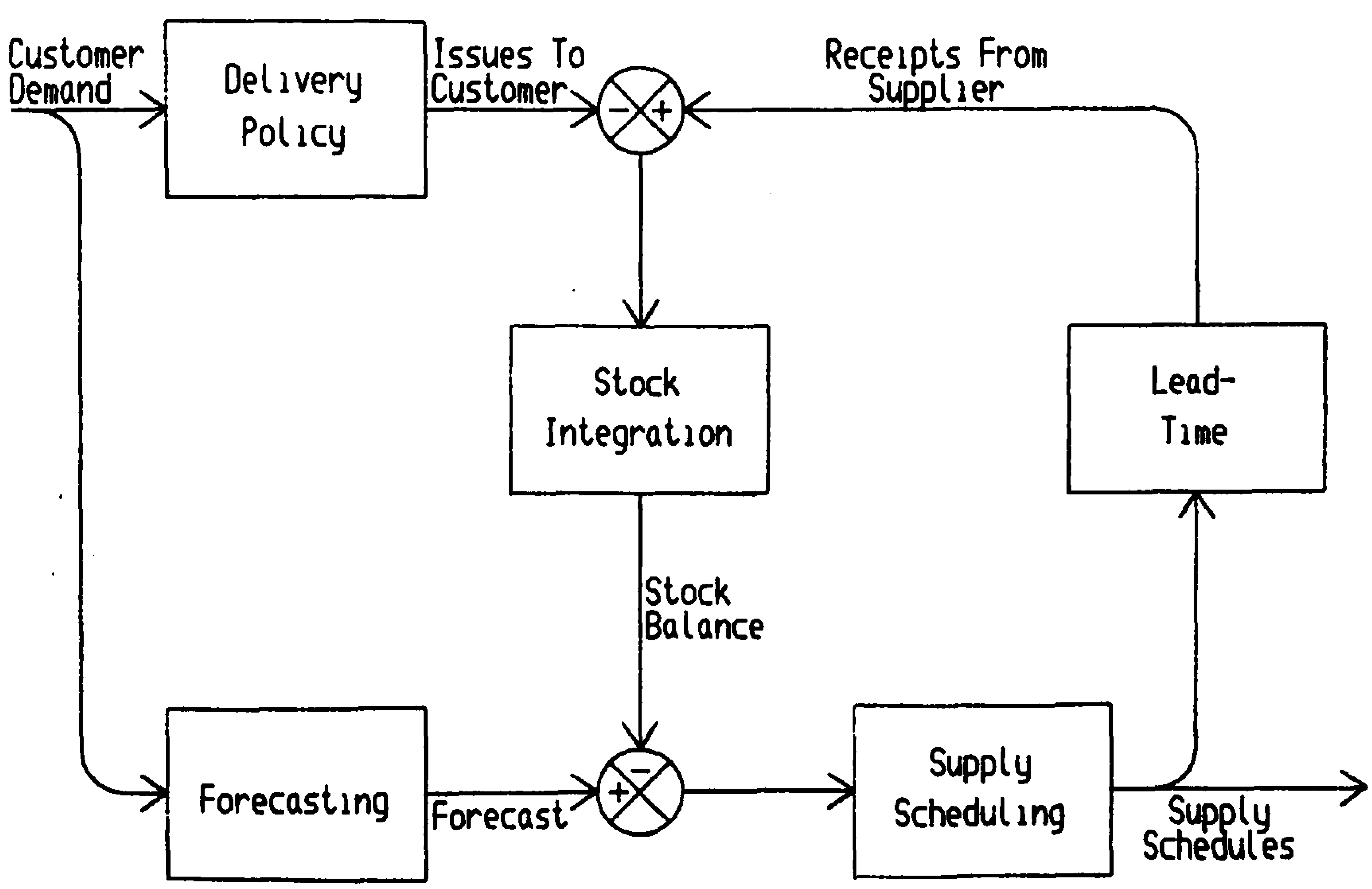

Figure 3.1.a : A Simple Inventory Control System.

Customers' demand is shown as input to the system and is subject to the system's policy on delivery. The current stock together with some forecast of demand must be taken into account in raising schedules to place upon the supplier. These schedules will be met after some lead-time, and result in receipts into stock. Receipts and issues to customers are integrated to determine the new stock.

3.2 The Model

It is clear that customer demand, issues to customers, forecasts, current stock balances, delivery schedules and receipts into stock are all time-series and thus that delivery policy, forecasting, scheduling, lead-times and stock integration constitute transformations of time-series. The z-transforms of the time-series are represented symbolically in Figure 3.2.a.

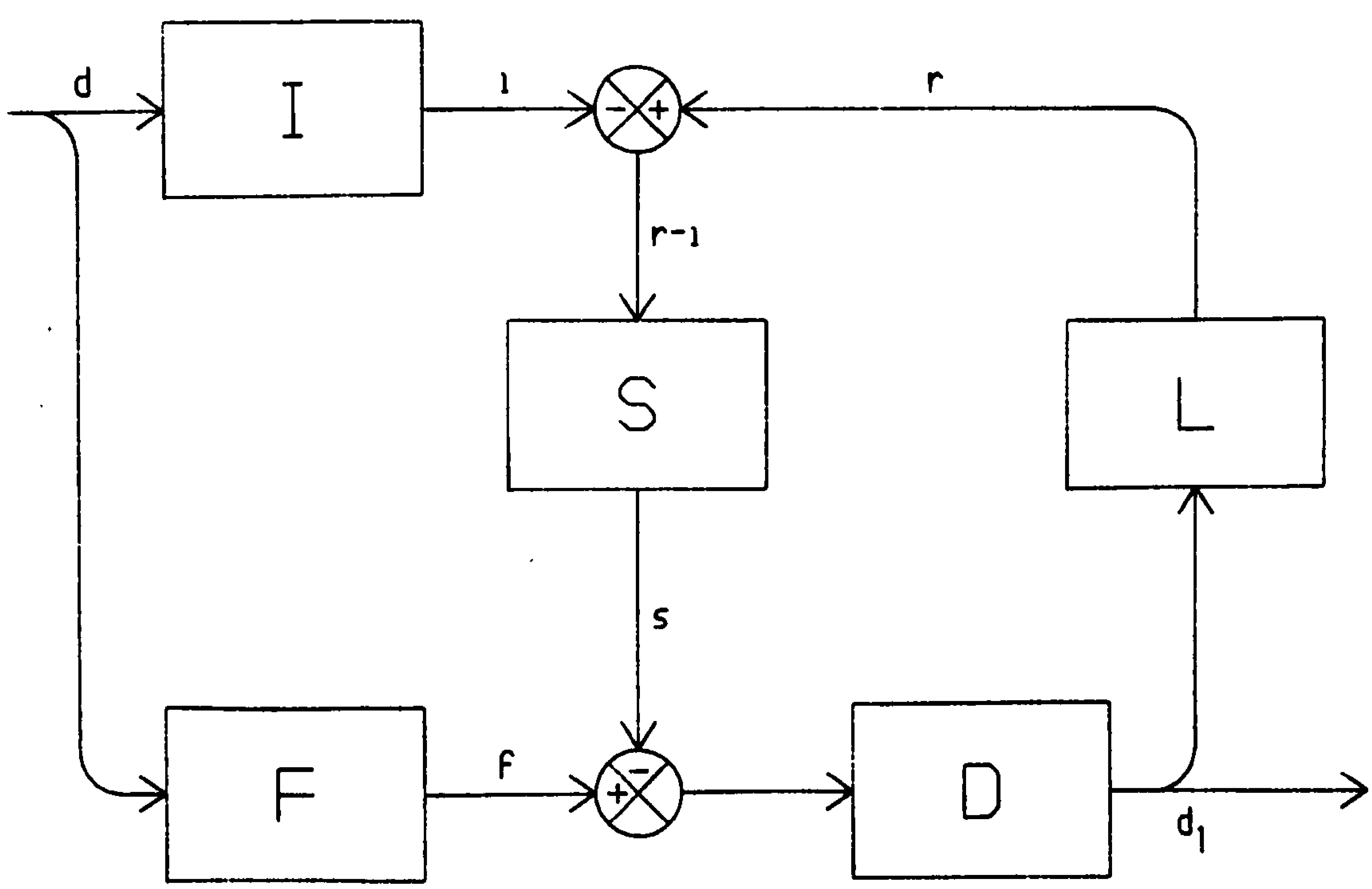

Figure 3.2.a : A Simple Inventory Control System (Symbolic Representation)

Customer Demand is represented by d.

Issues to Customers are represented by i. In its simplest form this time-series is merely a delayed version of d.

Forecast Gross Requirements are represented by f. When the delivery policy lead-time is equal to the supplier's schedule lead-time this can be identically equal to d. However the policy lead-time will normally be less than the supplier's lead-time, in which case some assumption must be made concerning future demand - predictions must be used for as yet unknown elements of d. It is possible that a forecast based on data other than past demand may be used. It would then be necessary to provide a second input to the system in which case the use of this model, and the general model described later, would be slightly modified.

Supplier's Delivery Schedules are represented by d_1. The letter d is used here as well as above because the delivery schedules output by this system are the demands input to the supplier's control system.

Receipts from Supplier are represented by r.

Stock Balance is represented by s.

The z-transfer functions of the system elements are also represented in the Figure (3.2.a).

Delivery Policy is represented by I. It will contain lead-time delays and any systematic spreading of deliveries.

Forecasting is represented by F. This generates a forecast gross requirement suitable for input to net scheduling.

Net Scheduling is represented by D. This receives as input the difference between a requirement generated by F and the current stock balance. D carries out schedule netting and any necessary delays and smoothing. A very close link exists between F and D. They are together the forecasting and scheduling routines and are better considered in this way than as two individual functions. They are separated by the point in their arithmetic where the current stock balance is subtracted from the gross requirement.

Stock Integration is represented by S. It receives as input the time-series r - i (the resultant of receipts and issues).

Supplier's lead-time is represented by L. This may be a simple delay, or, for a more difficult supplier, may contain some spreading, proportional change or other transformation.

We shall now derive system z-transfer functions for our system. Firstly we consider the z-transfer function from customer demand to delivery schedules. It is clear from Figure 3.2.a that:-

$$d_1 = (f - s)D$$
$$= dFD - (r - i)SD$$
$$= dFD - d_1LSD + dISD$$
$$= d(FD + ISD) - d_1LSD$$
$$\therefore d_1(1 + LSD) = d(FD + ISD)$$
$$\therefore d_1 = d\left(\frac{FD + ISD}{1 + LSD}\right)$$

Thus the system z-transfer function is:-

$$T(d,d_1) = \frac{d_1}{d} = \frac{FD + ISD}{1 + LSD} \qquad (3.2.1)$$

It should be remembered that this derivation has been carried out entirely in the z-domain. In the t-domain such manipulation of time-series and transformations is invalid.

Stock balance is a time-series internal to the system. However it is important, for system design, to be able to predict its behaviour. From Figure 3.2.a we deduce:-

$$s = (r - i)S$$
$$= d_1LS - dIS$$
$$= (f - s)DLS - dIS$$
$$\therefore s(1 + LSD) = d(FDLS - IS)$$
$$\therefore s = d\left(\frac{FDLS - IS}{1 + LSD}\right)$$

and so we have the system z-transfer function from customer demand to stock balance:-

$$T\ (d,s) = \frac{s}{d} = \frac{FDLS - IS}{1 + LSD} \qquad (3.2.2)$$

It must be remembered that, because the system is linear, stock may notionally become negative. This must be interpreted as the backlogging of orders. The effect of such backlogging, combined with the delivery policy, constitutes the system's service to the customer. As long as stock remains negative delays are imposed over and above the delivery policy determined for the system. It is possible, in certain environments, that such an additional delay may affect later customer demand, in which case a further feedback loop must be considered.

Both the system z-transfer functions derived above contain the same expression as denominator, namely 1 + LSD. This is inevitable as both d_1 and s lie on the same loop in the system, in this case the only loop.

3.3 The Effects of Random Noise

In a real environment random noise may be imposed at any or all of several points within the system. Although, in a stable system, recovery from any disturbance must eventually occur, evaluation of the system must include prediction of the effects of such noise. The distributions imposed upon time-series as a result of random noise generated elsewhere are critical to such factors as the necessary size of safety stocks, delivery performance to customers and the smoothness of delivery schedules. Since we are dealing with a linear system it is possible to consider each potential source of noise independently as their effects are strictly additive. We shall derive, as examples, system z-transfer functions to delivery schedules and current stock balance from three points where significant noise may be introduced.

Supplier Performance

We shall consider first the effects of random variation in the supplier's performance against our schedule. This is the most likely source of noise as it is beyond the control of those responsible for operating the system. Furthermore, it is the recipient of blame for most failures and its potential as a disrupting influence should be fully understood.

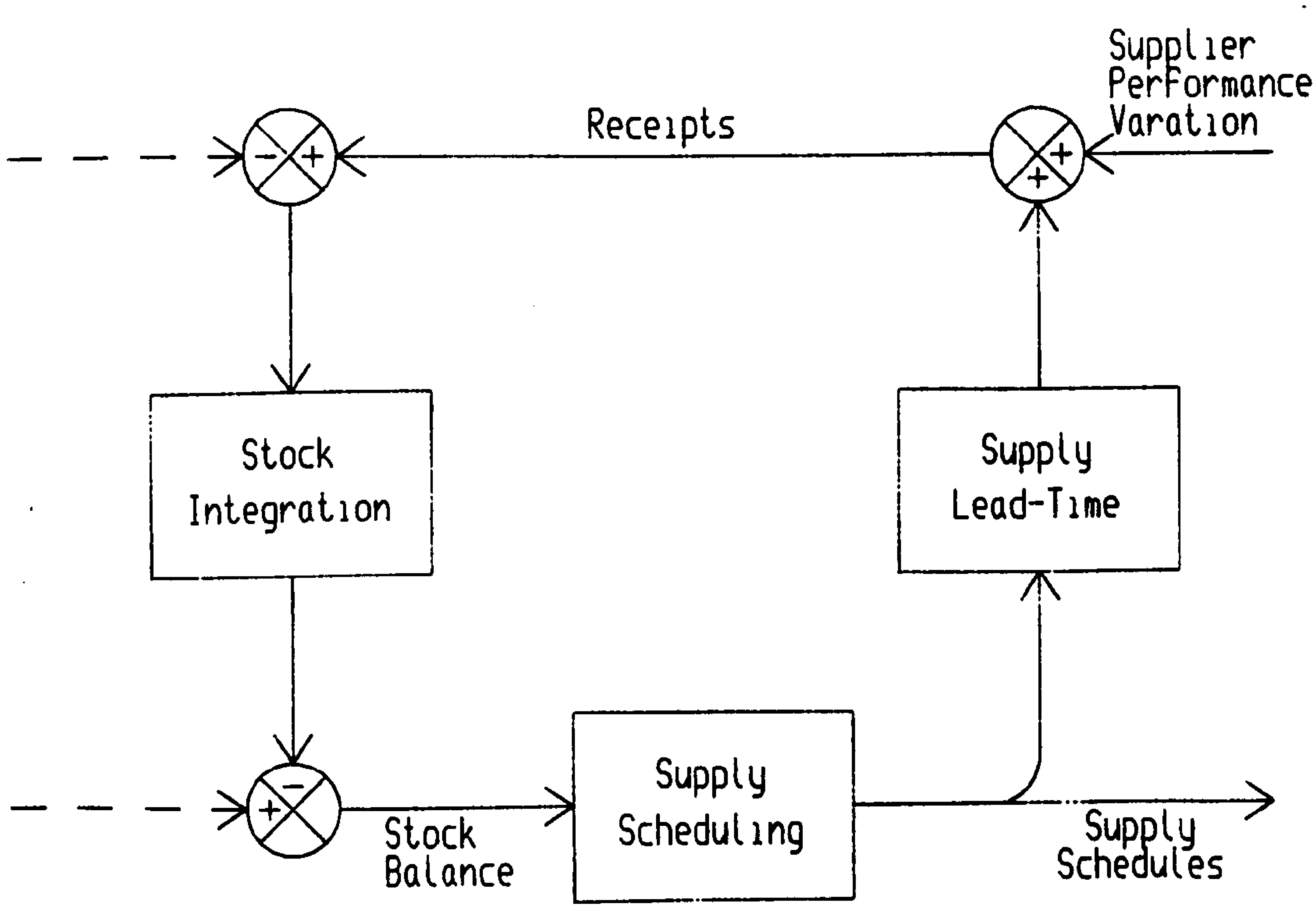

Figure 3.3.a : System Subject To Supplier Performance Variation

As the system is linear, customer demand, issues to customers and forecasts may be considered identically equal to zero without loss of generality. We may now reconstruct Figure 3.1.a omitting these and inserting the input noise which is added to receipts from supplier to form Figure 3.3.a. When the system elements are represented by their z-transforms (3.3.b) we can deduce:-

$$\begin{aligned} d_1 &= -sD \\ &= -rSD \\ &= -(\tilde{r} + d_1L)SD \\ &= -\tilde{r}SD - d_1LSD \end{aligned}$$

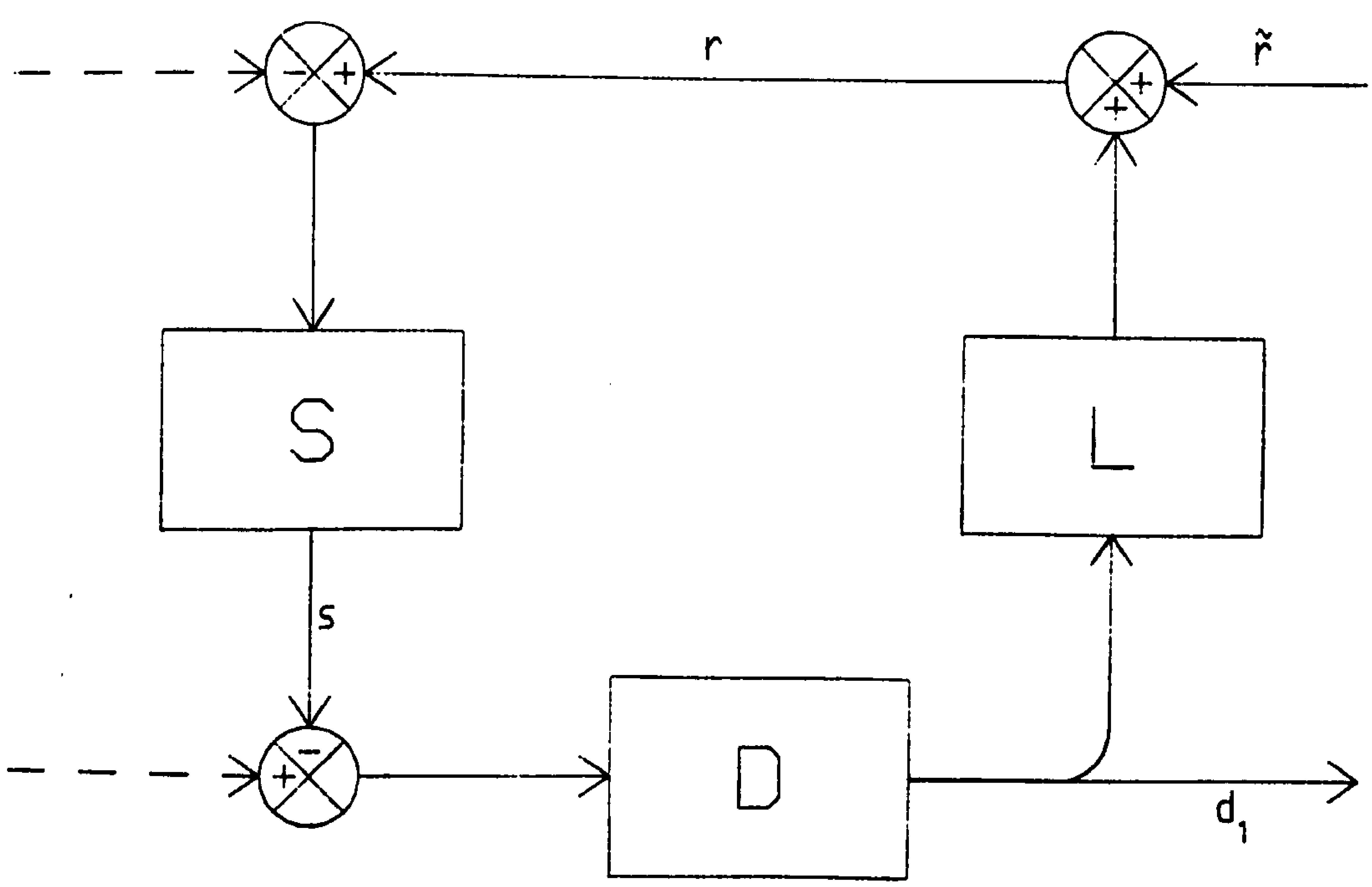

Figure 3.3.b : System Subject To Supplier Performance Variation (Symbolic Representation)

$$\therefore \quad d_1(1 + LSD) = -\tilde{r}SD$$

$$d_1 = \frac{-\tilde{r}SD}{1 + LSD}$$

So we have:-

$$T(\tilde{r},d_1) = \frac{d_1}{\tilde{r}} = \frac{-SD}{1 + LSD} \qquad (3.3.1)$$

This system z-transfer function represents the supplier's view of his customer as a "black box" which responds to any variation from schedule on the supplier's part.

Similarly, we can derive the system z-transfer function from $\tilde{r}$ to s so as to be able to describe the response of stock balance to delivery variability:-

$$T(\tilde{r},s) = \frac{S}{1 + LSD} \qquad (3.3.2)$$

By applying this to an expected distribution of deliveries we obtain the resulting distribution of stock balance and can thus set a safety stock to achieve any desired level of performance. If other random effects in the system also affect stock balance then all such distributions may be convoluted (because of system linearity) to provide a resultant stock balance distribution.

Stock Record Errors

Recording errors may be considered as a source of noise applied to stock balance. This is illustrated by Figure 3.3.c.

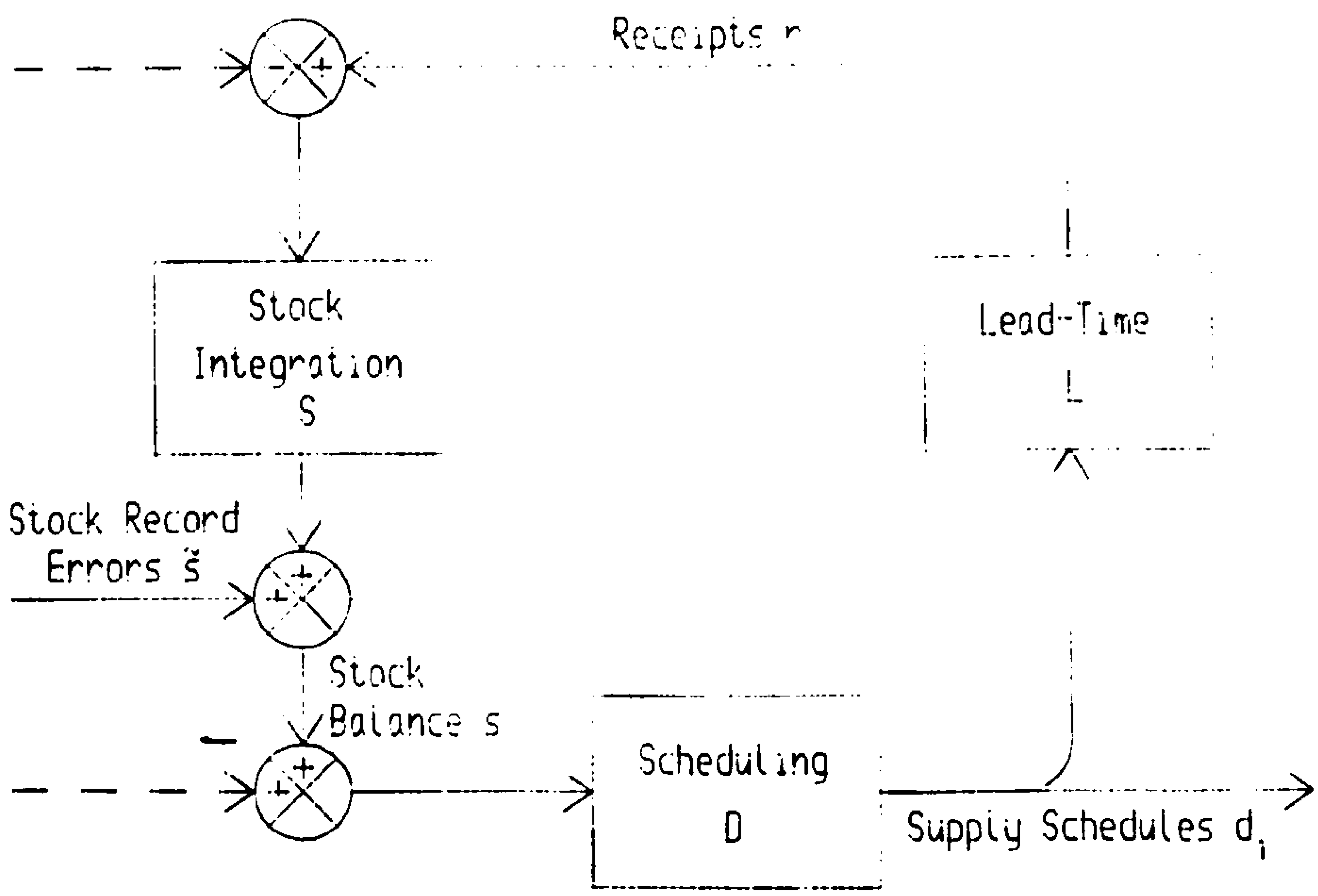

Figure 3.3.c : System Subject To Stock Record Errors

We derive as above:-

$$T(\tilde{s},d_1) = \frac{D}{1 + LSD} \qquad (3.3.3)$$

and:-

$$T(\tilde{s},s) = \frac{LSD}{1 + LSD} \qquad (3.3.4)$$

This latter result is of significance because it describes the full effect of recording errors upon stock-holding. Any excess or reduced stock-holding is the cost of recording inaccuracy. It is frequently thought that the limited cost of such errors does not justify better recording but it is doubtful that the full effect is always understood.

We can also consider random errors imposed upon issues to the customer consisting, for example, of miscounting and pilferage. It is clear, from Figure 3.3.b that such errors (let us call their z-transform $\tilde{i}$) affect the system in precisely the same way as $\tilde{r}$ except that $\tilde{i}$ is subtracted where $\tilde{r}$ is added into the input to S. Thus we have:-

$$T(\tilde{i},d_1) = \frac{SD}{1 + LSD} \quad (3.3.5)$$

and:-

$$T(\tilde{i},s) = \frac{-S}{1 + LSD} \quad (3.3.6)$$

The linearity of the system allows us a great deal of flexibility in handling such sources of error as delivery variability, issue counting errors, pilferage and stock loss due to deterioration since all impinge upon the system at the same point. As we have seen, all are subject to the same system z-transfer function. Where these factors can be isolated the system's response to their individual distributions can be predicted and, using linearity, the response distributions can be combined to predict overall response. If, however, it is simpler to determine the distribution of a combination of some or all of these factors then the system z-transfer functions can provide the combined output distributions. Of course, if individual distributions are known, we still have the choice of combining them (by convolution) before applying the system z-transfer functions.

3.4 A Specific Example

We now proceed to define the elements of the system modelled above. This demonstrates, in a simple case, the ease with which a system may be analysed before we introduce the complications of a multi-level, multi-product system. Although the underlying mathematics of the model is sophisticated, its potential application and interpretation are simple and elegant.

We define the system elements as follows:-

Delivery Policy is that deliveries shall be made during the period following receipt of the customer's order. Hence issues from stock are the same time-series as demand but delayed by one period.

Supplier's lead-time is 3 periods: deliveries shall be made in the third period after the schedule is placed. Receipts into stock are the same time-series as delivery schedules delayed by three periods.

Scheduling delivery schedules shall be calculated each period using the following expression:-

Forecast gross requirements for next 3 periods

- current stock
- delivery schedules already placed for next 2 periods

\+ safety stock

delivery schedule

We shall for the moment apply a safety stock of zero.

Forecasting we shall forecast the next three periods gross requirement very crudely by assuming that the current period's demand will be repeated in future periods.

Stock Integration adds the net receipts of the current period to the stock balance of the last period.

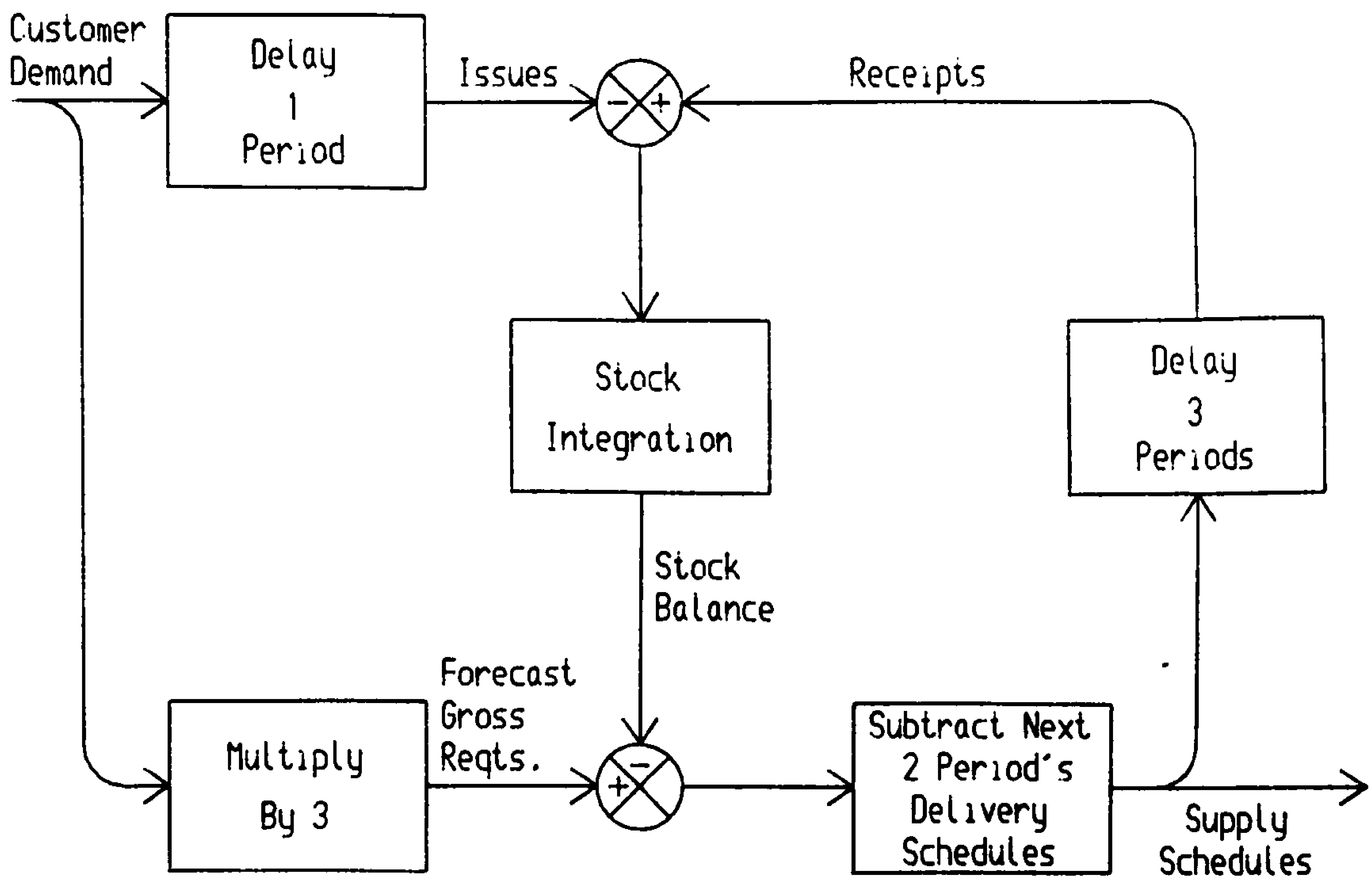

Figure 3.4.a : A Single Level Example

The block-diagram for this example can be drawn as above. We may now proceed to evaluate the z-transfer functions of the system's elements.

Delivery Policy the z-transfer function of a delay of n periods is z^{-n} so we have

$$I = z^{-1} \qquad (3.4.1)$$

Forecasting we need a forecast of the total demand over the next 3 periods and, since scalar multiplication in the t-domain is unchanged by transformation to the z-domain, we have:-

$$F = 3 \qquad (3.4.2)$$

Stock Integration may be represented by Figure 3.4.b. Here a new stock balance is calculated by adding net receipts (receipts - issues) for the current period to the last stock balance.

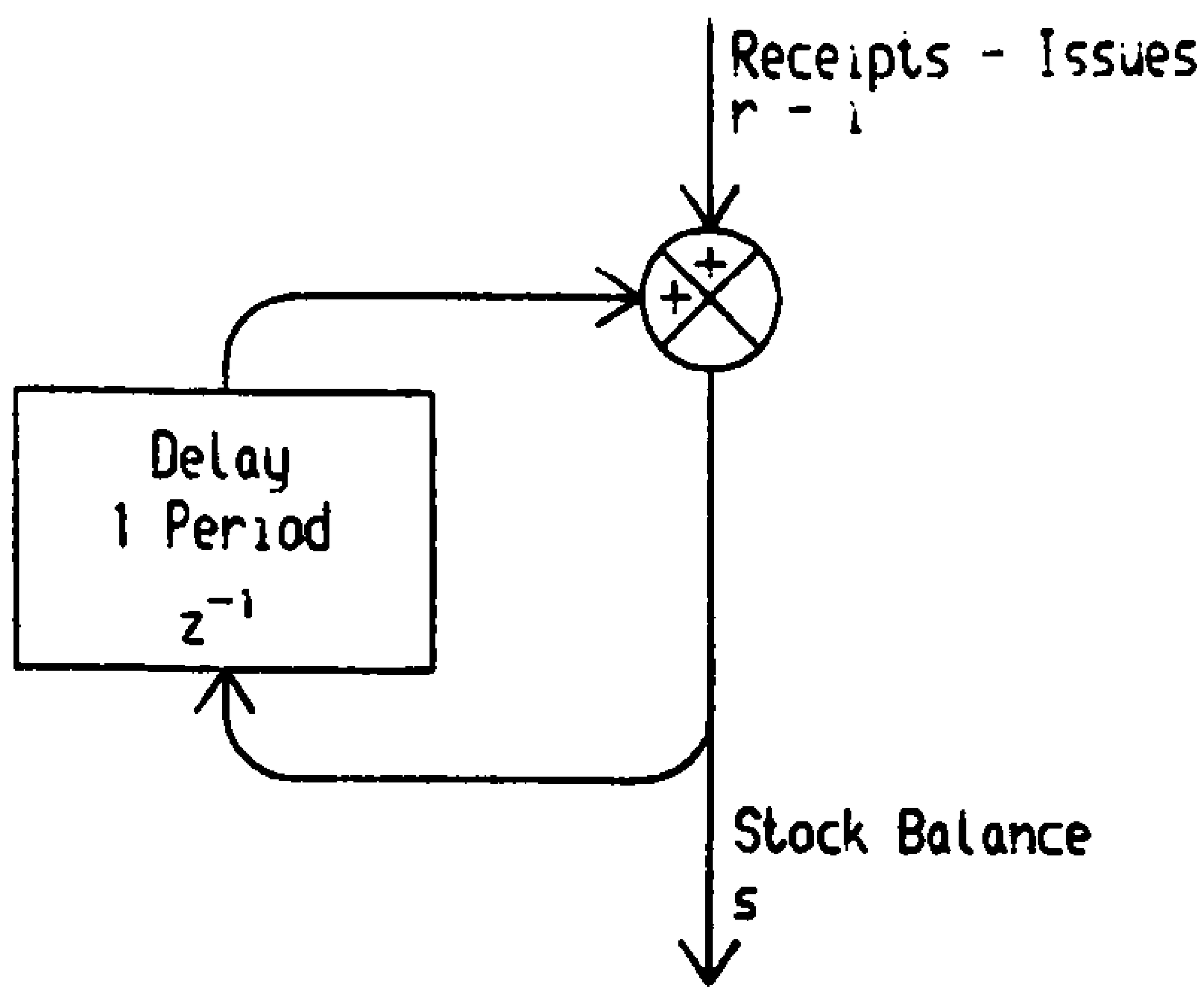

Figure 3.4.b : Stock Integration

From the diagram we see that:-

$$s = (r - i) + sz^{-1}$$

$$\therefore s(1 - z^{-1}) = r - i$$

$$\therefore s = (r - i) \frac{1}{1 - z^{-1}}$$

$$\therefore S = \frac{s}{r - i} = \frac{1}{1 - z^{-1}} \qquad (3.4.3)$$

We expand this, using the binomial theorem, as

$$S = 1 + z^{-1} + z^{-2} + z^{-3} + \ldots\ldots \qquad (3.4.4)$$

Since we know that z^{-n} delays a time-series by n periods it is clear that, when stock integration is applied to the time-series of net receipts, the resulting stock-balances are each the sums of all previous values of net receipts.

Net Delivery Scheduling is shown in Figure 3.4.c. Each schedule is used directly in calculating each of two subsequent schedules.

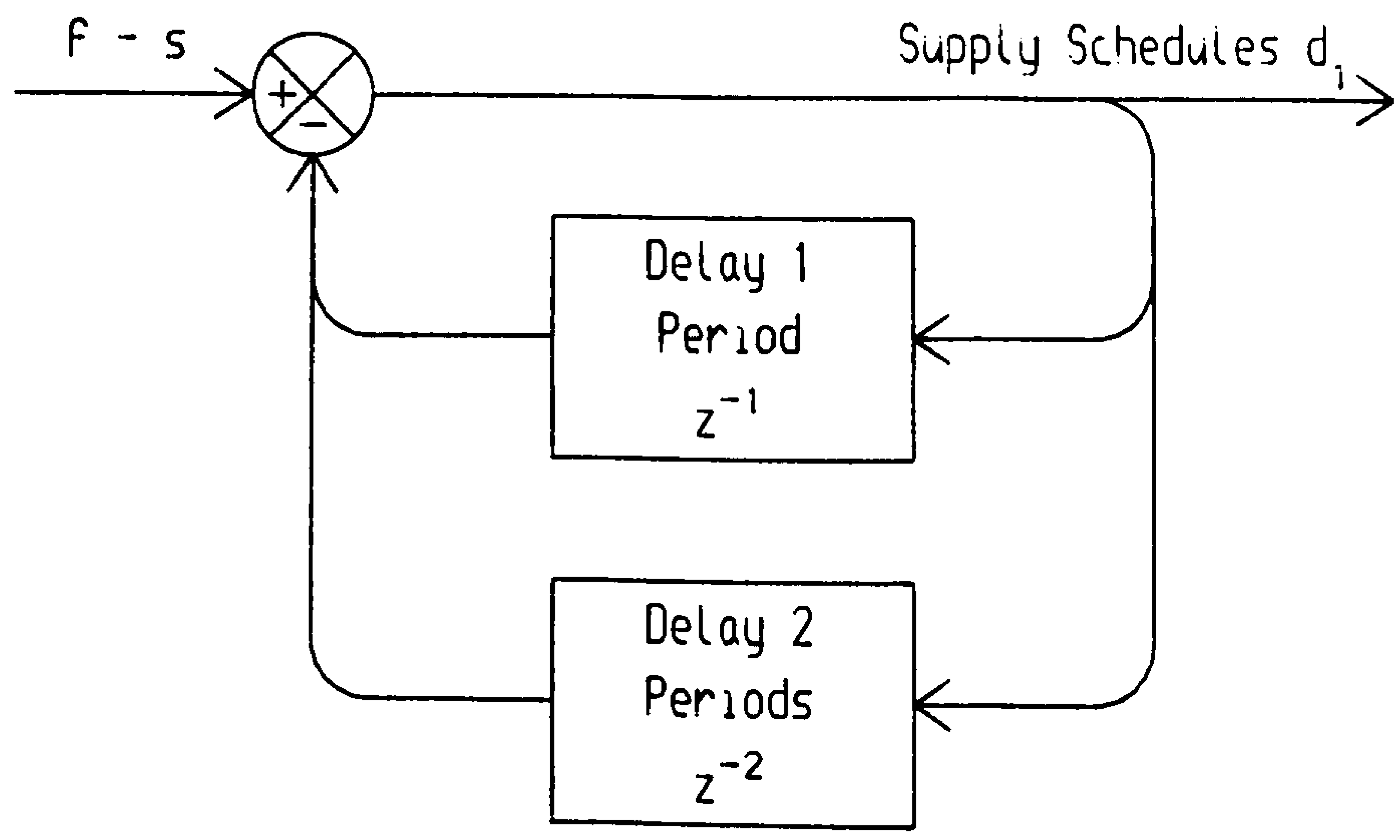

Figure 3.4.c : Net Delivery Scheduling

From the diagram:-

$$d_1 = (f - s) - d_1 z^{-1} - d_1 z^{-2}$$

$$\therefore f - s = d_1(1 + z^{-1} + z^{-2})$$

$$\therefore d_1 = (f - s)\left(\frac{1}{1 + z^{-1} + z^{-2}}\right)$$

$$\therefore D = \frac{d_1}{f - s} = \frac{1}{1 + z^{-1} + z^{-2}} \qquad (3.4.5)$$

This expands to:-

$$D = 1 - z^{-1} + z^{-3} - z^{-4} + z^{-6} - z^{-7} + \ldots\ldots \qquad (3.4.6)$$

We make two points to illustrate the interpretation of this z-transfer function. Firstly, the scheduling rule is such that each input affects schedules into the infinite future without damping - the system hunts. Secondly, every third period is totally unaffected. Both these effects will be modified when the scheduling rule is incorporated into the entire system because this will introduce a feedback loop from d_1 to f - s.

Supplier's lead-time is defined to be a delay of 3 periods so

$$L = z^{-3} \tag{3.4.7}$$

Now that we have evaluated the individual z-transfer functions we can use them to expand the system z-transfer functions derived above (3.2). From equation 3.2.1 we have:-

$$T\ (d,d_1) = \frac{FD + ISD}{1 + LSD}$$

$$= \frac{\left(\dfrac{3}{1 + z^{-1} + z^{-2}} + \dfrac{z^{-1}}{(1 - z^{-1})(1 + z^{-1} + z^{-2})}\right)}{\left(1 + \dfrac{z^{-3}}{(1 - z^{-1})(1 + z^{-1} + z^{-2})}\right)}$$

$$= \frac{3(1 - z^{-1}) + z^{-1}}{(1 - z^{-1})(1 + z^{-1} + z^{-2}) + z^{-3}}$$

$$= \frac{3 - 3z^{-1} + z^{-1}}{1 + z^{-1} + z^{-2} - z^{-1} - z^{-2} - z^{-3} + z^{-3}}$$

$$T\ (d,d_1) = 3 - 2z^{-1} \tag{3.4.8}$$

Similarly from equation 3.2.2 we can deduce:-

$$T\ (d,s) = \frac{-IS + FDLS}{1 + LSD}$$

$$= 2z^{-3} - z^{-2} - z^{-1} \tag{3.4.9}$$

As these two system z-transfer functions have reduced to polynomials in z^{-1} their interpretation is particularly simple. We recall (Chapter 2) that a z-transfer function is the z-transform of the system's response to a unit impulse and that multiplication by z^{-1} represents a unit delay in the t-domain. It is now clear from equation 3.4.8 that, in response to a unit impulse customer demand, a delivery schedule of 3 is immediately raised followed next period by a schedule of -2. Equation 3.4.9 shows that stocks become -1 on delivery to the customer and remain so until the next delivery schedule is met two periods later. Stock then increases to 2 before returning to zero on implementation of the negative schedule.

In a real system such negative figures may be impossible. They can, however, be understood in a number of different ways because the system is linear. We may consider the demand impulse to be super-imposed upon some larger, underlying demand pattern and thus the system z-transfer functions describe the changes in schedules and stocks. Negative stocks can be interpreted as the use of a policy safety stock or backlogging of customer demand.

Since we have seen that, if an impulse increase in customer demand occurs, it will result in a finite set of scaled, delayed impulses in both delivery schedules and stock balances, we can be satisfied that the system is stable as the effect of an impulse is bounded in time.

Besides requiring a system to be stable we must also consider the acceptability of its transient responses. We have looked at the system's responses to a unit impulse. The acceptability of these responses depends in part on the underlying demand pattern. It is unlikely that a supplier will accept negative schedules which must be interpreted as returns to him (there are exceptions such as "sale or return" agreements). We must therefore be satisfied that there is no significant probability of occurrence of an impulse large enough to generate a schedule reduction greater than the underlying schedule.

In order to achieve our stated policy on delivery to customers we must hold sufficient safety stock to absorb the stock reductions arising from any significantly probable impulse. Since the negative effect of an impulse lasts two periods we must provide safety stock equal to the sum of any two significantly probable consecutive impulses.

We shall now examine the system's responses to three other standard input patterns and in each case shall discuss the factors affecting the acceptability of these responses.

Unit step

The z-transform of the unit step can be obtained from z-transform tables as:-

$$d(z) = \frac{z}{z - 1}$$

We know from equation 3.4.8 that:-

$$d_1 = (3 - 2z^{-1})d$$

$$= (3 - 2z^{-1}) \left(\frac{z}{z - 1}\right)$$

$$= 3 \left(\frac{z}{z - 1}\right) - 2z^{-1} \left(\frac{z}{z - 1}\right)$$

This can be interpreted, just as for the unit impulse, as the sum of two scaled unit steps, one of which is delayed by one period. This is illustrated graphically (3.4.d).

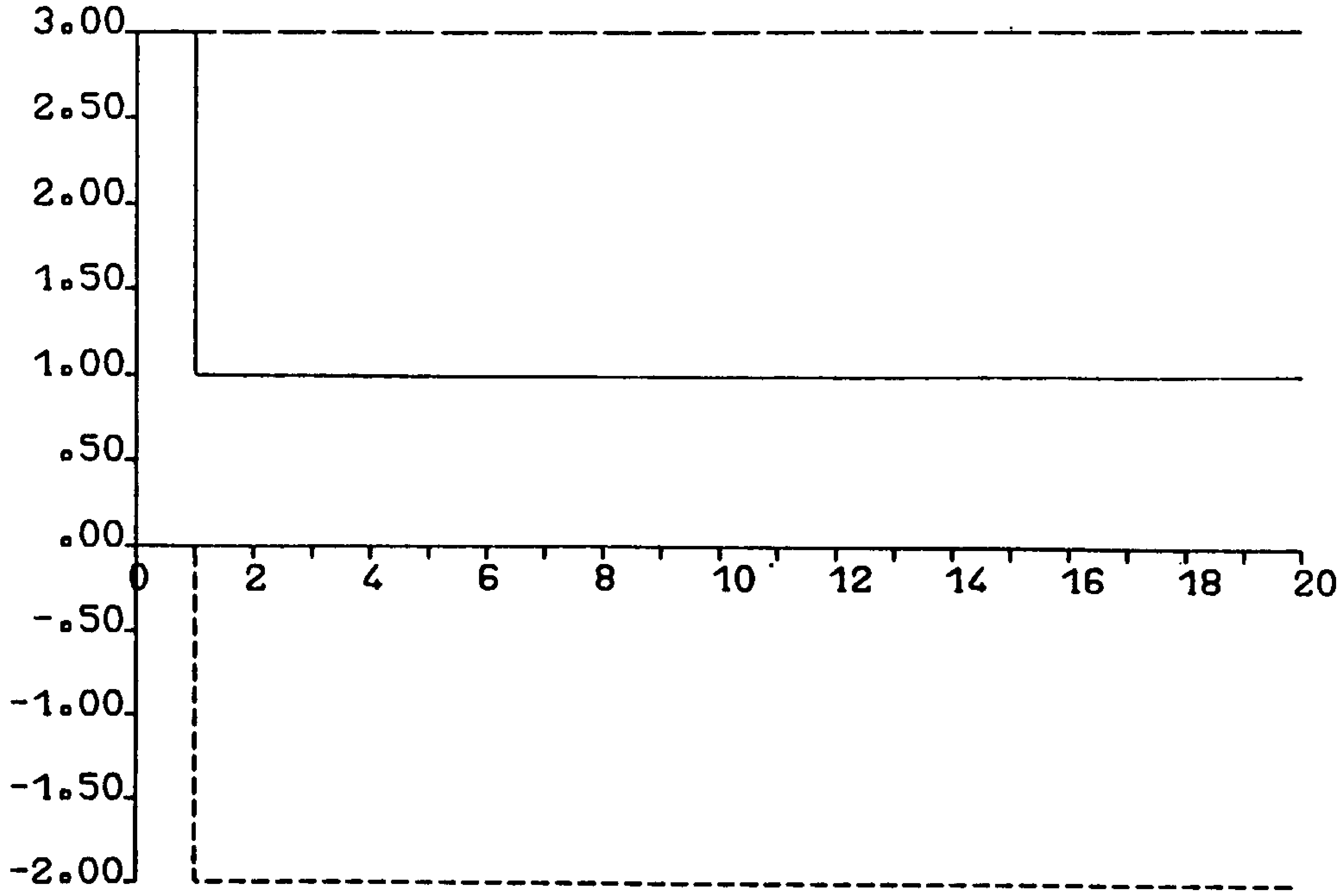

Figure 3.4.d : Schedule Response To Step

We know also, from equation 3.4.9 that:-

$$s = (2z^{-3} - z^{-2} - z^{-1})d$$

$$= (2z^{-3} - z^{-2} - z^{-1}) \left(\frac{z}{z - 1}\right)$$

$$= 2z^{-3} \left(\frac{z}{z - 1}\right) - z^{-2} \left(\frac{z}{z - 1}\right) - z^{-1} \left(\frac{z}{z - 1}\right)$$

This is shown in Figure 3.4.e

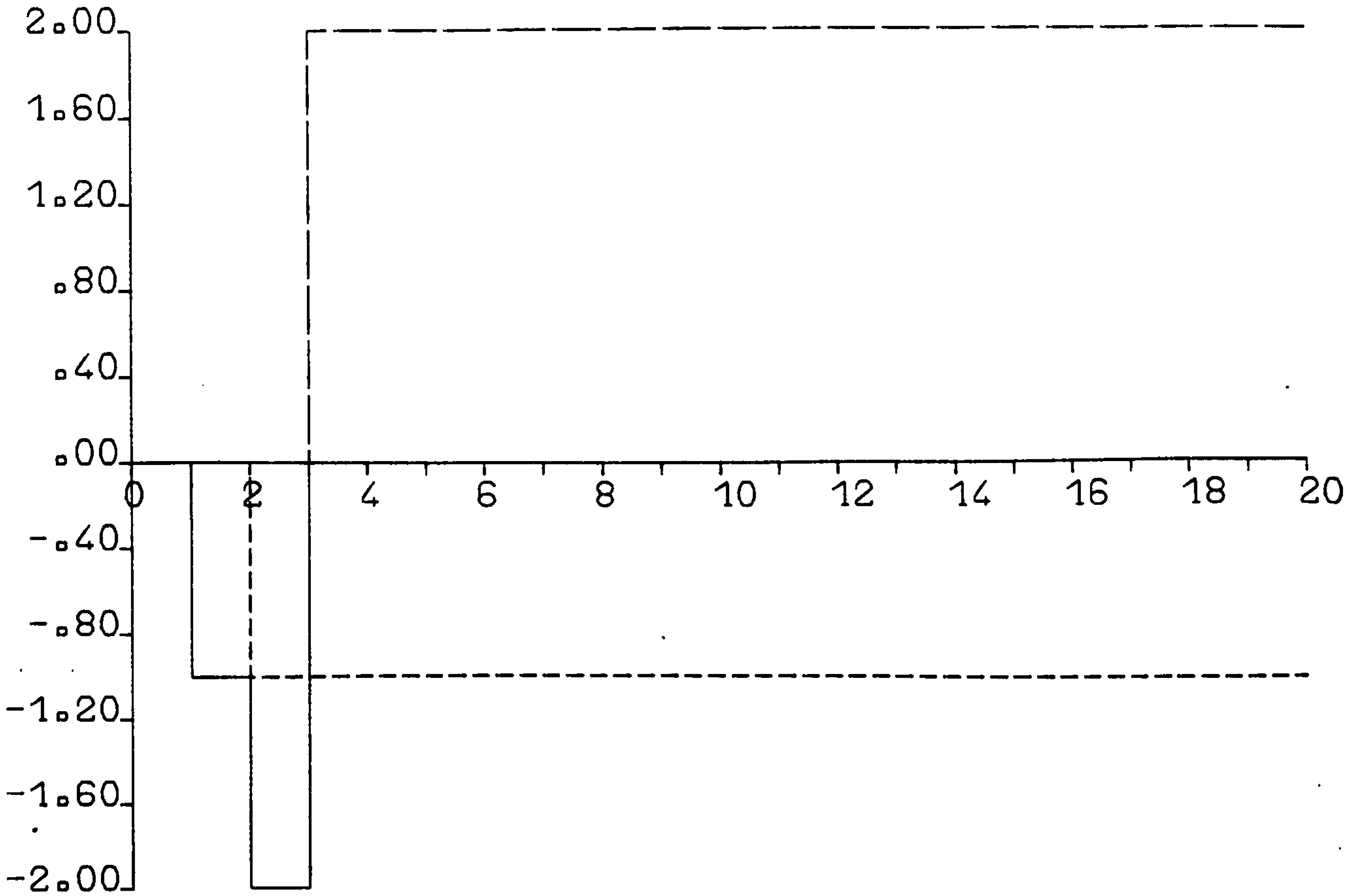

Figure 3.4.e : Stock Response To Step

Here the delivery schedule response is quite acceptable as the system first raises a large schedule to cover the early demands before supply reacts, then immediately settles to a steady level. The buffer stock needed to cover this can be determined from 3.4.e as 2A + B, where A and B are the magnitudes of any two significantly probable, consecutive step changes in customer demand. It is worth noting that no faster response, and therefore no smaller safety stock, is possible, given the constraint of supplier's lead-time.

Unit Ramp

The z-transform of the unit ramp can be obtained from z-transform tables as:-

$$d(z) = \frac{z}{(z - 1)^2}$$

We know (3.4.8) that:-

$$d_1 = (3 - 2z^{-1})d$$

$$= (3 - 2z^{-1}) \left(\frac{z}{(z - 1)^2}\right)$$

$$= \frac{3z - 2}{(z - 1)^2}$$

$$= \frac{(3z^2 - 3z + z)z^{-1}}{(z - 1)^2}$$

$$= \left(\frac{3z(z - 1) + z}{(z - 1)^2}\right) z^{-1}$$

$$= \frac{3z}{z - 1} + \left(\frac{z}{(z - 1)^2}\right) z^{-1}$$

This is the sum of a unit step multiplied by a scalar of 3 and a unit ramp, both delayed by one period, which can be illustrated graphically (3.4.f).

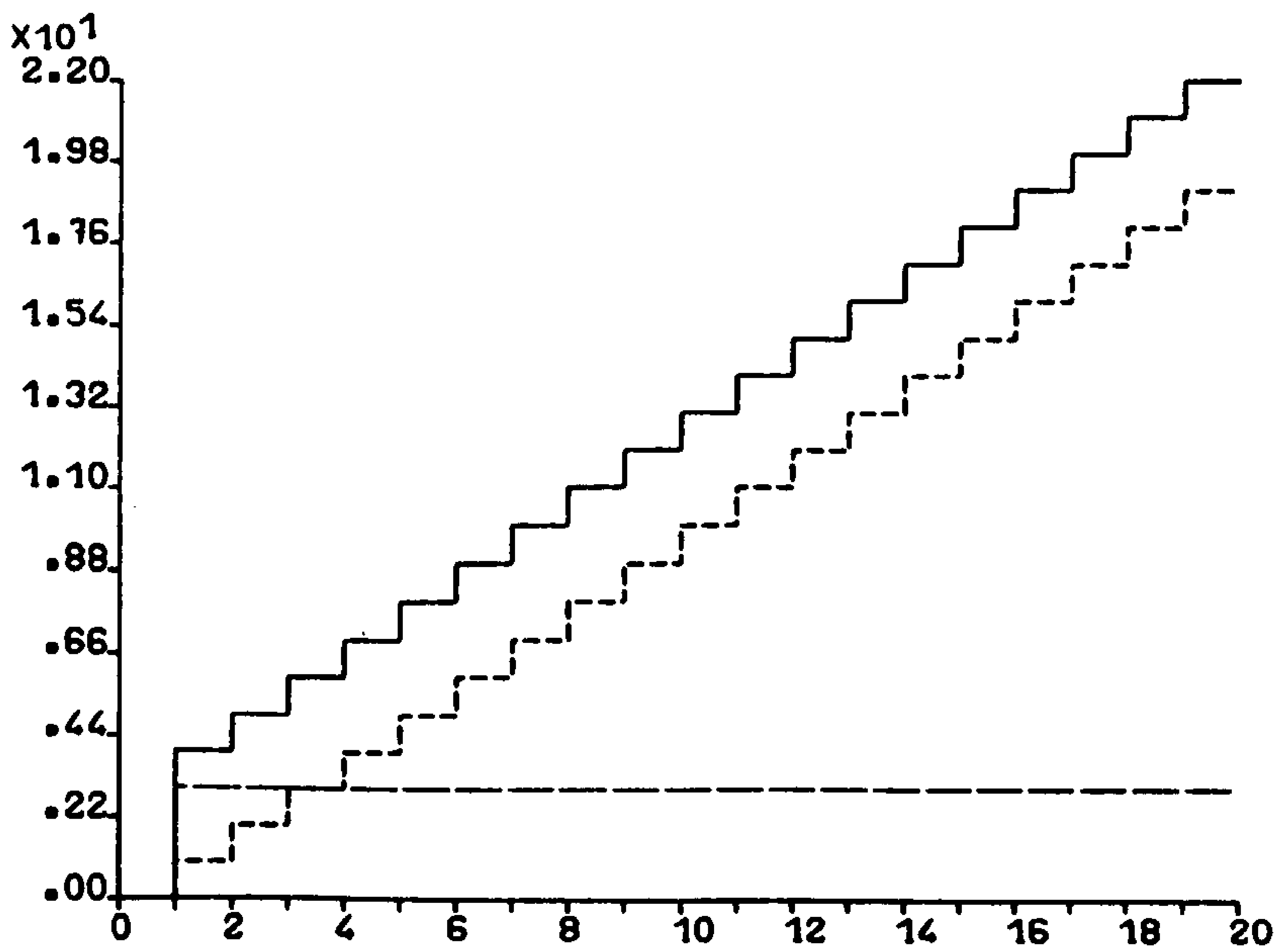

Figure 3.4.F : Schedule Response To Ramp

We also know (3.4.9) that:-

$$s = (2z^{-3} - z^{-2} - z^{-1})d$$

$$= (2z^{-3} - z^{-2} - z^{-1}) \left(\frac{z}{(z - 1)^2}\right)$$

$$= \frac{2z^{-2} - z^{-1} - 1}{(z - 1)^2}$$

$$= \frac{(-2z^{-2} - z^{-1})(z - 1)}{(z - 1)^2}$$

$$= \left(\frac{z}{(z - 1)}\right) (-2z^{-3} - z^{-2})$$

This is the sum of one unit step multiplied by -1 and delayed 2 periods and a second unit step multiplied by -2 and delayed 3 periods. This is plotted in 3.4.g.

Here we have a reasonable pattern of delivery schedules increasing at the same rate as demand after an initial jump. However, stock becomes negative and remains so implying either a permanent reduction in buffer stock or a permanent backlogging of demand. If these are unacceptable the system must be modified. To improve performance the forecasting mechanism must be altered to recognise and predict the continuing ramp.

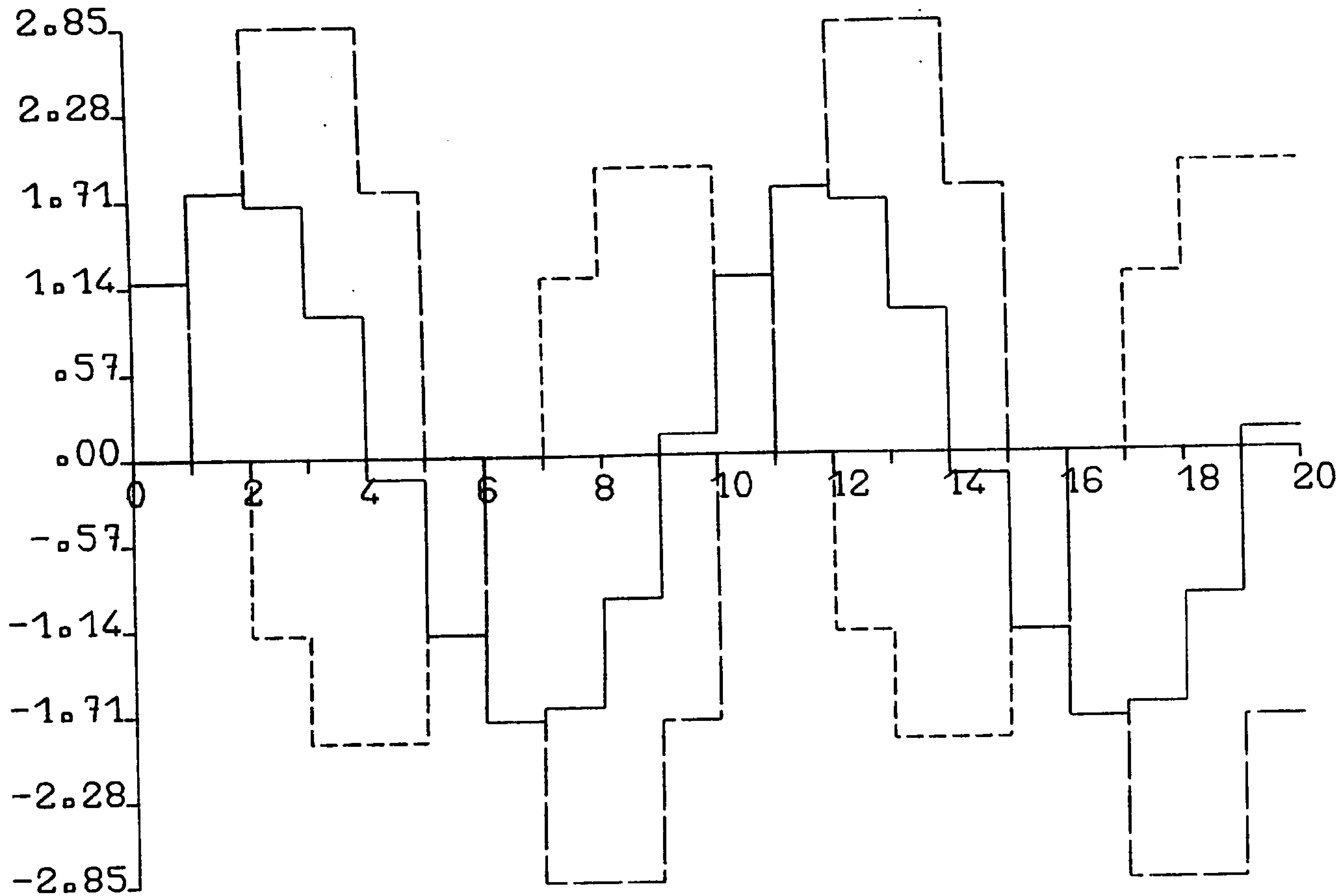

Figure 3.4.h : Schedule Response To Sine

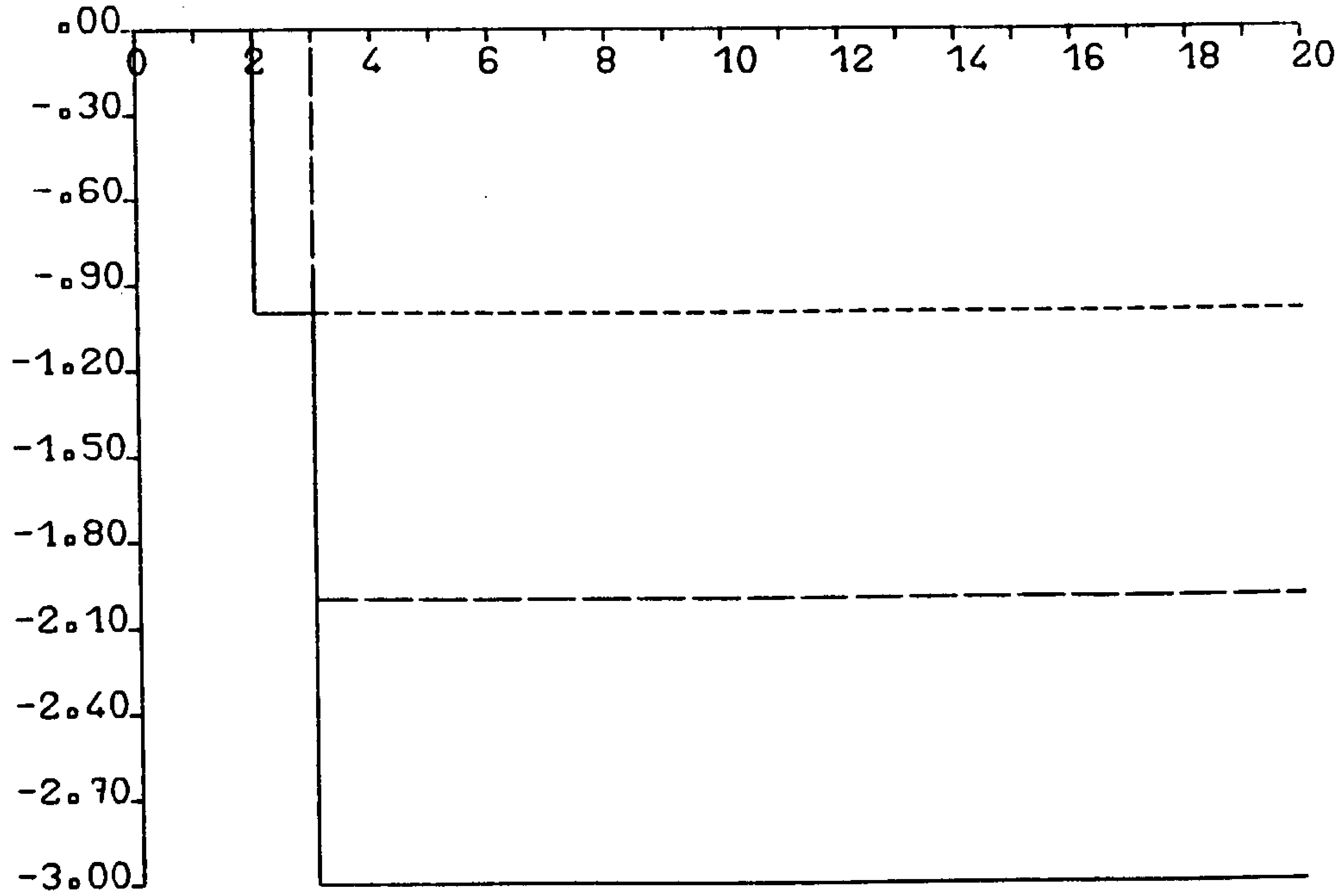

Figure 3.4.g : Stock Response To Ramp

Sinusoid

By application of the sampling theorem [Truxall 1955, pp. 505-506] we know that the demand time-series d(t) cannot reproduce any sinusoid whose frequency is greater than half the system review frequency. The system's fundamental frequency is 2π radians/period, and so any sinusoidal demand, $d(t) = a \sin \omega t$, input to the system need only be considered if its frequency ω is such that $\omega \leq \pi$.

We know from equation 3.4.8 that:-

$$d_1 = (3 - 2z^{-1})d$$

and so $d_1(t)$ is the sum of two scaled sine curves separated by a phase angle of ω radians. This is illustrated (3.4.h) with $\omega = \frac{\pi}{5}$. When $\omega = \pi$, the sampling limit, the two components of d_1 tend to reinforce because they are of opposite sign and have a phase difference of one period, so we have

$$d_1(t) = 5 \sin\pi t$$

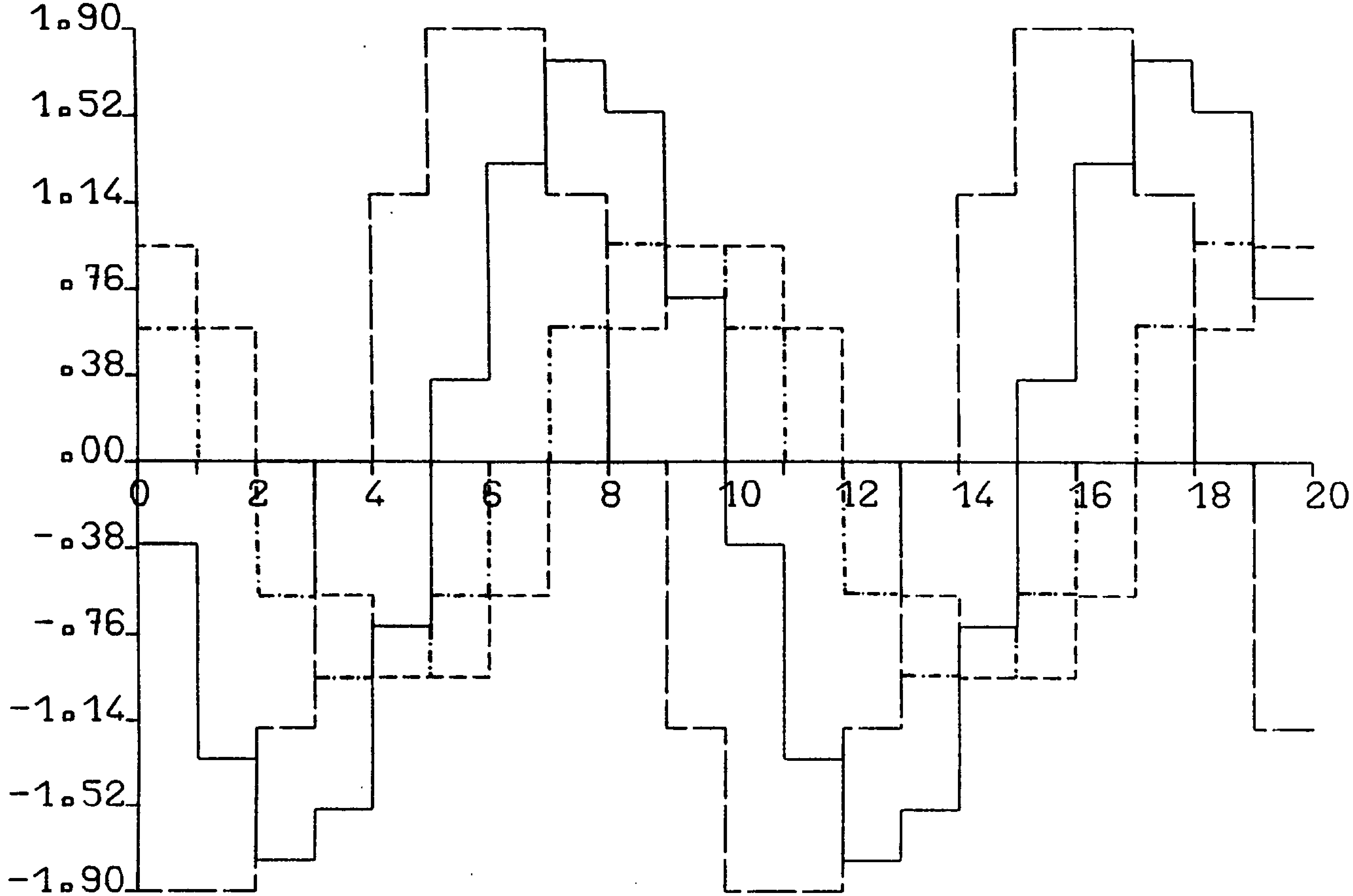

Figure 3.4.1 : Stock Response To Sine

As ω decreases the components tend to cancel and so

$$\lim_{\omega \to 0} d_1(t) = \sin\omega t$$

We can now interpret the response of $d_1(t)$ to any sinusoidal input and therefore to any cyclical input. The result will always be an amplified version of the demand time-series, having the same frequency. Low frequency input patterns will be amplified very little and will match the input waveform closely. High frequency demands will be amplified up to a maximum of 5 times and will suffer considerable waveform distortion through sampling.

We now examine the response of stock balance to sinusoidal demand. From 3.4.9:-

$$s = (2z^{-3} - z^{-2} - z^{-1})d$$

we know that s(t) is compounded of 3 consecutive sine curves separated by one period phase differences:-

$$s = 2\sin(\omega(t-3)) - \sin(\omega(t-2)) - \sin(\omega(t-1))$$

which we illustrate (3.4.i) with $\omega = \frac{\pi}{5}$.

If we allow ω to reach π, its maximum we find that:-

$$s = -2\sin\pi t - \sin\pi t + \sin\pi t$$

$$= -2\sin\pi t$$

As ω decreases the phase differences of one period lose significance so:-

$$\lim_{\omega \to 0} s(t) = 0$$

Thus stock response to sinusoidal input is also an amplified copy of demand. The maximum amplification, when $\omega = \pi$, is 2 and when frequency is low amplification tends to zero. The system sees a very low frequency demand as similar to a step and, as we have already seen, can maintain zero stocks.

Knowing that the system is going to exaggerate its response to the high frequency components of demand we are in a position to determine its acceptability. We must know the expected demand patterns well enough to evaluate the amplitude of any high frequency components. Where these are negligible the system's response will show a delayed copy of demand with negligible amplification. We may therefore wish to increase the review frequency so as to eliminate high frequency responses by reducing their frequency. However, we must also consider their elimination by reducing system frequency until they are unrecognisable. The choice of action will be governed by the entire frequency spectrum of demand.

By examining the system's response to a range of demand patterns we can decide upon its acceptability or the need for its modification. Analysis may point to specific modifications, for example, forecasting could be improved to detect and predict a ramp demand or system review frequency may be changed to improve response to cyclic elements of demand

Given some knowledge of the market where demands arise and past demand patterns it is possible to estimate, at least subjectively, the probability of occurrence of particular demand time-series, their amplitude relative to underlying demand patterns and their frequency spectra. From this knowledge we may decide the need for modification. In many cases the need for modification is expressible in economic terms. Where response to a probable pattern of demand results in negative stocks we must invest in sufficient buffer stocks and the cost of this stock-holding represents the cost of not modifying the system. Similarly, where the response is an impossibly negative delivery schedule which cannot be offset against underlying, steady-state components of delivery schedules we will see an undesired surplus stock.

The fact that our system is stable means that the effect of any noise impulse is bounded in time. However, noise is constantly generated and the noise response impulses will coincide at other points in the system. Each time-series present in the system will contain a linear combination of delayed, scaled copies of the noise time-series. Hence the system will transmit any noise distribution as the convolution of a number of scaled copies of this distribution.

We derived (3.3.1) the system z-transfer function $T(r,d_1)$ and we can now substitute the values of the z-transfer functions to obtain

$$T(\tilde{r},d_1) = \frac{-SD}{1+LSD}$$

$$= \frac{-\left(\frac{1}{1-z^{-1}}\right)\left(\frac{1}{1+z^{-1}+z^{-2}}\right)}{1+z^{-3}\left(\frac{1}{1-z^{-1}}\right)\left(\frac{1}{1+z^{-1}+z^{-2}}\right)}$$

$$= \frac{-1}{(1-z^{-1})(1+z^{-1}+z^{-2})+z^{-3}}$$

$$= \frac{-1}{1+z^{-1}+z^{-2}-z^{-1}-z^{-2}-z^{-3}+z^{-3}}$$

$$= -1$$

Thus the system z-transfer function from noise superimposed upon supplier's deliveries to delivery schedules is simply scalar multiplication by -1. The effect of such noise upon delivery schedules is a simple and immediate correction.

Similarly from 3.3.2:-

$$T(\tilde{r},s) = \frac{S}{1 + LSD}$$

$$= 1 + z^{-1} + z^{-2}$$

implying that each noise impulse will increase stock immediately and for the next two periods. The distribution of stock perturbations will therefore be the triple convolution:-

$$\tilde{s} = \tilde{r} + \tilde{r} + \tilde{r}$$

which is a flattened, spread version of the input noise distribution.

Let us take as an example a very simple form $\tilde{r}$:-

$\tilde{r}(-1) = .2$

$\tilde{r}(0) = .5$

$\tilde{r}(1) = .3$

$\tilde{r}(x) = 0$ for all $x>1$ or $x<-1$

Then we have:-

$$(\tilde{r} + \tilde{r})(-2) = .2 \times .2 = .04$$

$$(\tilde{r} + \tilde{r})(-1) = .2 \times .5 + .5 \times .2 = .2$$

$$(\tilde{r} + \tilde{r})(0) = .2 \times .3 + .5 \times .5 + .3 \times .2 = .37$$

$$(\tilde{r} + \tilde{r})(1) = .5 \times .3 + .3 \times .5 = .3$$

$$(\tilde{r} + \tilde{r})(2) = .3 \times .3 = .09$$

and so:-

$$(\tilde{r} + \tilde{r} + \tilde{r})(-3) = .2 \times .04 = .008$$

$$(\tilde{r} + \tilde{r} + \tilde{r})(-2) = .2 \times .2 + .5 \times .04 = .06$$

$$(\tilde{r} + \tilde{r} + \tilde{r})(-1) = .2 \times .37 + .5 \times .2 + .3 \times .04 = .186$$

$$(\tilde{r} + \tilde{r} + \tilde{r})(0) = .2 \times .3 + .5 \times .37 + .3 \times .2 = .305$$

$$(\tilde{r} + \tilde{r} + \tilde{r})(1) = .2 \times .09 + .5 \times .3 + .3 \times .37 = .279$$

$$(\tilde{r} + \tilde{r} + \tilde{r})(2) = .5 \times .09 + .3 \times .3 = .135$$

$$(\tilde{r} + \tilde{r} + \tilde{r})(3) = .3 \times .09 = .027$$

We are here using convolution of a finite set of distributions to arrive at a resultant distribution of response to noise. This method of analysis replaces the traditional simulation analysis necessary to evaluate the effect of noise within the system. The arithmetic of convolution, in spite of its simplicity, is extensive in real environments where noise distributions have a broad band of significantly probable values. Each convolution spreads the distribution and increases the arithmetic in the next (as is clear from the example above). It is, however, a process of fixed length (unlike simulation) and can be shortened in many cases by fixing a significance limit of probability. Furthermore, standard convolution programs are available for use in most computer installations. The program of Appendix II carries out convolution in an arithmetically simplified manner suitable for use in this context.

This method's application is rapid and cheap when compared with the extended process of system simulation. We are replacing the process of simulation of an entire, complex system by the numerical evaluation of its known response in terms of each time-series we wish to examine. Irrelevant parts of the system are not part of the computation, nor do we require protracted running of the model to ensure the occurrence of all combinations of circumstance.

CHAPTER 4

A GENERAL MODEL

4.1 Preamble

In the last chapter we demonstrated the use of z-transform control techniques in a single-product, single-level environment. We now proceed to the development of a general model introducing the complexities of a multi-product, multi-level production facility controlled by an integrated system. This facility assembles a range of finished products from sub-assemblies and components manufactured from raw materials. Products may share common parts at any level.

We shall describe two forms of information flow which together comprise the basis of most practical production control systems. These two flows will be explicitly included in the general model to be developed. We demonstrate that systems within the scope of this thesis can be modelled by some combination of sub-systems of three distinct forms. The first such sub-system is applied to each finished product, whilst one or other of the remaining two is applied to each sub-assmebly, component or material as dictated by the types of information flow used in controlling its production and stocking.

We can apply z-transform analysis to each sub-system in a manner which, particularly in the case of the finished product, is similar to Chapter 3. This analysis yields a sub-system z-transfer function for each finished product, but, because of commonality of use at lower levels, it cannot do so for the remaining sub-systems.

However, we can describe the response of any time-series in the sub-system in terms of all its inputs and so linearity allows us to analyse the response to any one sub-system input.

The use of the general analysis developed for each sub-system allows the response to any input to be traced through successive levels. Thus it is possible to generate a system z-transfer function between any two time-series in the system where causal relationship exists. Although the development of such a system z-transfer function is cumbersome in any general form, it is usually a simple matter to derive any desired function from the sub-system analysis in a given practical case. For this reason we confine analysis of the general model to analysis of the three sub-systems and show how they can be combined to form an analysis of the entire system. This forms a powerful tool in analysis of practical systems and we show its use in a very simple multi-product, multi-level example in Chapter 6.

4.2 Categories of Information Flow

Two categories of information flow form the basis of most multi-product, multi-level production control systems. Although z-transform techniques are equally applicable to both categories, the flows of information used to control production of parts below finished product level differ radically and so their control theory models must differ. We shall refer to the two categories as "cascaded systems" and "base information systems" and define these as follows:-

4.2.1 In a CASCADED SYSTEM, the *only* demand information used in controlling production of a part is the demand *for that part* generated at higher manufacturing levels.

4.2.2 In a BASE INFORMATION SYSTEM production of a part is controlled using demand for those *finished products* which contain the part.

Mixed systems can and do exist where some parts are controlled by base information systems and others by cascaded systems. This is particularly likely in large, complex organisations where inter-departmental relationships are not consistent. Such mixed systems are adequately modelled by an appropriate mixture of sub-system types.

4.3 The Environment of the Model

In this section we define the environment of the general model in terms of the structure of products whose assembly and manufacture are controlled.

4.3.1 There is a total range of M finished products which may have common components or materials at any stage of manufacture.

4.3.2 There is a total range of N parts used in the manufacture of these products. (Here and henceforth, the term "part" is used to denote any sub-assembly, component or material). This includes all parts whose manufacture or procurement is controlled by the system.

4.3.3 There exist K definable levels of manufacture of the product range. Thus each part can be considered as belonging to some level of the progression from raw material to finished product. We follow a widespread, though not universal, convention [British Standard 5191, 1975] of giving the least level number to finished product and increasing that number as we work toward raw material. We shall also adopt the almost universal parlance which describes levels near to finished product as high and those near to raw material as low. Despite the apparent inconsistency of giving the lowest numbers to the highest levels this combination of conventions is in widespread practical use.

For example, in an environment where raw materials are purchased and manufactured into components which are then assembled into a finished product we have three levels which we shall number from 0 to 2:-

Level 0 : finished product (highest level)

Level 1 : components

Level 2 : raw materials (lowest level)

4.3.4 Whilst in a real environment the majority of parts are used only at one level, there frequently exist parts used at several levels. These will tend to be such standard items as fasteners, spacers and the like, and we define their level as the lowest (i.e. greatest level number) at which they are used. To maintain consistency in the

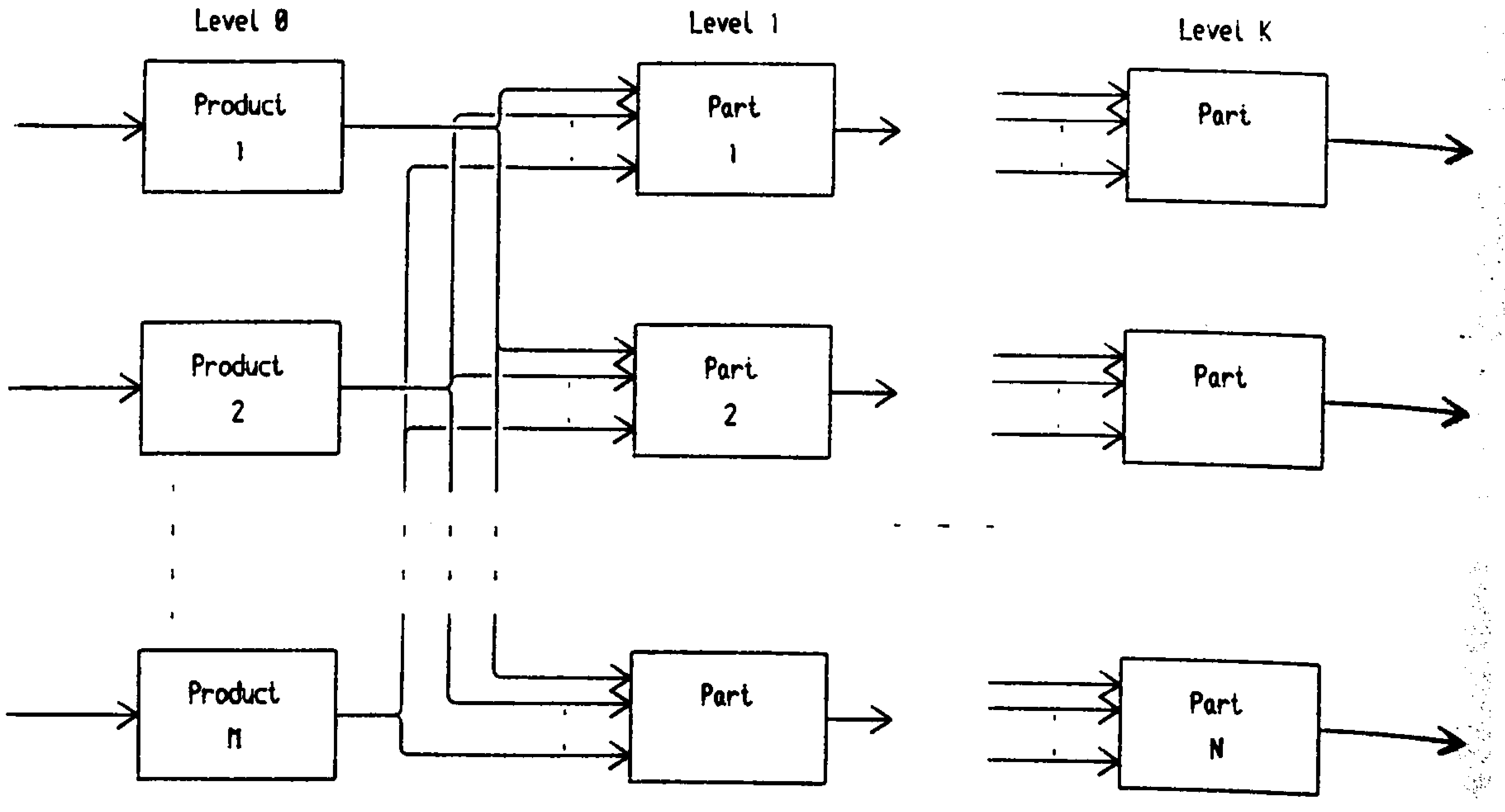

Figure 4.4.a : Decomposition Of A Cascaded System

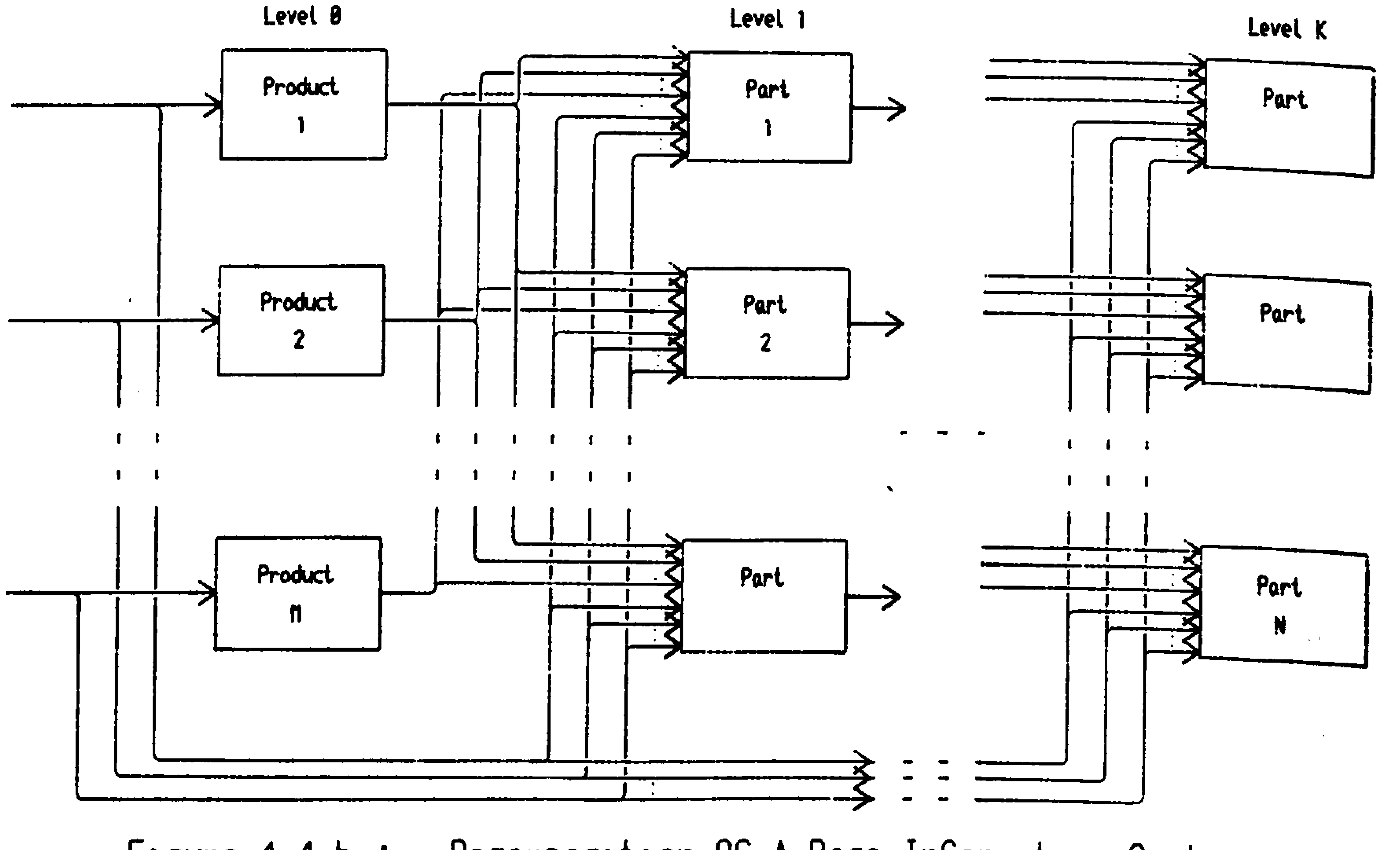

Figure 4.4.b : Decomposition Of A Base Information System

general model we forge the links from such parts to their uses more than one level up by introducing dummy parts at each intervening level. Although such dummies form an untidy aspect of the model, their effect on its analysis is negligible except as a simplification.

4.4 The Form of the Model

The model is formulated by decomposing the entire control system into a set of sub-systems, one for each product and part. The system category (base information, cascaded or mixed) will determine the patterns of information flow both between and within the sub-systems.

We show this decomposition for a cascaded system (Figure 4.4.a). In this general diagram it is clear that a flow of information is included from each sub-system to every sub-system at the next level down. In most real systems each sub-system will ignore many of these inputs as irrelevant. It is thus possible to simplify the appearance of the general diagram although its basic form remains unaltered.

The general diagram for a base information is also given (Figure 4.4.b). This is a more complicated representation as the number of information flows is far greater than for the cascaded system. Examination shows that the general form is the same as for the cascaded system except that each control sub-system receives additional information from product demands. The same practical considerations are applicable here.

The latter diagram (Figure 4.4.b) can also represent a mixed system. The parts subject to cascaded control will have all base information inputs multiplied by zero and the appropriate internal form of their sub-systems will be used. Hence we have a fully general decomposition, since it can be applied to any system in the environment defined in the last section (4.3).

Definition of a model of a real system must begin with the creation of an appropriate decomposition. Given z-transfer functions for each sub-system, from each input to its output, we may follow any causal sequence through the model. Thus we may determine any or all responses to a given input to the system. Our next task is therefore to derive the z-transfer functions relating to each sub-system. The general analysis of each of the three sub-system types (and the dummy) follows.

4.5 Symbolic Representation

The sub-systems described below as part of the general model are developed using a standard notation for z-transforms of time-series and z-transfer functions of system elements. This notation is as follows:-

z-transforms of time-series are represented by lower case letters.

z-transfer functions are denoted by capital letters.

The first subscript of a symbol denotes the manufacturing level at which the symbol applies.

The second subscript denotes the product or part to which the symbol refers.

A third subscript may be used to denote a product or part at the level above that identified by the first subscript.

We define our symbols as follows:-

Time-Series

d_{ij} for $i = 0$ is customer's demand for product j.
for $i > 0$ is demand for part j arising from level i-1.

r_{ij} represents receipts into stock of part/product j at level i.

s_{ij} represents current stocks of part/product j at level i.

f_{ij} represents forecast gross requirement for part/product j at level i.

z-transfer functions

I_{ijk} for $i = 0$ represents the application of a delivery policy to customer's demand for product j. When $i = 0$ k is omitted.
for $i > 0$ represents the use of part j at level i in meeting demand for production of part/product k at level i-1.

F_{ijk} for $i = 0$ represents generation of forecast gross requirements for product j. When $i = 0$ k is omitted.
for $i > 0$ in a base information sub-system, represents the generation of forecast gross requirements for part j at level i arising from customers' demand for model k.

for $i > 0$ in a cascaded sub-system, represents the generation of forecast gross requirements for part j at level i. This forecast is based entirely upon demand for part j and so k is omitted here.

S_{ij} represents stock integration for part j at level i.

D_{ij} represents net requirements scheduling for part j at level i.

L_{ij} represents the application of a production/ supply lead-time for part j at level i.

System z-transfer functions

The term $T(x,y)$ will denote the z-transfer function of the system when it is considered to have time-series x as input and y as output.

Sub-system z-transfer functions

The term $\zeta(x,y)$ will denote the z-transfer function of a sub-system when it is considered to have time-series x as input and y as output.

Noise distributions

Noise distributions will be denoted by the symbol ˜ over that representing the time-series upon which they are superimposed.

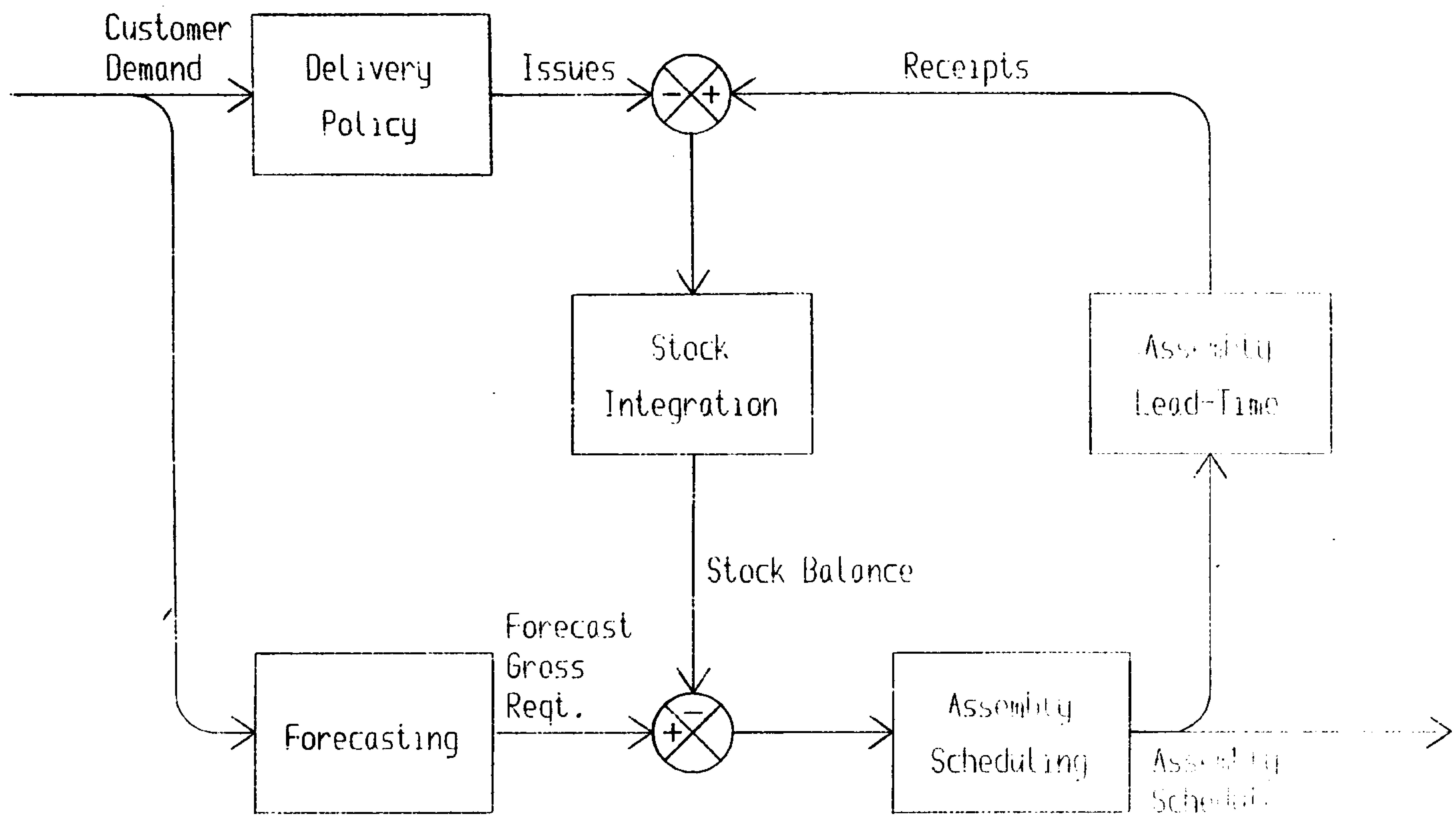

Figure 4.6.a : The Product Control Sub-System

4.6 The Product Control Sub-System

The product control sub-system, in response to customers' demand for the product, schedules its final assembly and so by implication generates demands for parts at level 1. This is illustrated in Figure 4.6.a from which it is clear that this sub-system is similar to the single level system we examined in the last chapter.

The diagram is redrawn (Figure 4.6.b) using the symbolic representation of the last section

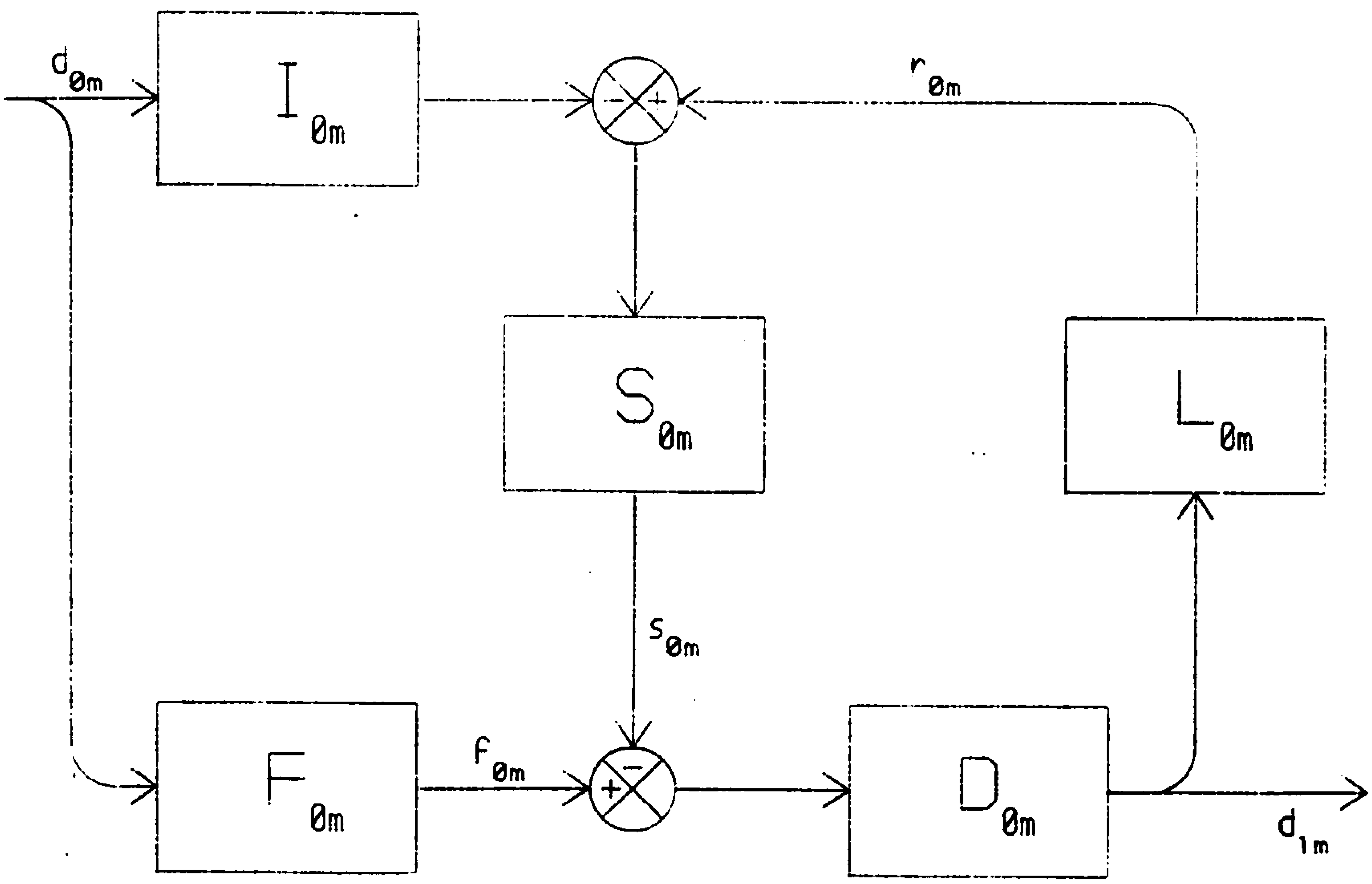

Figure 4.6.b : The Product Control Sub-System (Symbolic Representation)

In order to derive system z-transfer functions incorporating such a sub-system it is necessary to know the sub-system z-transfer function relating customer demand to demand for level 1 parts. This can be done using methods demonstrated in Chapter 3. Detailed proofs of this and subsequent results are given in Appendix II.

$$\zeta\ (d_{Om}, d_{1m}) = \frac{F_{Om}D_{Om} + I_{Om}S_{Om}D_{Om}}{1 + L_{Om}S_{Om}D_{Om}} \qquad (4.6.1)$$

The response of stock balance to customers' demand is given by:-

$$\zeta\ (d_{Om}, s_{Om}) = \frac{F_{Om}L_{Om}S_{Om}D_{Om} - I_{Om}S_{Om}}{1 + L_{Om}S_{Om}D_{Om}} \qquad (4.6.2)$$

We must also consider the effects upon both stock balance and assembly schedules of random noise introduced within the sub-system. We shall introduce noise at two points only: variability in customers' demand and supply variance against schedule. Representing the first noise distribution by $\tilde{d}_{Om}$ we have, from equations 4.6.1 and 4.6.2 respectively,

$$\zeta\ (\tilde{d}_{Om}, d_{1m}) = \frac{F_{Om}D_{Om} + I_{Om}S_{Om}D_{Om}}{1 + L_{Om}S_{Om}D_{Om}} \qquad (4.6.3)$$

$$\zeta(\tilde{d}_{Om}, s_{Om}) = \frac{F_{Om}L_{Om}S_{Om}D_{Om} - I_{Om}S_{Om}}{1 + L_{Om}S_{Om}D_{Om}} \qquad (4.6.4)$$

Supply variance $\tilde{r}_{Om}$ directly affects stock balance and so may also be used to encompass all sources of stock recording errors. Such a noise distribution may be determined as the convolution of all of these or as a compound distribution by empirical observation of their combined effect on stock balance. The sub-system response to this noise is described as follows:-

BEST COPY

AVAILABLE

Variable print quality

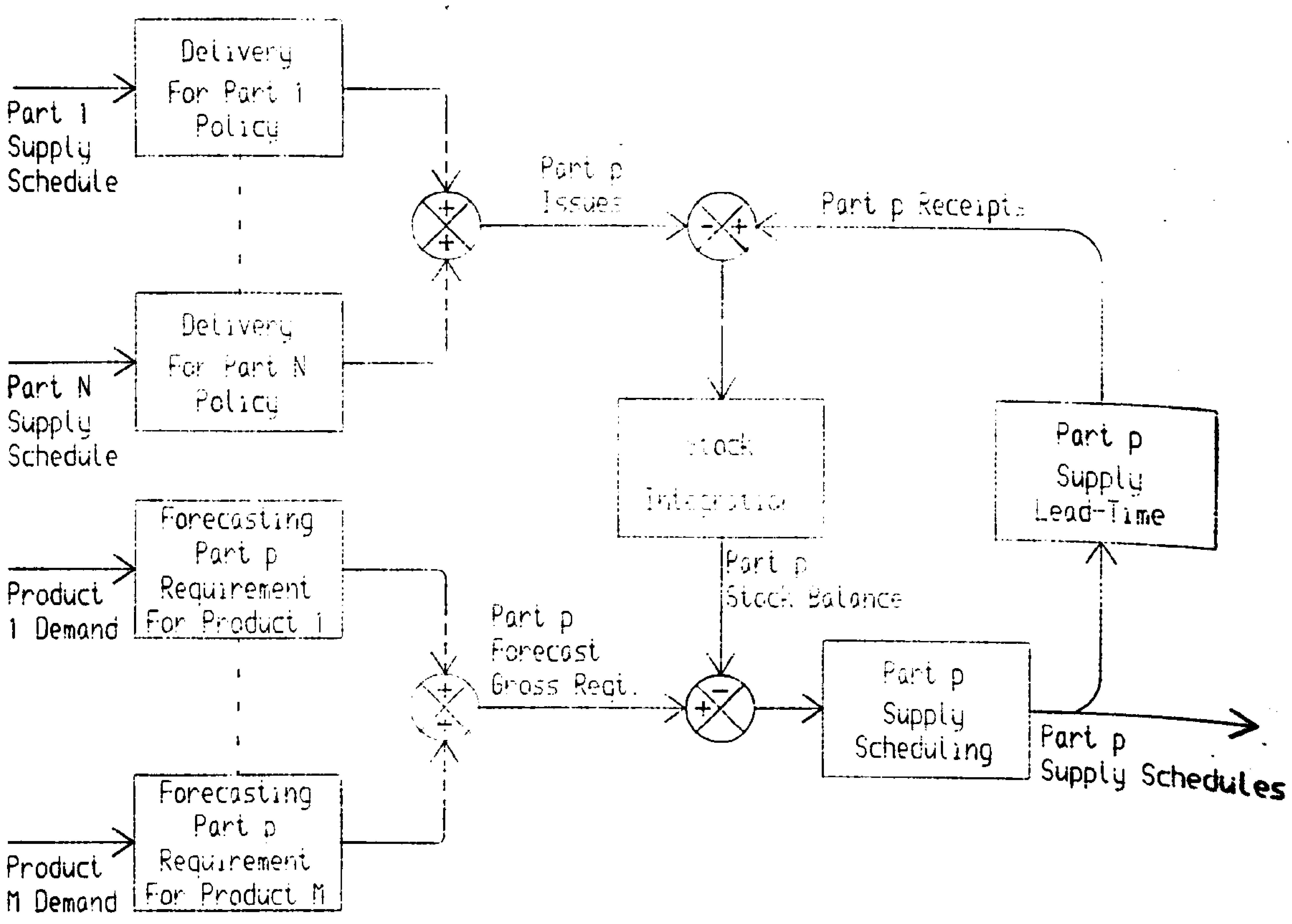

Figure 4.7.a : Part Control Sub-System - Base Information

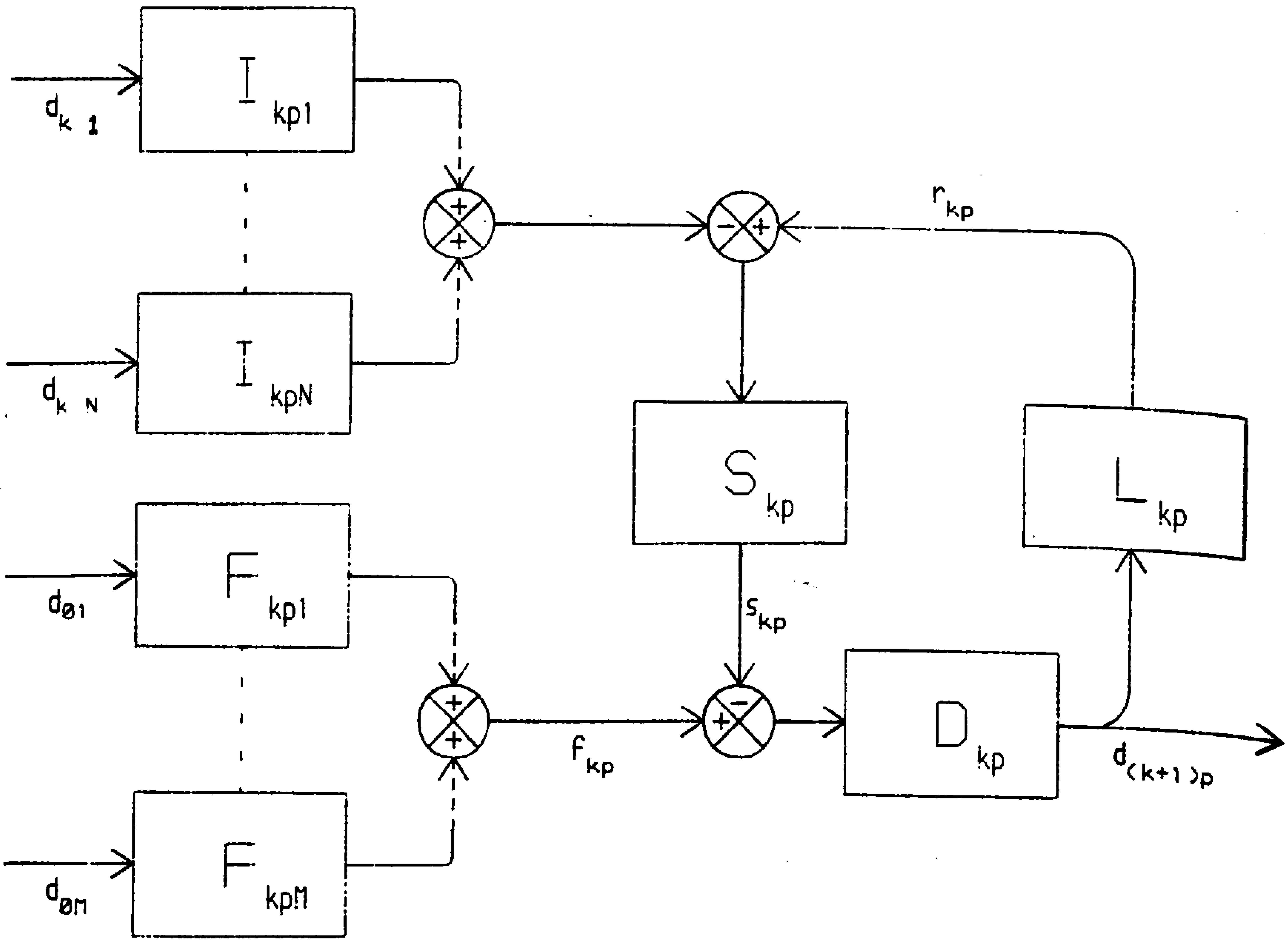

Figure 4.7.b : Part Control Sub-System - Base Information (Symbolic Representation)

$$\zeta(\tilde{r}_{0m}, d_{1m}) = \frac{- S_{0m} D_{0m}}{1 + L_{0m} S_{0m} D_{0m}} \qquad (4.6.5)$$

$$\zeta(\tilde{r}_{0m}, s_{0m}) = \frac{S_{0m}}{1 + L_{0m} S_{0m} D_{0m}} \qquad (4.6.6)$$

We could similarly derive the sub-system z-transfer functions for any causal relationship within the sub-system. However those derived above are the ones likely to be used most frequently.

4.7 The Part Control Sub-System - Base Information System

In the base information system two types of input are applied to each part control sub-system. These are demands for the part arising from production schedules at the next level up and demands for all products using the controlled part. Because the controlled part may be common to a range of products and to a range of higher level parts there may be several sources of each input type. A general form of this sub-system is illustrated (Figure 4.7.a) together with its symbolic representation (Figure 4.7.b).

We shall derive for this sub-system, not z-transfer functions but "source equations" which give rise to all sub-system z-transfer functions from their inputs $d_{01} \ldots d_{0M}$ and $d_{k\,1} \ldots d_{k\,N}$. These source equations include all inputs so the derivation of a specific z-transfer function is a simple matter of using system linearity to let all but the required input become zero.

We first derive the source equation from the demands for the controlled part to its supply schedules. For this, the first such equation, we show the proof.

$$d_{(k+1)p} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp} + \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.7.1)$$

Proof:-

$$d_{(k+1)p} = (f_{kp} - s_{kp})D_{kp}$$

$$= \sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp} - r_{kp}S_{kp}D_{kp} + \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}D_{kp}$$

$$= \sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp} - d_{(k+1)p}L_{kp}S_{kp}D_{kp} + \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}D_{kp}$$

$$\therefore\ d_{(k+1)p}(1 + L_{kp}S_{kp}D_{kp})$$

$$= \sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp} + \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}D_{kp}$$

$$\therefore\ d_{(k+1)p} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp} + \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proofs of subsequent source equations are to be found in Appendix I.

Similarly the part's stock balance responds to demands thus:-

$$s_{kp} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.7.2)$$

It is also possible to derive the sub-system's response to any noise input. We give below the sub-system's z-transfer functions from any of the two types of input (finished product demand and part demand) to its output (supply schedules) as these are necessary for evaluation of the total system response to noise.

$$\zeta\ (\tilde{d}_{Om}, d_{(k+1)p}) = \frac{F_{kpm}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.7.3)$$

$$\zeta\ (\tilde{d}_{kn}, d_{(k+1)p}) = \frac{I_{kpn}S_{kp}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.7.4)$$

We also show the response of stock balance to such noise, as this is a major factor in determining necessary safety stocks.

$$\zeta\ (\tilde{d}_{Om}, s_{kp}) = \frac{F_{kpm}D_{kp}L_{kp}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.7.5)$$

$$\zeta\ (\tilde{d}_{kn}, s_{kp}) = \frac{-I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.7.6)$$

4.8 The Part Control Sub-System - Cascaded System

In the cascaded system only one type of input is applied to the part control sub-system, this being demand arising from production schedules for parts at the next level up. Because the part may be common to a range of higher level parts there may be several such sources of input. This is illustrated in Figure 4.8.a whilst in Figure 4.8.b we substitute symbols for the system elements. In a similar manner to the base information system we can derive source equations which give rise to all the sub-system z-transfer functions from their inputs, $d_{k1} \ldots d_{kN}$. As before we give the source equations from the part demands to supply schedules and to stock balance.

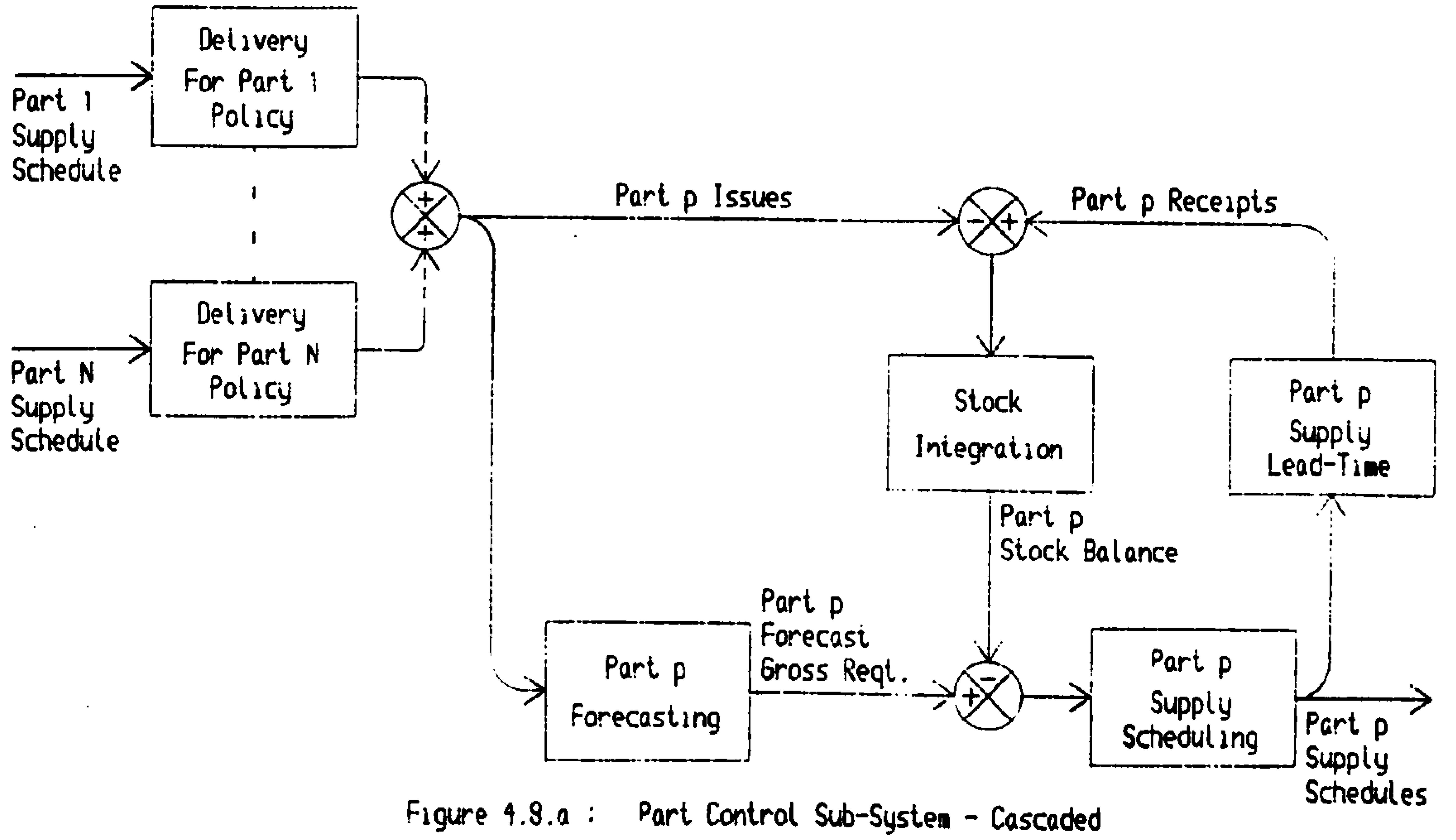

Figure 4.8.a : Part Control Sub-System - Cascaded

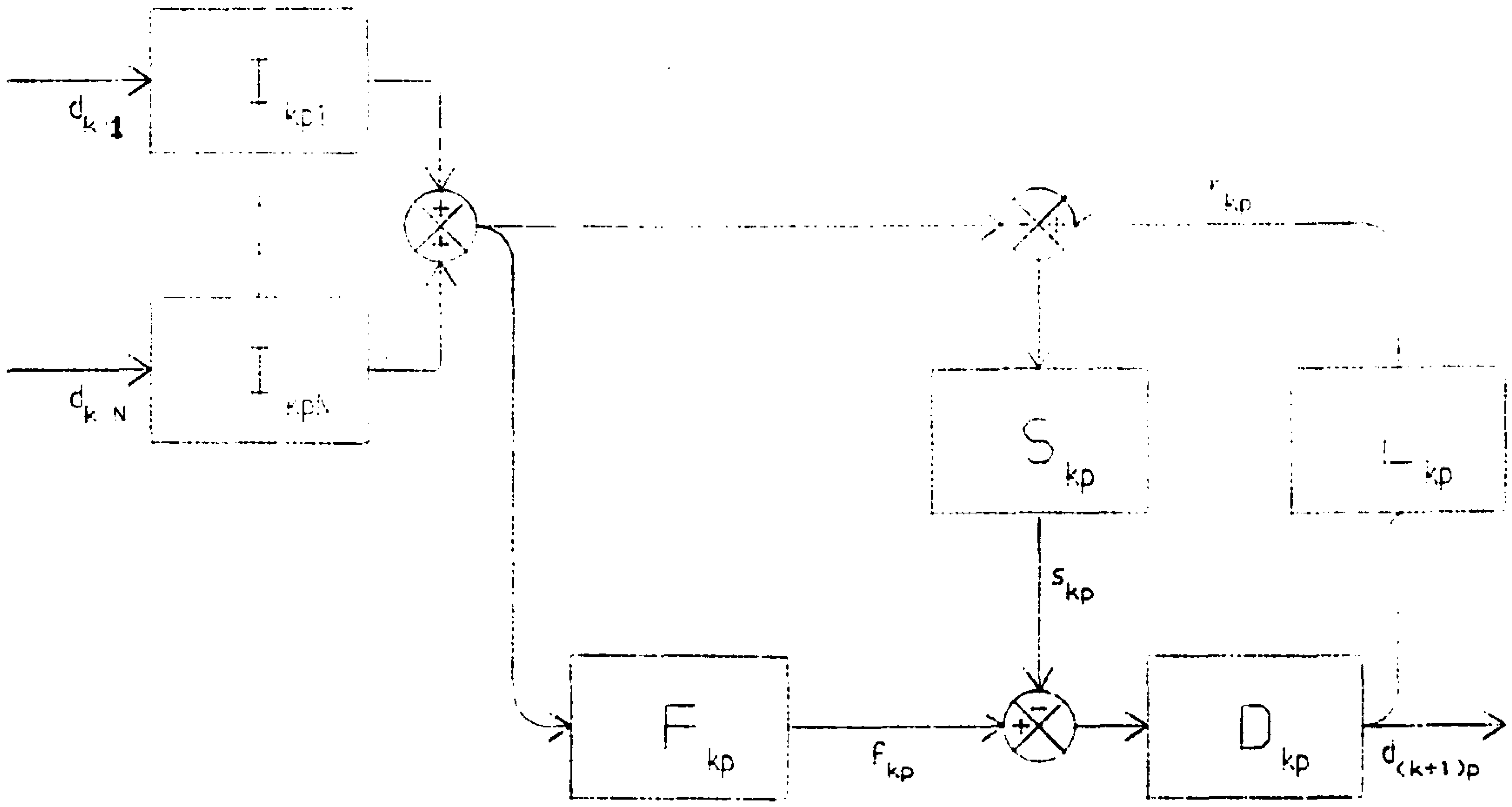

Figure 4.8.b : Part Control Sub-System - Cascaded (Symbolic Representation)

$$d_{(k+1)p} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \quad (4.8.1)$$

$$s_{kp} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \quad (4.8.2)$$

The response to noise which we gave for the base information sub-system are here replaced by the following z-transfer functions.

$$\zeta\ (\tilde{d}_{kn}, d_{(k+1)p}) = \frac{(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.8.3)$$

$$\zeta\ (\tilde{d}_{kn}, s_{kp}) = \frac{(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}} \qquad (4.8.4)$$

4.9 Dummy Part Control Sub-System

When a part at level k_2 is used directly in satisfying demand for production of a part at level k_1 where $k_2 \neq k_1 + 1$, we may introduce as a notational convenience a dummy part control sub-system at each intervening level. The purpose of this sub-system is purely to maintain the rigour of notation used in the model by ensuring that each part is issued for demands at the next highest level only.

Figure 4.9.a represents the action of this dummy sub-system. So that it has no effect upon the demand input, its sub-system z-transfer function is $\zeta\ (d_{kp}, d_{(k+1)p}) = 1$.

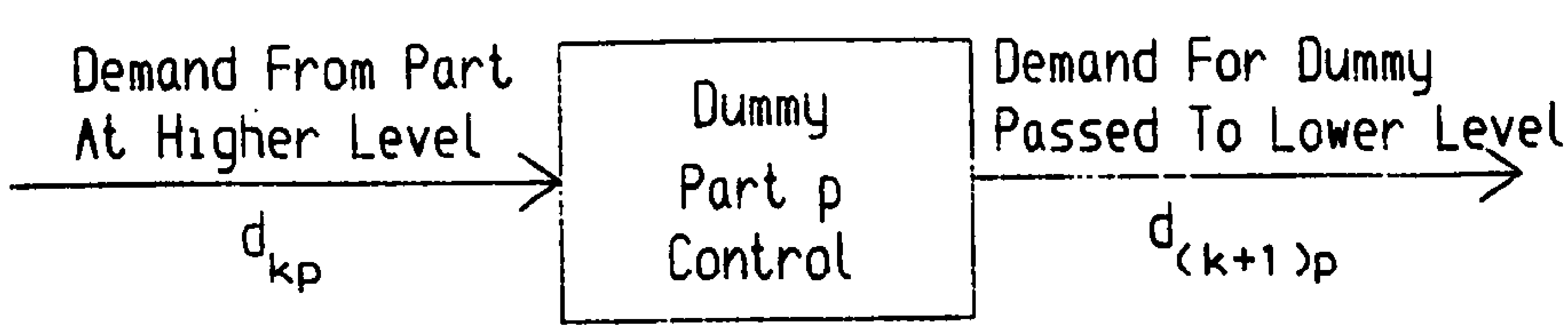

Figure 4.9.a : Dummy Part Control Sub-System

4.10 Overall System Stability

In Chapter 2 we saw that it is a necessary and sufficient condition for system stability that all the roots of the system's characteristic equation lie within the unit circle of the z-plane; the characteristic equation is derived by setting equal to zero the denominator of a system z-transfer function when this is formulated as a quotient of polynomials in z^{-1}.

We note that every sub-system z-transfer function in the general model shows the same form of denominator, namely $1 + L_{kp}S_{kp}D_{kp}$. However, this need not be a polynomial and, in practice, it rarely is. It is thus not an equivalent stability condition for the sub-system that $1 + L_{kp}S_{kp}D_{kp} = 0$ has roots within the unit circle.

Examination of Figure 4.4.b shows a general flow of information from left to right (i.e. from finished product control down to successively lower level control sub-systems). This is because we have assumed that net scheduling will take into account the known abilities of production at lower levels to derive at least feasible schedules. All predictable factors are included and so the only remaining effect which may be imposed by lower level sub-systems is random variation. This is better represented by the imposition of noise on the receipt time-series. Thus by suitable definition of system elements we have eliminated the need to represent feedback loops in the overall system. This is in fact a realistic formulation of many practical systems where feedback is present within each control sub-system but applies between sub-systems only in the event of a non-linearity (e.g. a stockout or overload on production resources). Such non-linearity can only be modelled by the generation of negative stocks.

Because of this one-way flow of information we may derive a general system z-transfer function between any two time-series, g_{jp} and g_{kq}, lying in different sub-systems provided that a true causal relationship exists:-

$$T(g_{jp},g_{kq}) = \zeta(g_{jp},d_{(j+1)p}) \sum_i \{\zeta_{i1}\zeta_{i2} \cdots \zeta_{i(k-j-1)}\zeta(d_{iq},g_{kq})\} \quad (4.10.1)$$

where: $\zeta(g_{jp},d_{(j+1)p})$ is the sub-system z-transfer function from the input time-series to the output of its sub-system.

$\zeta_{i1}\zeta_{i2} \cdots \zeta_{i(k-j-1)}$ is an information route through the full system from the sub-system controlling part p to that controlling part q.

$\zeta(d_{iq},g_{kq})$ is the sub-system z-transfer function from the input to the part q sub-system from route i to the required output time-series g_{kq}.

$\sum_i$ represents summation over all information routes.

Thus every system z-transfer function is a sum of products of sub-system z-transfer functions. If all sub-system z-transfer functions are defined as quotients of polynomials, then derivation of any system z-transfer function will also yield such a quotient. Each route through the system will yield as denominator a product of sub-system z-transfer function denominators, and the system z-transfer function will have their lowest common denominator as its denominator. Hence any system z-transfer function has as denominator some product of sub-system z-transfer function denominators and any roots of the system z-transfer function characteristic equations are also roots of the sub-system z-transfer function characteristic equations. It is thus a sufficient condition of total system stability that all roots of all sub-system z-transfer function characteristic equations lie within the unit circle.

If the roots of any characteristic equation of any z-transfer function of any sub-system lie outside the unit circle, that sub-system is unstable and its instability will manifest itself in any information route through the sub-system. In this event, the complete system cannot be described as stable. Thus it is necessary for complete system stability that all roots of all sub-system z-transfer function characteristic equations lie within the unit circle.

We note, however, that stability in this context implies only that the effect of any impulse input is bounded in time. The transient response to such an impulse or to other inputs may still be unacceptable, as may system response to random noise.

4.11 Transient Responses and Responses to Random Noise

We saw in the last section (Equation 4.10.1) the derivation of any system z-transfer function. We can use this to determine the system's transient response to any input or its response to any source of noise. We note that in general it is not sufficient to use sub-system z-transfer functions as these may not completely describe the relationship between input and output.

4.12 Summary

In this Chapter we have shown a general decomposition of a wide range of production control systems together with the general forms of each of the elements of that decomposition. We have further shown how these elements can be recombined to analyse the entire system.

CHAPTER 5

PRODUCTION CONTROL SYSTEM ELEMENTS

5.1 Preamble

In this Chapter we derive the z-transfer functions of a selection of product delivery policies, forecasting systems, net scheduling systems and part delivery patterns such as may commonly be found in practical production control systems. These are presented for two reasons: firstly to demonstrate that the method has sufficient range of application to be a practical tool, and secondly to provide a basic library of functions for general use.

The catalogue presented can in no way be regarded as exhaustive. However, we attempt to give general versions of a reasonable range of the more commonly applied systems.

We present in Appendix IV a short list of z-transforms of time-series for use in transient analysis, comprehensive tables are published elsewhere [Beightler et al., 1961 & Bishop, 1975].

The catalogue is presented in four sections corresponding to the system elements which may arise in the general model of Chapter 4. (The form of stock integration is fixed and was examined in 3.4.3).

5.2 Product Delivery Policies

We consider that there are only two ways in which a product delivery policy may act upon orders: namely by delaying delivery for some finite period or by spreading deliveries over a finite number of periods. These two can be combined both in practice and in theory to provide a single general system element encompassing all policies.

If customers' demand for product m in period t is $d_{Om}(t)$ then deliveries against $d_{Om}(t)$ shall be $i_{Om}(t+\delta) = \phi_\delta d_{Om}(t)$ where δ is an integral number of periods not less than zero and not greater than some finite maximum Δ, and where ϕ_δ is the fraction of demand to be delivered after a delay of Δ periods. (Thus $\phi_0 + \phi_1 + \ldots + \phi_\Delta = 1$ unless there is a policy of over or under delivery). This is illustrated in Figure 5.2.a.

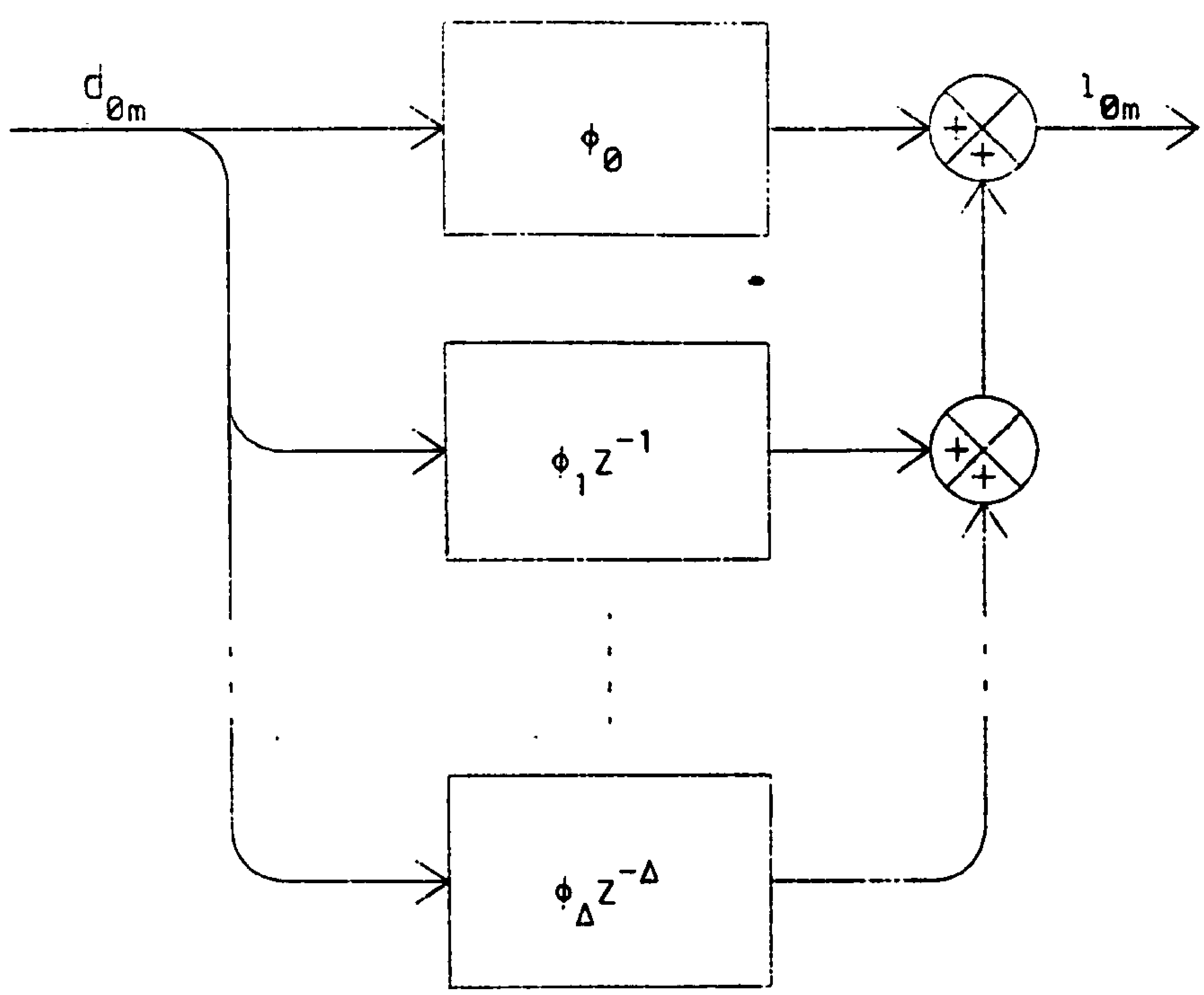

Figure 5.2.a : Product Delivery Policies

From this it is clear that:-

$$i_{Om} = d_{Om}\phi_O + d_{Om}\phi_1 z^{-1} + \dots + d_{Om}\phi_\Delta z^{-\Delta}$$

and so we have a general z-transfer function:-

$$I_{Om} = \frac{i_{Om}}{d_{Om}} = \sum_{\delta=0}^{\Delta} \phi_\delta z^{-\delta} \qquad (5.2.1)$$

5.3 Forecasting Systems

The purpose of forecasting is to provide a gross requirement over some future period. The duration of this period is determined by the form of scheduling routine, so here we present forecasting routines which output a forecast for P future periods. Where the forecast takes no account of trend this is simply P times the current forecast; trend predictors will need the use of more complex accumulations.

Weighted Moving Average

This forecasting technique is commonly used because of its apparent simplicity. Procedurally it is in fact more complex than exponential smoothing methods, but its superficial ease of comprehension makes it a popular clerical technique.

The procedure is to average demands over some fixed, finite, past period H + 1, and to use this average as a forecast. We generalise this by allowing the predicted average to be a weighted average of past demands:

$$f_{ij}(t) = P \sum_{h=0}^{H} \phi_h d_{ij}^{(t-h)}$$

where ϕ_h are weighting factors and $\sum_{h=0}^{H} \phi_h = 1$.

This is illustrated in Figure 5.3.a.

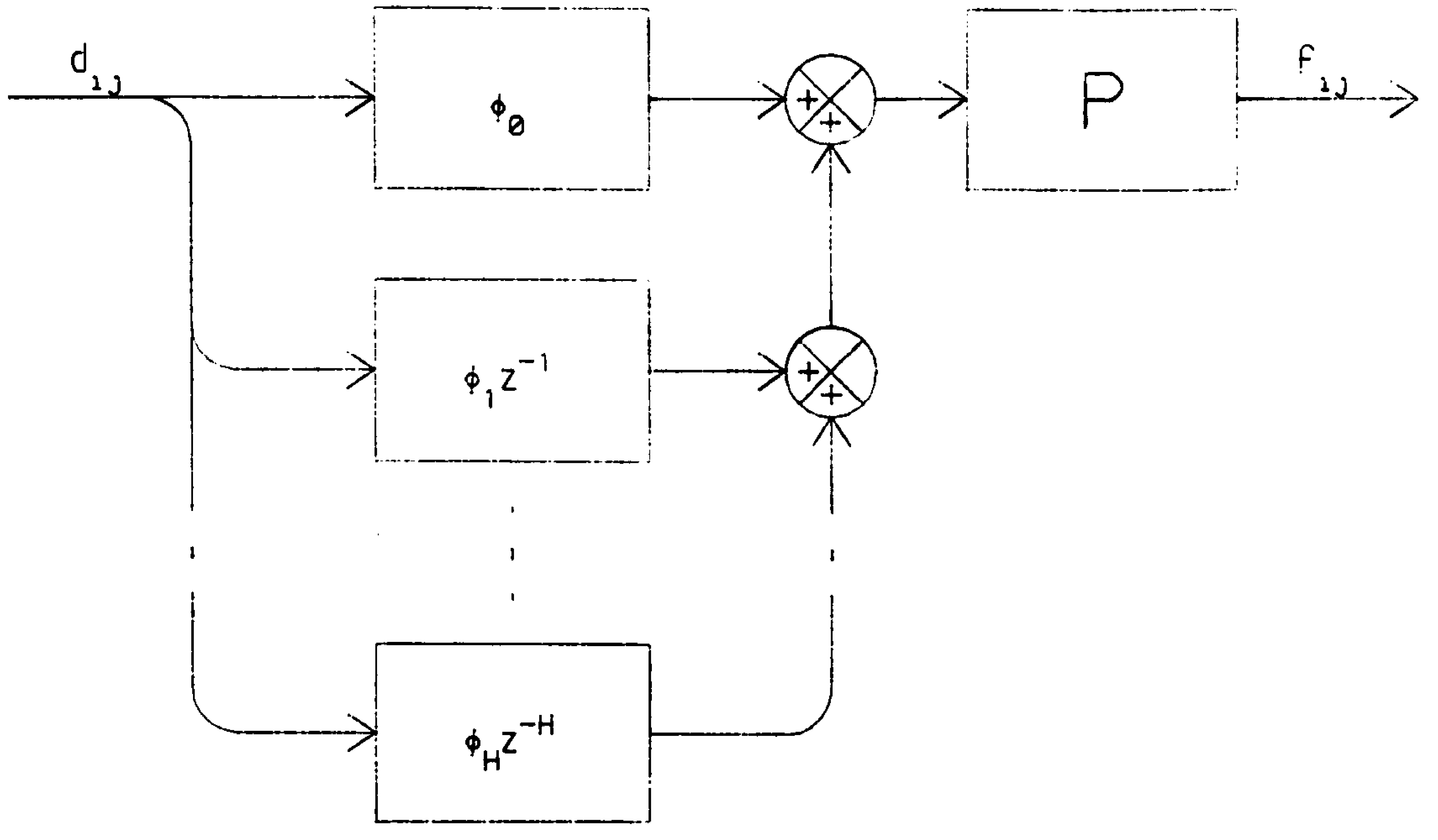

Figure 5.3.a : Weighted Moving Average

From the figure it is clear that:-

$$f_{ij} = P\{\phi_0 + \phi_1 z^{-1} + \dots + \phi_H z^{-H}\} d_{ij}$$

and so the z-transfer function for weighted moving average forecasting is:-

$$F_{ij} = \frac{f_{ij}}{d_{ij}} = P \sum_{h=0}^{H} \phi_h z^{-h} \qquad (5.3.1)$$

Note that this contains, as special cases, both unweighted moving averages, where $\phi_0 = \phi_1 = \dots = \phi_H = \frac{1}{H+1}$, and the trivial forecasting system which assumes that the present demand level will continue indefinitely; in this case $H = 0$ and $\phi_0 = 1$.

Single Exponential Smoothing

The procedure of single exponential smoothing is one of the simplest statistical forecasting techniques. The exponentially smoothed forecast of demand for the next P periods is:-

$$f_{ij}(t) = P\{\alpha d_{ij}(t) + (1 - \alpha)\ \frac{f_{ij}(t-1)}{P}\ \}$$

for some α normally such that $0 < \alpha < 1$. We illustrate this procedure in Figure 5.3.b.

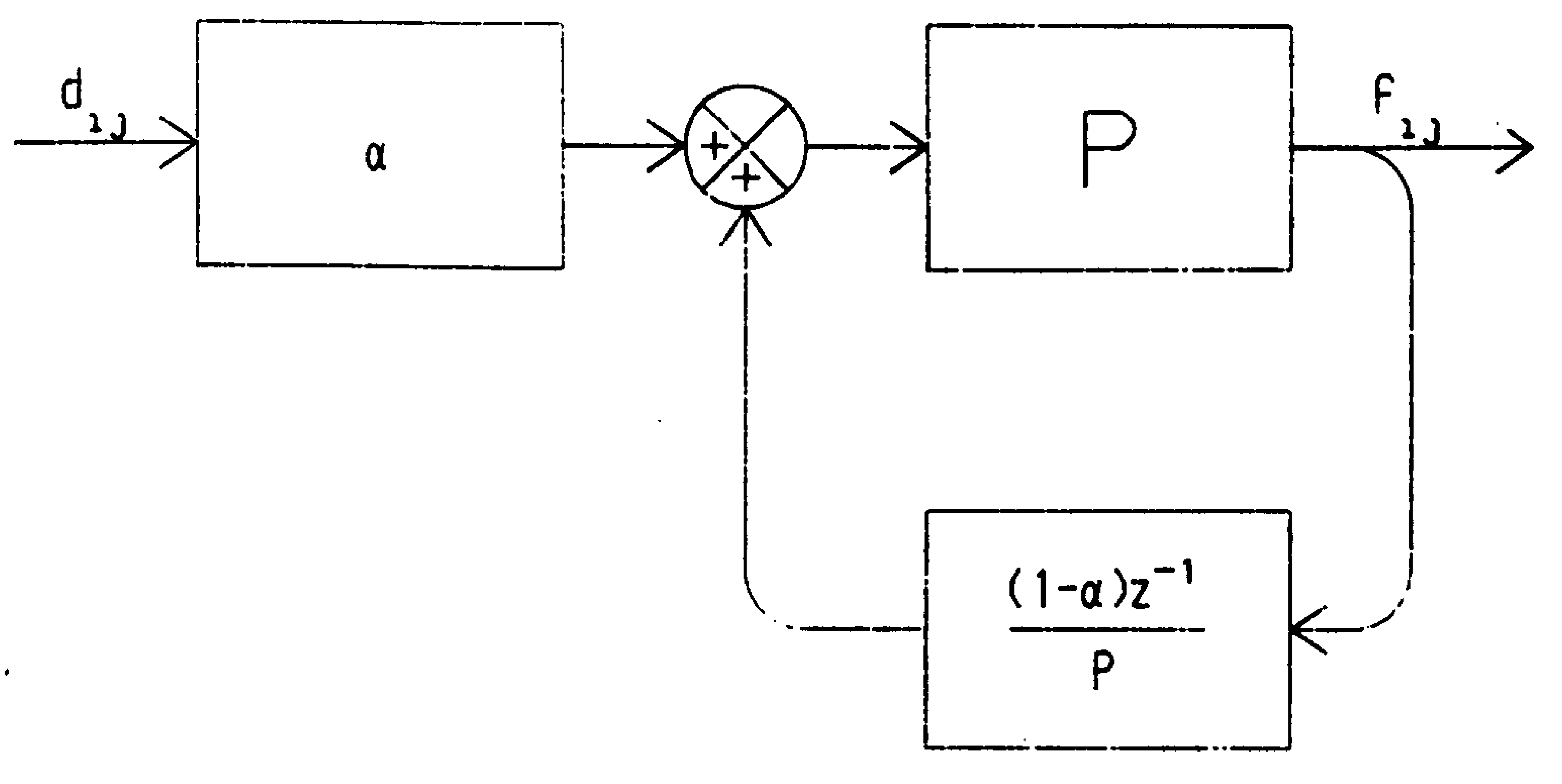

Figure 5.3.b : Single Exponential Smoothing

From the diagram we have:-

$$f_{ij} = P\{\alpha d_{ij} + (1 - \alpha)\frac{f_{ij}}{P} z^{-1}\}$$

$$\therefore F_{ij} = \frac{f_{ij}}{d_{ij}} = \frac{P\alpha}{1 - (1 - \alpha)z^{-1}} \qquad (5.3.2)$$

Double Exponential Smoothing

This sophistication of exponential smoothing enables a forecast to detect an underlying trend in the demand time-series. It uses the difference between the exponentially smoothed average of past demand and the exponentially smoothed average of this average as a trend predictor. The procedure is described in full by [Brown, 1963] where the following algorithm is given:-

$$\sigma_{ij}(t) = \alpha d_{ij}(t) + (1 - \alpha)\sigma_{ij}(t-1)$$

$$\rho_{ij}(t) = \alpha\sigma_{ij}(t) + (1 - \alpha)\rho_{ij}(t-1)$$

$$\hat{a}_{ij}(t) = 2\sigma_{ij}(t) - \rho_{ij}(t)$$

$$\hat{b}_{ij}(t) = \frac{\alpha}{1-\alpha}(\sigma_{ij}(t) - \rho_{ij}(t))$$

$$\hat{x}_{ij}(t+\mu) = \hat{a}_{ij}(t) + \mu\hat{b}_{ij}(t) \qquad \text{for all } \mu \geq 0$$

where $\sigma_{ij}(t)$ is the exponentially smoothed average of demands to time t.

$\rho_{ij}(t)$ is the exponentially smoothed average of σ_{ij} to time t.

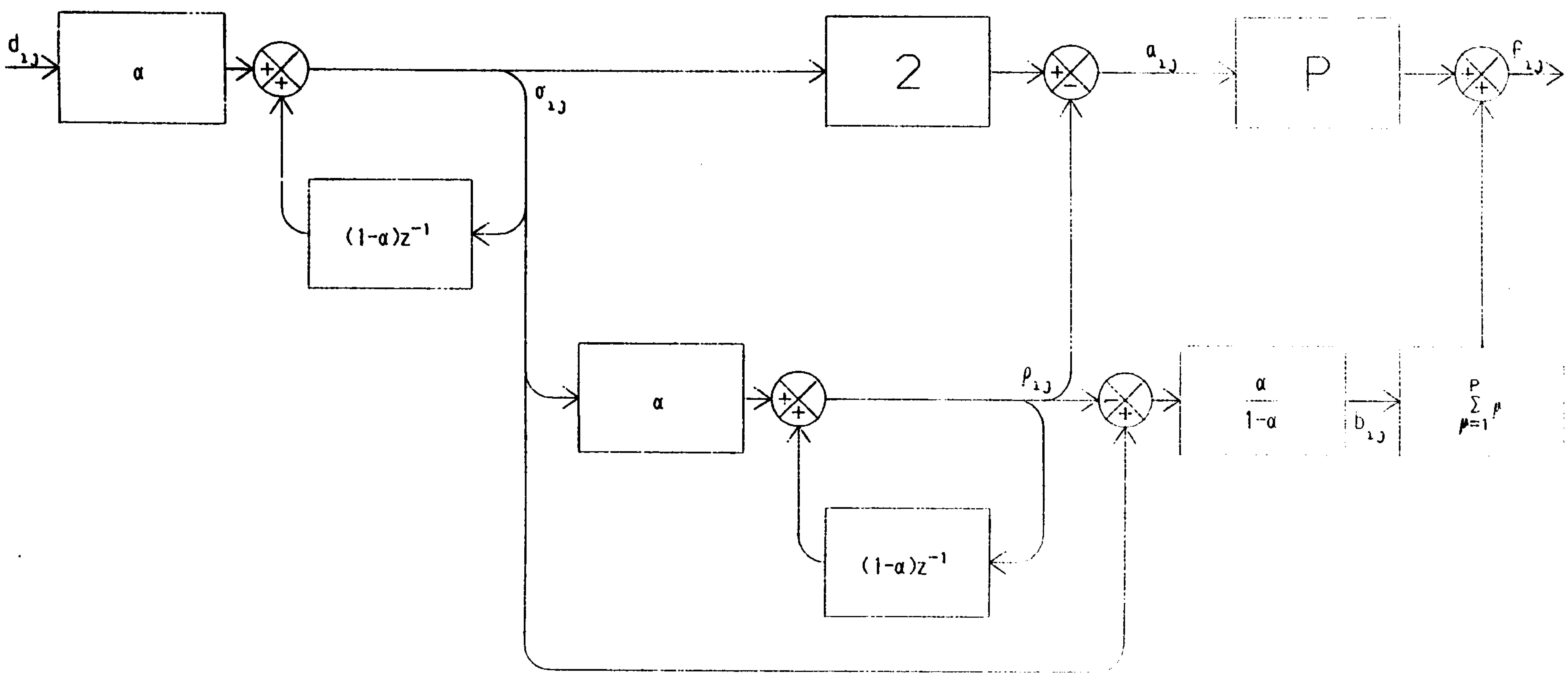

Figure 5.3.c : Double Exponential Smoothing

$\hat{a}_{ij}(t)$ is the expected value of demand at time t (= $\hat{x}_{ij}(t)$).

$\hat{b}_{ij}(t)$ is the trend predictor at time t.

$\hat{x}_{ij}(t+\mu)$ is the forecast made at time t for period t + μ.

From this it is clear that the forecast of total demand over the next P periods, for use in scheduling is:-

$$f_{ij}(t) = \sum_{\mu=1}^{P} \hat{x}_{ij}(t+\mu) = P\hat{a}_{ij} + \hat{b}_{ij}(t) \sum_{\mu=1}^{P} \mu$$

This figure can be represented by Figure 5.3.c whence we can see that in the z-domain:-

$$f_{ij} = \hat{b}_{ij} \sum_{\mu=1}^{P} + P\hat{a}_{ij}$$

$$= (\sigma_{ij} - \rho_{ij}) \frac{\alpha}{1-\alpha} \sum_{\mu=1}^{P} \mu + (2\sigma_{ij} - \rho_{ij})P$$

$$= (2P + \frac{\alpha}{1-\alpha} \sum_{\mu=1}^{P} \mu)\sigma_{ij} - (P + \frac{\alpha}{1-\alpha} \sum_{\mu=1}^{P} \mu)\rho_{ij}$$

But $\rho_{ij} = \dfrac{\alpha\sigma_{ij}}{1 - (1 - \alpha)z^{-1}}$ (single exponential smoothing)

$$\therefore f_{ij} = (2P + \frac{\alpha}{1-\alpha} \sum_{\mu=1}^{P} \mu)\sigma_{ij} - \frac{(P + \frac{\alpha}{1-\alpha} \sum_{\mu=1}^{P} \mu)\alpha\sigma_{ij}}{1 - (1 - \alpha)z^{-1}}$$

$$= \left\{ \frac{(2P + \frac{\alpha}{1-\alpha} \sum_{\mu=1}^{P} \mu)(1 - (1 - \alpha)z^{-1}) - \alpha(P + \frac{\alpha}{1-\alpha} \sum_{\mu=1}^{P} \mu)}{1 - (1 - \alpha)z^{-1}} \right\} \sigma_{ij}$$

But $\sigma_{ij} = \dfrac{\alpha d_{ij}}{1 - (1-\alpha)z^{-1}}$ (single exponential smoothing)

$$f_{ij} = \alpha d_{ij} \left\{ \frac{2P + \frac{\alpha}{1-\alpha}\sum_{\mu=1}^{P} \mu - 2P(1-\alpha)z^{-1} - \alpha z^{-1}\sum_{\mu=1}^{P} \mu - \alpha P - \alpha\left(\frac{\alpha}{1-\alpha}\right)\sum_{\mu=1}^{P} \mu}{(1-(1-\alpha)z^{-1})^2} \right\}$$

$$= \alpha d_{ij} \left\{ \frac{(2-\alpha)P + \alpha\sum_{\mu=1}^{P} \mu - (2P(1-\alpha) + \alpha\sum_{\mu=1}^{P} \mu)z^{-1}}{(1-(1-\alpha)z^{-1})^2} \right\}$$

and so we have:-

$$F_{ij} = \frac{f_{ij}}{d_{ij}} = \frac{\alpha\{(2-\alpha)P + \alpha\sum_{\mu=1}^{P} \mu - (2P(1-\alpha) + \alpha\sum_{\mu=1}^{P} \mu)z^{-1}\}}{(1-(1-\alpha)z^{-1})^2} \qquad (5.3.3)$$

We note that the characteristic equation here, $(1 - (1-\alpha)z^{-1})^2 = 0$, has the same roots as for single exponential smoothing, and so the same stability condition will apply (i.e. $0<\alpha<2$).

Box Jenkins Models

Box and Jenkins [Box et. al., 1970] describe time-series containing stochastic elements in terms of "auto-regressive integrated moving averages"; this term is usually abbreviated to ARIMA. Using the notation conventionally employed an ARIMA is of the form:-

$$\Phi_p(B)\nabla^d x(t) = \theta_q(B)\varepsilon(t)$$

where x(t) is the time-series at time t.

$\varepsilon(t) = x(t) - \hat{x}(t)$ where $\hat{x}(t)$ is the expected value of the time-series at time t; thus $\varepsilon(t)$ is the stochastic component of the time-series.

B is the backward translation operator such that

$$Bx(t) = x(t-1).$$

∇ is the backward difference operator such that

$$\nabla x(t) = x(t) - x(t-1).$$

Φ_p is a polynomial of degree p with coefficients $\phi_0, \phi_1, \ldots \phi_p$.

Θ_q is a polynomial of degree q with coefficients $\theta_0, \theta_1, \ldots \theta_q$.

The literature implies limits on the ranges p, d, q and the coefficients of Φ and Θ. Within these limits the precise values chosen determine the model's rate of response to differing orders of trend. Selection of precise values must be made in the context of the system where the model is to be used as a forecasting agent. It is fortunate therefore that the predictor derived from this model is a linear transformation for which a z-transfer function is easily derived. The use of this within a complete system model allows examination of stability, transient response and noise transmission of the whole system and so assists selection of p, d, q, Φ and Θ.

The backward translation operator B clearly has z-transfer function z^{-1}, whilst the backward difference operator ∇ has z-transfer function $1 - z^{-1}$. We can therefore rewrite the above expression in the z-domain as:-

$$\Phi_p(z^{-1})(1 - z^{-1})^d x(z) = \Theta_q(z^{-1})\varepsilon(z)$$

and so:-

$$x(z) = \frac{\Theta_q(z^{-1})}{\Phi_p(z^{-1})(1 - z^{-1})^d} \varepsilon(z)$$

Any quotient of polynomials is expressible as an infinite polynomial so we can rewrite this as:-

$$x(z) = \Psi(z^{-1})\varepsilon(z)$$

for some infinite polynomial $\Psi(z^{-1}) = \dfrac{\Theta_q(z^{-1})}{\Phi_p(z^{-1})(1 - z^{-1})^d}$

Returning to the t-domain we have:-

$$x(t) = \psi_0\varepsilon(t) + \psi_1\varepsilon(t-1) + \psi_2\varepsilon(t-2) + \ldots$$

and similarly

$$x(t+1) = \psi_0\varepsilon(t+1) + \psi_1\varepsilon(t) + \psi_2\varepsilon(t-1) + \ldots$$

$$x(t+2) = \psi_0\varepsilon(t+2) + \psi_1\varepsilon(t+1) + \psi_2\varepsilon(t) + \ldots$$

etcetera.

In using the ARIMA as a predictor it is necessary to assume that the stochastic element, $\varepsilon(t)$, is zero for all future times: that is, to predict $x(T+1)$ at time T we assume $\varepsilon(t) = 0$ for all $t > T$. Thus the last result, used as a predictor, takes the form:-

$$x(t+1) = \psi_1\varepsilon(t) + \psi_2\varepsilon(t-1) + \psi_3\varepsilon(t-2) + \ldots$$

$$x(t+2) = \psi_2\varepsilon(t) + \psi_3\varepsilon(t-1) + \psi_4\varepsilon(t-2) + \ldots$$

$$X(t+3) = \psi_3\varepsilon(t) + \psi_4\varepsilon(t-1) + \psi_5\varepsilon(t-2) + \ldots$$

etcetera.

Hence the total forecast demand over P future periods,

$x(t+1) + x(t+2) + \ldots + x(t+P)$, is:-

$$f_{ij}(t) = \sum_{\mu=1}^{P} x(t+\mu)$$

$$= \sum_{\mu=1}^{P} \{\psi_{\mu}\varepsilon(t) + \psi_{\mu+1}\varepsilon(t-1) + \psi_{\mu+2}\varepsilon(t-2) + \ldots\}$$

$$= \sum_{\mu=1}^{P} \sum_{\lambda=0}^{\infty} \psi_{\mu+\lambda} \varepsilon(t-\lambda)$$

In the z-domain we thus have:-

$$f_{ij}(z) = \sum_{\mu=1}^{P} \sum_{\lambda=0}^{\infty} \psi_{\mu+\lambda} z^{-\lambda}\varepsilon(z)$$

Figure 5.3.d represents this forecasting system.

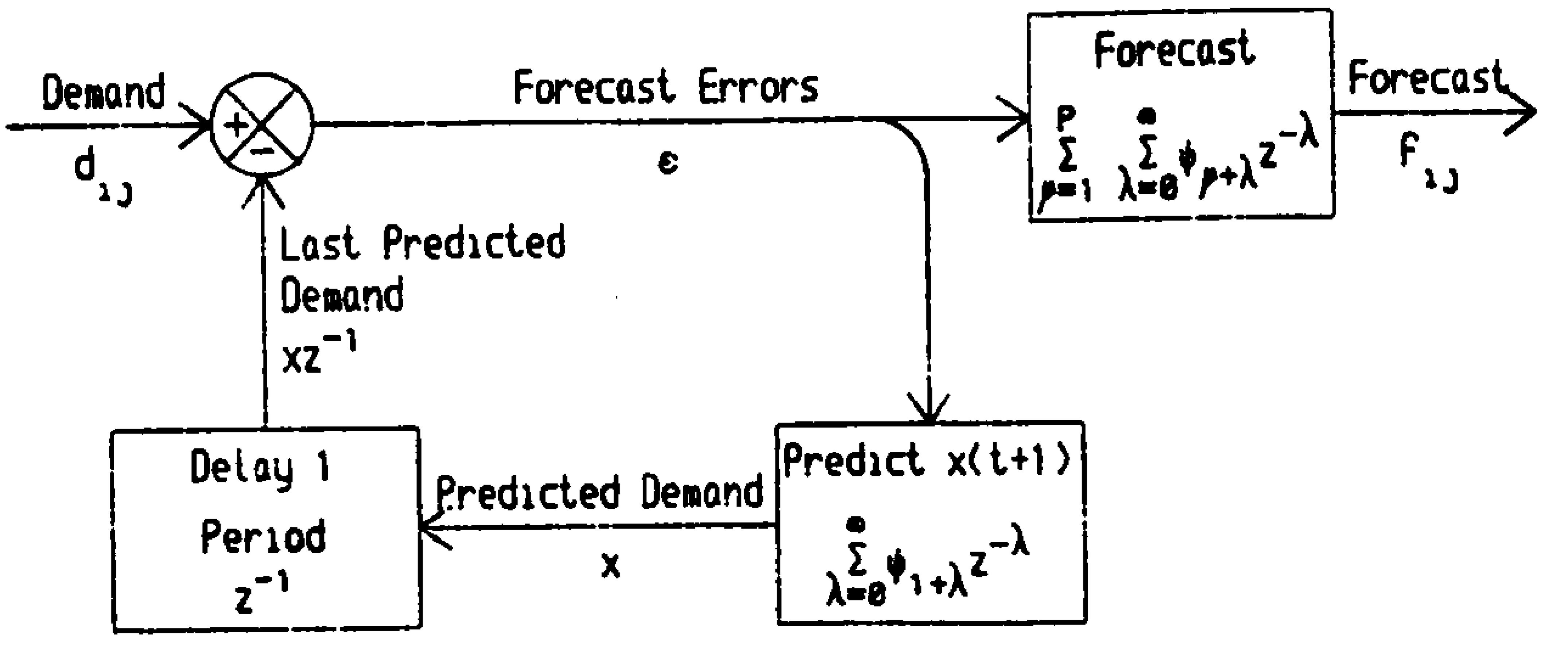

Figure 5.3.d : Box-Jenkins Forecasting

The development of the z-transfer function F_{ij} is now straight-forward. From the diagram:-

$$\varepsilon = d_{ij} - z^{-1}x$$

$$= d_{ij} - z^{-1}\left(\sum_{\lambda=0}^{\infty} \psi_{1+\lambda} z^{-\lambda}\right)\varepsilon$$

$$\therefore \varepsilon = \frac{d_{ij}}{1 + \sum_{\lambda=1}^{\infty} \psi_{\lambda} z^{-\lambda}}$$

But $f_{ij} = \sum_{\mu=1}^{P} \sum_{\lambda=0}^{\infty} \psi_{\mu+\lambda} z^{-\lambda} \varepsilon$

$$= \frac{\sum_{\mu=1}^{P} \sum_{\lambda=0}^{\infty} \psi_{\psi+\lambda} z^{-\lambda}}{1 + \sum_{\lambda=1}^{\infty} \psi_{\lambda} z^{-\lambda}} d_{ij}$$

Hence we have:-

$$F_{ij} = \frac{\sum_{\mu=1}^{P} \sum_{\lambda=0}^{\infty} \psi_{\mu+\lambda} z^{-\lambda}}{1 + \sum_{\lambda=1}^{\infty} \psi_{\lambda} z^{-\lambda}} \qquad (5.3.4)$$

This is the most intricate forecasting z-transfer function derived, and we should expect this as Box-Jenkins in the general case is a complicated technique. In practical cases the values of p, d and q are normally very small (values of 1, 2 and 3 are common), and since future forecasts are unlikely to be based upon errors in the distant past we can expect values of ψ_λ to converge rapidly to zero.

5.4 Net Scheduling

Scheduling systems may or may not embody some form of production smoothing rules. It is necessary to distinguish between production smoothing rules and any forecast smoothing mechanism. Forecast smoothing is intended to eliminate from a forecast the effects of random fluctuations superimposed on an underlying demand pattern. Such fluctuations must be met from a pre-set safety stock which is maintained by the scheduling system. Production smoothing rules are designed to reduce the rate of change of schedules, and for such rules to be effective adequate provision must be made to support smoothing stocks.

Net Scheduling Without Smoothing

We assume that the production lead-time is P periods. That is, a schedule generated at time t will be delivered in the period leading up to time t + P. A net schedule is calculated by subtracting from the forecast of demand over the coming P periods both the current stock balance and the schedules already placed for delivery during the intervening P - 1 periods. This value may be modified by some scalar factor σ to provide safety stocks. Remembering that the first subtraction (stock balance from forecast) is an element of the overall part block diagram, we can represent this process as in Figure 5.4.a.

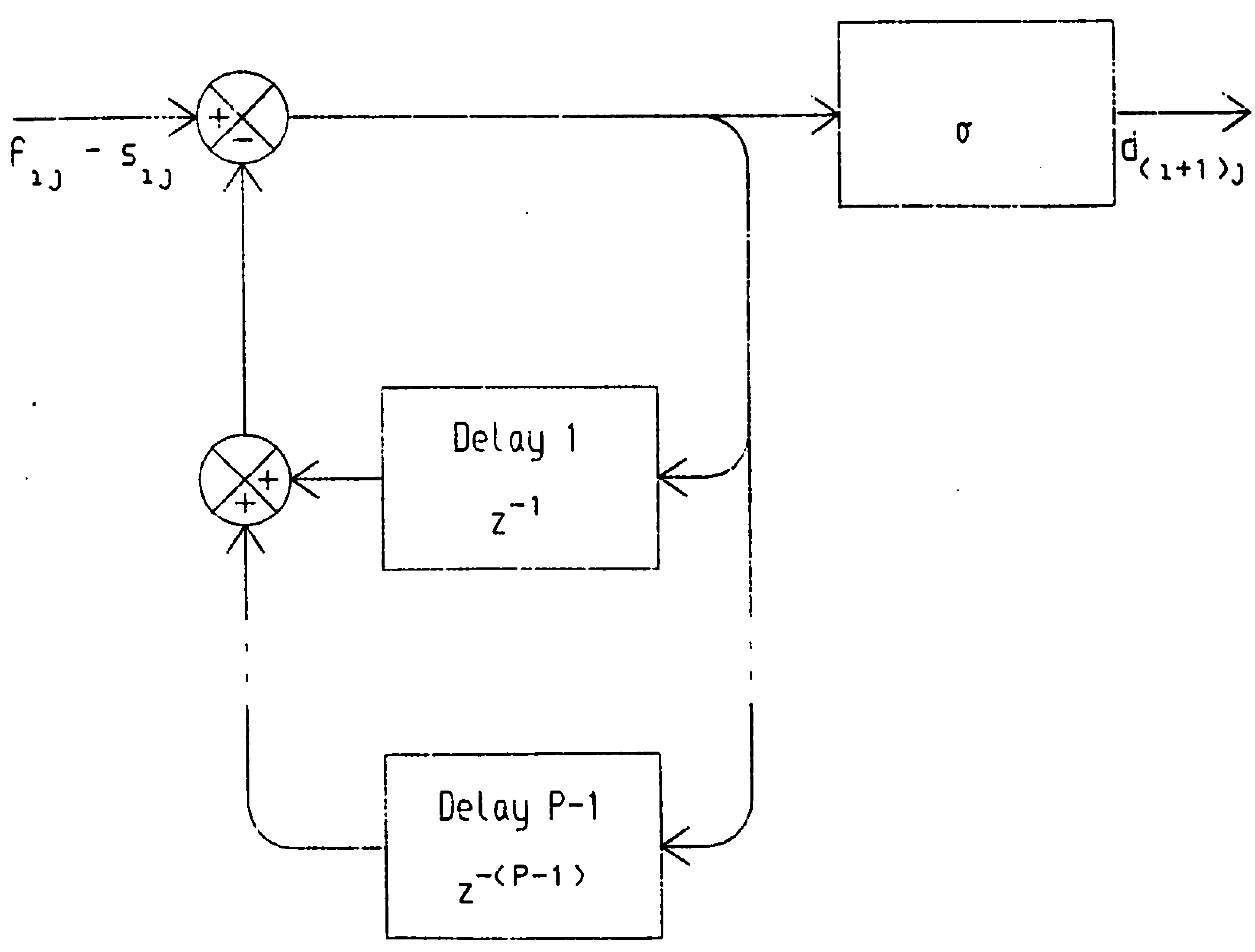

Figure 5.4.a : Net Scheduling Without Smoothing

From the figure it is clear that:-

$$d_{(i+1)j} = \sigma\{f_{ij} - s_{ij} - d_{(i+1)j}z^{-1} - \dots - d_{(i+1)j}z^{-(P-1)}\}$$

$$= \sigma(f_{ij} - s_{ij}) - \sigma d_{(i+1)j}(z^{-1} + z^{-2} + \dots + z^{-(P-1)})$$

ans so we have a z-transfer function:-

$$D_{ij} = \frac{d_{(i+1)j}}{f_{ij} - s_{ij}} = \frac{\sigma}{1 + \sigma(z^{-1} + z^{-2} + \dots + z^{-(P-1)})} \qquad (5.4.1)$$

Table 5.4.b shows the stability conditions which apply for lead times up to 3 periods. For higher values of P other algorithms (Chapter 2) may be used to determine the conditions under which scheduling is stable.

Lead-time P	Stability Condition
1	All values of σ.
2	$\lvert\sigma\rvert < 1$.
3	$\left\lvert \frac{2\sigma}{\sigma \pm \sqrt{\sigma^2 - 4\sigma}} \right\rvert < 1$

Table 5.4.b

Net Scheduling With Smoothing

We assume a lead-time of P periods and first calculate an unsmoothed schedule w, modified by a safety factor σ. Here however we shall attempt to smooth production by reducing the change from the last schedule to the newly calculated schedule by a fixed proportion α of this change. Thus the actual schedule to be placed is calculated as:-

$$d_{(i+1)j}(t) = \sigma w - \alpha(\sigma w - d_{(i+1)j}(t-1))$$

We represent this process in Figure 5.4.c.

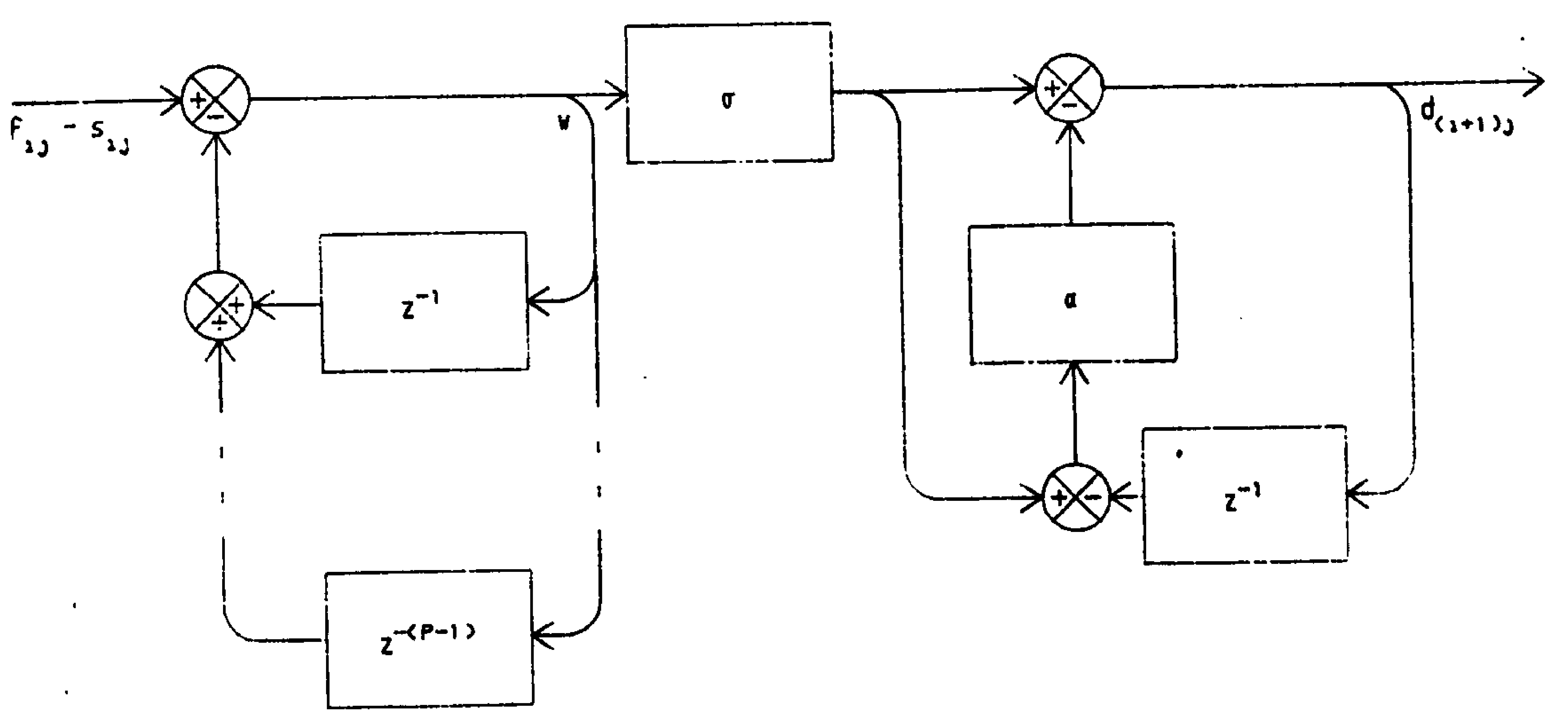

Figure 5.4.c : Net Scheduling With Smoothing

From the figure we see:-

$$d_{(i+1)j} = \sigma w - \alpha(\sigma w - d_{(i+1)j}\, z^{-1})$$

$$= (1 - \alpha)\sigma w + \alpha d_{(i+1)j} z^{-1}$$

$$= \sigma(1 - \alpha)\{f_{ij} - s_{ij} - d_{(i+1)j} z^{-1} - d_{(i+1)j} z^{-2} - \dots$$

$$\dots\ d_{(i+1)j} z^{-(P-1)}\} + \alpha d_{(i+1)j} z^{-1}$$

$$= \sigma(1 - \alpha)(f_{ij} - s_{ij}) + d_{(i+1)j}\{\alpha z^{-1} - \sigma(1 - \alpha)(z^{-1} + z^{-2} + \dots$$

$$\dots + z^{-(P-1)}\}$$

Thus we have the z-transfer function:-

$$D_{ij} = \frac{d_{(i+1)j}}{f_{ij} - s_{ij}} = \frac{\sigma(1 - \alpha)}{1 - \alpha z^{-1} + \sigma(1-\alpha)\{z^{-1} + z^{-2} + \dots + z^{-(P-1)}\}} \qquad (5.4.2)$$

Net Scheduling With Arrears/Over-deliveries

Here we consider a mechanism where outstanding schedules are maintained by accumulating from period to period the difference between scheduled and delivered quantities. A new schedule is then calculated by subtracting from forecast demand both current stock balance and this outstanding schedule figure. This process is illustrated by Figure 5.4.d. In this case we have, for simplicity, applied no schedule smoothing although it is quite straightforward to apply smoothing rules as in Figure 5.4.c.

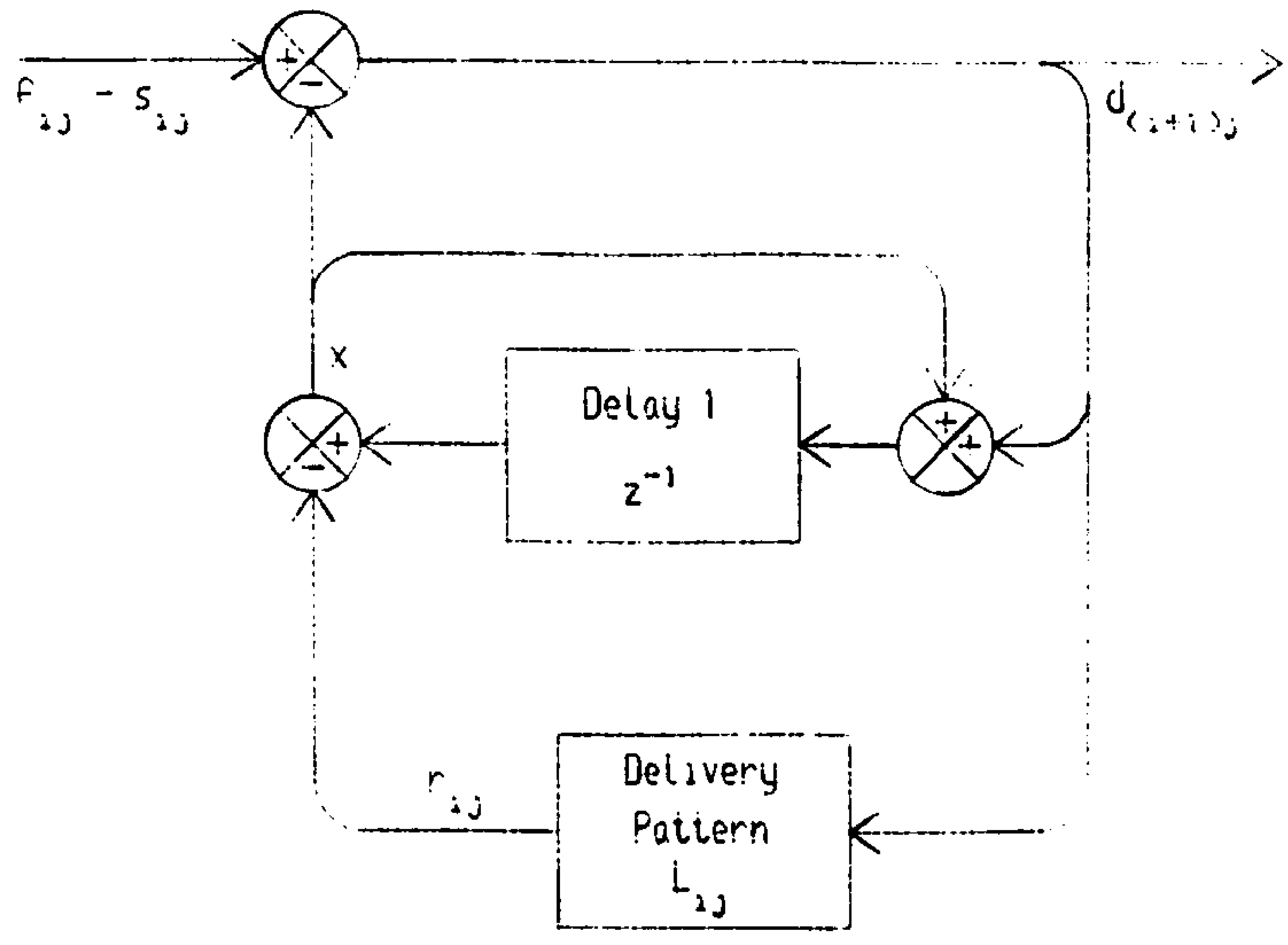

Figure 5.4.d : Net Scheduling With Arrears/Overdeliveries

L_{ij} is the same z-transfer function as appears elsewhere in the part block diagram transforming $d_{(i+1)j}$ to r_{ij}.

Note that the time-series x is introduced only as a temporary notational convenience and represents outstanding schedules in the following derivation of the z-transfer function D_{ij}.

$$x = z^{-1}(x + d_{(i+1)j}) - L_{ij}d_{(i+1)j}$$

$$\therefore \quad x = \frac{z^{-1} - L_{ij}}{1 - z^{-1}} \, d_{(i+1)j}$$

But $d_{(i+1)j} = f_{ij} - s_{ij} - x$

$$= f_{ij} - s_{ij} - \frac{z^{-1} - L_{ij}}{1 - z^{-1}} \, d_{(i+1)j}$$

$$d_{(i+1)j} \left(1 + \frac{z^{-1} - L_{ij}}{1 - z^{-1}}\right) = f_{ij} - s_{ij}$$

$$\therefore \quad d_{(i+1)j} \left(\frac{1 - L_{ij}}{1 - z^{-1}}\right) = f_{ij} - s_{ij}$$

Thus the z-transform D_{ij} is:-

$$D_{ij} = \frac{d_{(i+1)j}}{f_{ij} - s_{ij}} = \frac{1 - z^{-1}}{1 - L_{ij}} \quad (5.4.3)$$

Since this z-transfer function has characteristic equation $1 - L_{ij} = 0$ (see 5.5 for form of L_{ij}), the system's stability depends upon the form of the delivery pattern L_{ij}.

N.B. It is essential in examining the response of a system of this nature to random noise imposed upon receipts from suppliers, that the noise input $\tilde{r}_{ij}$ must be included twice: once (as standard) before receipts are added to stock; secondly before receipts are deducted from outstanding schedules. As a result the net scheduling response to $\tilde{r}_{ij}$ cancels the effect of such noise on stock balance and so d_{ij} becomes independent of $\tilde{r}_{ij}$.

5.5 Part Delivery Patterns

In the last section (5.4) we examined net scheduling systems which included smoothing mechanisms. Such smoothing is a deliberate act in planning part supplies. We now examine the manner in which the supplier responds to schedules. Ideally the supplier will respond precisely to the schedules raised, and the delivery transform is a simple delay. In practice deliveries are likely to be spread over a number of periods; the spread may contain both a random element and an underlying pattern of delayed proportional deliveries. We may consider the random element by adding a noise input to receipts into stock, whilst a general description of the underlying pattern is contained in the z-transfer function now derived.

If the schedule raised at time t is $d_{(i+1)j}(t)$ then the delivery against this schedule at time $t + \delta$ shall be:

$$r_{ij}(t+\delta) = \phi_\delta \, d_{(i+1)j}(t)$$

where ϕ_δ is the fraction of a schedule to be delivered after δ periods. Thus, unless there is a deliberate policy of over - or under-delivery on the supplier's part:

$$\phi_0 + \phi_1 + \ldots + \phi_\Delta = 1$$

This process is represented by Figure 5.5.a.

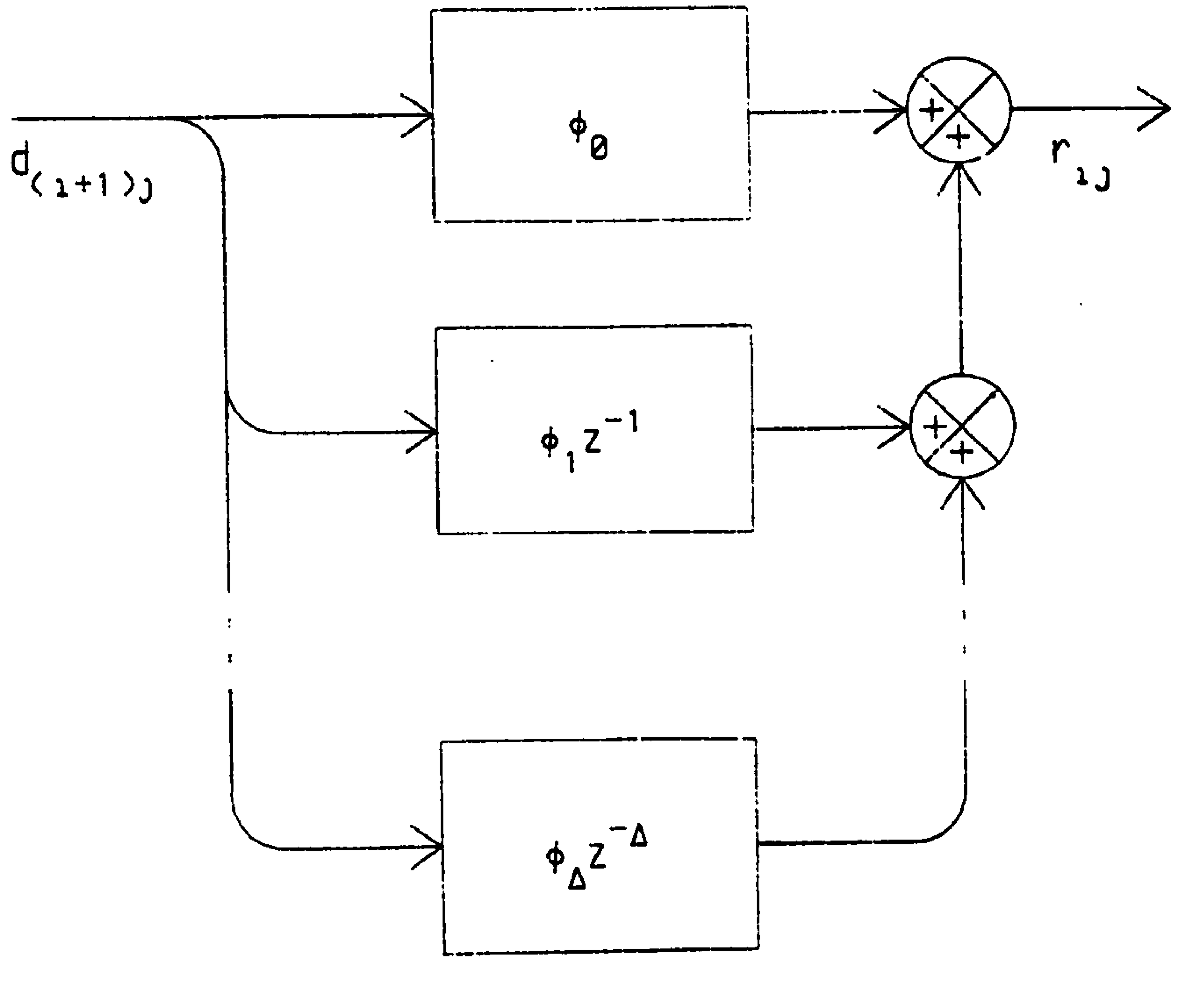

Figure 5.5.a : Part Delivery Pattern

From the figure:-

$$r_{ij} = d_{(i+1)j}\phi_0 + d_{(i+1)j}\phi_1 z^{-1} + \ldots + d_{(i+1)j}\phi_\Delta z^{-\Delta}$$

and so we have the z-transfer function:-

$$L_{ij} = \frac{r_{ij}}{d_{(i+1)j}} = \sum_{\delta=0}^{\Delta} \phi_\delta z^{-\delta} \qquad (5.5.1)$$

and by suitable selection of the values of ϕ_δ we can use this to represent any delivery pattern.

CHAPTER 6

AN EXAMPLE OF THE APPLICATION OF THE GENERAL MODEL

6.1 Preamble

We shall now apply the general model to an example of a multi-product, multi-level system. As this example is included to illustrate the use of the model a simple production environment has been adopted.

6.2 The Production Environment

We consider a system controlling the manufacture of two products (M = 2) comprised of five parts (N = 5) at three levels (K = 3). The product structure is illustrated in Figure 6.2.a which shows the relationship between products A and B and their parts U, V, W, X and Y.

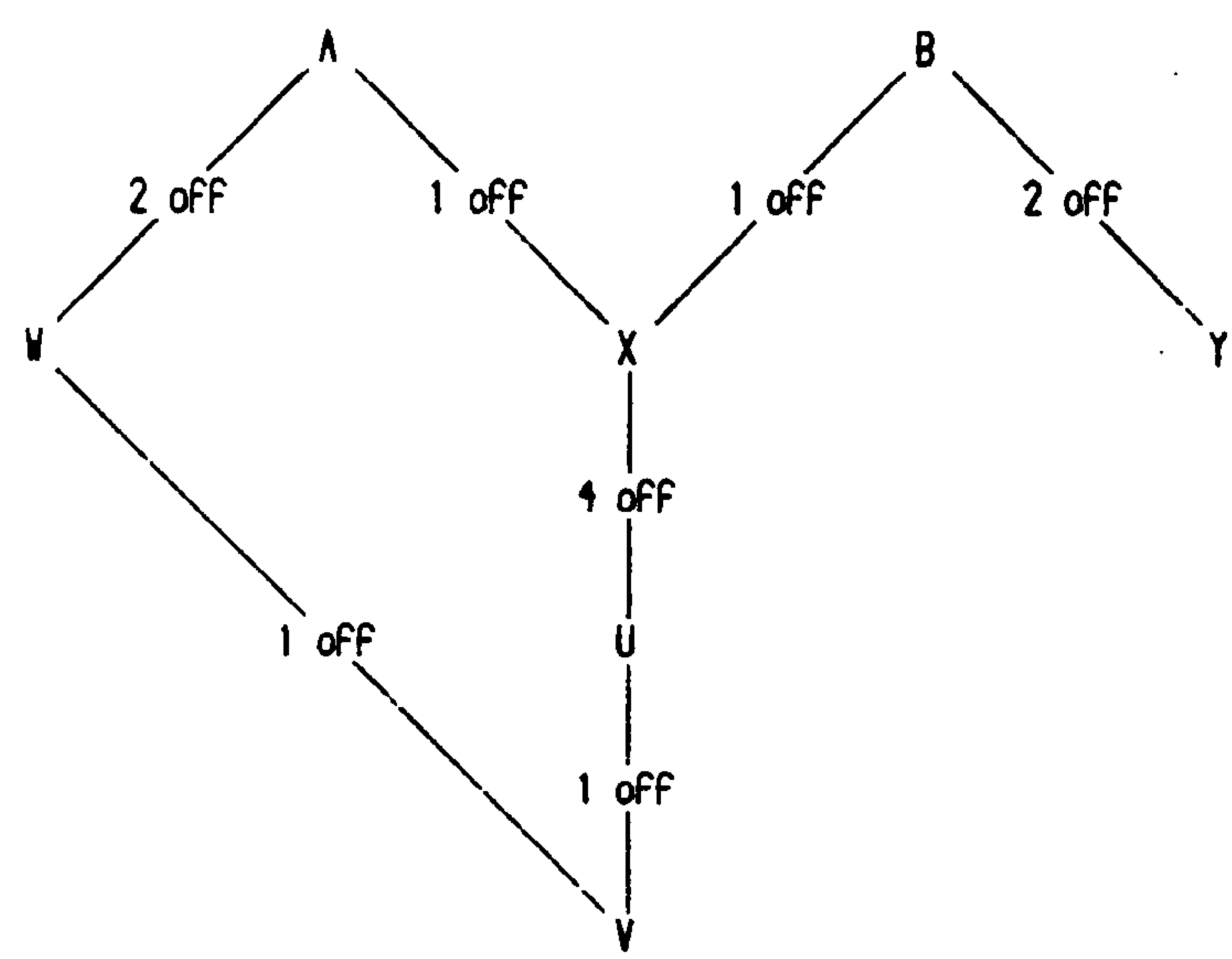

Figure 6.2.a : Product/Part Structure

We shall assume that the manufacturing units are so integrated that it is sensible to use a base information system and so demand forecasts for A and B are used in scheduling deliveries of U and V as well as the level 1 parts. Figure 6.2.b illustrates the decomposition of the system (cf. general decomposition : 4.4.b).

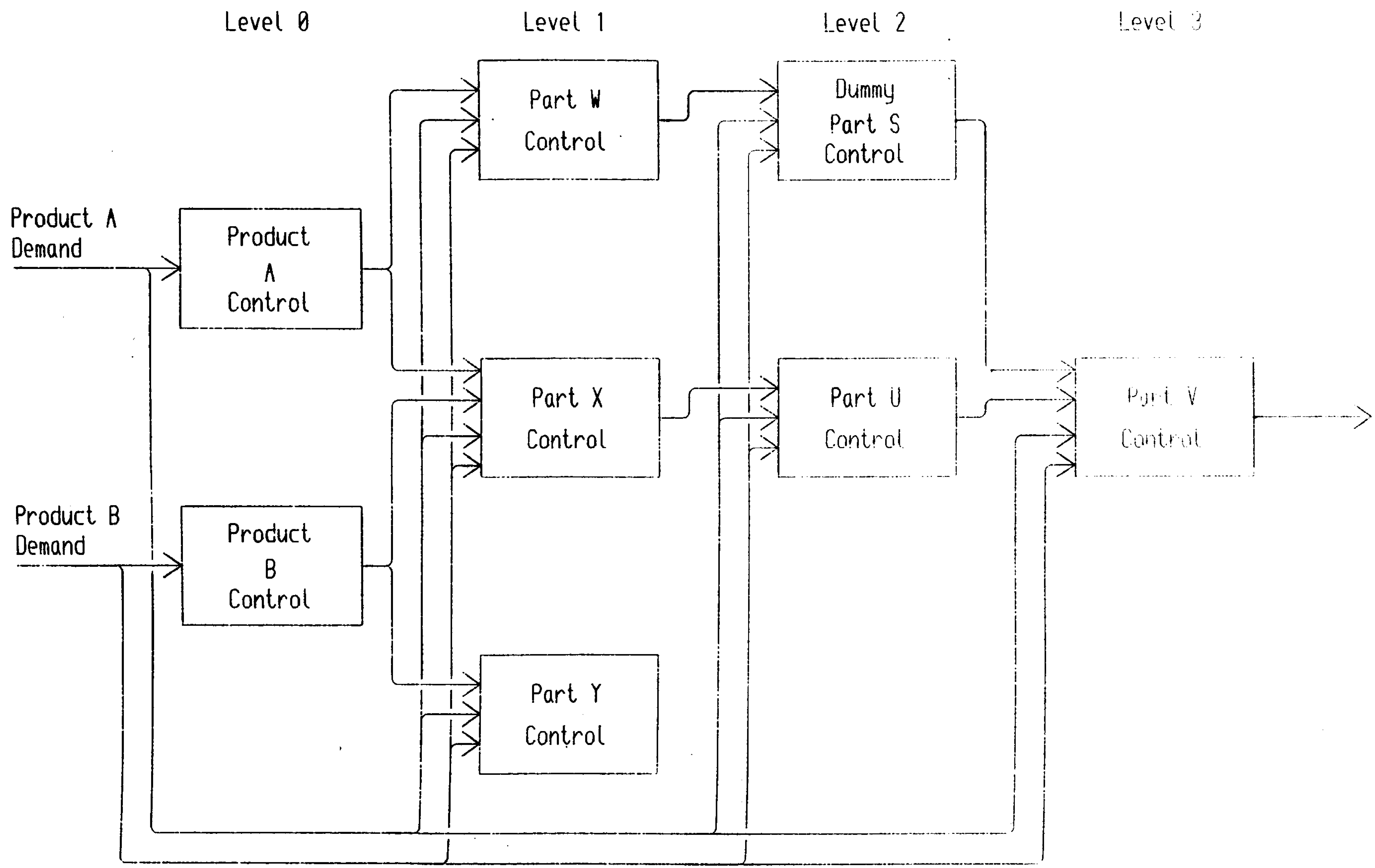

Figure 6.2.b : System Decomposition

Note that a dummy part, S, has been introduced at level 2 to form the link between V at level 3 and W at level 1.

6.3 The Proposed Control System

The following production control system is proposed.

Delivery to customers will be made from stock allowing only an administrative and transport delay. Thus orders placed for A and B in one period will be delivered in the next.

The production and supply lead-time (i.e. schedule lead-time) for each part and product are as shown in Table 6.3.a.

The number of each part used in the manufacture of one unit at the level above is also shown in Table 6.3.a.

Demand for products A and B will be forecast by exponentially smoothed average of past demands using smoothing constants α_A and α_B respectively.

The net requirement schedule for each part will be calculated as the precise quantity needed to meet forecast demand during the part's schedule lead-time:

total forecast gross requirement over lead-time

- current stock balance
- deliveries expected against previous schedules

We are thus attempting no form of production smoothing. We include no safety or buffer stock elements as linearity allows us to interpret a negative stock balance as an inroad into safety or buffer stock.

PART/PRODUCT	USED TO MAKE	NUMBER OFF	SCHEDULE LEAD-TIME
A	-	-	1
B	-	-	1
W	A	2	2 3
X	A, B	1	1 2
Y	B	2	3
U	X	4	4
S	W	1	0
V	U, S	1	2

Figure 6.3.a: Product Structure & Lead-Times

6.4 Modelling The System

Now we may define the z-transfer functions of the individual system elements within each sub-system.

The declared policy of delivery to customers gives us:-

$$I_{OA} = I_{OB} = z^{-1} \tag{6.4.1}$$

Stock integration always has the z-transfer function

$\dfrac{1}{1 - z^{-1}}$ so we have:-

$$S_{OA} = S_{OB} = S_{1W} = S_{1X} = S_{1Y} = S_{2U} = S_{3V} = \frac{1}{1 - z^{-1}} \tag{6.4.2}$$

Since the schedule lead-time on both products matches the customer delivery delay of one period, input customer demand forms an adequate forecast of demand over the lead-time. Expected deliveries against previous schedules will be zero because of immediate delivery. Hence:-

$$F_{OA} = F_{OB} = 1 \tag{6.4.3}$$

and

$$D_{OA} = D_{OB} = 1 \tag{6.4.4}$$

Schedule lead-times are stated in Table 6.3.a so we have:-

$$\begin{aligned} L_{OA} = L_{OB} = L_{1X} &= z^{-1} \\ L_{1W} = L_{3V} &= z^{-2} \\ L_{1Y} &= z^{-3} \\ L_{2U} &= z^{-4} \end{aligned} \tag{6.4.5}$$

We assume that in the case of all parts the schedule lead-time consists of a planning lead-time of one period before production commences and a production lead-time. Thus for each product and part the constituents at the next level down are issued from stock one period after the delivery schedule is raised. The system elements transforming input schedules to issues from stock are therefore multiplication by number off and a delay of one period, giving:-

$$\begin{aligned} I_{1WB} = I_{1YA} = I_{2UW} = I_{2UY} &= 0 \\ I_{1XA} = I_{1XB} = I_{3VU} = I_{3VS} &= z^{-1} \\ I_{1WA} = I_{1YB} &= 2z^{-1} \\ I_{2UX} &= 4z^{-1} \end{aligned} \tag{6.4.6}$$

All forecasts used in part scheduling are of the form of Section 5.3.2. Each forecast is multiplied by the quantity of the part required for the corresponding model. Hence:-

$$F_{1WA} = \frac{4\alpha_A}{1 - (1 - \alpha_A)z^{-1}}$$

$$F_{1WB} = 0$$

$$F_{1XA} = \frac{\alpha_A}{1 - (1 - \alpha_A)z^{-1}}$$

$$F_{1XB} = \frac{\alpha_B}{1 - (1 - \alpha_B)z^{-1}}$$

$$F_{1YA} = 0$$

$$F_{1YB} = \frac{6\alpha_B}{1 - (1 - \alpha_B)z^{-1}} \qquad (6.4.7)$$

$$F_{2UA} = \frac{16\alpha_A}{1 - (1 - \alpha_A)z^{-1}}$$

$$F_{2UB} = \frac{16\alpha_B}{1 - (1 - \alpha_B)z^{-1}}$$

$$F_{3VA} = \frac{12\alpha_A}{1 - (1 - \alpha_A)z^{-1}}$$

$$F_{3VB} = \frac{8\alpha_B}{1 - (1 - \alpha_B)z^{-1}}$$

The net scheduling procedures used are of the form of Section 5.4.1 and so:-

$$D_{1W} = \frac{1}{1 + z^{-1}}$$

$$D_{1X} = 1$$

$$D_{1Y} = \frac{1}{1 + z^{-1} + z^{-2}}$$

$$D_{2U} = \frac{1}{1 + z^{-1} + z^{-2} + z^{-3}} \qquad (6.4.8)$$

$$D_{3V} = \frac{1}{1 + z^{-1}}$$

6.5 System Stability

We showed in the general model that all sub-system z-transfer functions had in their formulae the denominator 1 + LSD (Chapter 4). Let us therefore represent any one of these as the quotient $\frac{G}{1 + LSD}$. Substituting the system elements derived above in the denominator using x to represent schedule lead-time and y to represent number off we have:-

$$\frac{G}{1 + LSD} = \frac{G}{1 + z^{-x}\left(\frac{1}{1 - z^{-1}}\right)\left(\frac{1}{\sum_{i=0}^{x-1} z^{-i}}\right)}$$

$$= \frac{G(1 - z^{-1})(\sum_{i=0}^{x-1} z^{-i})}{(1 - z^{-1})(\sum_{i=0}^{x-1} z^{-i}) + z^{-x}}$$

$$= \frac{G(1 - z^{-1})(\sum_{i=0}^{x-1} z^{-i})}{\sum_{i=0}^{x-1} z^{-i} - \sum_{i=1}^{x} z^{-i} + z^{-x}}$$

$$= G(1 - z^{-1})(\sum_{i=0}^{x-1} z^{-i}) \qquad (6.5.1)$$

Above (6.4) we expressed all the sub-system z-transfer functions as quotients of polynomials in z^{-1} and amongst these only three forms of non-trivial denominators occurred. These were in stock integration $\frac{1}{1 - z^{-1}}$, net scheduling $\frac{1}{\sum_{i=0}^{x-1} z^{-i}}$ and forecasting $\frac{yx\alpha}{1 - (1 - \alpha)z^{-1}}$. Where either of the first two of these occurs in G its denominator is cancelled by one of the factors in 6.5.1. Thus the only terms which remain in sub-system z-transfer function denominators are the two forecasting denominators. Using the arguments in section 4.10 it is therefore a necessary and sufficient condition for system stability that the equation

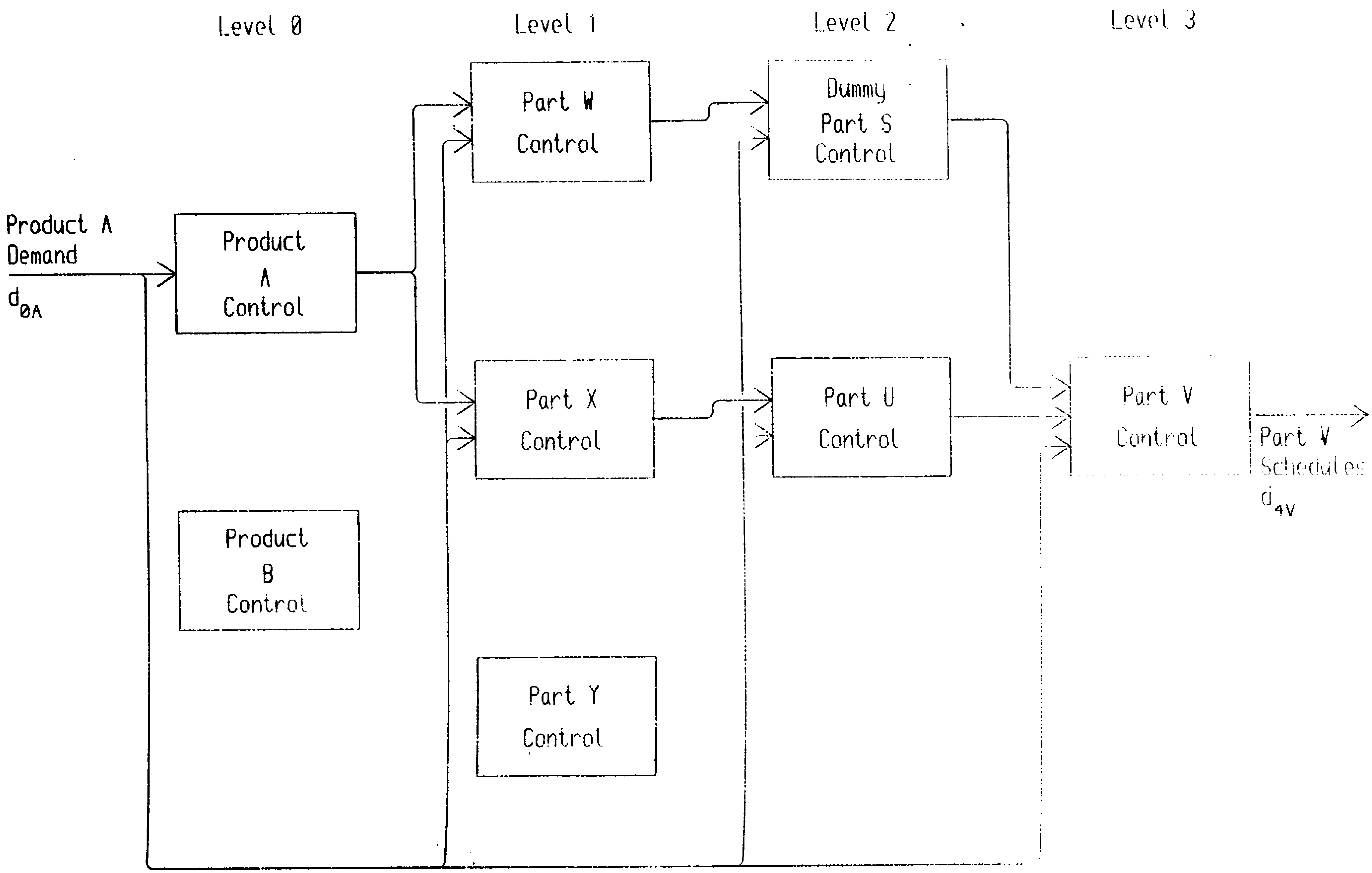

Figure 6.6.a : Causal Routes From d_{0A} To d_{4V}

$$(1 - (1 - \alpha_A)z^{-1})(1 - (1 - \alpha_B)z^{-1}) = 0$$

has roots within the unit circle. The roots are $z = 1 - \alpha_A$ and $z = 1 - \alpha_B$. Hence for stability we must have:-

$$-1 < 1 - \alpha_A < 1 \quad \text{and} \quad -1 < 1 - \alpha_B < 1$$

i.e. $0 < \alpha_A < 2$ and $0 < \alpha_B < 2.$ (6.5.2)

6.6 System z-Transfer Functions

In order to examine either transient responses or responses to noise it is necessary to use the appropriate system z-transfer functions. We show here the derivation of one of these, namely $T(d_{0A}, d_{4V})$. We shall use equation 4.10.1 to combine the z-transfer functions of all causal routes from demand for product A to delivery schedules for part V.

We can summarise the causal routes as in Figure 6.6.a. In Figure 6.6.b we show separate sub-system z-transfer functions for each input of a sub-system to its output. Because of system linearity the output of each sub-system is the sum of the inputs multiplied by the appropriate z-transfer functions. By inspection of the diagram we can see that there are seven distinct causal routes through the system. Each of these can be represented by a product of the sub-system z-transfer functions it includes. The system z-transfer function from d_{0A} to d_{4V} is the sum of these products.

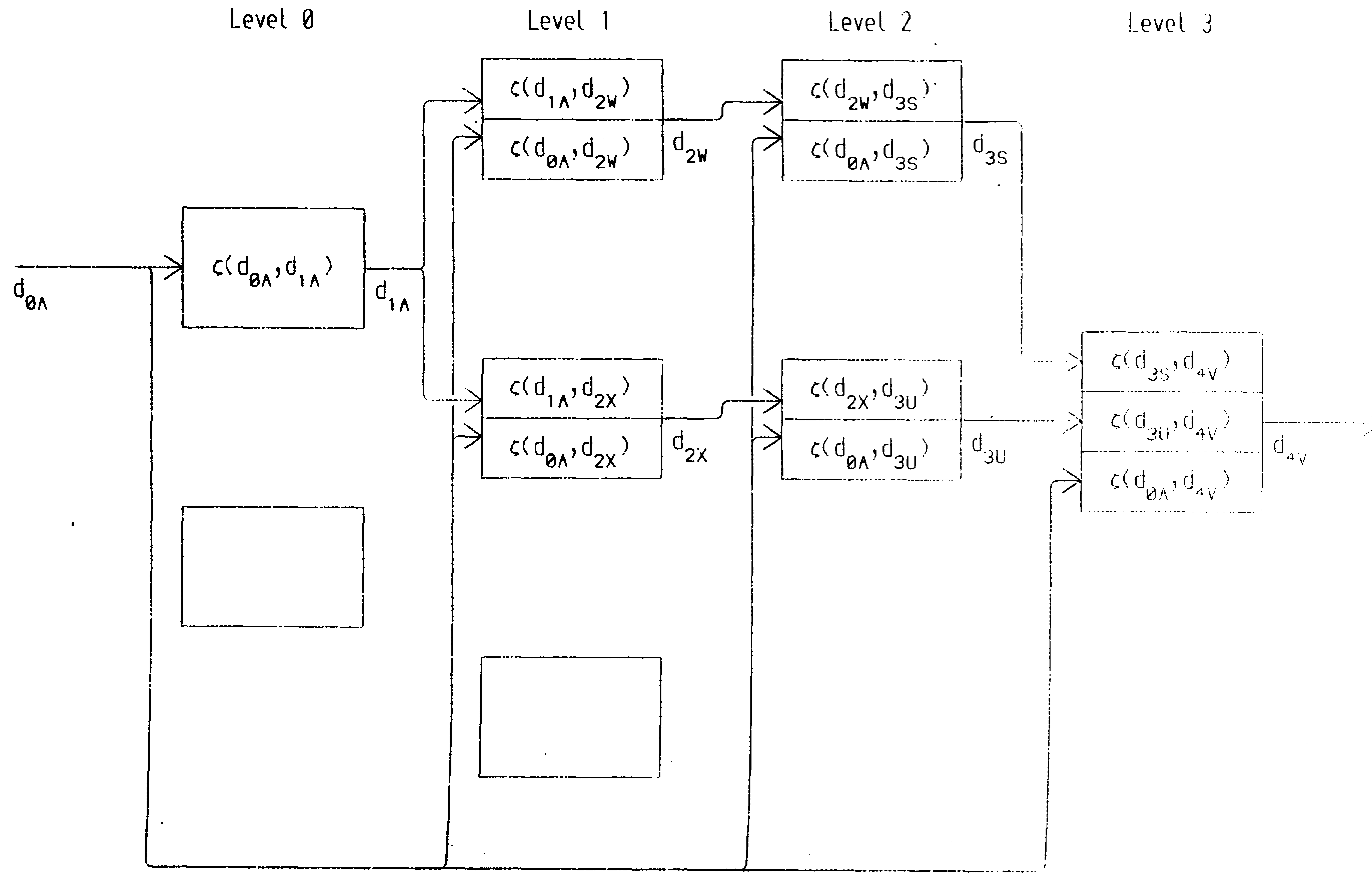

Figure 6.6.b : Sub-System z-Transfer Functions

$$T(d_{0A},d_{4V}) = \zeta(d_{0A},d_{1A})\zeta(d_{1A},d_{2W})\zeta(d_{2W},d_{3S})\zeta(d_{3S},d_{4V})$$

$$+ \zeta(d_{0A},d_{2W})\zeta(d_{2W},d_{3S})\zeta(d_{3S},d_{4V})$$

$$+ \zeta(d_{0A},d_{3S})\zeta(d_{3S},d_{4V})$$

$$+ \zeta(d_{0A},d_{1A})\zeta(d_{1A},d_{2X})\zeta(d_{2X},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+ \zeta(d_{0A},d_{2X})\zeta(d_{2X},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+ \zeta(d_{0A},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+ \zeta(d_{0A},d_{4V}) \tag{6.6.1}$$

We note that, although in this case it is a simple matter to identify the full range of causal routes by inspection, in a more complex example this becomes less practicable and more subject to error. For this reason we demonstrate a more systematic approach to the derivation of 6.6.1:

$$d_{4V} = d_{3S}\zeta(d_{3S},d_{4V}) + d_{3U}\zeta(d_{3U},d_{4V}) + d_{0A}\zeta(d_{0A},d_{4V})$$

$$= \{d_{2W}\zeta(d_{2W},d_{3S}) + d_{0A}\zeta(d_{0A},d_{3S})\}\zeta(d_{3S},d_{4V})$$

$$+ \{d_{2X}\zeta(d_{2X},d_{3U}) + d_{0A}\zeta(d_{0A},d_{3U})\}\zeta(d_{3U},d_{4V})$$

$$+ d_{0A}\zeta(d_{0A},d_{4V})$$

$$= \{d_{1A}\zeta(d_{1A},d_{2W}) + d_{0A}\zeta(d_{0A},d_{2W})\}\zeta(d_{2W},d_{3S})\zeta(d_{3S},d_{4V})$$

$$+ \{d_{1A}\zeta(d_{1A},d_{2X}) + d_{0A}\zeta(d_{0A},d_{2X})\}\zeta(d_{2X},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+ d_{0A}\{\zeta(d_{0A},d_{3S})\zeta(d_{3S},d_{4V}) + \zeta(d_{0A},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+ \zeta(d_{0A},d_{4V})\}$$

$$= d_{OA}\zeta(d_{OA},d_{1A})\zeta(d_{1A},d_{2W})\zeta(d_{2W},d_{3S})\zeta(d_{3S},d_{4V})$$

$$+ d_{OA}\zeta(d_{OA},d_{1A})\zeta(d_{1A},d_{2X})\zeta(d_{2X},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+ d_{OA}\{\zeta(d_{OA},d_{2W})\zeta(d_{2W},d_{3S})\zeta(d_{3S},d_{4V})$$

$$+\zeta(d_{OA},d_{2X})\zeta(d_{2X},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+\zeta(d_{OA},d_{3S})\zeta(d_{3S},d_{4V}) \quad +\zeta(d_{OA},d_{3U})\zeta(d_{3U},d_{4V})$$

$$+\zeta(d_{OA},d_{4V})\}$$

$$\therefore\ T(d_{OA},d_{4V}) = \frac{d_{4V}}{d_{OA}}$$

A B C D

$$= \zeta(d_{OA},d_{1A})\zeta(d_{1A},d_{2W})\zeta(d_{2W},d_{3S})\zeta(d_{3S},d_{4V})$$

A G H I

$$+ \zeta(d_{OA},d_{1A})\zeta(d_{1A},d_{2X})\zeta(d_{2X},d_{3U})\zeta(d_{3U},d_{4V})$$

E C D

$$+ \zeta(d_{OA},d_{2W})\zeta(d_{2W},d_{3S})\zeta(d_{3S},d_{4V})$$

J H I

$$+ \zeta(d_{OA},d_{2X})\zeta(d_{2X},d_{3U})\zeta(d_{3U},d_{4V})$$

Dummy → O D

$$+ \zeta(d_{OA},d_{3S})\zeta(d_{3S},d_{4V})$$

K I

$$+ \zeta(d_{OA},d_{3U})\zeta(d_{3U},d_{4V})$$

F

$$+ \zeta(d_{OA},d_{4V})$$

This expression is precisely 6.6.1.

The general sub-system blocks analysed in sections 4.6, 4.7 and 4.9 allow us to give values to each of the z-transfer functions in this expression using the system elements derived above (6.4).

It is then a straightforward matter of substitution to complete the evaluation of $T(d_{0A},d_{4V})$:

$$T(d_{0A},d_{4V}) = \frac{\left\{ \begin{array}{c} 12\alpha_A + 8\alpha_A z^{-1} + (2 - 16\alpha_A)z^{-2} + \\ (2 - 2\alpha_A)z^{-3} - (4 - \alpha_A)z^{-4} \end{array} \right\}}{1 - (1 - \alpha_A)z^{-1}} \qquad (6.6.2)$$

This is now in standard form as a quotient of polynomials. It can easily be rewritten as an infinite power series, convergent provided that the stability conditions (6.5.2) are met. From this expansion we may use the methods illustrated in the example of Chapter 3 to examine in detail both transient response and response to random inputs.

The task of evaluating 6.6.1 to achieve 6.6.2, although straightforward, is tedious as is the work of deriving transient responses using 6.6.2. For this reason the computer program of Appendix I was developed to carry out the arithmetic involved. The use of the program speeds analysis and restricts the scope for error.

6.7 Transient and Noise Responses

We now use the program of Appendix I to examine graphically the transient responses and response to noise of this system. The graphs presented all show part V responses to product A demand; both stock and schedule responses are included. Each graph shows responses for $\alpha = 0.1$ (continuous line), $\alpha = 0.3$ (broken line) and $\alpha = 0.5$ (dotted line).

Graphs 6.7.a show responses for the system as described earlier in this Chapter. In 6.7.b we have increased, by one period, the lead-times for each of the parts W, X and Y and made only those system changes directly linked to their lead-times.

Graphs 6.7.c arise from a cascaded system applied to the production environment described in 6.2 and graphs 6.7.d are then derived, as for 6.7.b by extending the lead-times of parts W, X and Y.

The noise distributions are all responses to the same distribution of random demand for product A:-

Probability of $-1 <$ noise impulse <0 is 0.5

Probability of $0 <$ noise impulse <1 is 0.5.

This is a simple distribution but it nevertheless allows us to make a first qualitative evaluation of the alternatives. A broader, more detailed, distribution will produce similar responses and may be used at a later stage of analysis, perhaps to determine precise safety stock sizes, or in choosing between alternative systems where other criteria are inconclusive.

A number of points are apparent, simply from close examination of the graphs.

6.7.1 For both cascaded and base information systems, and for all values of the smoothing constant α, extended lead-times give "poorer" responses in all respects. That is, the initial response to perturbation is of greater magnitude (i.e. step is overshot, sine is amplified, etc.), decay to steady state is slower and the noise response distribution is broader.

6.7.2 For given lead-times the cascaded system gives poorer response than the base information system. Again the initial response to perturbation is more exaggerated in the cascaded system and decay is slower.

6.7.3 In all cases, high values of the forecast smoothing constant α give a more rapid decay to steady state and a broader noise response distribution. In almost all cases the high values of α also give greater initial response. The exceptions here are stock response to a step in demand with the shorter lead-times. If large steps in demand are probable more detailed analysis of this response would be required.

We may therefore draw the following conclusions.

6.7.4 There is no operating advantage in applying longer lead-times for parts W, X and Y regardless of the control system adopted. The disadvantage is quantified and can be compared, say, with the capital costs of plant needed to achieve the shorter lead-times.

6.7.5 There is no advantage in operating a cascaded system. If organisational constraints preclude a base information system, the cost of these constraints is quantified.

6.7.6 The choice of forecast smoothing constant is dependent upon factors such as stock holding costs and market volatility and seasonality. Where the system must respond rapidly to changes in demand pattern and return to a steady state quickly then a high value of α is indicated. The price of this fast response is seen to be the high safety stock needed to cover the initial wild fluctuations in stock when perturbation occurs and to absorb the broader distribution of response to random elements in demand. There may well be costs attached to the rapid changes in production levels as represented by schedule responses. A small value of α will make smaller inroads into safety stock but will take longer to achieve a steady state after perturbation. This may lead to backlogging of orders. Low values of α require less safety stock since their noise distributions are narrower. Where seasonal demand patterns occur, the lower values of α are much more conservative in response. Low values of α provide some smoothing of schedules.

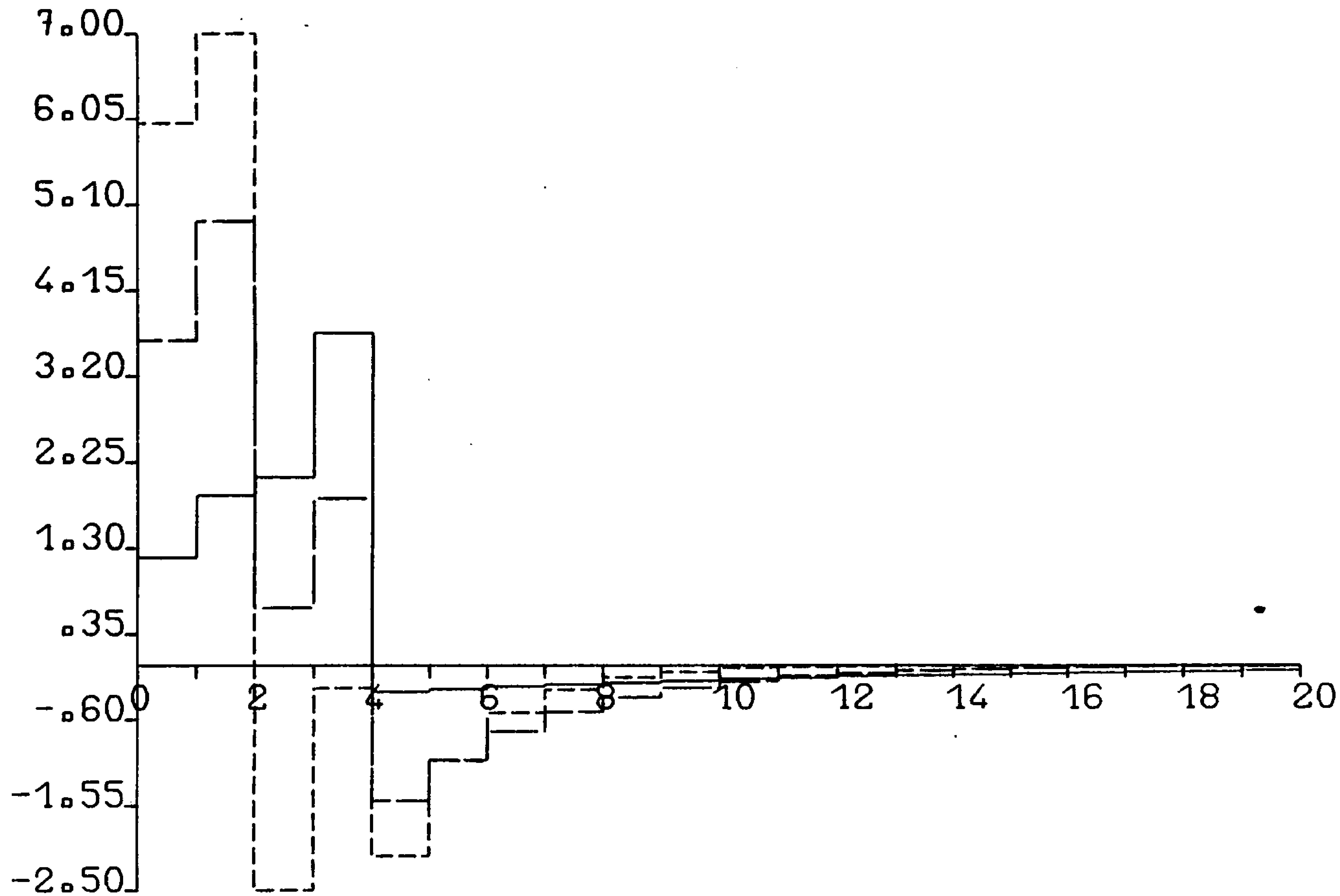

Figure 6.7.a(i) : Base Information; Short Leads; Schedule Impulse Response
— α = 0.1 —— α = 0.3 --- α = 0.5

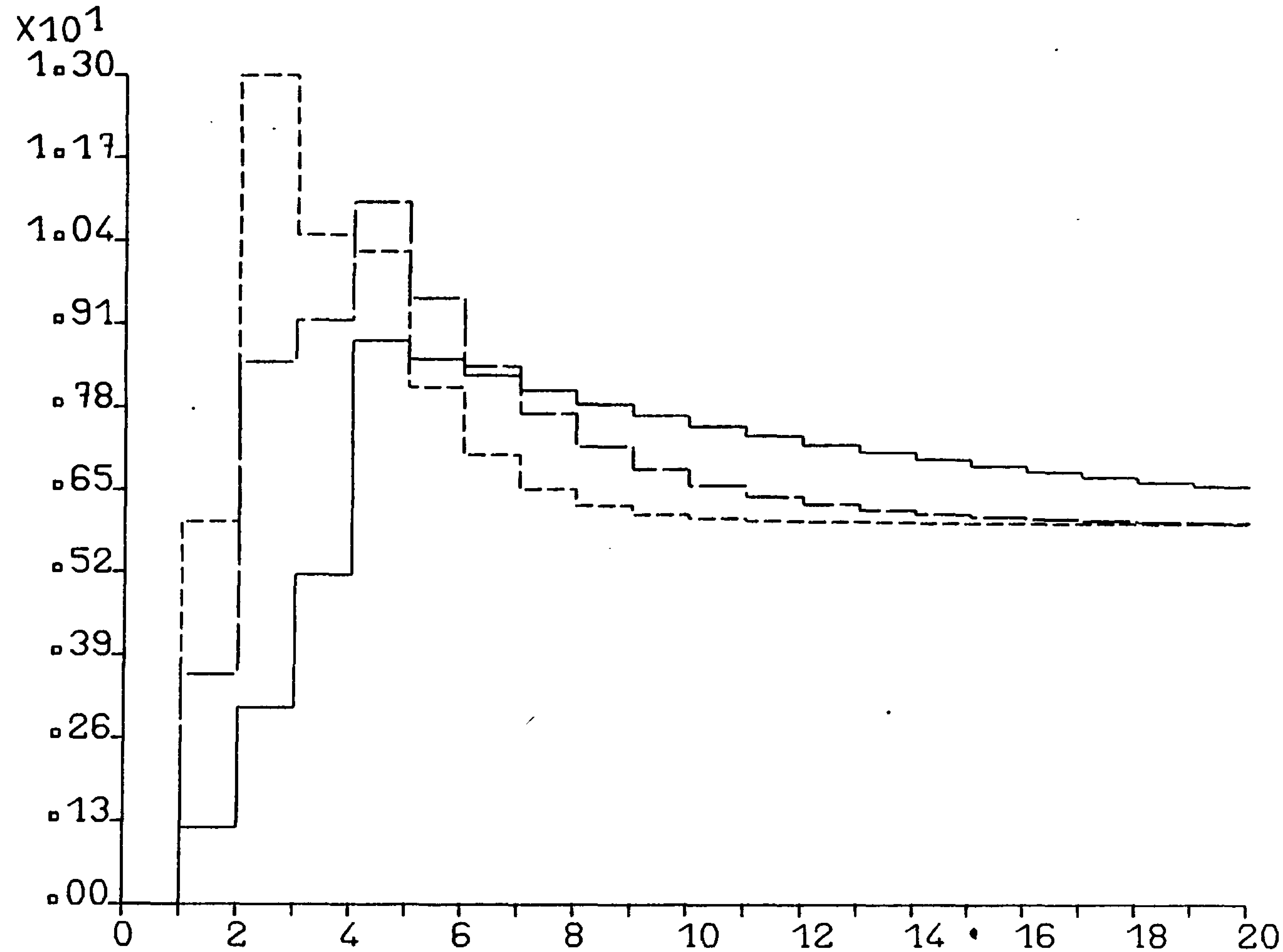

Figure 6.7.a(ii) : Base Information; Short Leads; Schedule Step Response

— α = 0.1   — — α = 0.3   - - - α = 0.5

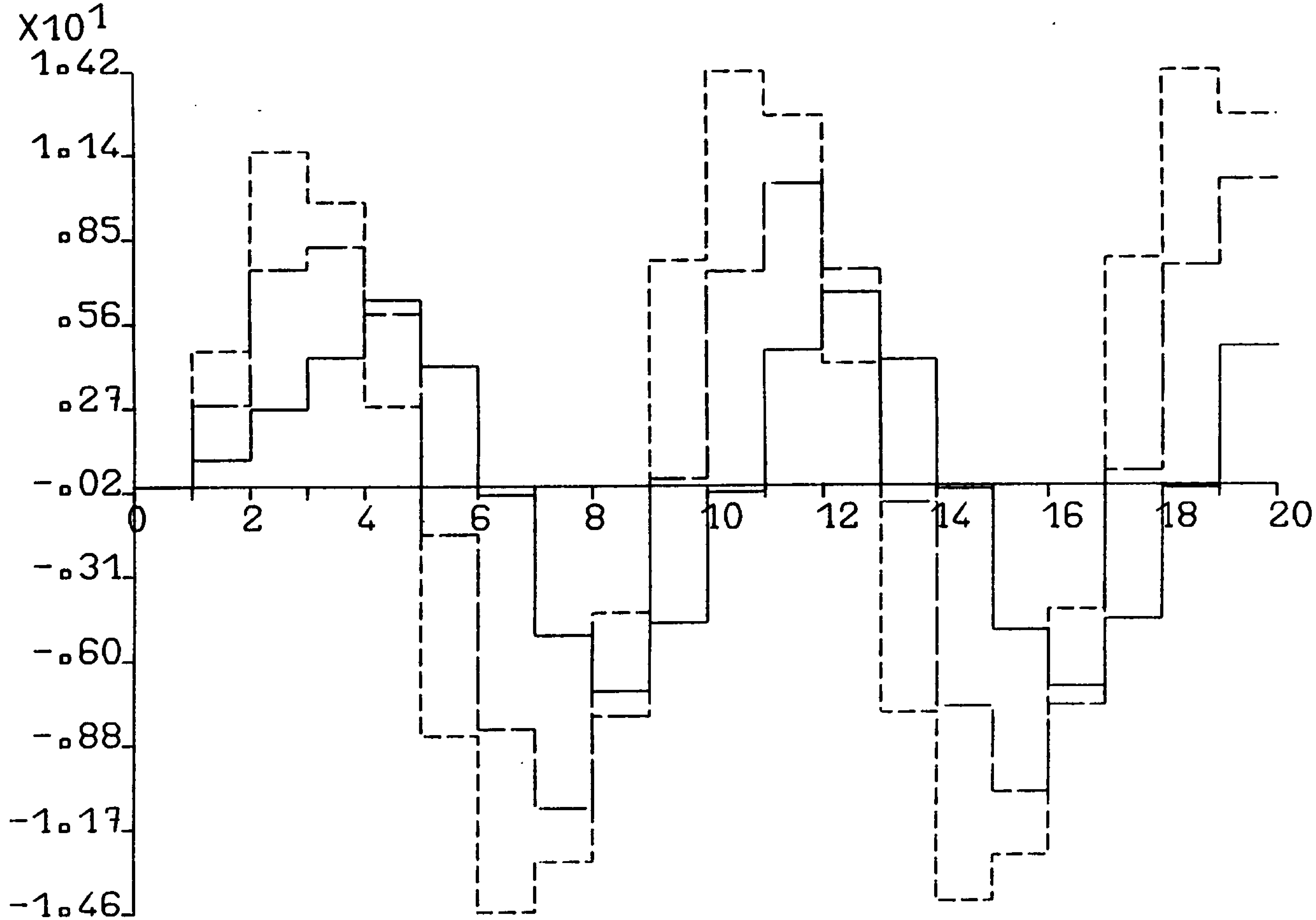

Figure 6.7.a(iii) : Base Information; Short Leads; Schedule Sine Response

— $\alpha = 0.1$ —— $\alpha = 0.3$ - - - $\alpha = 0.5$

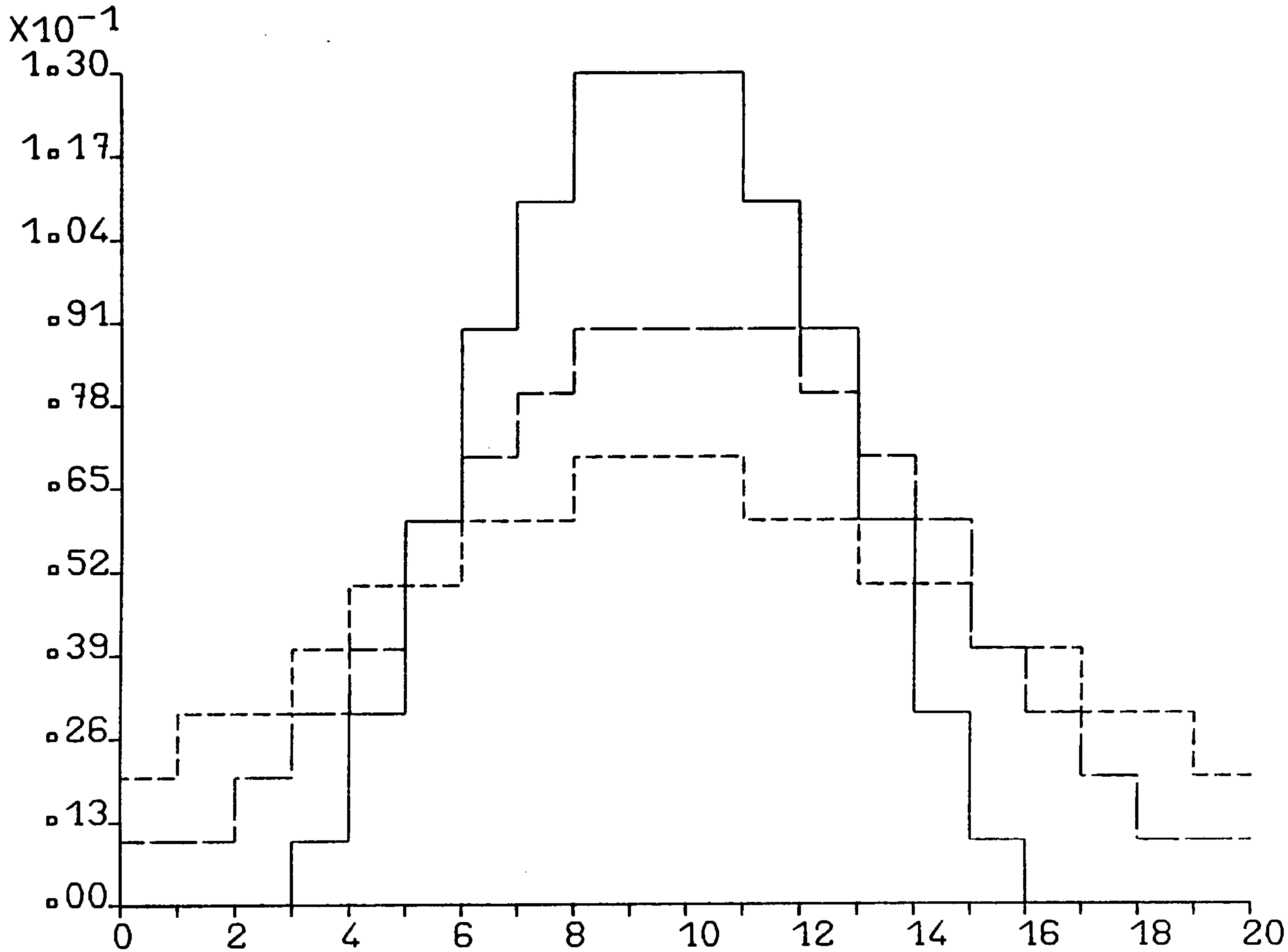

Figure 6.7.a(iv) : Base Information; Short Leads; Schedule Noise Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

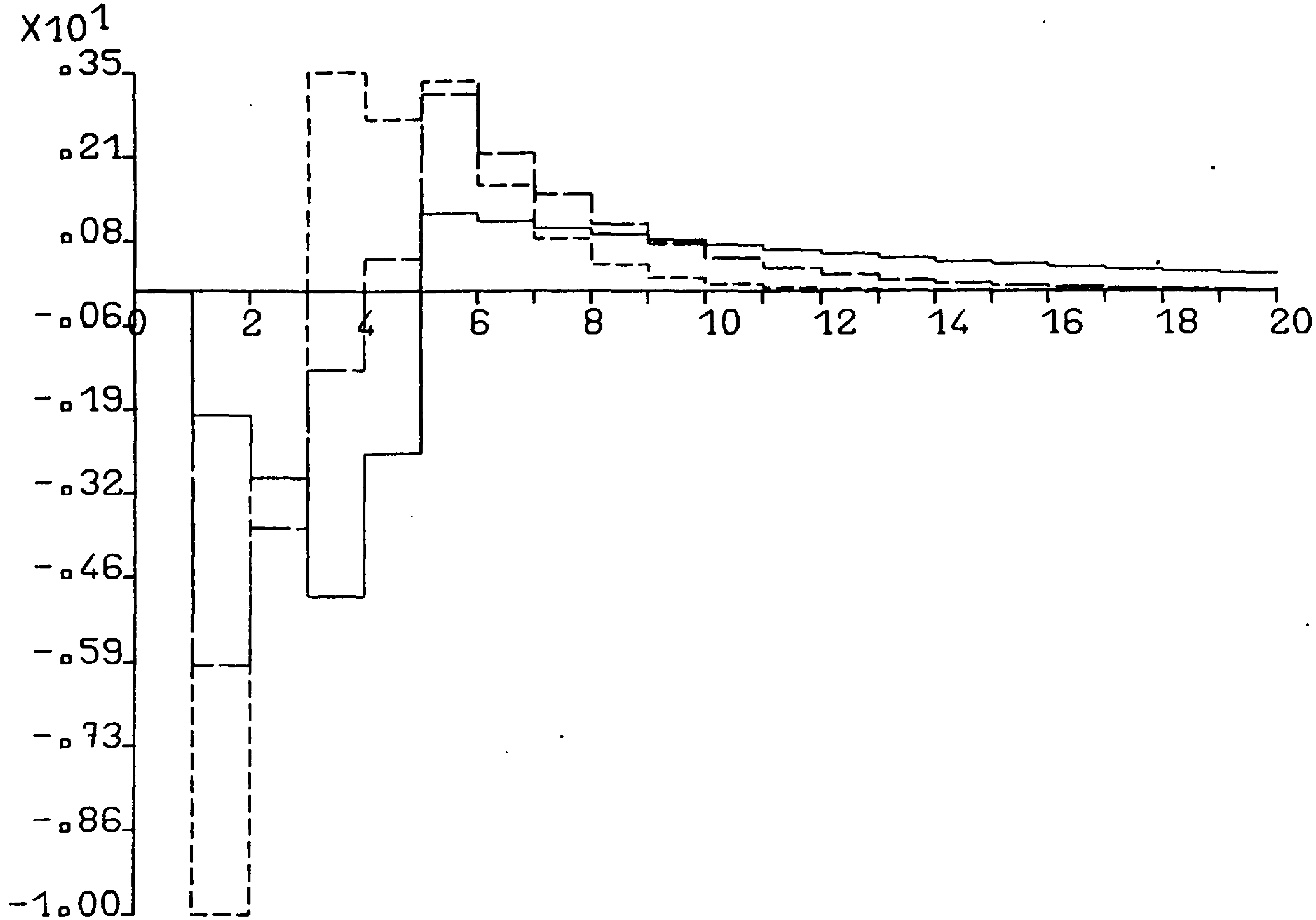

Figure 6.7.a(v) : Base Information; Short Leads; Stock Impulse Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

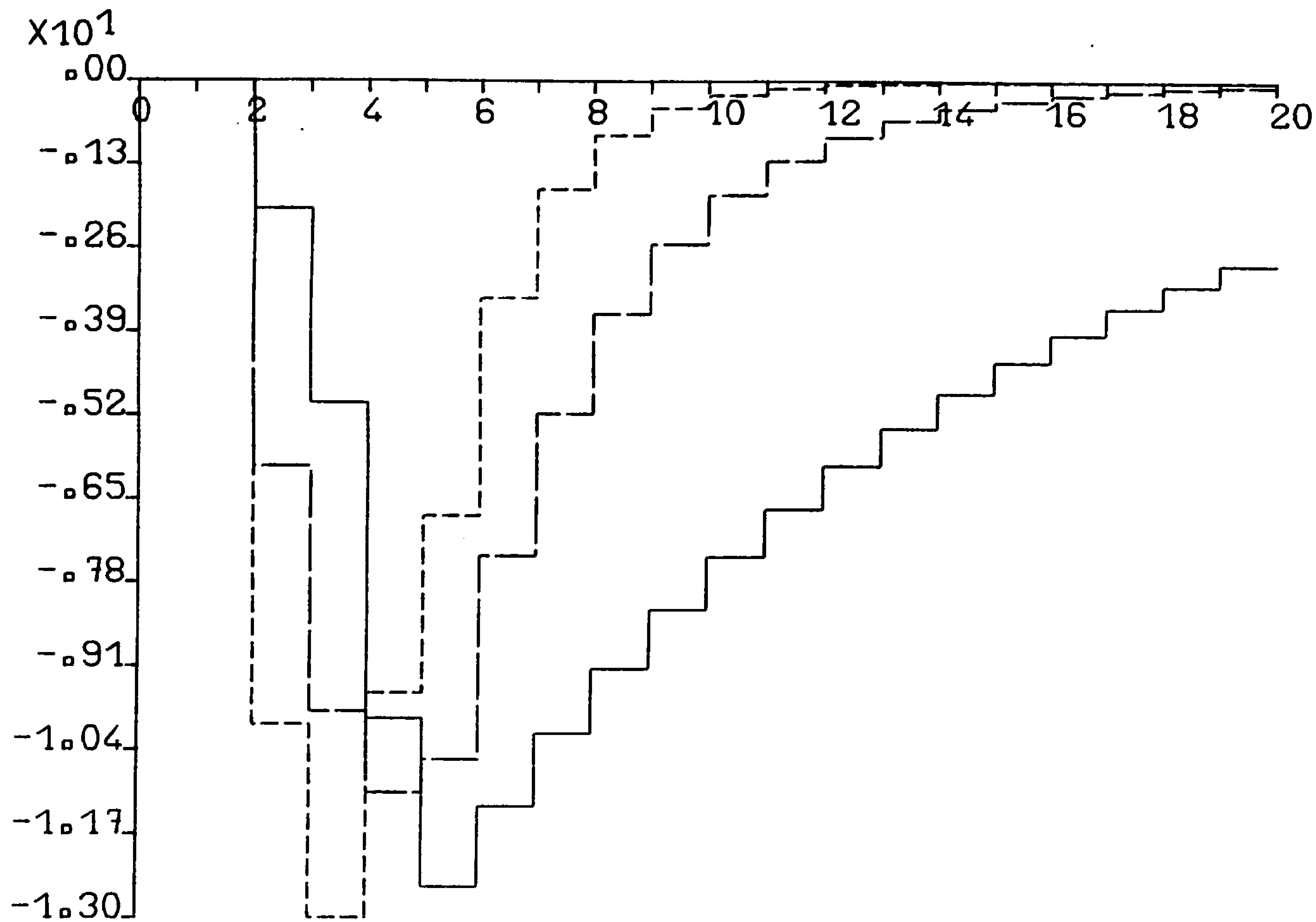

Figure 6.7.a(vi) : Base Information; Short Leads; Stock Step Response

—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

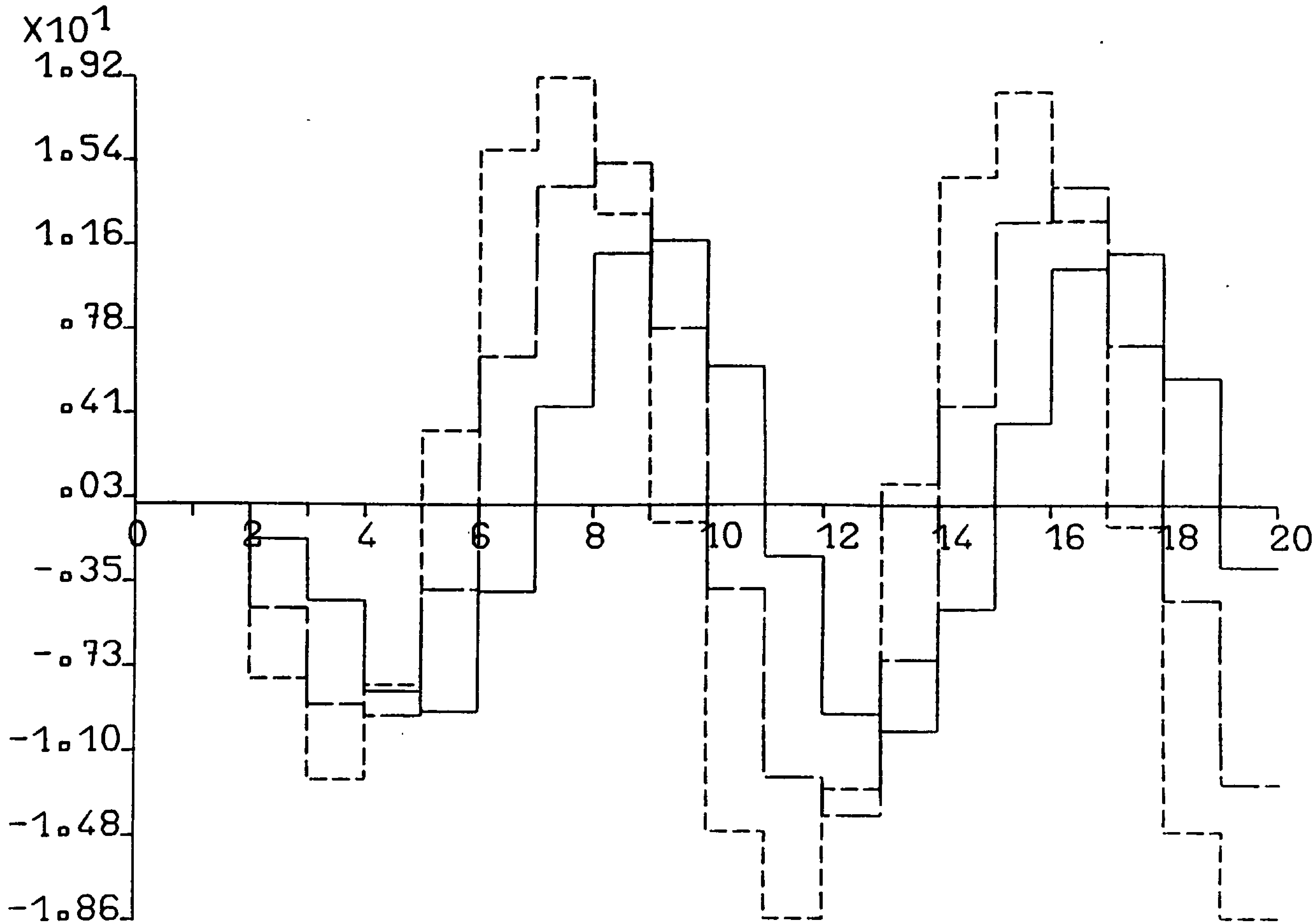

Figure 6.7.a(vii) : Base Information; Short Leads; Stock Sine Response
— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

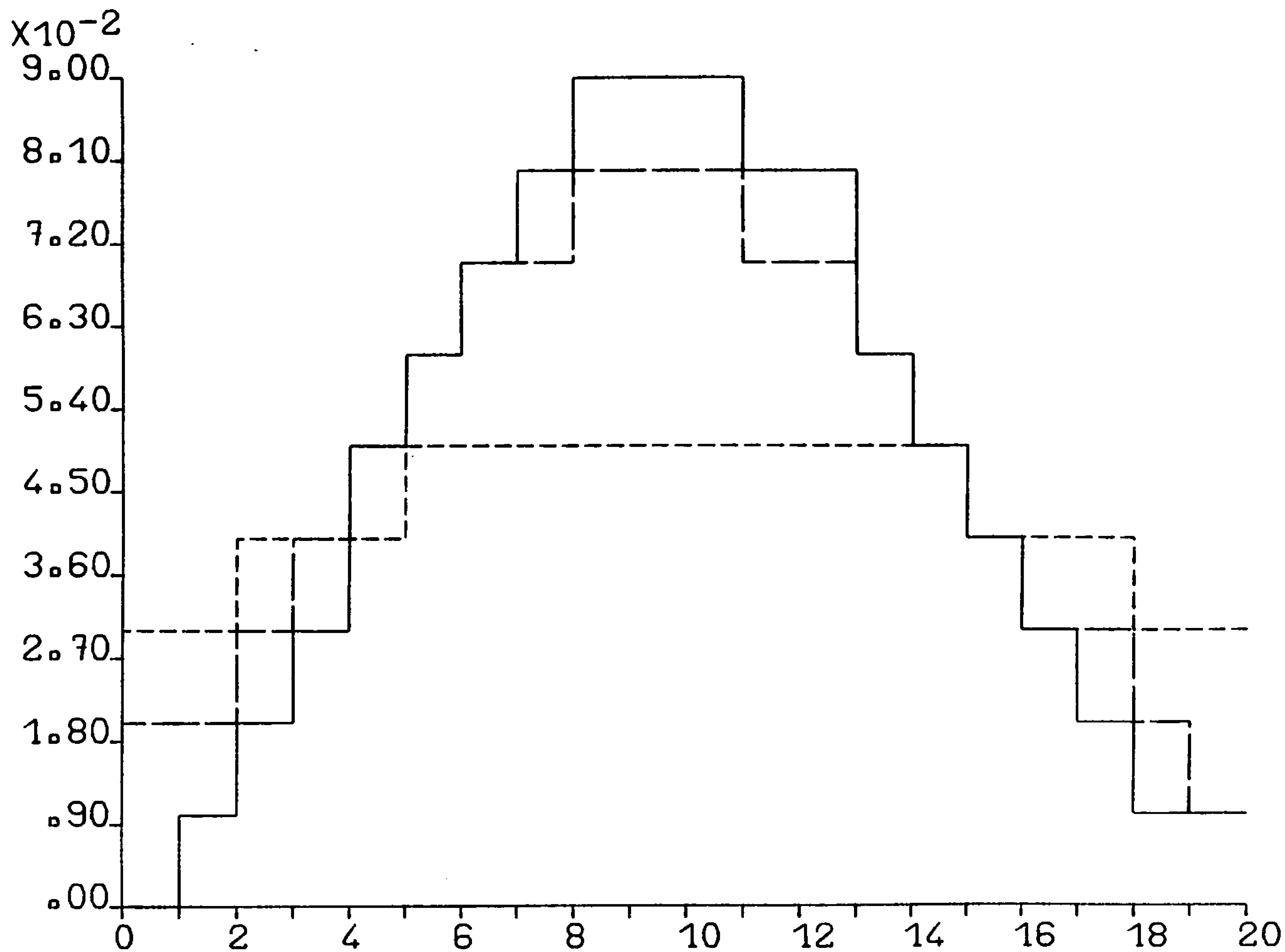

Figure 6.7.a(viii) : Base Information; Short Leads; Stock Noise Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

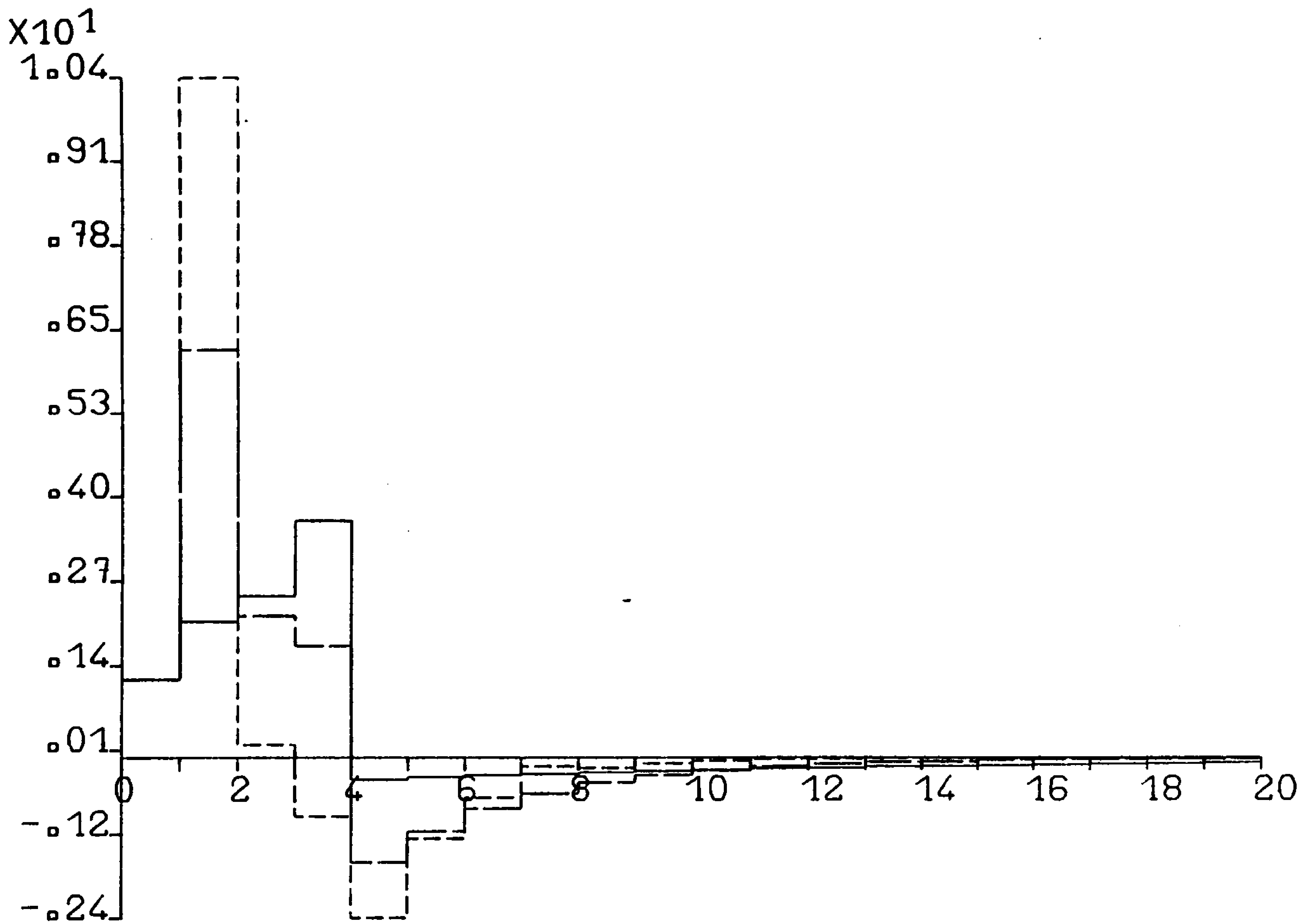

Figure 6.7.b(i) : Base Information; Long Leads; Schedule Impulse Response
—— α = 0.1 — — α = 0.3 - - - α = 0.5

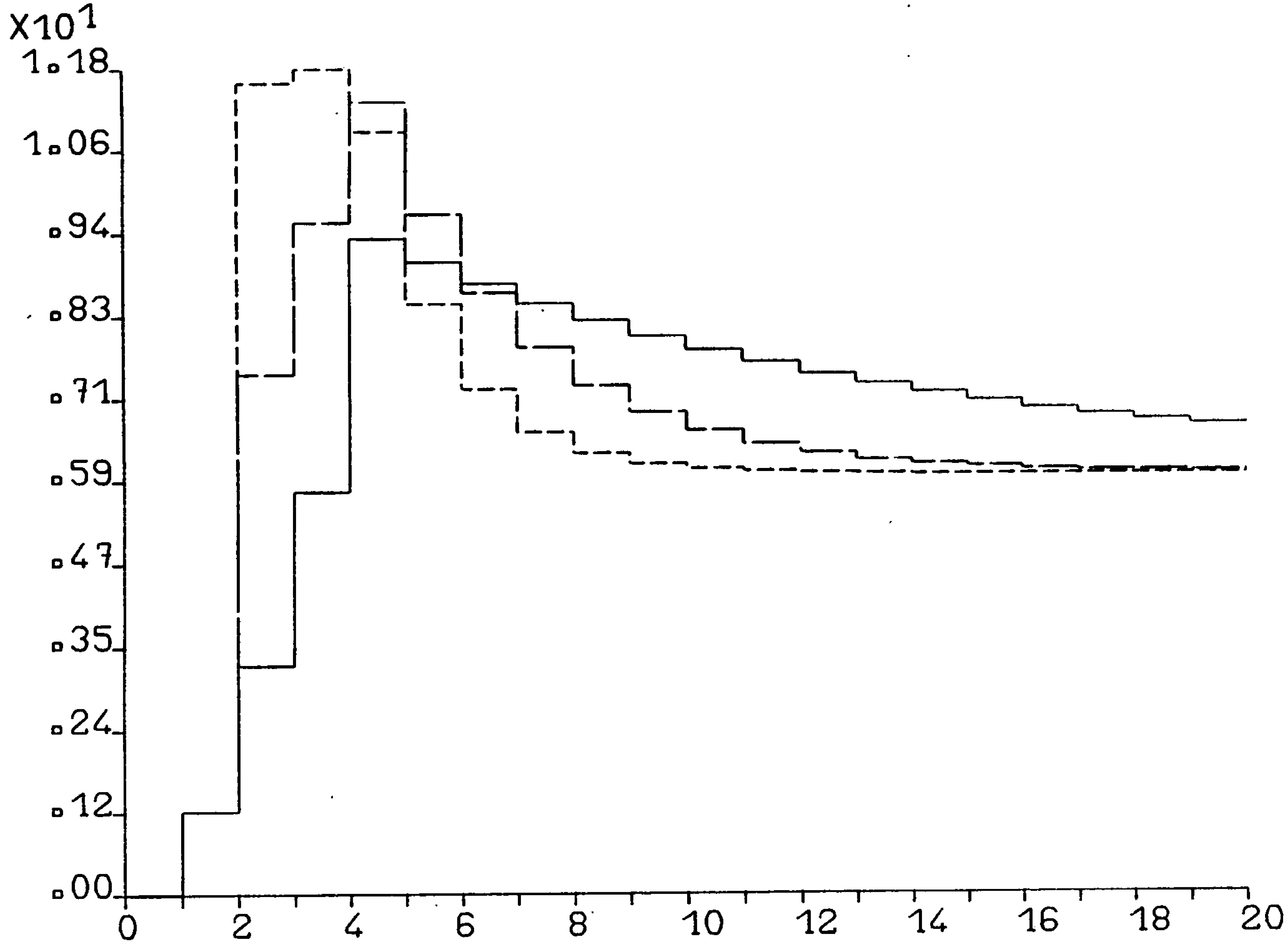

Figure 6.7.b(ii) : Base Information; Long Leads; Schedule Step Response
—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

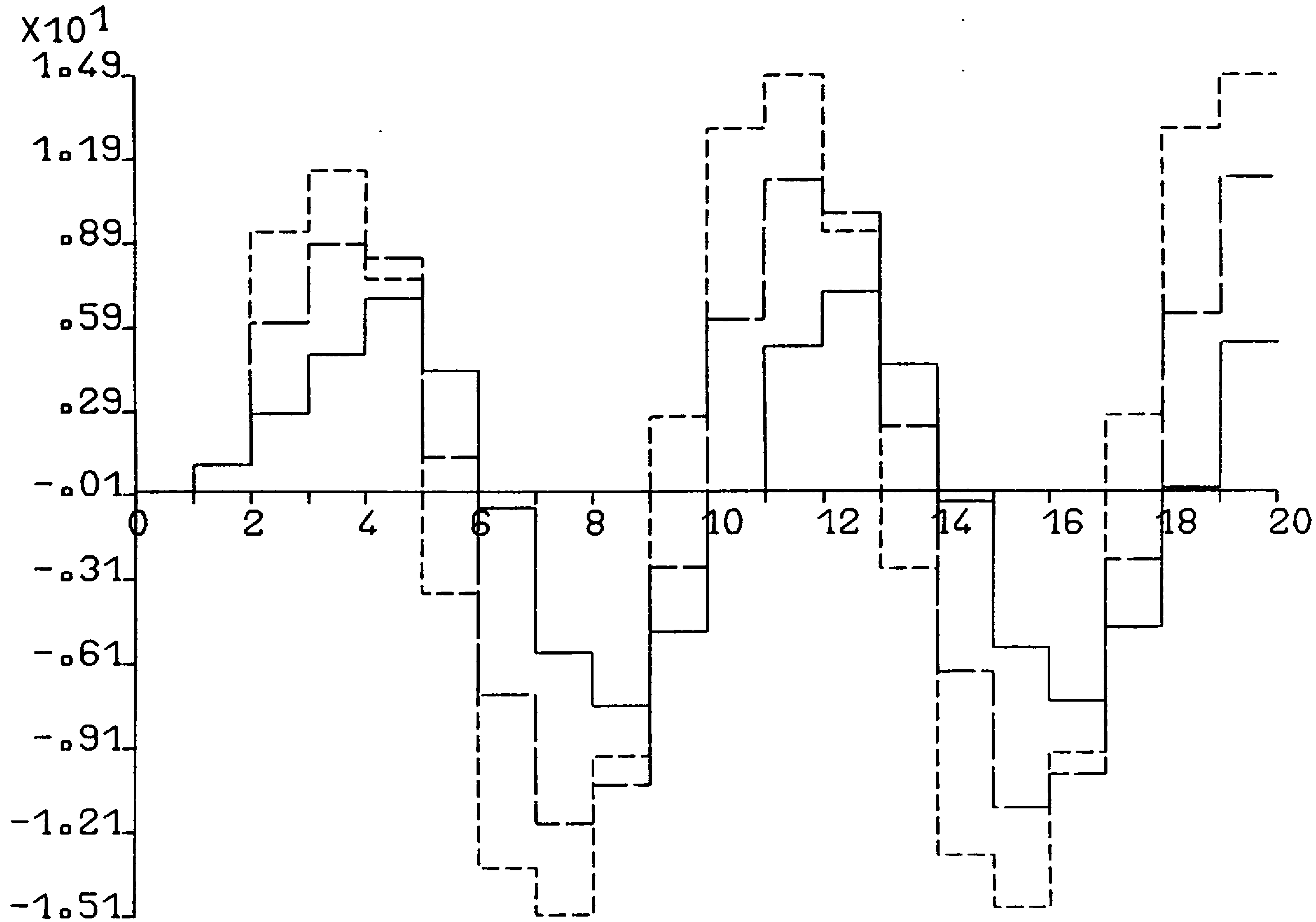

Figure 6.7.b(iii) : Base Information; Long Leads; Schedule Sine Response
— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

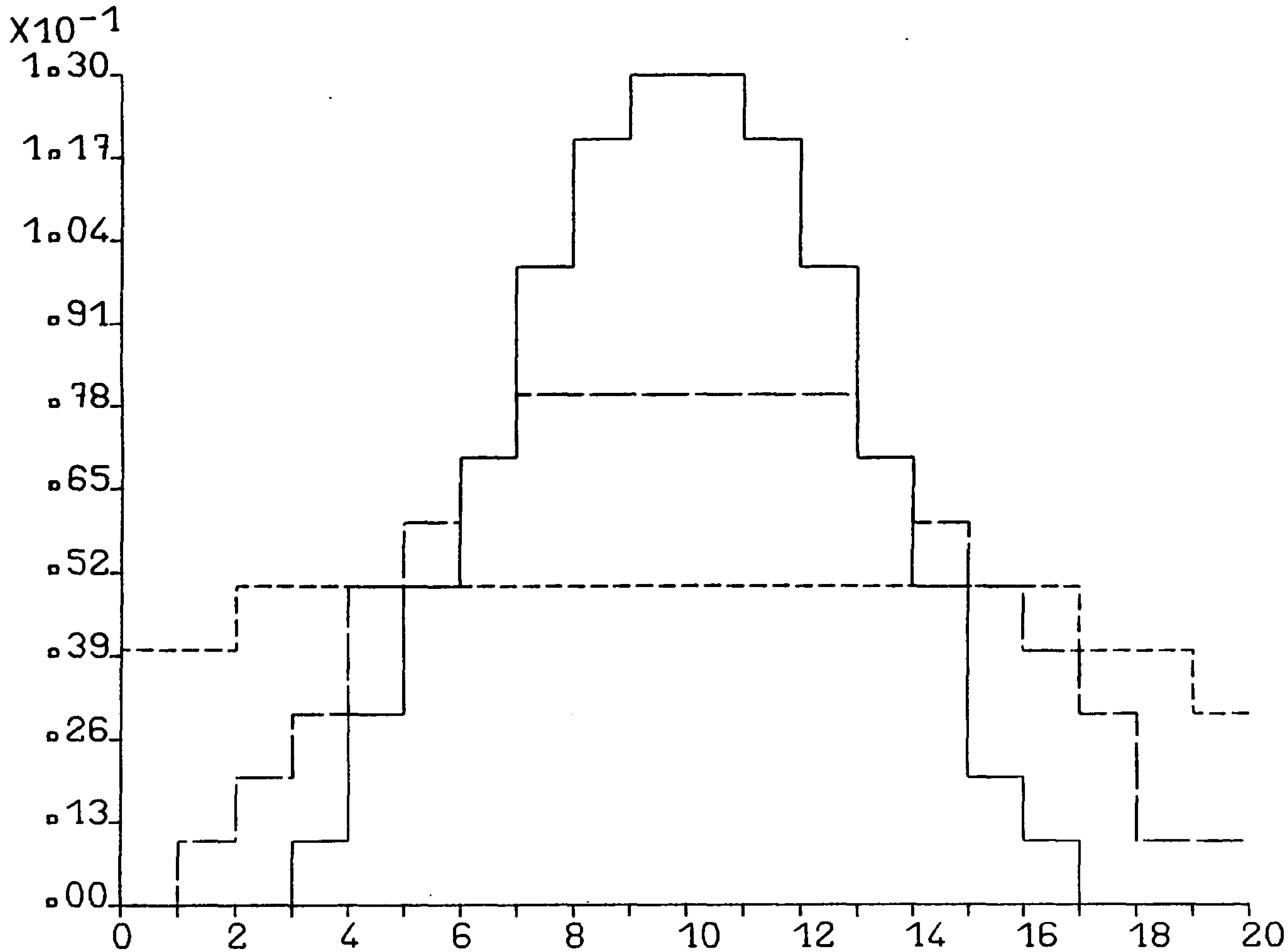

Figure 6.7.b(iv) : Base Information; Long Leads; Schedule Noise Response

—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

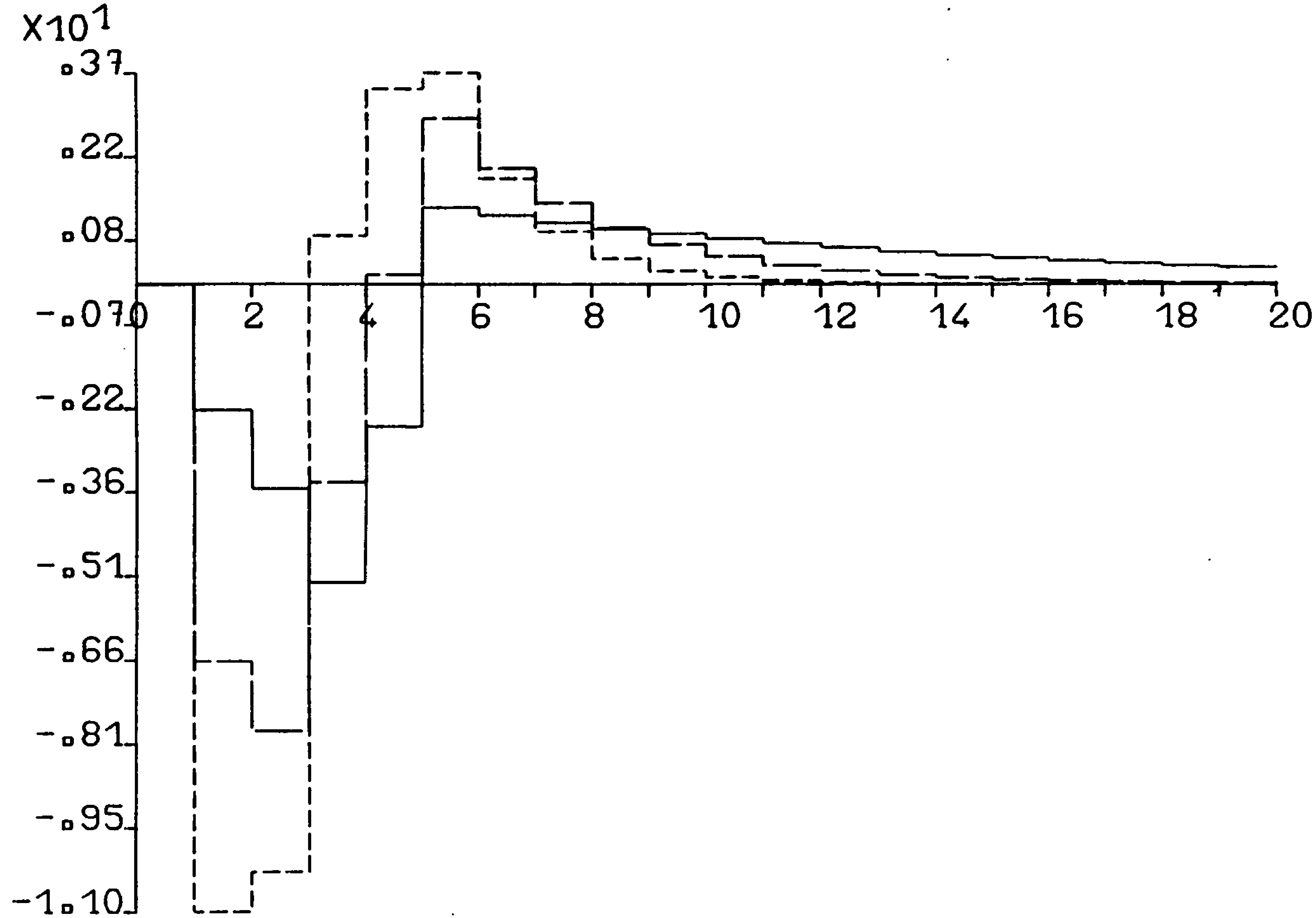

Figure 6.7.b(v) : Base Information; Long Leads; Stock Impulse Response
— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

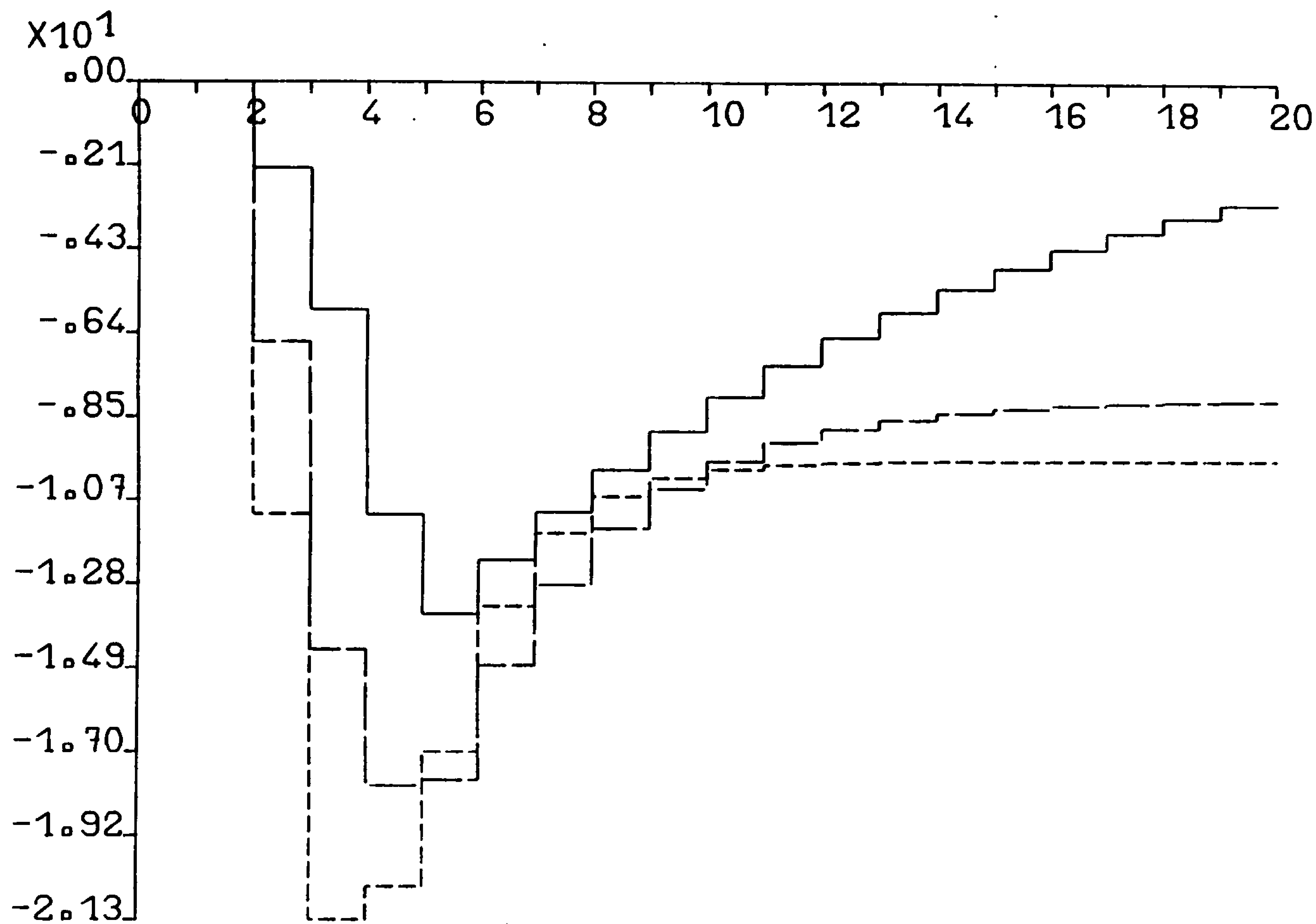

Figure 6.7.b(vi) : Base Information; Long Leads; Stock Step Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

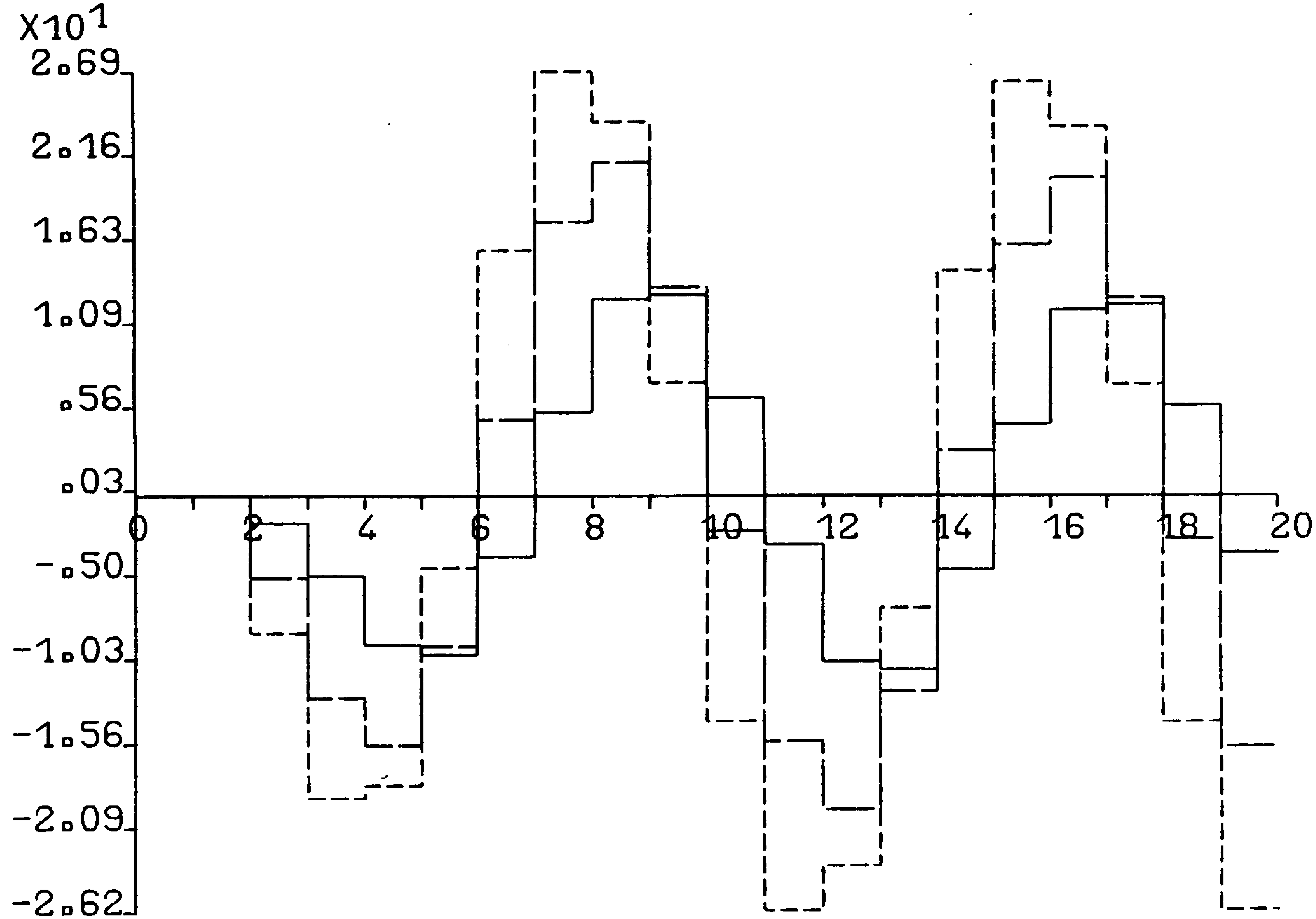

Figure 6.7.b(vii) : Base Information; Long Leads; Stock Sine Response

—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

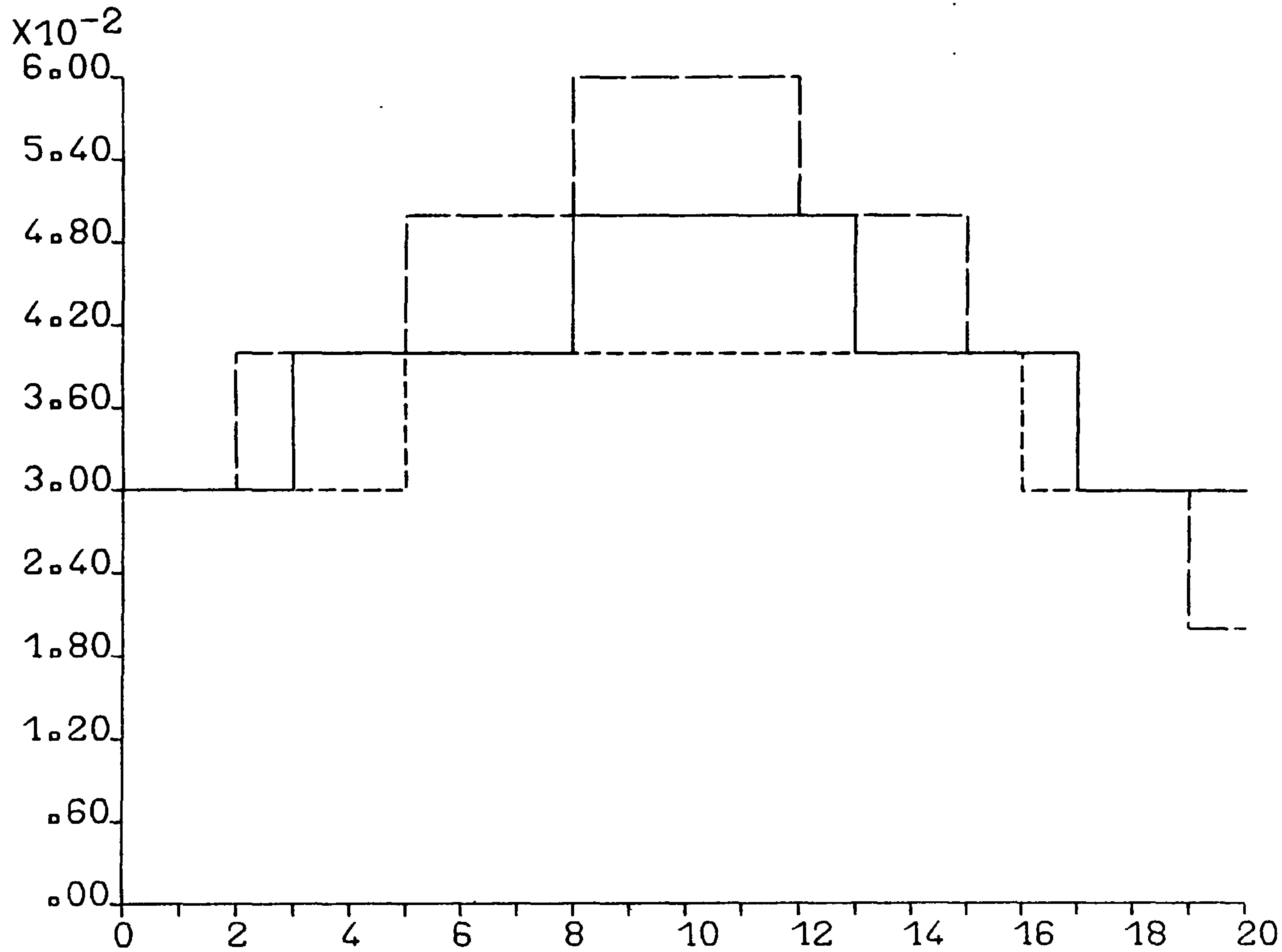

Figure 6.7.b(viii) : Base Information; Long Leads; Stock Noise Response
—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

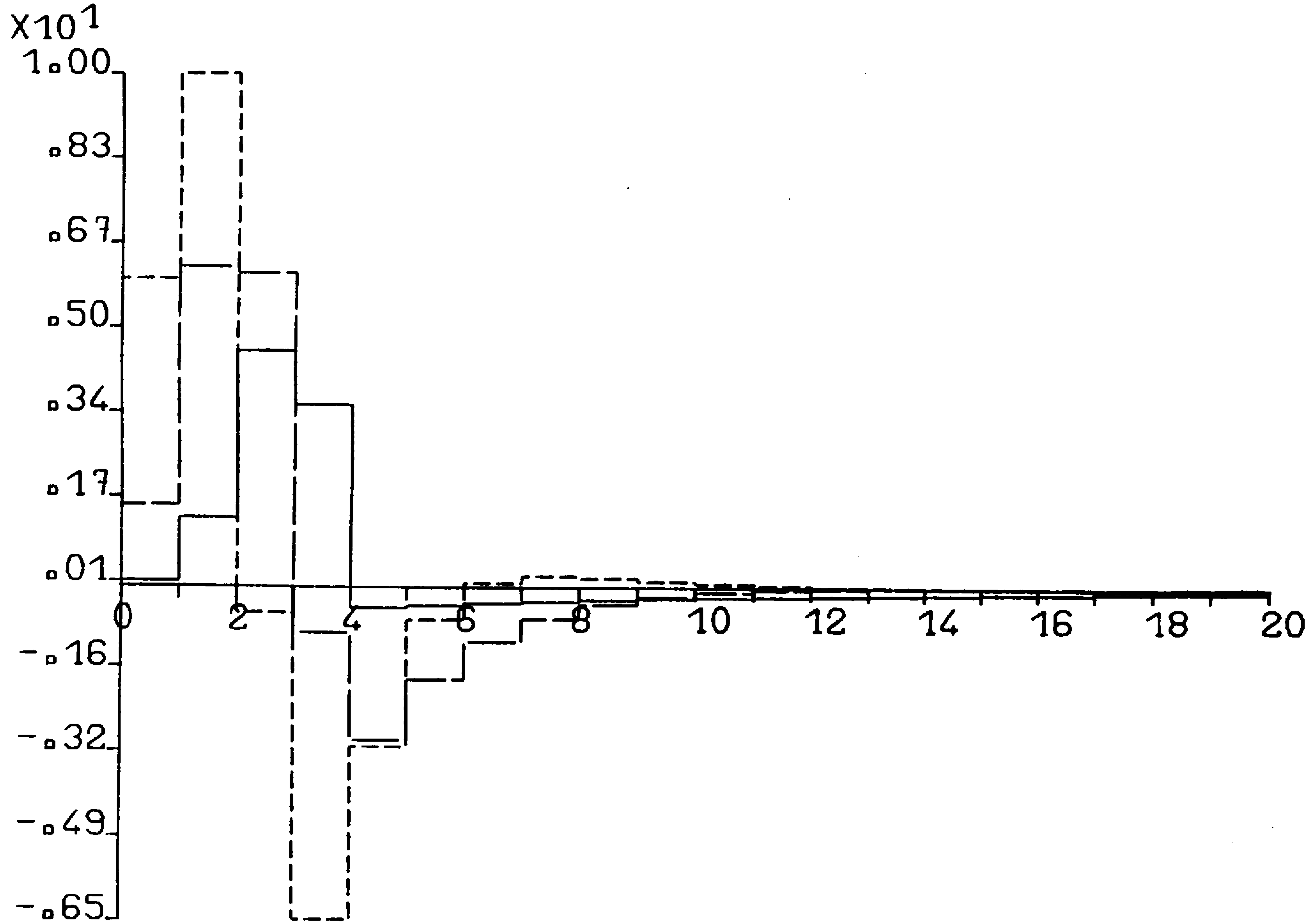

Figure 6.7.c(i) : Cascaded; Short Leads; Schedule Impulse Response

— α = 0.1 — — α = 0.3 - - - α = 0.5

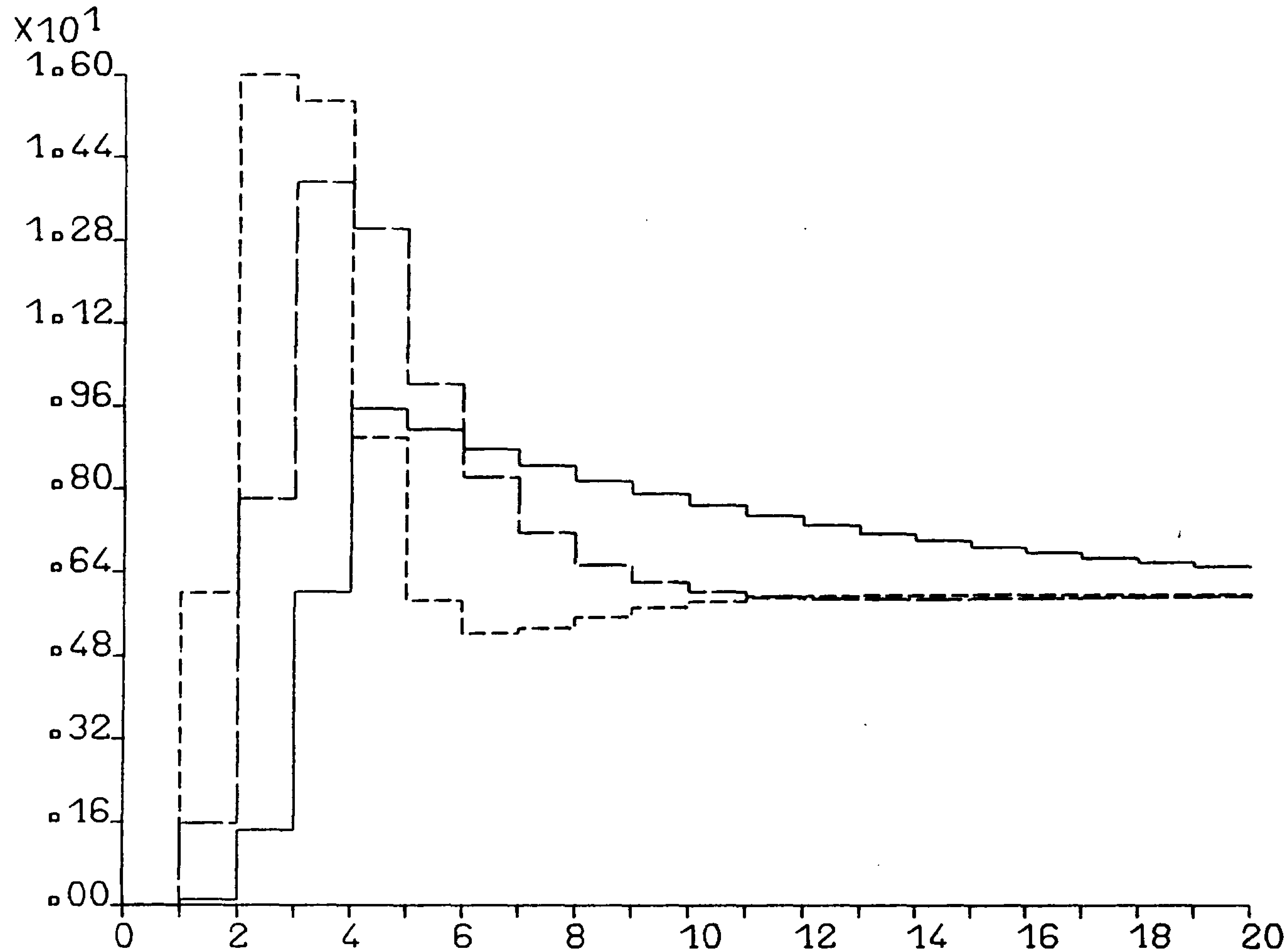

Figure 6.7.c(ii) : Cascaded; Short Leads; Schedule Step Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

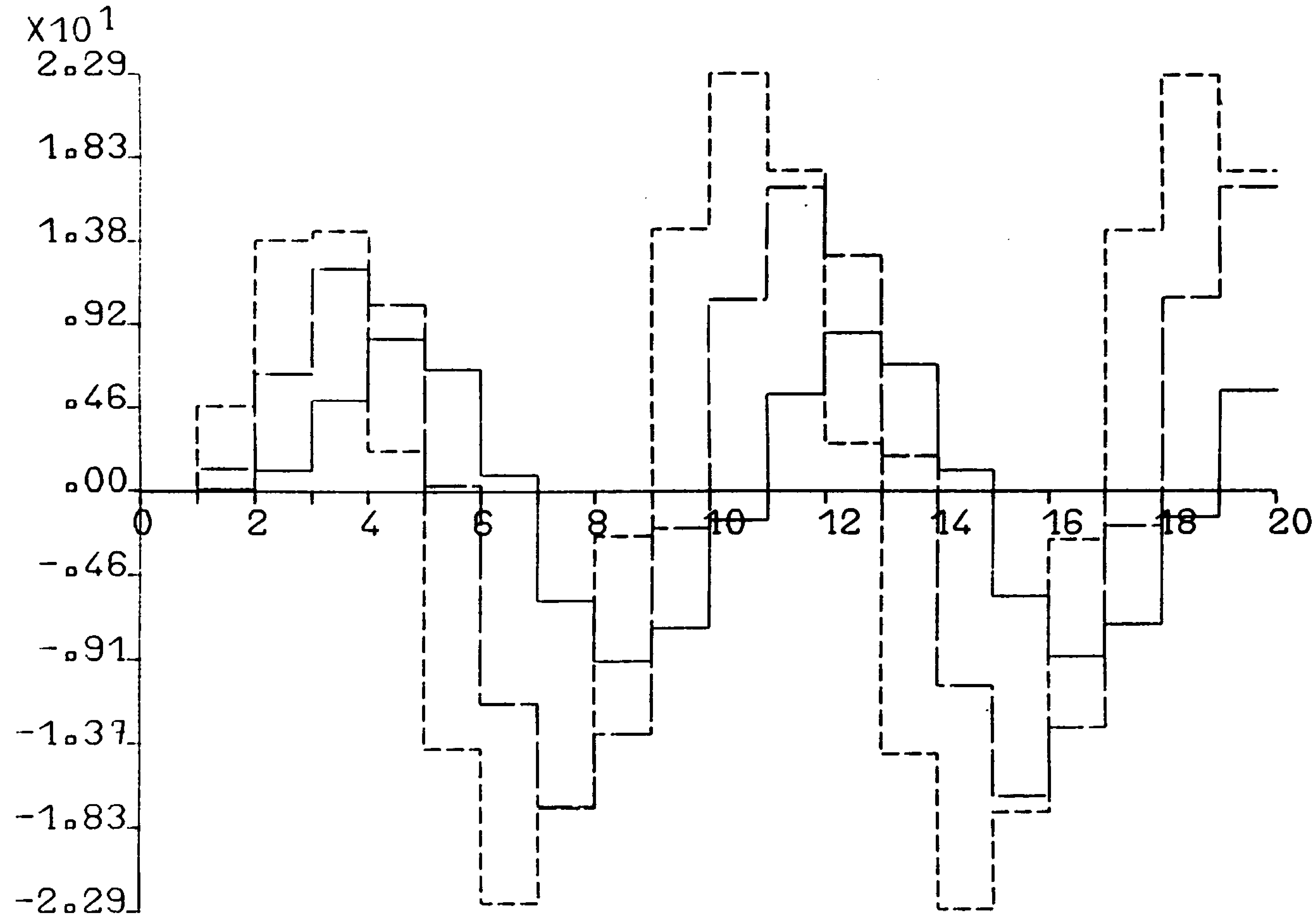

Figure 6.7.c(iii) : Cascaded; Short Leads; Schedule Sine Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

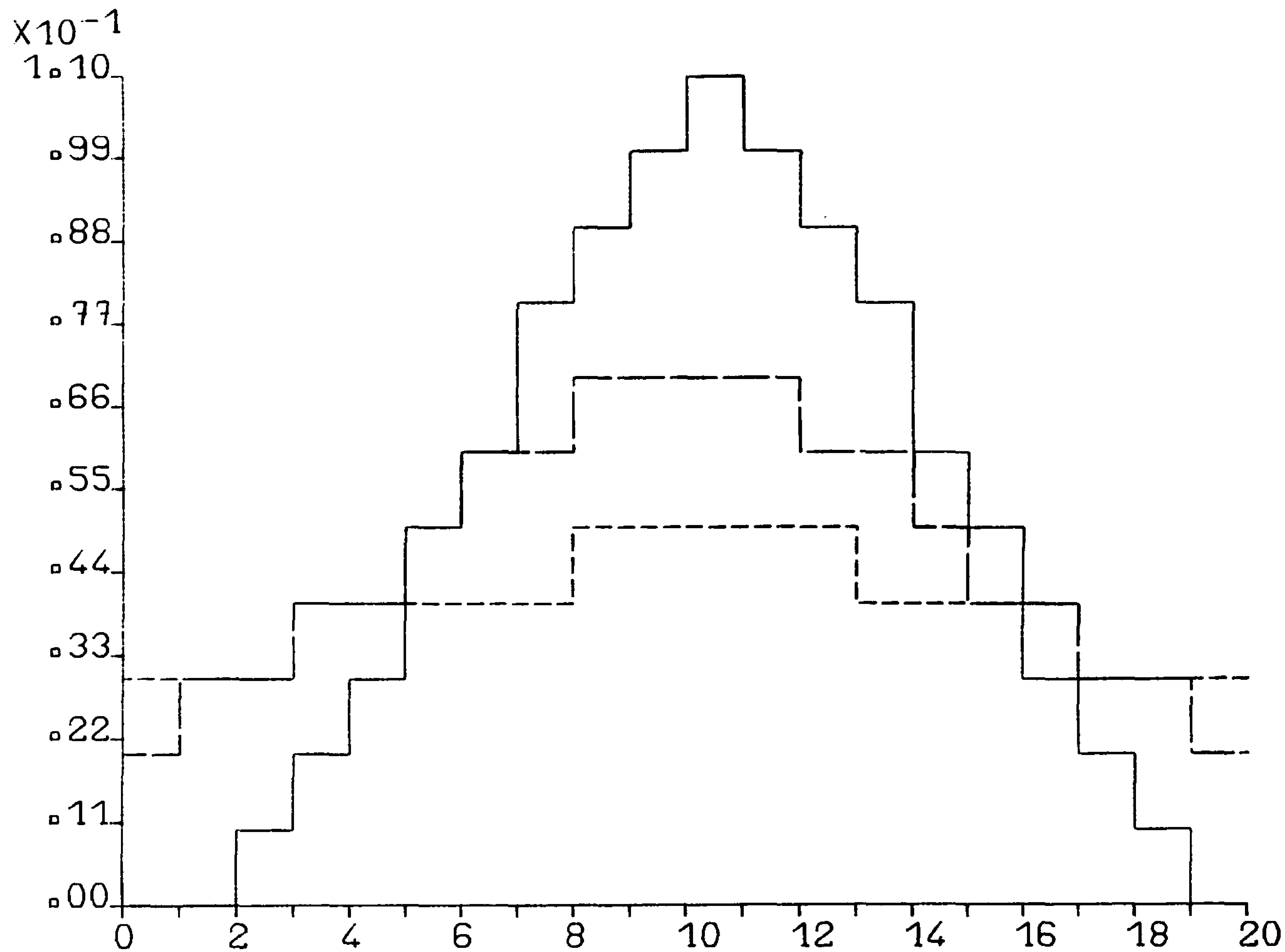

Figure 6.7.c(iv) : Cascaded; Short Leads; Schedule Noise Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

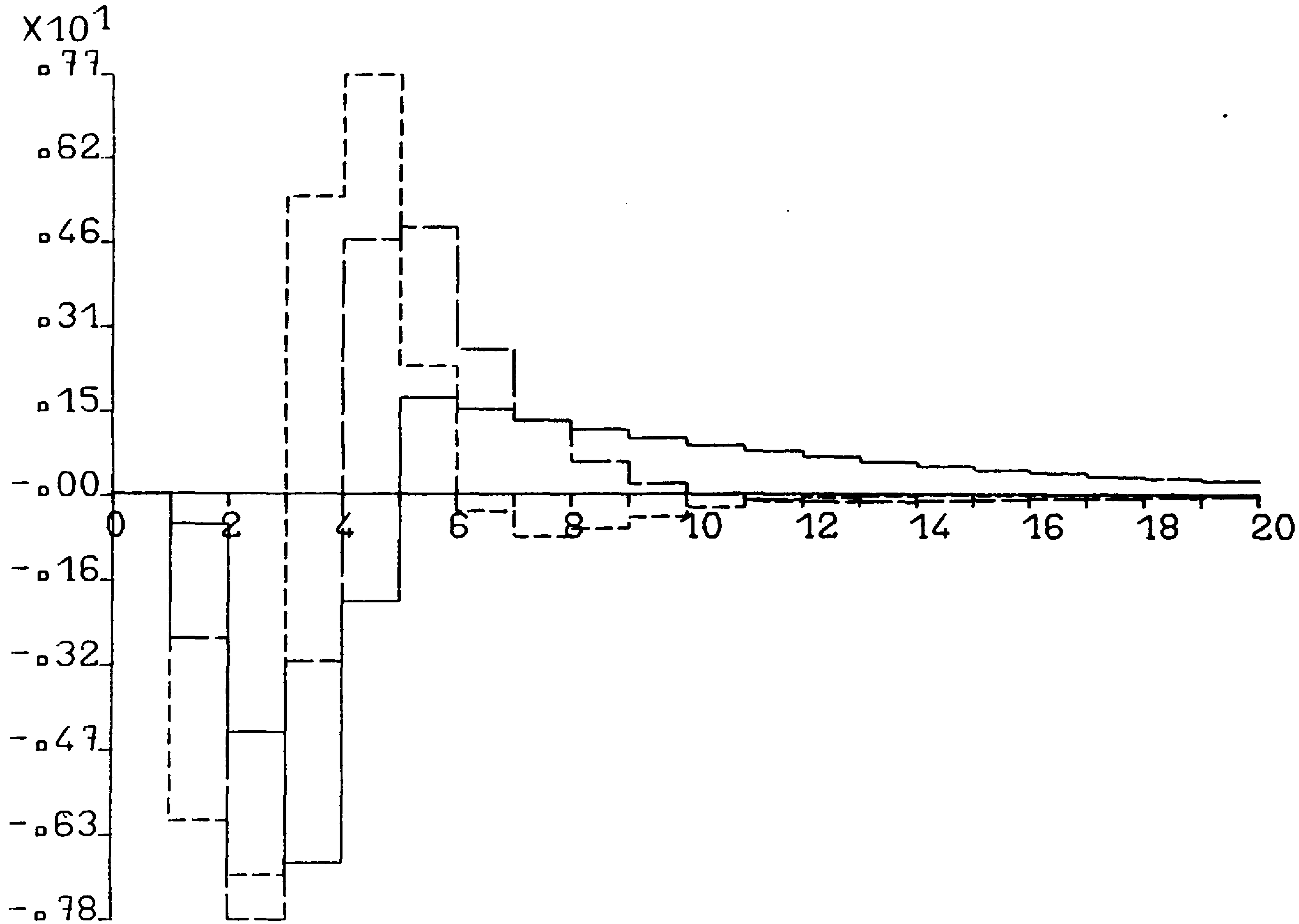

Figure 6.7.c(v) : Cascaded; Short Leads; Stock Impulse Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

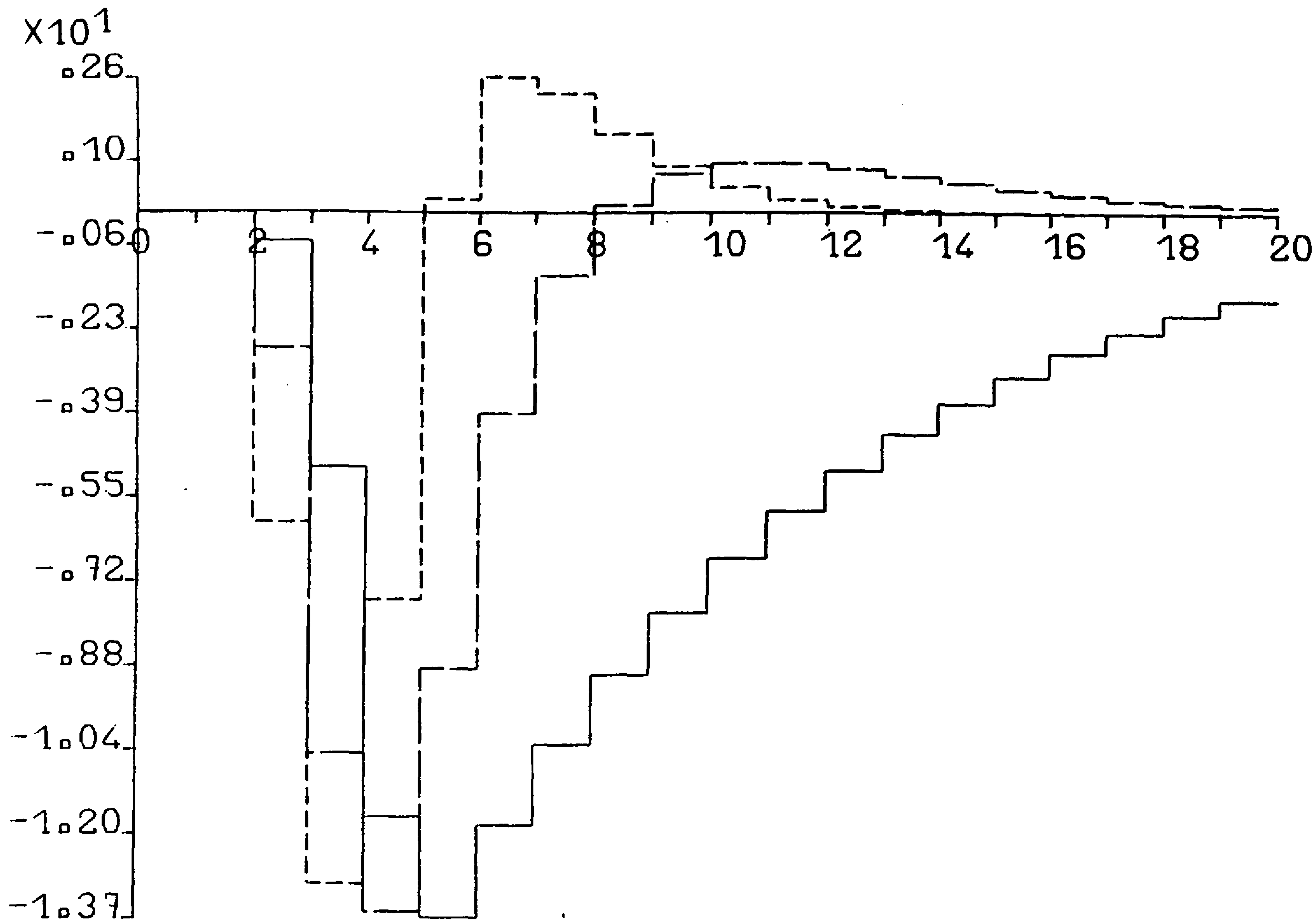

Figure 6.7.c(vi) : Cascaded; Short Leads; Stock Step Response
— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

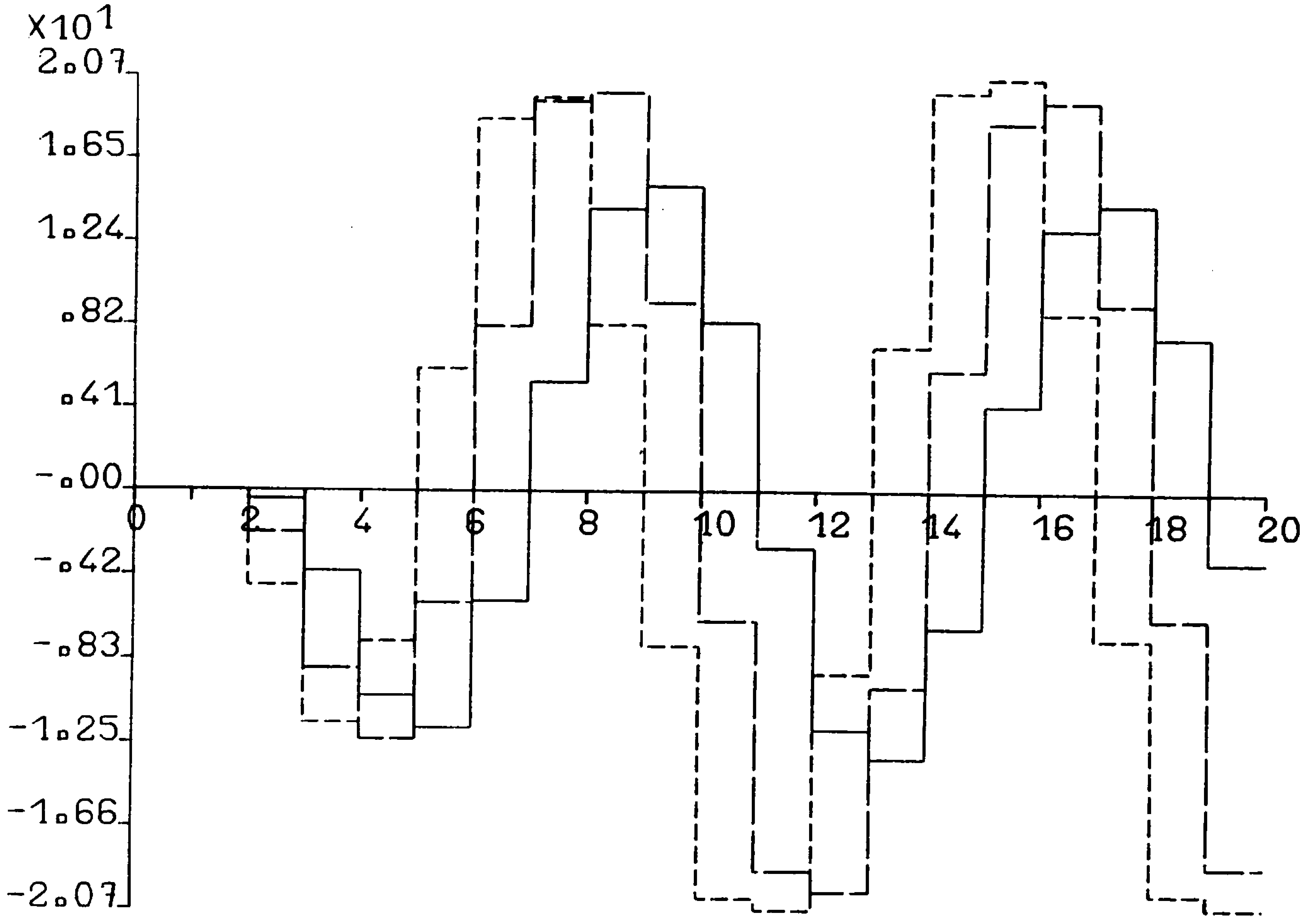

Figure 6.7.c(vii) : Cascaded; Short Leads; Stock Sine Response
—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

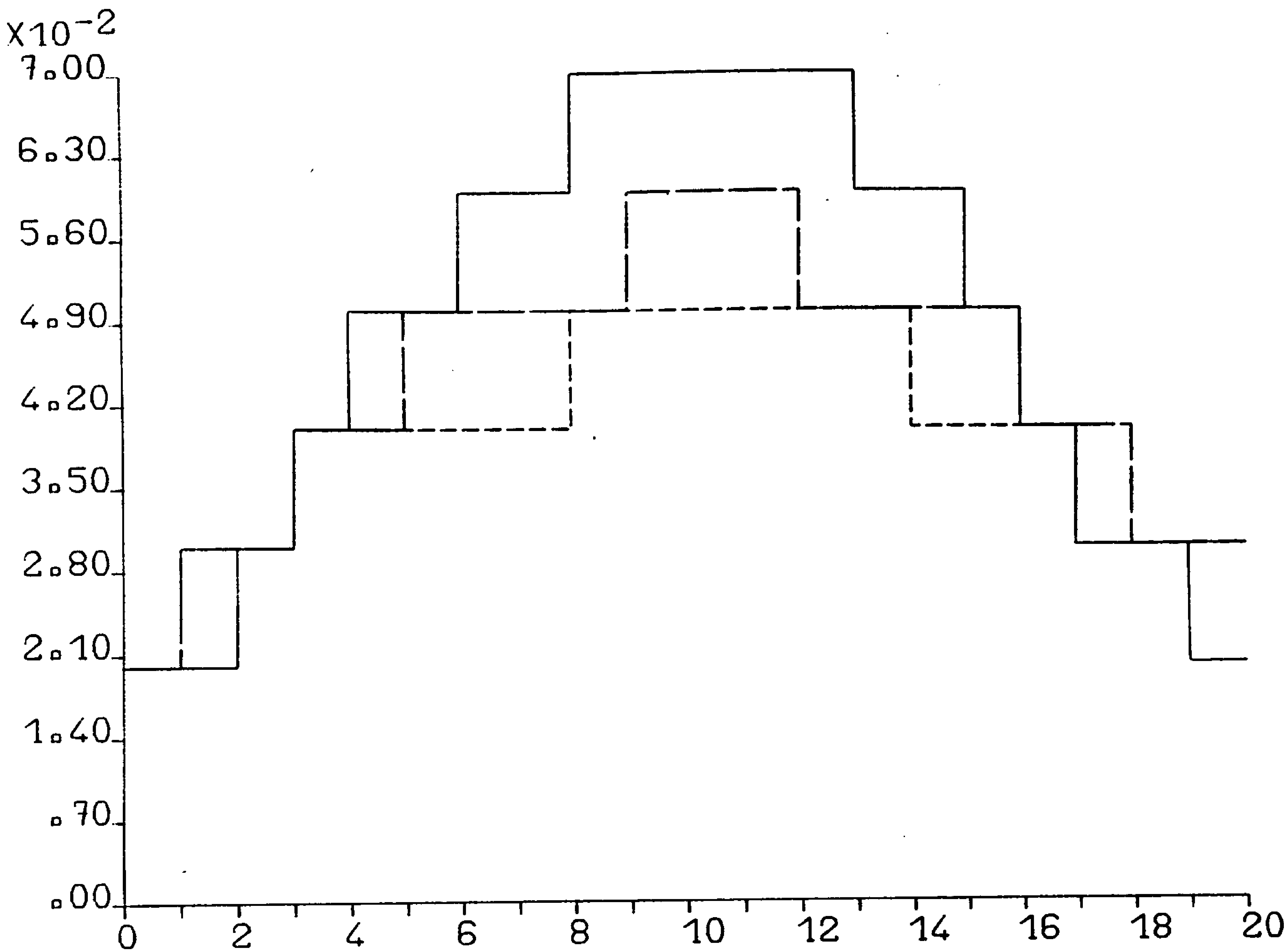

Figure 6.7.c(viii) : Cascaded; Short Leads; Stock Noise Response
—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

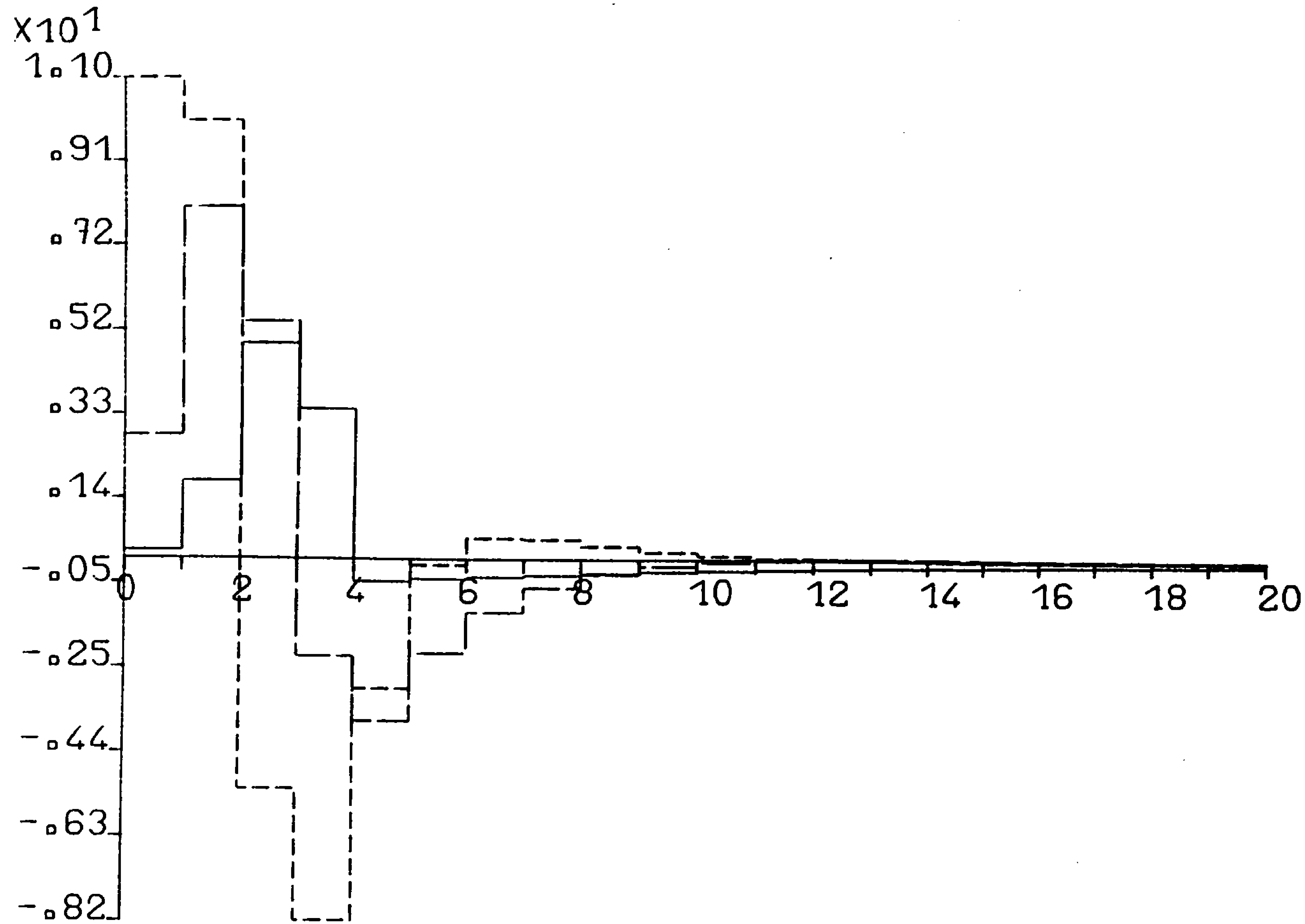

Figure 6.7.d(i) : Cascaded; Long Leads; Schedule Impulse Response

— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

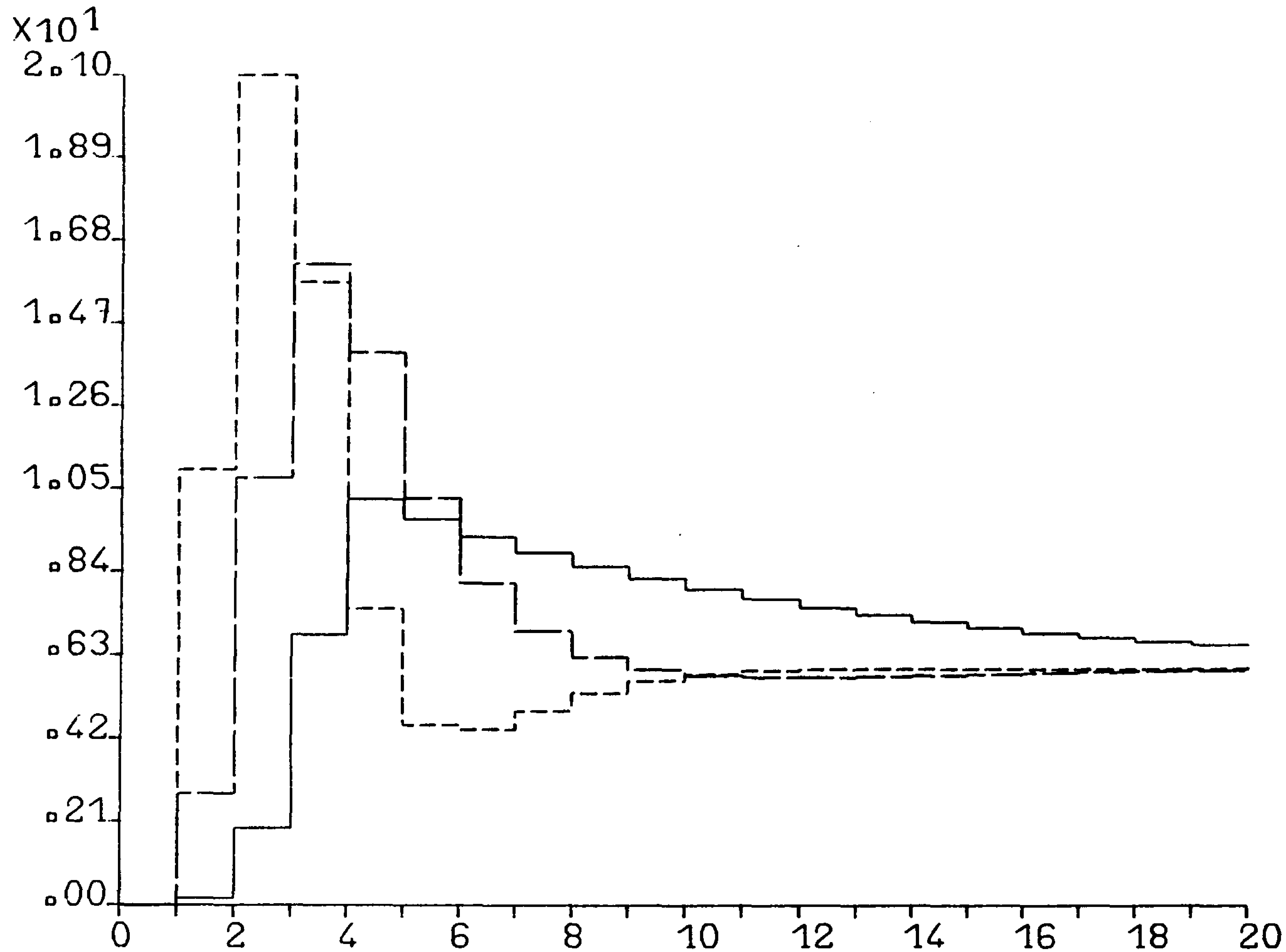

Figure 6.7.d(ii) : Cascaded; Long Leads; Schedule Step Response
— α = 0.1 — — α = 0.3 - - - α = 0.5

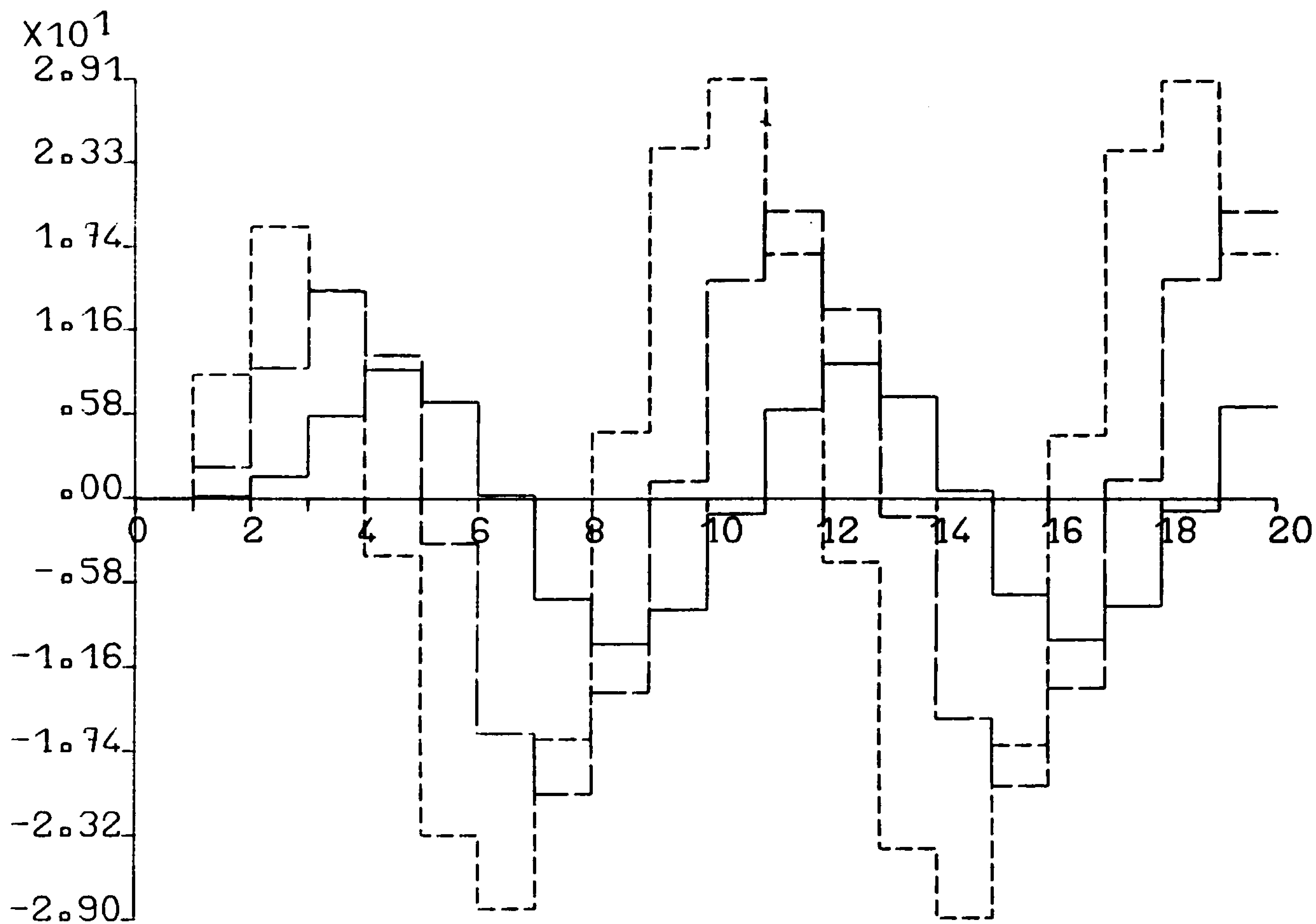

Figure 6.7.d(iii) : Cascaded; Long Leads; Schedule Sine Response
— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

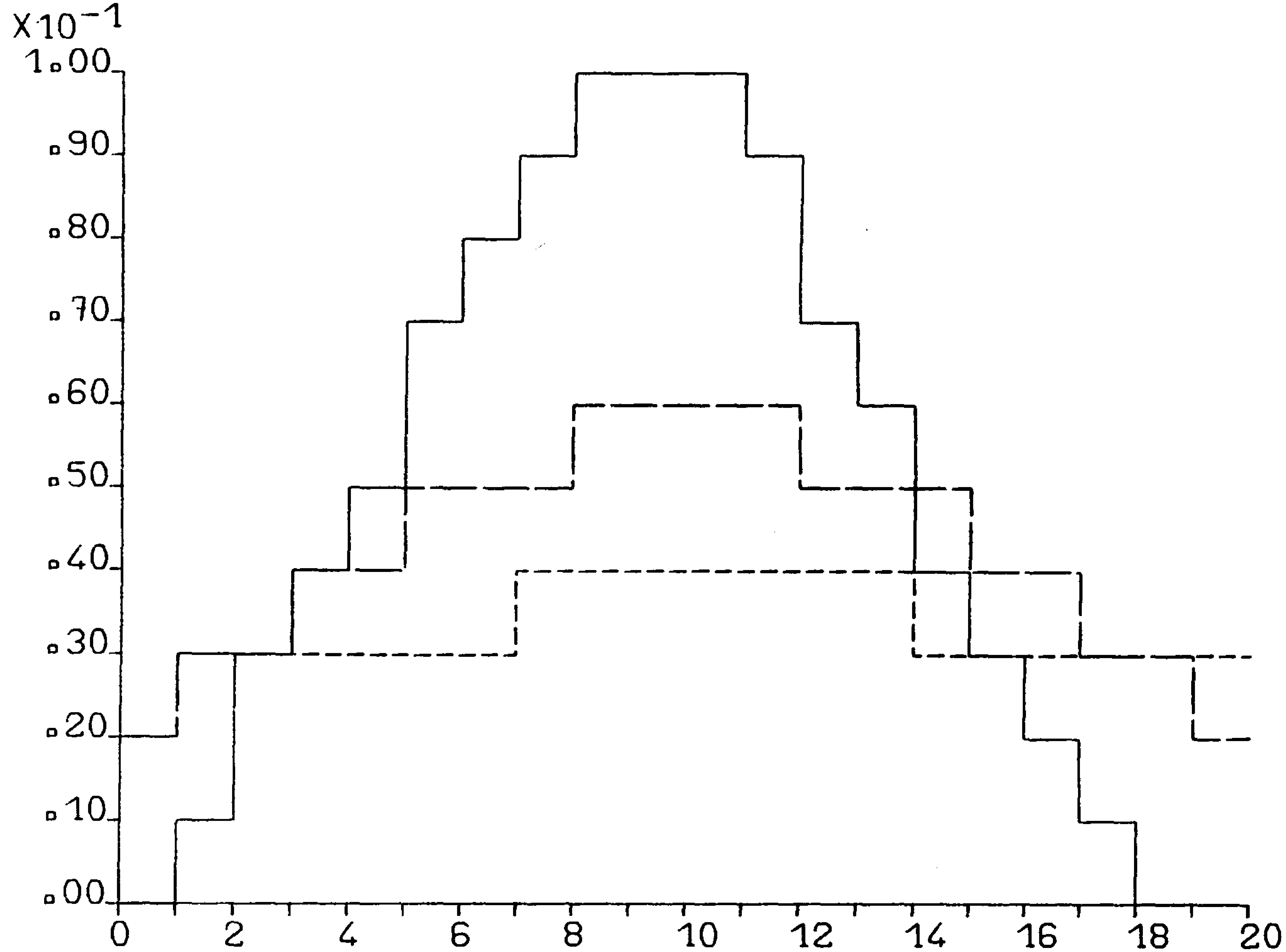

Figure 6.7.d(iv) : Cascaded; Long Leads; Schedule Noise Response
—— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

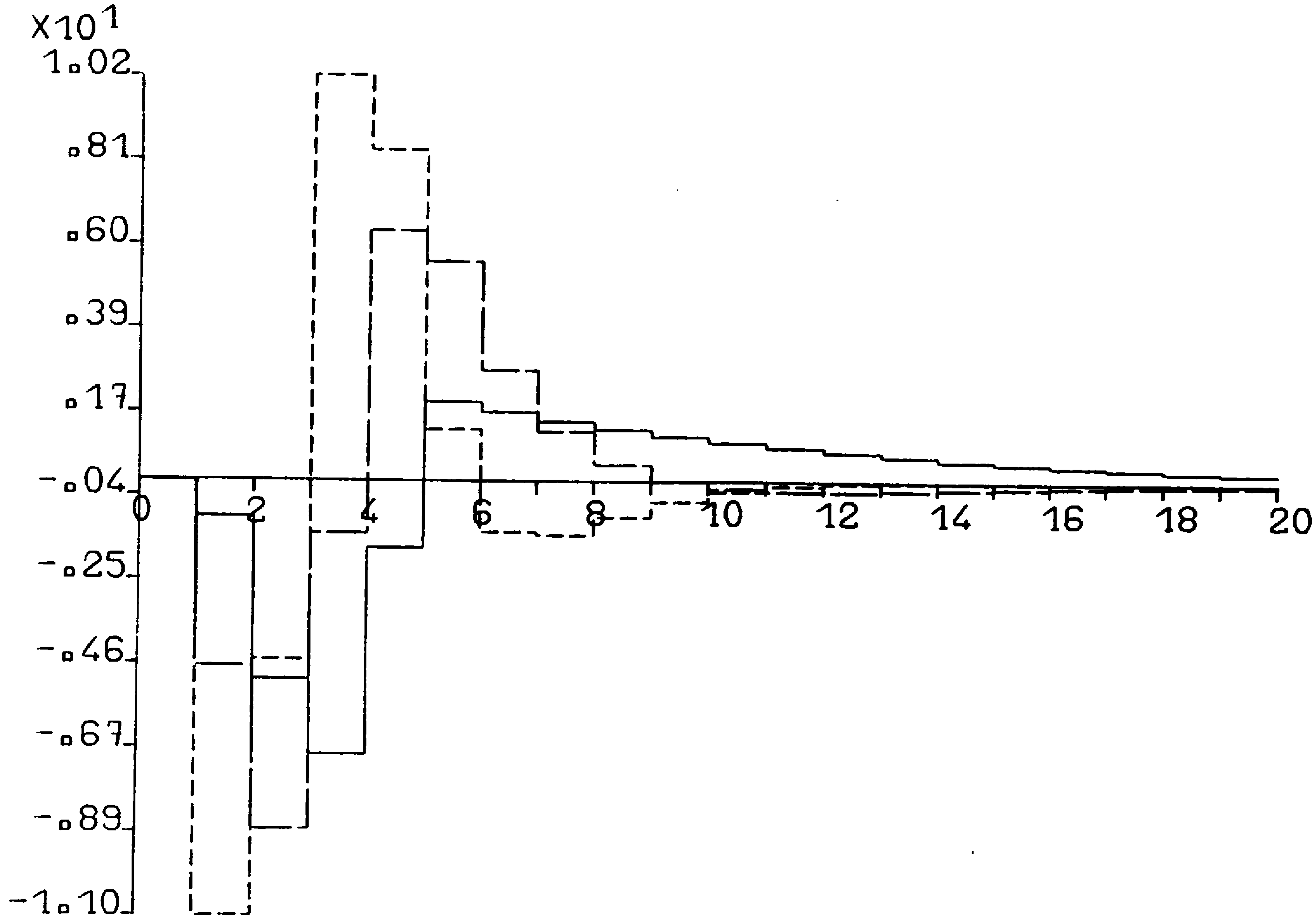

Figure 6.7.d(v) : Cascaded; Long Leads; Stock Impulse Response

— $\alpha = 0.1$ —— $\alpha = 0.3$ --- $\alpha = 0.5$

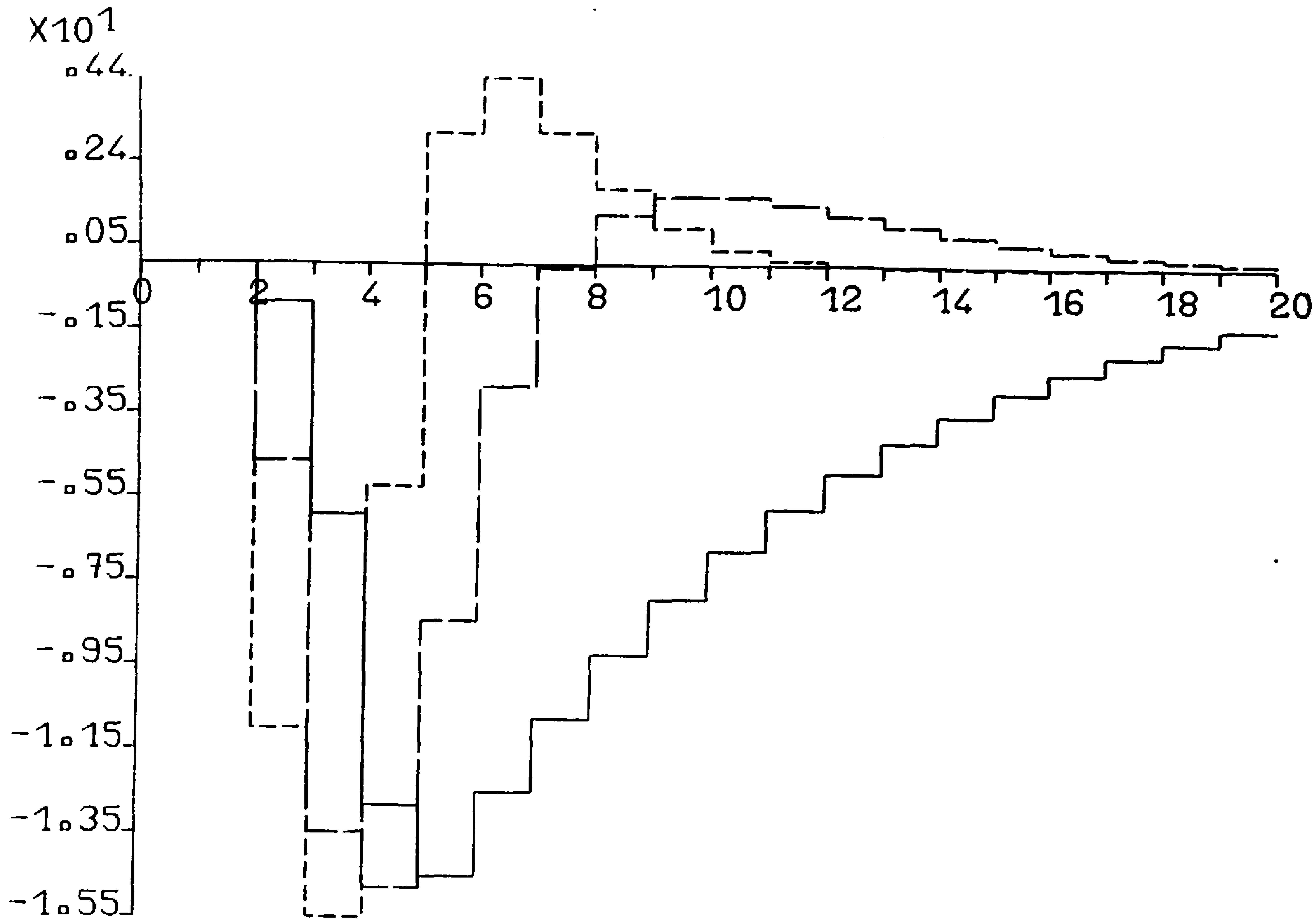

Figure 6.7.d(vi) : Cascaded; Long Leads; Stock Step Response

— $\alpha = 0.1$ —— $\alpha = 0.3$ --- $\alpha = 0.5$

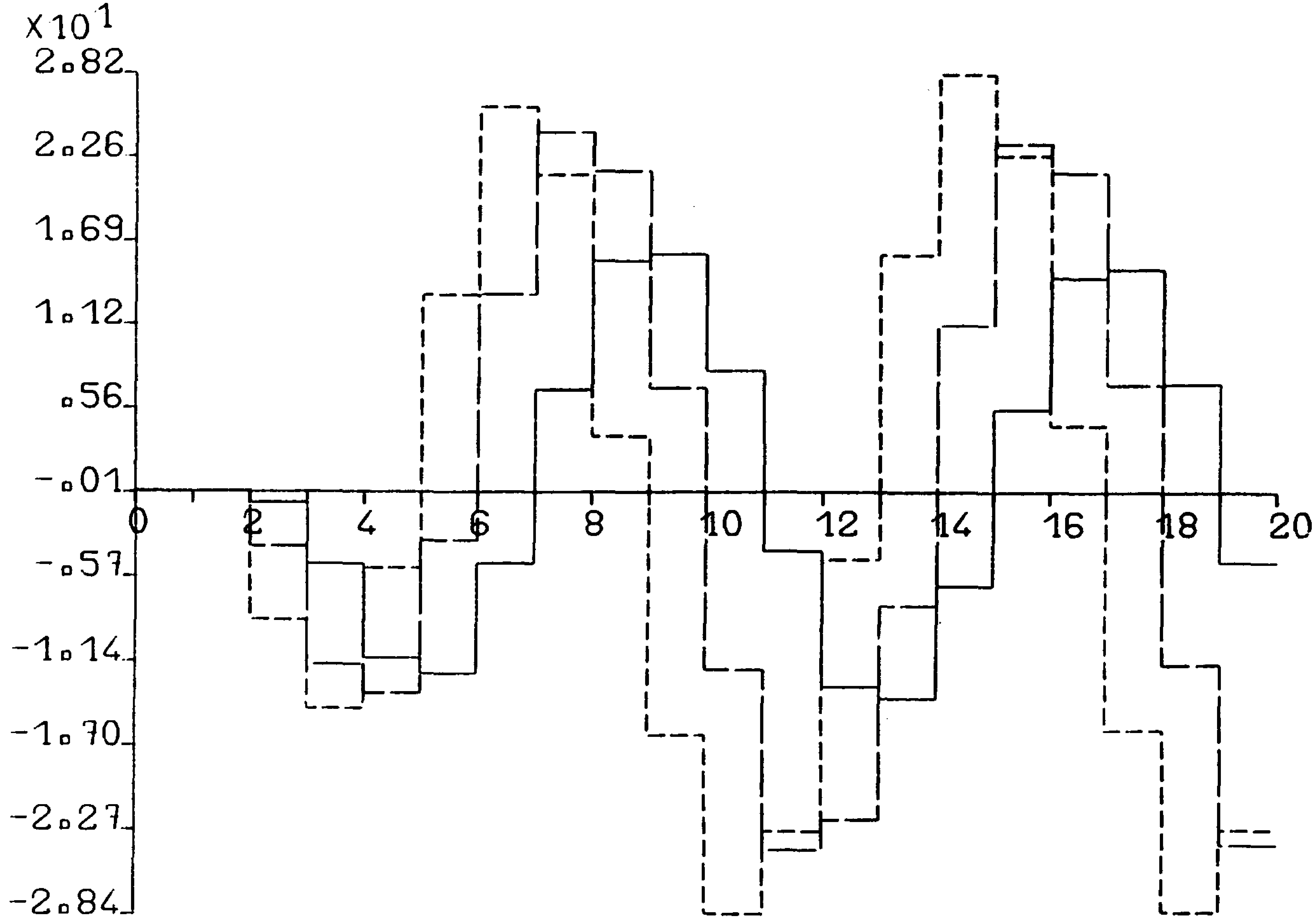

Figure 6.7.d(vii) : Cascaded; Long Leads; Stock Sine Response
— $\alpha = 0.1$ — — $\alpha = 0.3$ - - - $\alpha = 0.5$

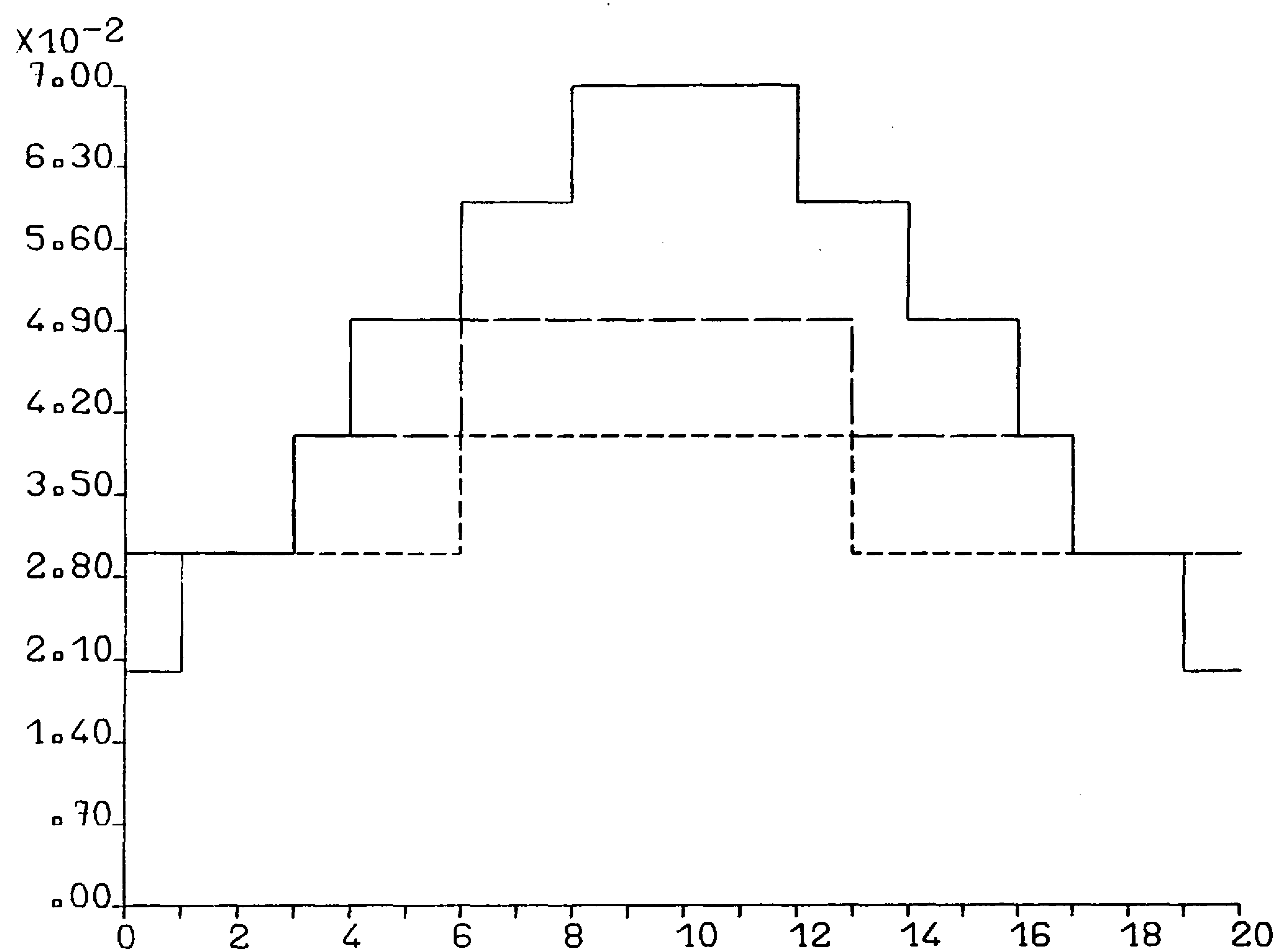

Figure 6.7.d(viii) : Cascaded; Long Leads; Stock Noise Response
— $\alpha = 0.1$ —— $\alpha = 0.3$ --- $\alpha = 0.5$

CHAPTER 7

APPLICATION TO THE DESIGN OF A REAL SYSTEM

7.1 Preamble

In this Chapter we show the application of some of the preceding theory to an actual production control system design problem. This problem arose in a consumer durable manufacturing company whose main product line comprised some 2,000 models assembled from a range of approximately 6,000 materials, components and sub-assemblies. Each model required about 100 distinct parts. Thus the complexity is significant but not extreme.

The company was organised as two distinct divisions enjoying considerable autonomy. One of these was responsible for the production and sale of a specific large range of components, for which the second was a major, but by no means the only, customer. The second division was responsible for production and sale of the finished product. To this end it was concerned with the manufacture of many components from raw materials and the use in assembly of parts from the component division and other external sources.

Within the assembly division, whose activities were diverse, the organisation was very complex. It had been created over many years by growth and adaptation rather than by design and the control systems had grown with the organisation. As a result there were gaps in communication and some parts of the control system were mutually incompatible. This organisation and its control system derived great inertia from its complexity: it was very difficult to understand (or even find!) the entire control system and any major change undertaken in the light of such ignorance would have been potentially disastrous.

Physical modernisation of the factory was planned (and is now being carried out) and the installation of new control systems would become necessary. Many system changes would be linked to physical changes, but others would not. Final decoupling points in assembly had been defined as the last stocking points, on any route to final assembly, where each part was not destined for some specified assembly batch with pre-determined assembly timetable. The set of such points became known as *"the ring"*. All control within the ring is inextricably tied to the physical characteristics of the factory, but the control of provisioning into the ring stores, in the case of bought-out parts supplied from the component division or elsewhere is not so tied.

Although it was thus technically possible to begin the development of improved provisioning systems, problems arose from inter-departmental relationships which could not, instantly, be changed. Careful negotiation was needed before development could commence, and such development work was usually confined to a restricted range of provisioning activity. As a result, opportunities for such development arose piecemeal and usually with a demand to "do something immediately". Spans of a few weeks for system planning, design and implementation were typical. The problems of meeting such deadlines were compounded by the need to link new systems with hitherto untouched elements of existing systems, still beyond the permitted scope of investigations.

A modelling technique was needed to predict system behaviour. Because of its iterative nature digital simulation is too slow a technique to allow analysis of system design in this context, and, when carried out mechanically, it gives little insight into the dynamics of the system. Since control analysis can be a rapid, "back-of-an-envelope" method it proved extremely useful and the insight gained by using this manual technique suggested improvements to design during modelling. Perhaps most importantly we were able to handle elements of the system where we had little or no understanding. It was possible to use analysis of the system where such elements were undefined and to impose restrictions upon their acceptable responses. In this way we could ask precisely the relevant questions about these elements, determining their acceptability or the need to either modify the current design or to enforce change beyond the initial scope of the project.

We describe the process of development of one such sub-system in the remainder of this Chapter.

7.2 The Sub-System

We considered the provisioning of parts to final assembly stock from the component manufacturing division. Because of the special relationship between these two divisions a number of anomalies arose. The greatest of these was the form in which demand was fed from one to the other, which resulted in a mixed base information and cascaded system.

Product demand could be fed simultaneously to the sub-system controlling final assembly and to the supplier's production control system. It was regarded by the latter as a broad forecast of demand to come. Actual demand was subsequently imposed by the assembly control sub-system but this data was available too late to be of use in committing the supplier's production. Thus it was necessary to design a stable system controlling supplier's production on the basis of product demand, yet capable of meeting subsequently derived assembly demands.

To add to the difficulties we also found that the final assembly control sub-system was poorly understood and, that for internal company reasons its re-design or even overt detailed investigation at that time was precluded. All that was known was that it provided part demand two periods after input of model demands. We were thus obliged to treat the sub-system as a "black box" transforming the product demand in some partly random manner to generate part demands upon the supplier. Consequently a major portion of the project had to be the inclusion of compensators in the provisioning system so as to maintain stability under this randomly imposed load.

Figure 7.2.a takes a simple example of a single part, X, used in the assembly of a single product, A. We can base the investigation upon this simplified example without loss of generality by regarding "product A" as the entire range of products which use part X, the assembly control "black box" is then the combined control of this product range. Again, without loss of generality, we assume that each product requires one part X so that a demand for product A represents an equal demand for part X. The remaining elements of the diagram all relate to part X and are precisely those we would expect to find. The only point of note is that part X demand forecasting has, because of the mixed base information/cascaded nature of the system, two inputs.

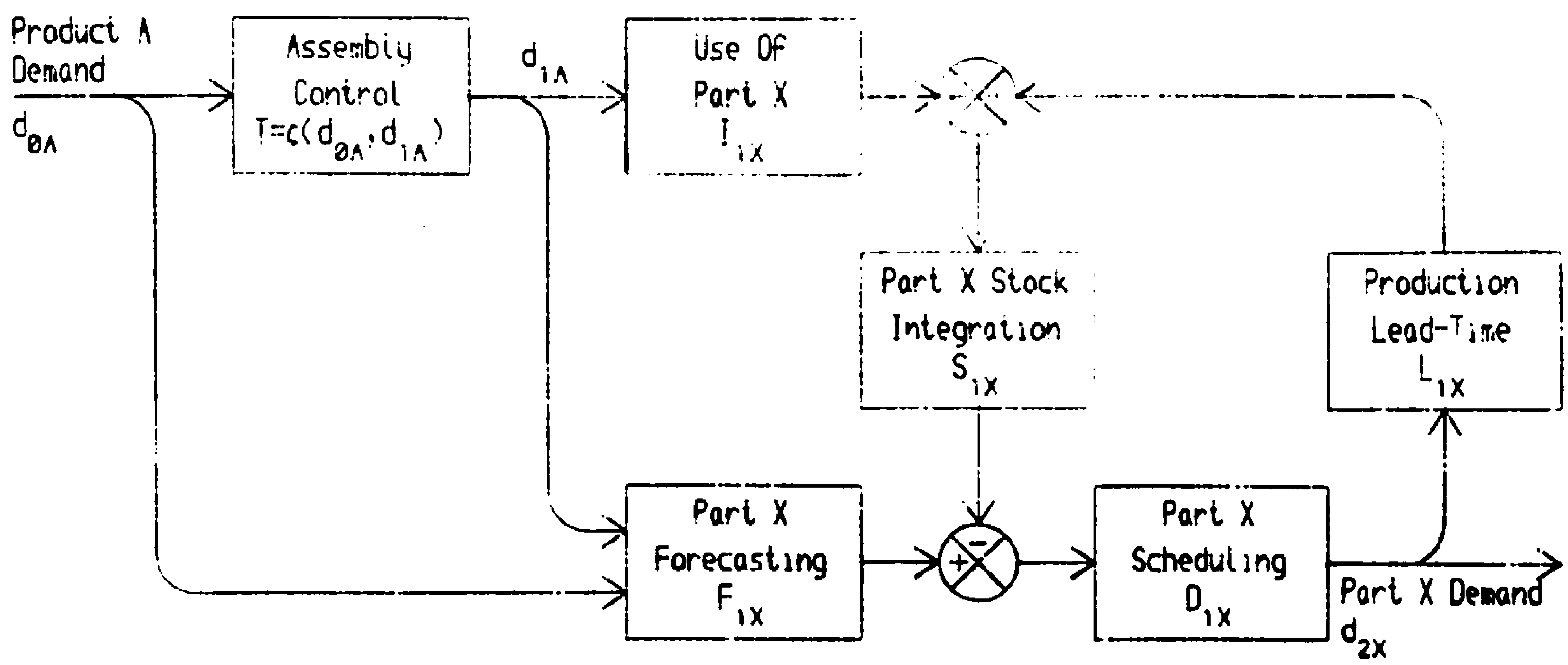

Figure 7.2.a : The Sub-System

The natures of the production and political environments imposed fixed forms on some elements of this system and these are noted below:-

$\zeta(d_{OA},d_{1A})$: this is the "black box" already described. We will henceforth denote this z-transfer function by T.

S_{1X} : this should be the standard z-transfer function for stock integration $\frac{1}{1 - z^{-1}}$. However, although appropriate stock recording systems were under development they were not expected to be available to this control system in time for its implementation. We therefore used a system incorporating no feedback of stock records. We shall actually use $S_{1X} = \frac{1}{1 - z^{-1}}$ to derive current stock balances but scale s_{1X} by zero before subtraction from forecasts.

I_{1X} : delivery from supplier's finished stock is made in the period immediately following delivery schedule (d_{1A}') generation. Thus $I_{1X} = z^{-1}$.

L_{1X} : production lead-time is fixed for each part by its engineering characteristics. Taking delivery lead-time as λ for part X then $L_{1X} = z^{-\lambda}$.

D_{1X} : since stock balance was unavailable netting became impossible and we took $D_{1X} = 1$.

It is now clear that the only system element available for modification was forecasting, F_{1X}. Several forms of this element were considered and these are described below. The descriptions are presented in the order in which they arose and were examined.

7.3 Proposal I

The first proposal investigated was that each period, the new schedule be calculated as the gross requirement to meet product A demand modified by the difference $d_{1A}(t) - f_{1X}(t-2)$, that is the error in the forecast made two periods ago for demand this period (Figure 7.3.a).

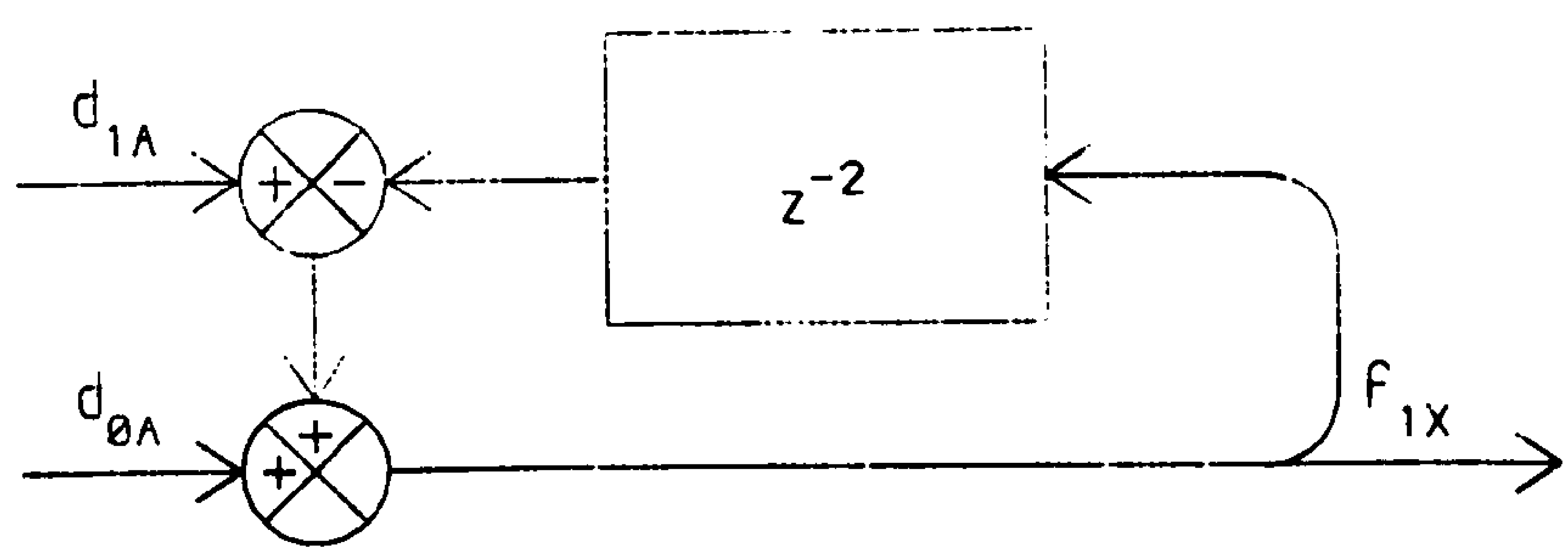

Figure 7.3.a : Proposal I

Since F_{1X} has two inputs it cannot be represented entirely by one z-transfer function, but is completely described by the two z-transfer functions $F_{1X}(d_{1A},f_{1X})$ and $F_{1X}(d_{0A},f_{1X})$. These in turn are contained in the same equation derived below:

$$f_{1X} = d_{0A} + (d_{1A} - f_{1X}z^{-2})$$

$$\therefore f_{1X}(1 + z^{-2}) = d_{0A} + d_{1A}$$

$$\therefore f_{1X} = \frac{d_{0A} + d_{1A}}{1 + z^{-2}} \quad (7.3.1)$$

We note that both z-transfer functions derived from 7.3.1 have characteristic equation $1 + z^{-2} = 0$ whose roots lie upon the unit circle. Thus F_{1X} introduces instability into the system.

Here we had a system proposal which superficially seemed adequate in that the forecast to be used in scheduling was always arithmetically correct. Each schedule accurately reflected the best known future demands but, when considered dynamically, the system was unstable : any discrepancy between f_{1X} and d_{1X} would result in a permanent undamped oscillation in f_{1X}, and a random sequence of such forecast errors would accumulate in f_{1X} which will diverge in a random walk.

From this point three paths were apparent. These were considered as proposals II, III and IV.

7.4 Proposal II

There is a strong tradition within the company of using data accumulated from some datum point for many and varied purposes, so it was not surprising that, as soon as the inadequacy of proposal I was exposed, the use of accumulated demand and forecast data to generate forecasts was suggested. This was far more the next idea in production management's repertoire than anything arising out of the analysis of proposal I.

Thus we considered calculating a new forecast by adding to the product demand the difference between accumulated part demand and accumulated forecasts for the corresponding period (Figure 7.4.a).

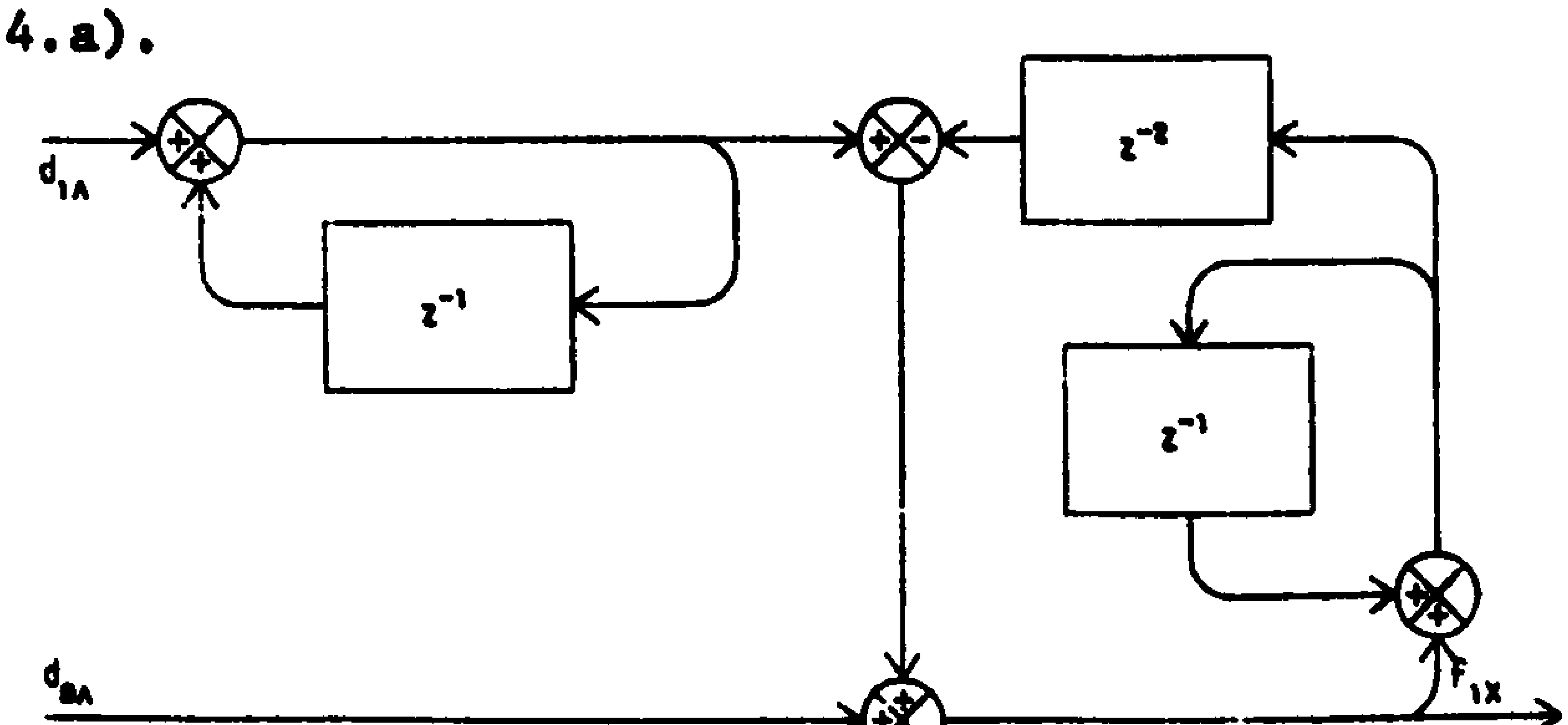

Figure 7.4.a : Proposal II

From the diagram:

$$f_{1X} = d_{0A} + \frac{d_{1A}}{1 - z^{-1}} - \frac{z^{-2}f_{1X}}{1 - z^{-1}}$$

$$\therefore\ (1 - z^{-1})f_{1X} = d_{0A}(1 - z^{-1}) + d_{1A} - z^{-2}f_{1X}$$

$$\therefore\ f_{1X}(1 - z^{-1} + z^{-2}) = d_{0A}(1 - z^{-1}) + d_{1A}$$

and so we derive the source equation:-

$$f_{1X} = \frac{d_{0A}(1 - z^{-1}) + d_{1A}}{1 - z^{-1} + z^{-2}} \qquad (7.4.1)$$

which as in the case of proposal I (7.3.1) contains both $F_{1X}(d_{0A},f_{1X})$ and $F_{1X}(d_{1A},f_{1X})$. These two have the characteristic equation:

$$1 - z^{-1} + z^{-2} = 0$$

This has roots $z^{-1} = \dfrac{1 \pm i\sqrt{3}}{2}$ which lie on the unit circle, and hence the corresponding values of z lie upon the unit circle. Thus this proposal was unstable in just the same way as proposal I and was rejected.

7.5 Proposal III

The effect of a characteristic equation root lying on the unit circle is to introduce an undamped oscillation resulting from any perturbation. We directly use the characteristic equation to introduce an appropriate form of damping to stabilise the system of proposal I.

Proposal I had characteristic equation $1 + z^{-2} = 0$ with roots $\pm i$ upon the unit circle. Let $0 < \mu < 1$ and take $z = -i\mu$.

Then $z^{-1} = \frac{i}{\mu}$ and $z^{-2} = \frac{-1}{\mu^2}$, so to achieve stability we could adopt a system with characteristic equation:-

$$\frac{1}{\mu^2} + z^{-2} = 0$$

To achieve this without changing the basic form of equation 7.3.1 we could use the forecasting system:-

$$f_{1X} = \frac{d_{0A} + d_{1A}}{\mu^{-2} + z^{-2}}$$

i.e. $f_{1X} = \mu^2(d_{0A} + d_{1X} - z^{-2}f_{1A})$

However, although this is stable it is also arithmetically wrong! It damps the whole of demand as well as the "error" term and so despite its stability under-orders.

The above does suggest that by damping only the forecast error term a sensible system may be obtained (Figure 7.5.a).

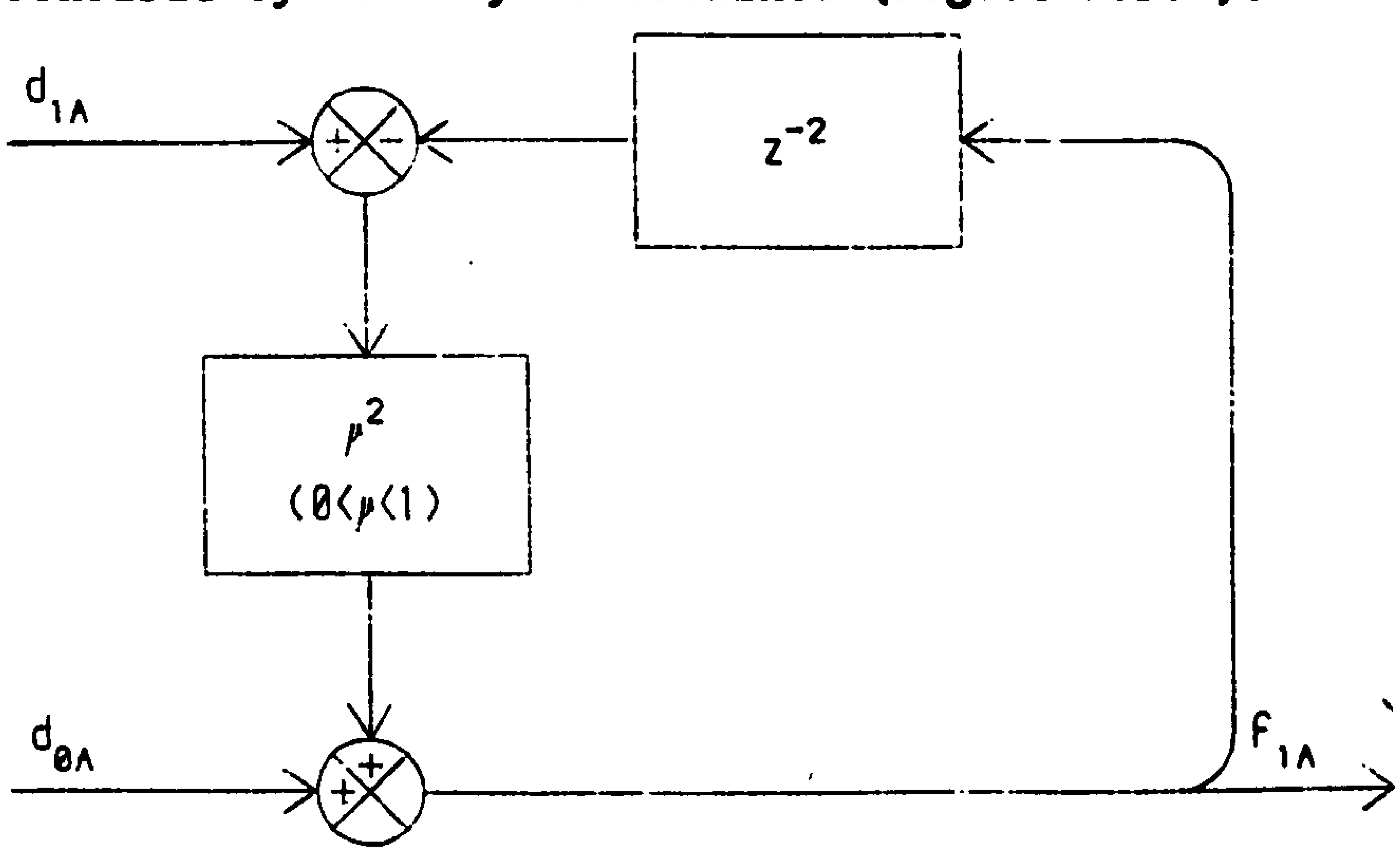

Figure 7.5.a : Proposal III

Here it is clear that:-

$$f_{1X} = d_{OA} + \mu^2(d_{1A} - z^{-2}f_{1X})$$

$$\therefore (1 + \mu^2 z^{-2})f_{1X} = d_{OA} + \mu^2 d_{1A}$$

$$\therefore f_{1X} = \frac{d_{OA} + \mu^2 d_{1A}}{1 + \mu^2 z^{-2}} \qquad (7.5.1)$$

and this, a source equation for $F_{1X}(d_{OA},f_{1X})$ and $F_{1X}(d_{1A},f_{1X})$, has characteristic equation:-

$$1 + \mu^2 z^{-2} = 0$$

$$z^{-2} = -\frac{1}{\mu^2}$$

$$z^{-1} = \pm i \sqrt{\frac{1}{\mu^2}}$$

$$z = \pm i\mu \quad .$$

This lies within the unit circle so we have a stable system.

Allowing $\zeta(d_{OA},d_{1A})$ to be its simplest deterministic form we can derive the system z-transfer function $T(d_{OA},f_{1X})$ as follows (see Figure 7.2.a for all causal routes):

$$T(d_{OA},f_{1X}) = F_{1X}(d_{OA},f_{1X}) + TF_{1X}(d_{1A},f_{1X})$$

$$= \frac{1 + z^{-2}\mu^2}{1 + z^{-2}\mu^2} = 1$$

Hence the forecast is untransformed demand. It is therefore only stochastic or spreading components of T which may invalidate the proposal. To examine the effect of a stochastic element we obtain from 7.5.1:-

$$F_{1X}(d_{1A},f_{1X}) = \frac{\mu^2}{1 + \mu^2 z^{-2}}$$

$$= \mu^2 - \mu^4 z^{-2} + \mu^6 z^{-4} - \dots\dots$$

Thus the response of f_{1X} to a unit impulse in d_{1A} is the time-series:-

$$\mu^2,\ -\mu^4,\ \mu^6,\ -\mu^8,\ \dots\dots$$

This is a geometric progression and as such has the infinite sum:-

$$\mu^2 \lim_{n\to\infty} \left\{\frac{1 - (-\mu^2)^n}{1 + \mu^2}\right\} = \frac{\mu^2}{1 + \mu^2}$$

Thus the net effect of a unit impulse in d_{1A} is a total increase of $\frac{\mu^2}{1 + \mu^2}$ in the total forecast. For any real value of μ this is less than 1. Hence perturbations introduced by assembly scheduling will be inadequately reflected in the forecast.

7.6 Proposal IV

It is apparent from the above that there is something inherently wrong in the solutions tried so far. The fault lies in that the wrong error term is being used. We are considering the forecast as directly comparable with delivery schedules, and compensating accordingly, but in doing this we introduce comparisons with old compensations. We should regard the model demand, d_{OA}, as a predictor for part demand d_{1A} and compensate for errors in this predictor (Figure 7.6.a).

We can derive a source equation as follows:-

$$f_{1X} = d_{OA} + d_{1A} - d_{OA}z^{-2}$$

$$f_{1X} = d_{OA}(1 - z^{-2}) + d_{1A} \quad (7.6.1)$$

This sub-system responds to a unit impulse in d_{OA} by a positive then a negative unit impulse in f_{1X}, and to a unit impulse in d_{1A} by a positive unit impulse in f_{1X}. Thus we have a simple and elegant solution achieving the shortest possible response delays.

We proceed to examine the response of the complete system when this form of F_{1X} is employed. Using the methods of Chapter 4 on Figure 7.2.a we can now easily show that:-

$$T(d_{OA},d_{2X}) = \frac{d_{2X}}{d_{OA}}$$

$$= \frac{(F_{1X}(d_{OA},f_{1X}) + TF_{1X}(d_{1A},f_{1X}) + TI_{1X}S_{1X})D_{1X}}{1 + L_{1X}S_{1X}D_{1X}}$$

Which, since $S_{1X} = 0$ and $D_{1X} = 1$ is:-

$$T(d_{OA},d_{2X}) = F_{1X}(d_{OA},f_{1X}) + TF_{1X}(d_{1A},f_{1X})$$

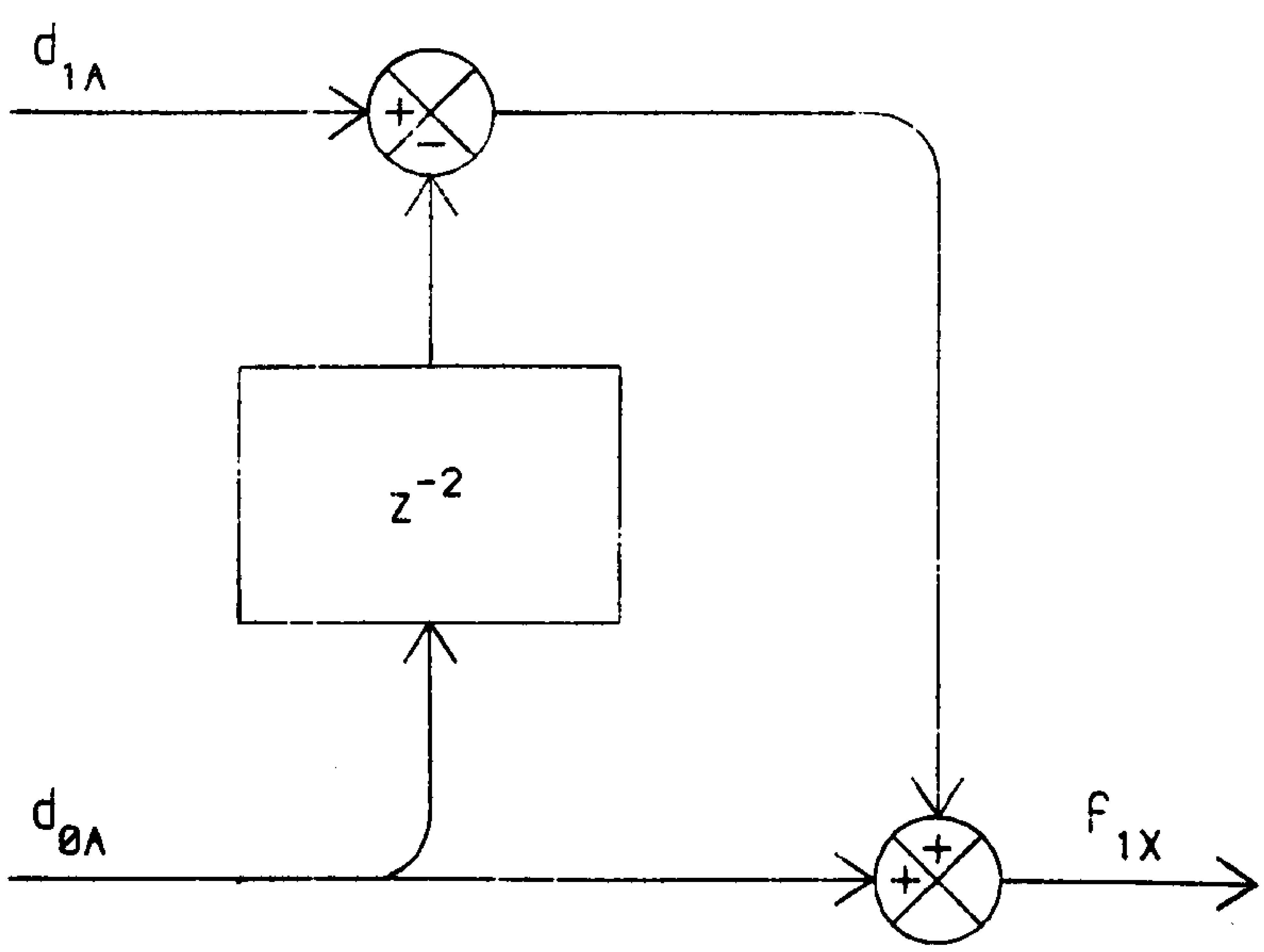

Figure 7.6.a : Proposal IV

so substituting from 7.6.1 we have:-

$$T(d_{OA}, d_{2X}) = 1 - z^{-2} + T \qquad (7.6.2)$$

Thus we know that the system as a whole is stable provided only that T introduces no instability.

We can also predict the possible responses to certain forms of T. We designed F_{1X} by assuming T to contain a delay of two periods. Substituting $T = z^{-2}$ in 7.6.2 we have $T(d_{OA}, d_{2X}) = 1$ and so if our assumption were accurate the system would achieve the ideal of raising part production schedules equal to model demands.

Returning to Figure 7.2.a and letting S'_{1X} be the first, non-zero component of S_{1X} (i.e. $S'_{1X} = \dfrac{1}{1 - z^{-1}}$) we see that:-

$$s_{1X} = S'_{1X}(-d_{OA}TI_{1X} + d_{2X}L_{1X})$$

$$= d_{OA}S'_{1X}(-TI_{1X} + T(d_{OA}, d_{2X})L_{1X})$$

so by substitution:-

$$T(d_{OA}, s_{1X}) = \frac{s_{1X}}{d_{OA}} = \frac{-Tz^{-1} + (1 - z^{-2} + T)z^{-\lambda}}{1 - z^{-1}} \qquad (7.6.3)$$

This has characteristic equation $1 - z^{-1} = 0$ which has roots on the unit circle. Thus stock balance has only limiting stability. This is unavoidable since we cannot use feedback to control stocks. Noise introduced anywhere but a causal route from d_{OA} to d_{1A} will result in a permanent modification to stock balance. Only by using stock balances in deriving production schedules can this be avoided; this is precluded and within this constraint we can hope for no better than limiting stability.

We have now established that our system is as stable as it can be and that the response of production schedules for part X are reasonable. We could however, proceed little further without some knowledge of the assembly scheduling z-transfer function, $T = \zeta(d_{0A}, d_{1A})$. As noted above it was politically impractical to carry out any detailed investigation into this sub-system whose behaviour could only be predicted on the basis of its historical output.

A detailed, though admittedly largely subjective, comparison of model demands and assembly schedules over the range of parts concerned suggested that T could be regarded as composed of three parts:-

(1) A scalar multiplication of model demand. This factor was generally less than one since there was a tradition of loading the factory beyond its capacity in the fond belief that this increased achieved performance!

(2) A delay of two periods inherent in the assembly scheduling procedure.

(3) An element which could only be interpreted as random noise generated by the scheduling procedures to facilitate assembly.

No claim is made that it was technically impossible to achieve far deeper understanding of assembly scheduling. It was however precluded politically. This and the need for rapid provision of results justified the use of such a simplified output analysis of T.

Such an estimate was sufficient in practice to allow us to proceed considerably further in predicting the system's responses. We looked at the system in relation to a single part, for the same analysis could be extended to all parts in the controlled range. We examined deterministic response to standard model demand patterns, the buffer stocks which were necessary to support the system and the response to the stochastic component of T which gave rise to a need for safety stocks.

The actual numerical values concerned are unimportant since, in dealing with linear systems, we can scale the results as we please. The form of the noise distribution component of T is important since the shape of the distribution is passed on distorted by convolution through the system. We considered a part whose mean demand was 10 per period and applied deterministic analyses from this base. The first scalar multiplication component of T was taken to be 0.9, whilst the third component is shown in the table (Figure 7.6.b).

NOISE IMPULSE	<-3	-3	-2	-1	0	1	2	>2
PROBABILITY	0	.05	.10	.20	.25	.29	.11	0

Figure 7.6.b. Distribution of noise components of T.

7.7 Response to Standard Deterministic Demand Patterns

Demand Impulse

Substituting $T = 0.9z^{-2}$ (its deterministic component) in 7.6.3 we have:-

$$T(d_{0A}, s_{1X}) = \frac{-Tz^{-1} + (1 - z^{-2} + T)z^{-\lambda}}{1 - z^{-1}}$$

$$= \frac{-0.9z^{-3} + z^{-\lambda}(1 - z^{-2} + 0.9z^{-2})}{1 - z^{-1}}$$

$$= \frac{-0.9z^{-3} + z^{-\lambda} - 0.1z^{-\lambda-2}}{1 - z^{-1}}$$

$$= (-0.9z^{-3} + z^{-\lambda} - 0.1z^{-\lambda-2})(1 + z^{-1} + z^{-2} + ...)$$

We can interpret this, term by term, as follows:-

$z^{-\lambda}$ represents the arrival of parts as a result of the production schedule raised when the demand impulse occurs. Delivery occurs one lead-time later.

$-0.9z^{-3}$ represents the use of the part in response to assembly schedules.

$0.1z^{-\lambda-2}$ represents the arrival of a correction schedule raised immediately the assembly order is known and delivered λ periods later.

$(1 - z^{-1})^{-1}$ is stock integration. It perpetuates the effect of each of the other terms and it is clear that after $\lambda + 2$ periods the three perpetuated terms have coefficients totalling zero and stock returns to rest.

Alternatively we can multiply out the expression above and plot coefficients of powers of z^{-1} to represent the variations of stock with time.

$$T(d_{0A},s_{1X}) = -0.9z^{-3} - 0.9z^{-4} - 0.9z^{-5} - 0.9z^{-6} - \ldots\ldots$$

$$+ z^{-\lambda} + z^{-\lambda-1} + z^{-\lambda-2} + z^{-\lambda-3} + z^{-\lambda-4} + \ldots\ldots$$

$$- 0.1z^{-\lambda-2} - 0.1z^{-\lambda-3} - 0.1z^{-\lambda-4} - \ldots\ldots$$

$$= -0.9z^{-3} - 0.9z^{-4} - 0.9z^{-5} - 0.9z^{-6} - \ldots\ldots$$

$$+ z^{-\lambda} + z^{-\lambda-1} + 0.9z^{-\lambda-2} + 0.9z^{-\lambda-3} + \ldots\ldots$$

So taking $\lambda = 2$ periods we have:-

$$T(d_{0A},s_{1X}) = z^{-2} + 0.1z^{-3} \qquad (7.7.1)$$

From this we can plot Figure 7.7.a.

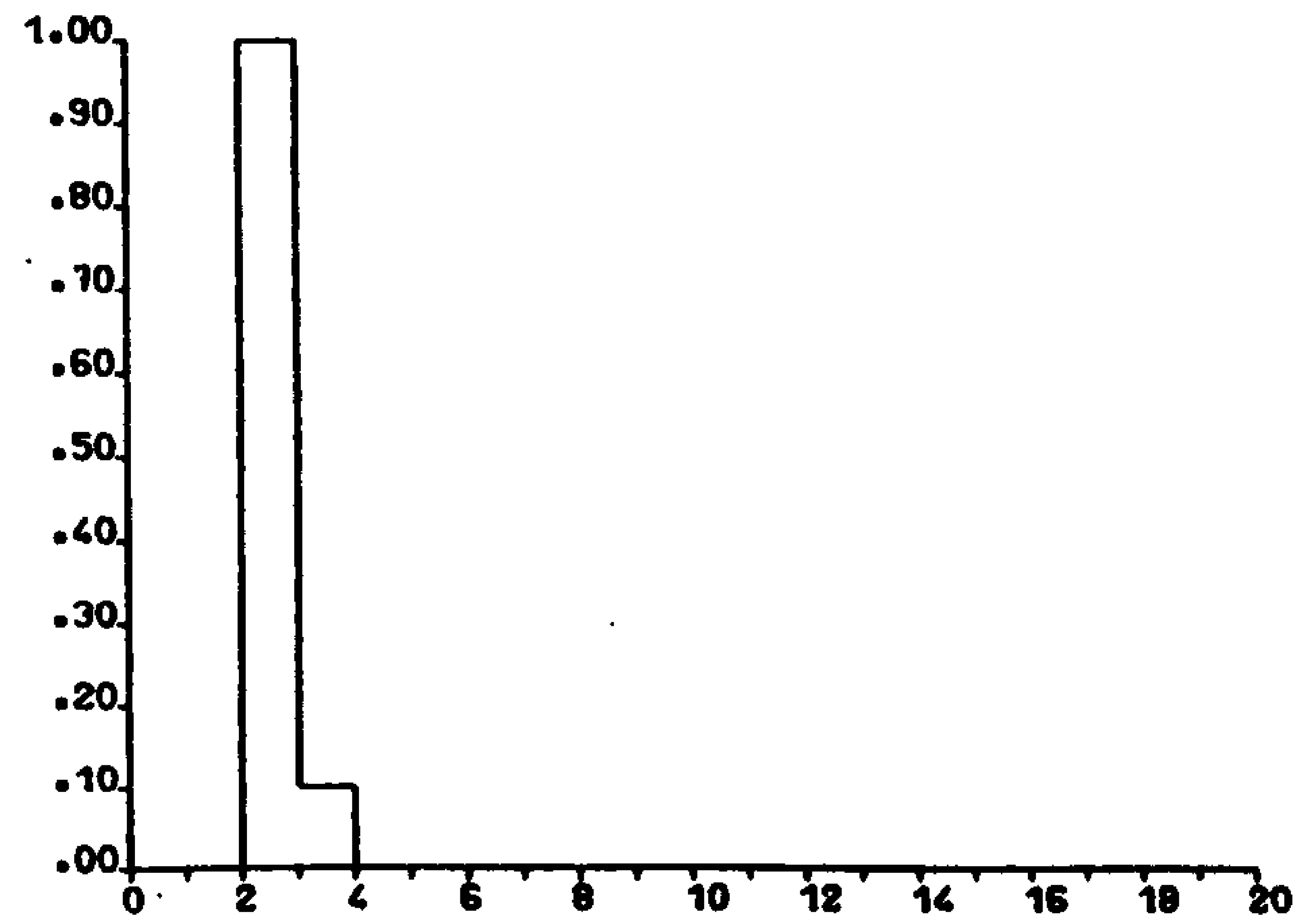

Figure 7.7.a : Stock Response To Impulse

To examine response of production schedules to a unit impulse we have, from 7.6.2:-

$$T(d_{0A},d_{2X}) = 1 - z^{-2} + T$$

$$= 1 - z^{-2} + 0.9z^{-2}$$

$$= 1 - 0.1z^{-2}$$

This is an immediate schedule in response to demand followed by a correction when the assembly schedule is available (Figure 7.7.b).

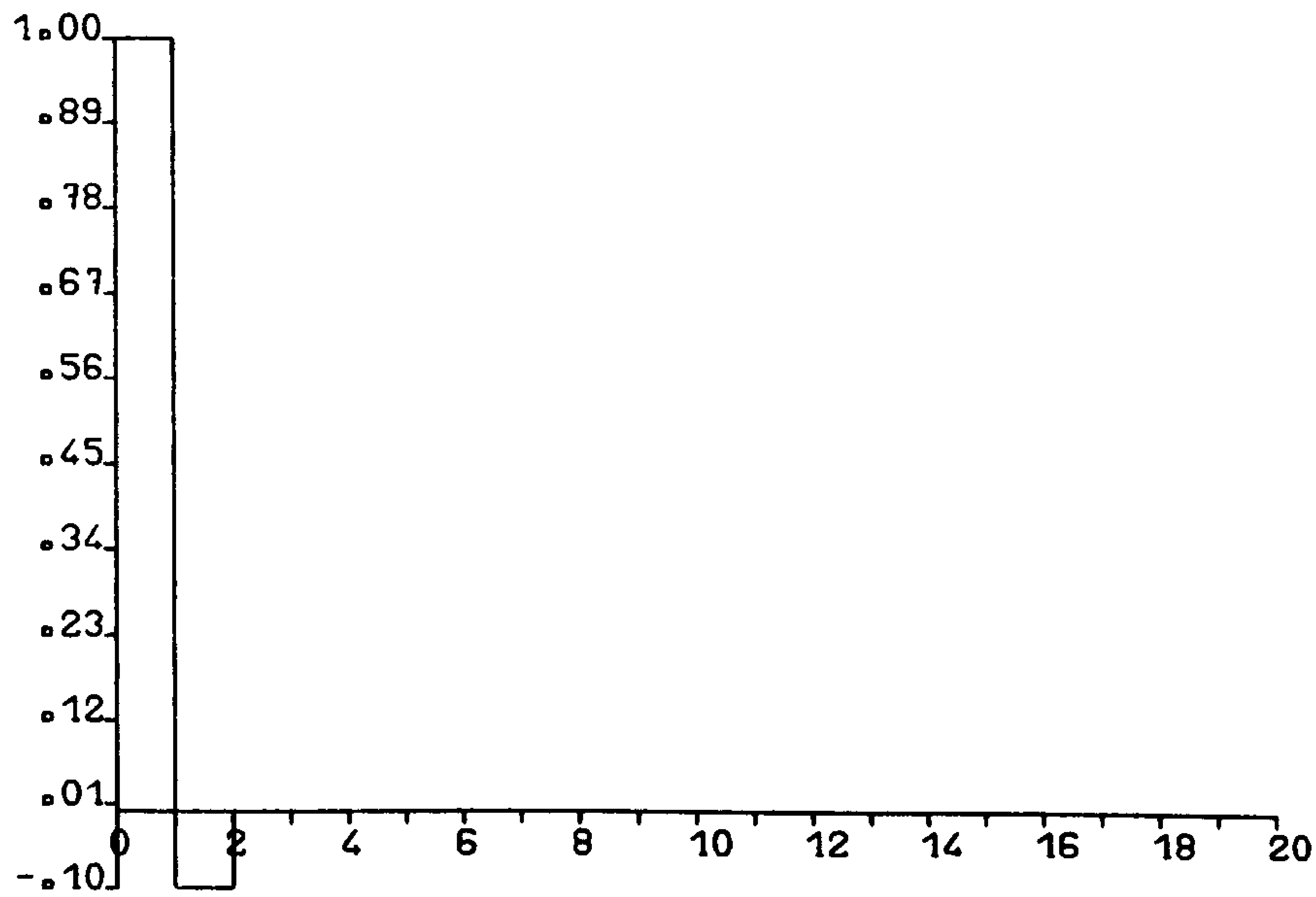

Figure 7.7.b : Schedule Response To Impulse

Demand Step

Stock response to a step is obtained by multiplying $T(d_{OA}, s_{1X})$ by the z-transform of a unit step, $(1 - z^{-1})^{-1}$, so:-

$$T(d_{OA}, s_{1X})(1 - z^{-1})^{-1} = \frac{-Tz^{-1} + z^{-\lambda}(1 - z^{-2} + T)}{(1 - z^{-1})^2}$$

Thus taking $T = 0.9z^{-2}$ and $\lambda = 2$, step response is (as in 7.7.1):

$$(z^{-2} + 0.1z^{-3})(1 - z^{-1})^{-1}$$

$$= z^{-2} + 1.1z^{-3} + 1.1z^{-4} + \ldots.. \qquad (7.7.2)$$

which is represented by Figure 7.7.c.

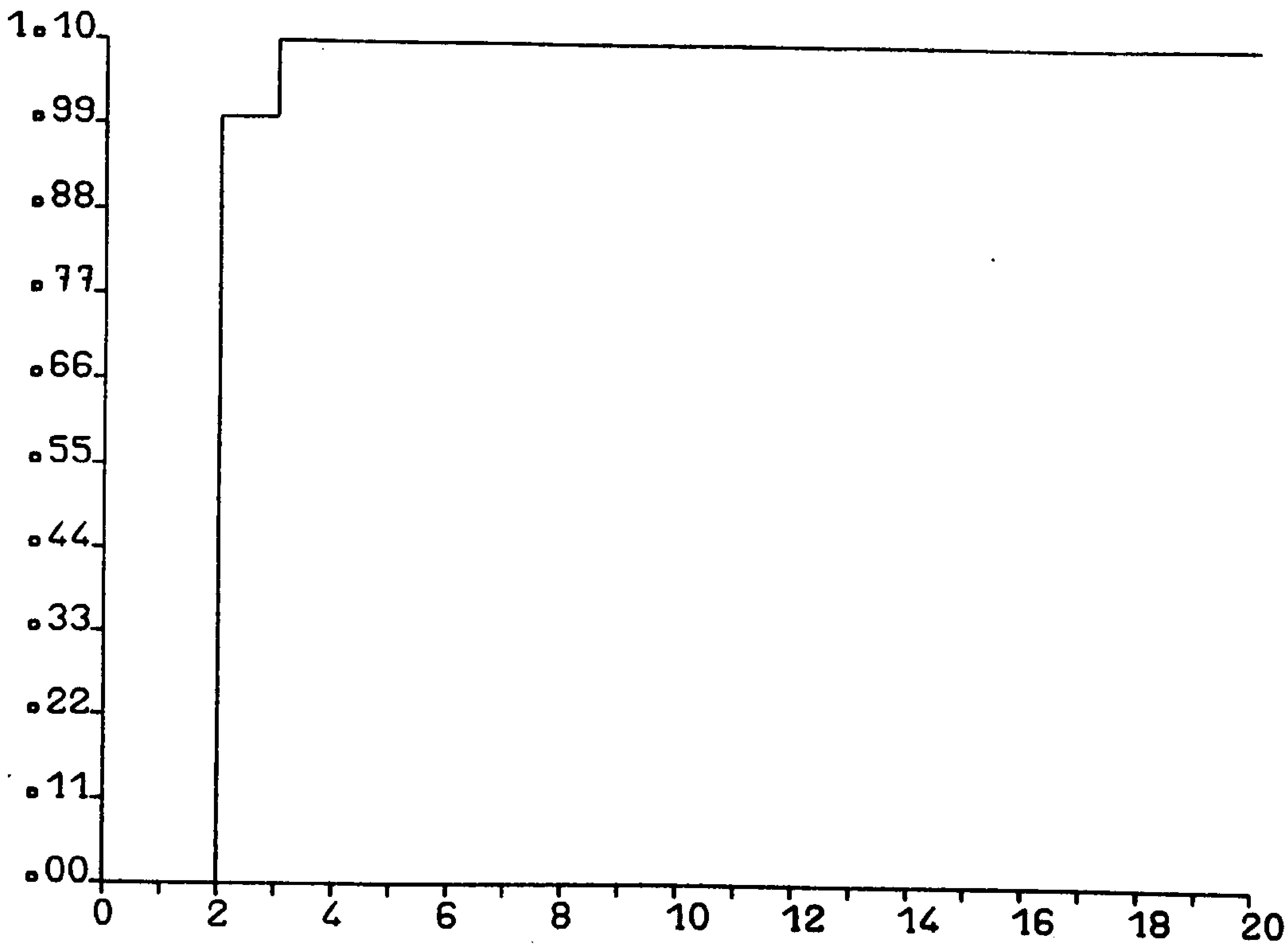

Figure 7.7.c : Stock Response To Step

Similarly from 7.6.2, and substituting $T = 0.9z^{-2}$, response of production schedules to step demand is:-

$$T(d_{OA},d_{2X})(1 - z^{-1})^{-1} = (1 - 0.1z^{-2})(1 - z^{-1})^{-1}$$

$$= 1 + z^{-1} + 0.9z^{-2} + 0.9z^{-3} + \ldots\ldots$$

(Figure 7.7.d)

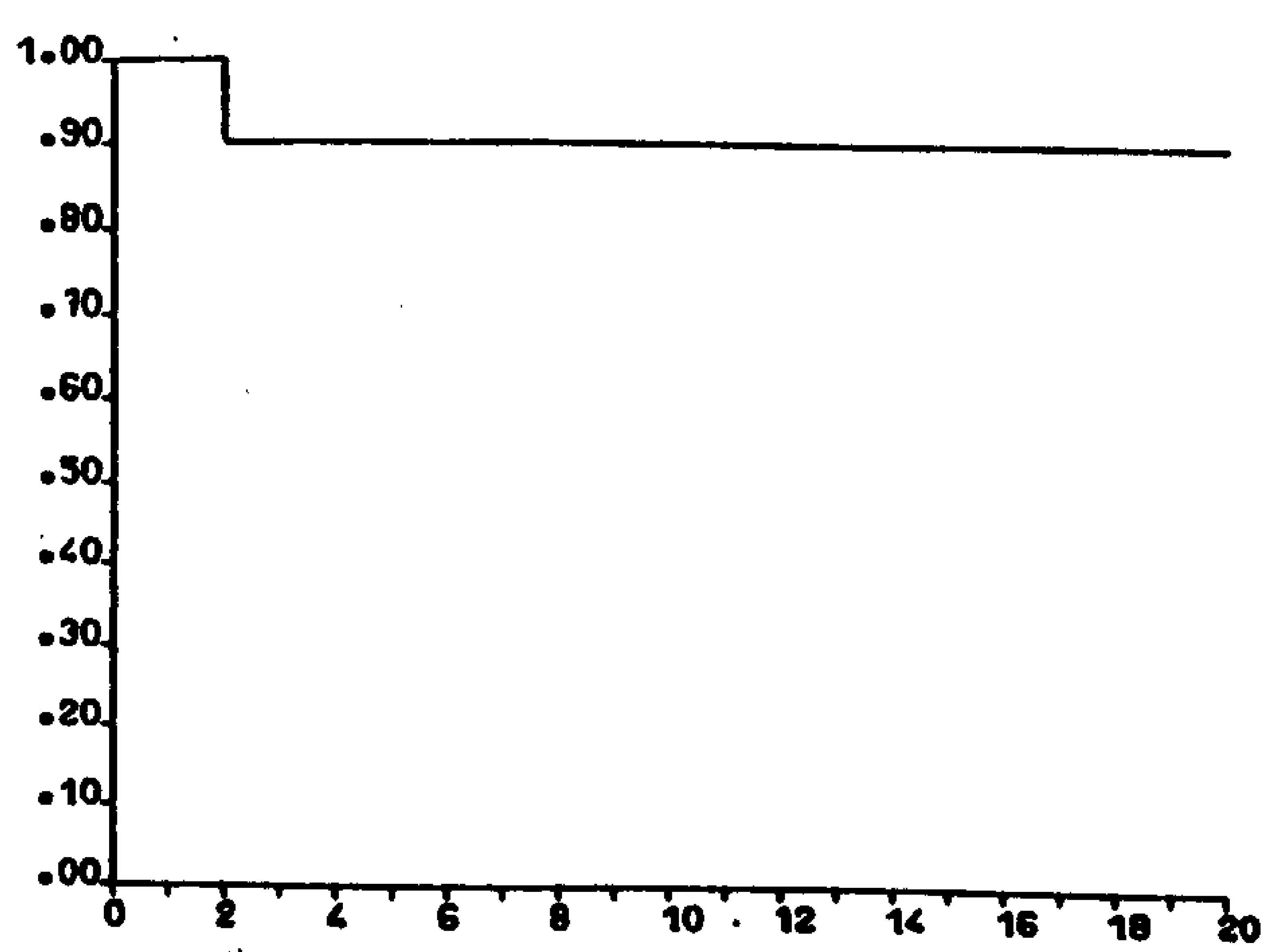

Figure 7.7.d : Schedule Response To Step

Thus we see that after generating an initial overorder for two periods the system continues to order correctly. However, because stock is not fed back, no compensation for the overorder ever occurs.

Demand Ramp

A ramp has z-transform $\frac{z^{-1}}{(1 - z^{-1})^2}$ so the stock response to a ramp is, from 7.6.2 and substituting $T = 0.9z^{-2}$ and $\lambda = 2$:

$$z^{-1}(z^{-2} + 1.1z^{-3} + 1.1z^{-4} + \ldots)(1 - z^{-1})^{-1}$$

$$= z^{-3} + 2.1z^{-4} + 3.2z^{-5} + 4.3z^{-6} + \ldots..$$

which is illustrated (Figure 7.7.e).

Similarly we derive production schedule response as:-

$$z^{-1} + 2z^{-2} + 2.9z^{-3} + 3.8z^{-4} + 4.7z^{-5} + \ldots..$$

which is represented by Figure 7.7.f.

Sinusoidal Demand

Using the computer program of Appendix I and substituting $T = 0.9z^{-2}$ and $\lambda = 2$ we derived the stock and production schedule responses to sinusoidal demand as shown in Figures 7.7.g and 7.7.h.

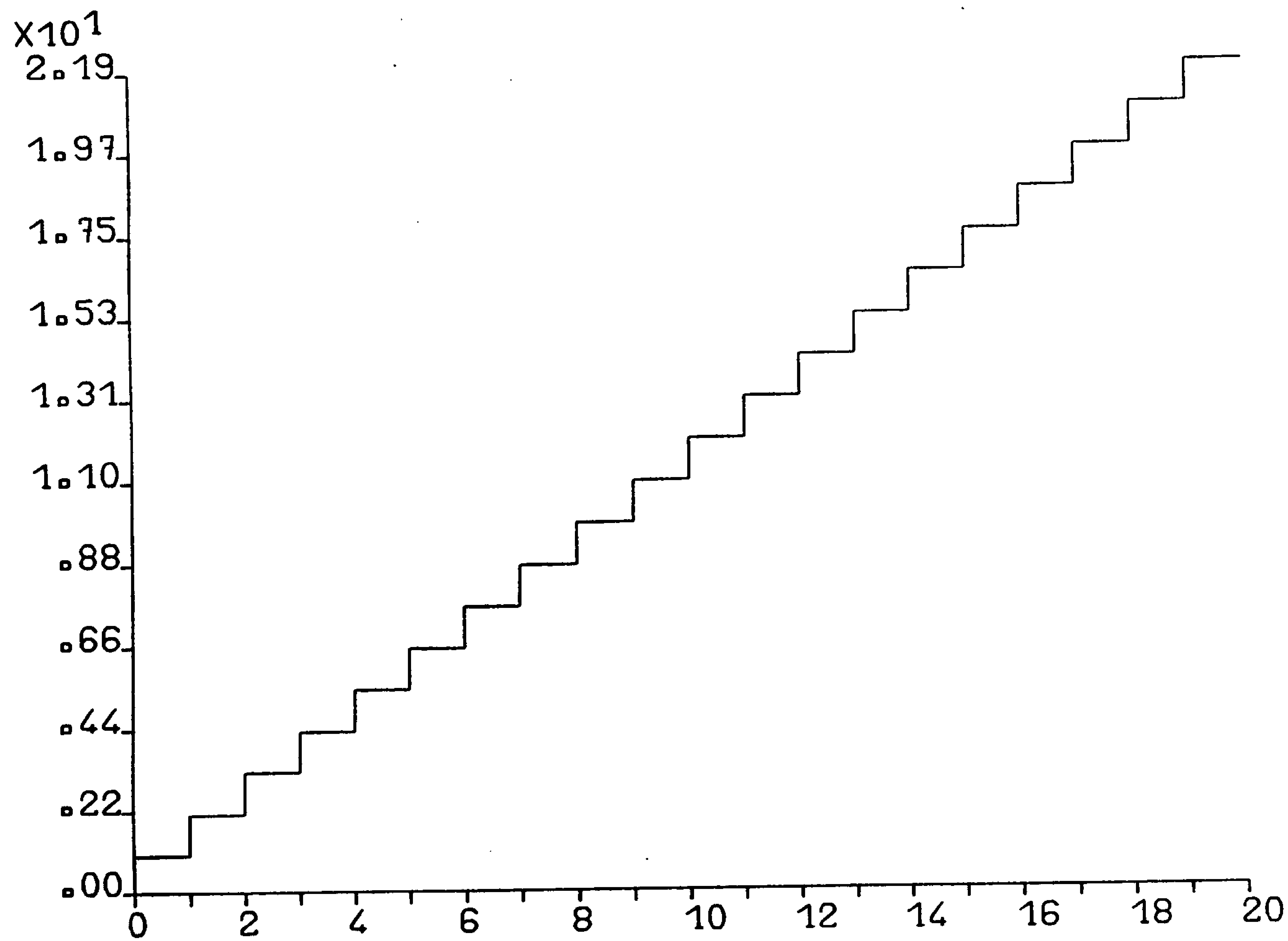

Figure 7.7.e : Stock Response To Ramp

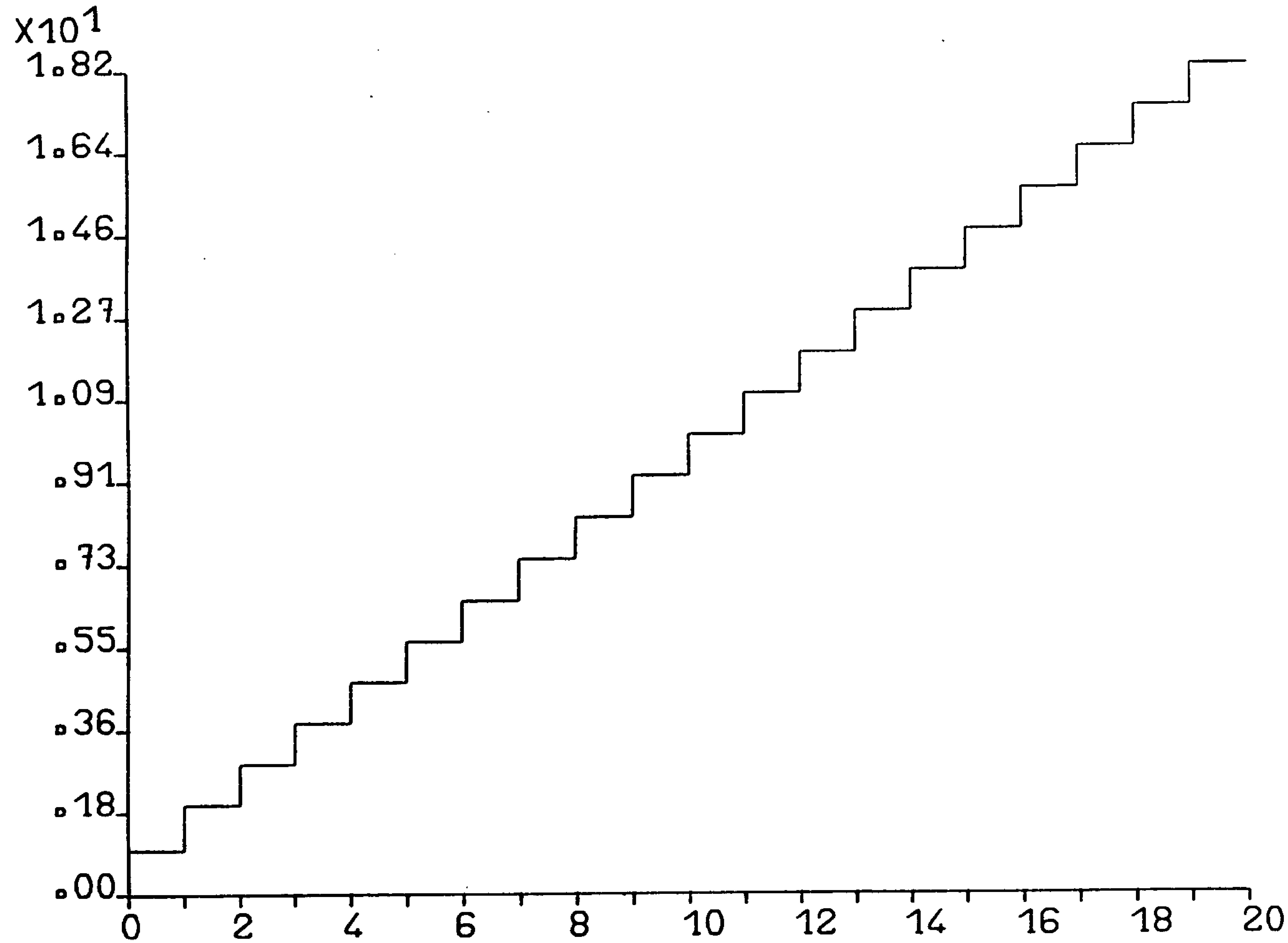

Figure 7.7.f : Schedule Response To Ramp

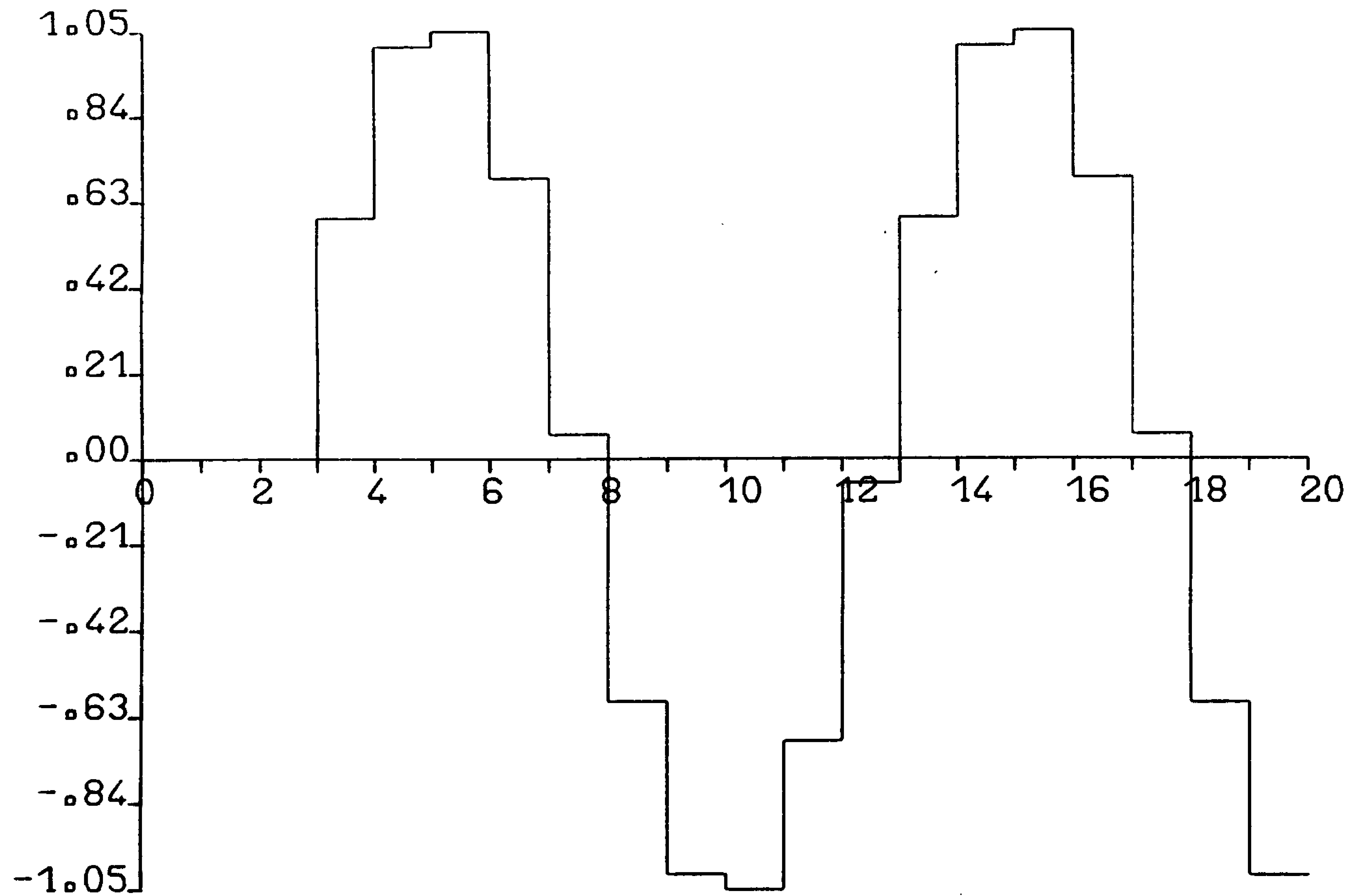

Figure 7.7.g : Stock Response To Sine

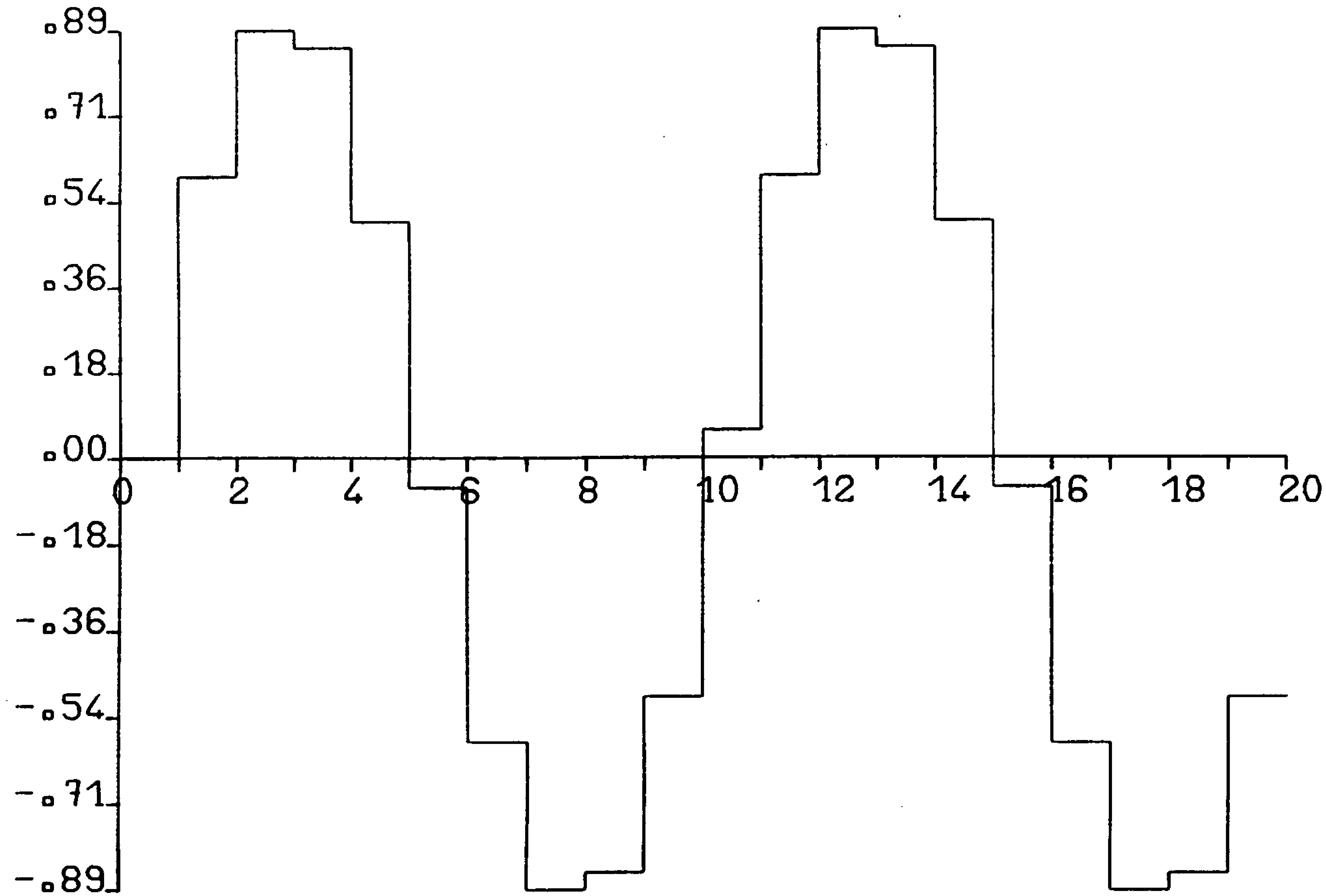

Figure 7.7.h : Schedule Response To Sine

7.8 Response to the Stochastic Component of Assembly Scheduling

It is a simple matter to derive $T(\tilde{d}_{1A}, s_{1X})$ to be:-

$$T(\tilde{d}_{1A}, s_{1X}) = \frac{z^{-\lambda} - z^{-1}}{1 - z^{-1}}$$

$$= - z^{-1} - z^{-2} - \ldots - z^{-\lambda+1}$$

Thus stock balance has a distribution which is the convolution of $\lambda - 1$ copies of the assembly scheduling distribution. For $\lambda = 2$ this is simply a copy.

When we look at $T(\tilde{d}_{1A}, d_{2X})$ we find it is 1. Thus noise is passed unmodified to the supplier.

7.9 Response to Random Elements of Model Demand

For most parts we found that their commonality to a wide range of models resulted in a very steady demand pattern. A mean demand rate was maintained and the period fluctuations were treated as random noise.

We derived a distribution of model demand for part X whose mean periodic total demand was 10. This was estimated in much the same way as the noise component of assembly scheduling, for similar reasons (Section 7.6).

This distribution is illustrated in Figure 7.9.a.

NOISE IMPULSE	<-2	-2	-1	0	1	2	>2
PROBABILITY	0	0.10	0.25	0.30	0.25	0.10	0

Figure 7.9.a. Distribution of noise component of model demand for part X.

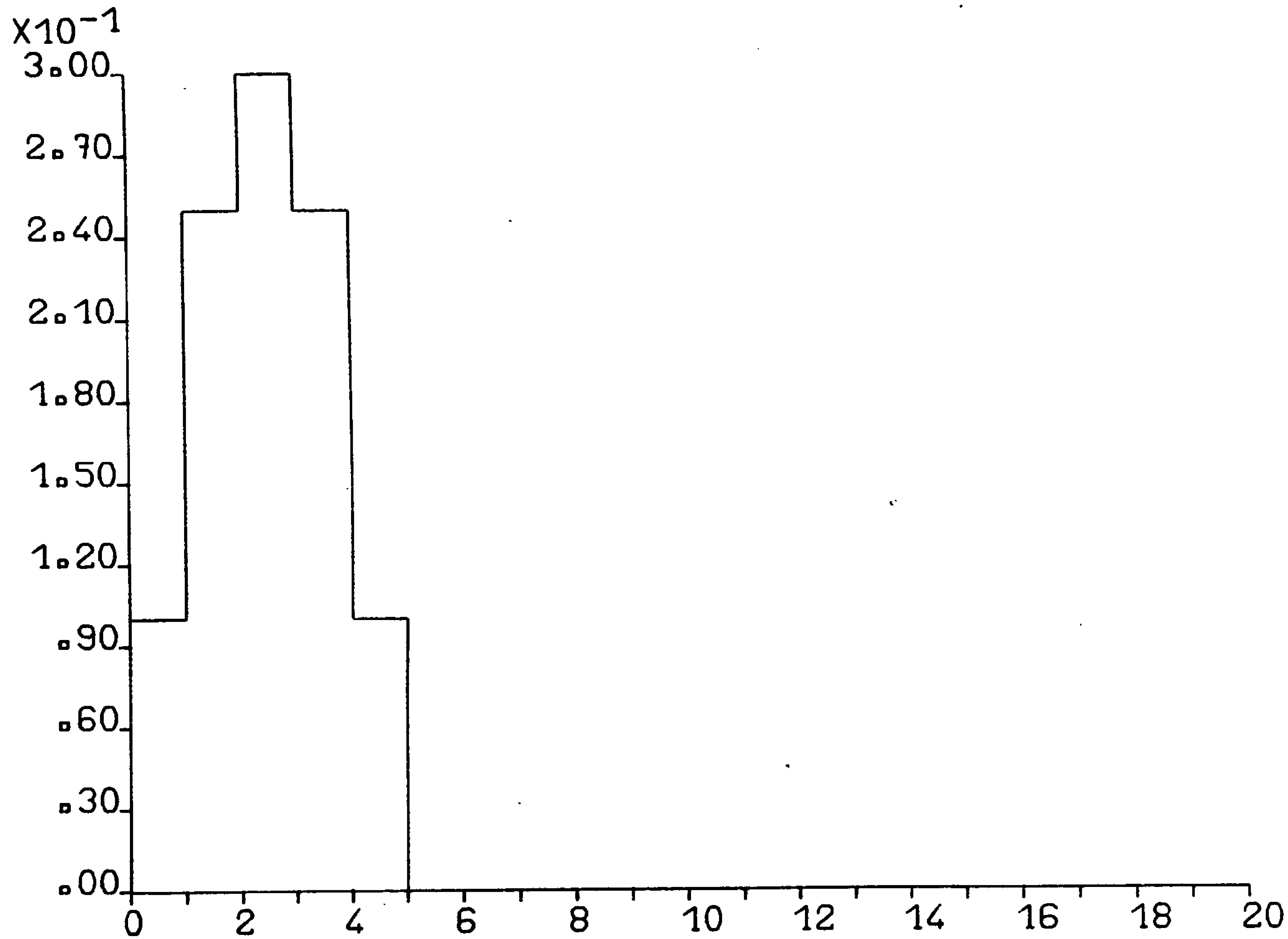

Figure 7.9.b : Schedule Response To Noise

From 7.6.3 we knew that stock would perform a random walk in response to demand noise.

From 7.6.2 and substituting $T = 0.9z^{-2}$ we knew that:-

$$T(d_{OA}, d_{2X}) = 1 - 0.1z^{-2}$$

so that the response of production schedules for part X to such noise is the convolution of the distribution of 7.9.a and a version of itself where the impulses are scaled −0.1. Using the program of Appendix I we saw the production schedule distribution to be as in Figure 7.9.b.

7.10 Conclusions

The above analysis achieved, in a practical sense, two things. Firstly we were guided away from proposed scheduling systems which contained unnecessary inherent failings; secondly we demonstrated that a system where stock balances are not fed back cannot achieve full stability. We note that either of these could have been achieved by other means but the z-transform techniques provided a framework approach to the problems which could be applied very rapidly during the course of a few hours discussions.

The result of the whole exercise was that the final proposal was implemented. The instability in stock balance was noted and to retain control a routine of occasional stock reconciliation was superimposed as a temporary measure.

The procedure operated as described for a year by which time it was possible to start implementing a more satisfactory solution. As a result of deeper understanding we were able to promote organisational changes making possible full control of stocks and a recording system adequate to allow development to include feedback loops from stock balance.

CHAPTER 8

CONCLUSIONS

8.1 Application

The range of applications of discrete linear control theory is limited to those systems where the constraints of linearity and discreteness are satisfied. A great many production and inventory control systems, particularly the larger, more complex ones, are discrete. A significant range of such systems can be represented by a linear model, either because they are entirely linear, or because the model's behaviour beyond the limits of system linearity is open to sensible interpretation.

It is possible to construct a general model representative of a broad range of the multi-product, multi level production and inventory control systems which are in use in manufacturing industry. This general model is the composition of models of subsystems each of which controls a single part or product. The composition is finite and well defined (4.10.1). The subsystem models can also be decomposed to represent the individual elements of the part control system. A range of such elements has been derived and the process of derivation has been demonstrated. Any discrete linear element can be modelled thus.

8.2 Results

We have demonstrated, in some detail, the use of discrete linear control models. Results in terms of system stability, response to standard inputs, and response to imposed random noise are readily obtained. These are of use in system selection, in selection of system parameters (e.g. smoothing constants) and in predicting safety stock requirements.

The construction of a model is essentially a process of drawing block diagrams representing information flow. This process itself facilitates understanding of the system and can frequently enhance communication between the modeller and managers responsible for system operation. This advantage is shared by simulation techniques.

The arithmetic involved in modelling can, when necessary, be assisted by computer. The model is easily amended to represent a modified system and the necessary re-computation is restricted to the scope of the system changes. The computer model can be amended interactively (in response to its own output, perhaps), thus allowing very rapid system design. The technique also lends itself to the generation of easily assimilated graphical results.

It is often easy both to determine stability and to interpret z-transforms as time-series by inspection. In such cases computer assistance may be unnecessary and a useful manual analysis can be accomplished rapidly. The very process of modelling may suggest improvement of the modelled systems (Chapter 7).

8.3 Implementation

The use of discrete linear control theory described in Chapter 7 was successful within the constraints placed upon study at that time. Major modernisation of both plant and control systems is now proceeding and the technique continues to be employed in system analysis and design.

8.4 Possible Research Applications

Discrete linear control theory may prove to be a useful tool in the following areas:-

8.4.1: The study of the effects of misinformation (particularly partly systematic misinformation) upon complex systems.

8.4.2: The study of the effects of using differing planning horizons within complex systems.

8.5 Possible Extensions of the Modelling Technique

8.5.1: It may be possible to model certain classes of non-linear systems (e.g. piecewise linear systems) using techniques based upon discrete linear control theory.

8.5.2: Use of the "modified z-transform" [Truxal, 1955] would allow analysis of a model's behaviour between period ends.

8.5.3: Use of array-processing techniques may increase the capacity and versatility of the possible computer models: we may model larger systems and process faster in this way; the model's operating characteristics may allow greater flexibility in running.

8.6 Summary

Discrete linear control theory is applicable in modelling a significant range of multi-product, multi-level production and inventory control systems. Models so derived can be and have been of value in analysis and design of practical working systems, and may also be used in theoretical studies. Where applicable the technique is worthy of consideration alongside other methods in modelling such systems.

APPENDIX I

INTERACTIVE PROGRAM

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                        C
C      THIS PROGRAM HAS BEEN WRITTEN IN FORTRAN IV FOR PORTABILITY.      C
C                                                                        C
C      THE INPUT AND OUTPUT CHANNELS ARE SET AT THE START OF THE PROGRAM.C
C      ALL THE POLYNOMIAL ARRAYS ARE DECLARED IN THE MAIN SEGMENT AND THEY AND C
C      THEIR SIZES ARE PASSED TO THE SUBROUTINES AS PARAMETERS SO THAT IF IT C
C      IS WISHED TO CHANGE THE SIZE OF THE POLYNOMIAL ARRAYS IT IS ONLY  C
C      NECESSARY TO CHANGE CODE AT THE START OF PROGRAM                  C
C                                                                        C
C      CHARACTER HANDLING USES A1 FORMAT. TWO ROUTINES ARE SUPPLIED AT THE C
C      END OF THE PROGRAM, ONE FOR COPYING A CHARACTER FROM ONE LOCATION TO C
C      ANOTHER AND THE OTHER FOR COMPARING TWO CHARACTERS FOR EQUALITY (THESE C
C      ARE MACHINE DEPENDENT).                                           C
C                                                                        C
C      PLOTTING USES THE GINO-F LIBRARY. BEFORE THE FIRST PICTURE IS DRAWN C
C      A CALL IS MADE TO ROUTINE PIC1ST TO INITIALISE THE PLOTTING DEVICE, C
C      THIS IS DEVICE DEPENDENT AND AN APPROPRIATE SUBROUTINE PIC1ST SHOULD C
C      BE SUBSTITUTED BY THE IMPLEMENTING SITE. CALLS TO GINO-F ROUTINES TO C
C      PLOT THE GRAPHS ARE MADE FROM SUBROUTINE SQUIGL THEY SHOULD NOT NEED C
C      TO BE ALTERED. IF PLOTTING HAS BEEN REQUESTED DURING THE RUNNING OF C
C      THE PROGRAM SUBROUTINE DEVEND IS CALLED BEFORE EXITING, THIS OCCURS C
C      IN SUBROUTINE REPLY FOR EXIT ON RECEIPT OF AN 'END' COMMAND AND IN C
C      SUBROUTINE ERSTOP FOR EXIT AFTER A FATAL ERROR.                   C
C                                                                        C
C      THE OUTPUT FORMATS CONTAIN 2 CHARACTERS WHICH ARE NOT STANDARD FORTRAN C
C      THEY ARE ? (QUESTION MARK) AND : (COLON), THESE SHOULD BE REPLACED IF C
C      THEY CAUSE ANY INCONVENIENCE.                                     C
C                                                                        C
C      THE PROGRAM IS DESIGNED TO RUN INTERACTIVELY, HOWEVER IT MAY BE RUN C
C      OFFLINE IF THE FIRST DATA CARD CONTAINS THE WORD 'NO'.            C
C      WHEN RUNNING INTERACTIVELY PROMPTS WILL BE GIVEN WHENEVER NECESSARY. C
C      THE COMMAND 'END' WILL STOP THE PROGRAM AT ANY TIME.              C
C      SPACE OR NEWLINE ACT AS SEPARATORS FOR COMMANDS AND PARAMETERS.   C
C      ALL COMMANDS AND ANSWERS MAY BE ABBREVIATED TO THE FIRST 3 CHARACTERS. C
C                                                                        C
C      ALL POLYNOMIALS ARE REFERENCED BY NAME AND NAMES MAY BE UP TO 12  C
C      CHARACTERS IN LENGTH AND CAN INCLUDE ANY CHARACTERS EXCEPT SPACES. C
C      A GRAPH IS GIVEN A TITLE WHICH MUST BE 20 CHARACTERS IN LENGTH, SPACES C
C      ARE ALLOWED IN THE TITLE.                                         C
C      THE REPLY TO ANY QUESTION STARTING 'HOW MANY POLYNOMIALS' MUST BE AN C
C      UNSIGNED INTEGER (ZERO IS ALLOWED).                               C
C      OTHER NUMBERS AS PARAMETERS ARE CONSIDERED AS REALS. THEY MAY BE  C
C      SIGNED OR UNSIGNED AND MAY OR MAY NOT INCLUDE A DECIMAL POINT BUT C
C      MAY NOT CONTAIN AN EXPONENT.                                      C
C                                                                        C
C      THE FOLLOWING DATA IS REQUIRED :                                  C
C         INTERACTIVE RUNNING? (ANSWER - YES OR NO)                      C
C         HOW MANY PLOYNOMIALS ARE ABOUT TO BE ENTERED?                  C
C         SOURCE OF POLYNOMIALS? (ANSWER - FILE OR STREAM)               C
C           NAME OF POLYNOMIAL          )                                C
C           HASH-TOTAL                  )  SUPPLIED FOR EACH POLYNOMIAL  C
C           COEFFICIENTS                )  ABOUT TO BE ENTERED           C
C         COMMANDS TO MANIPULATE THE POLYNOMIALS                         C
C                                                                        C
```

```
C*************************************************************************C
C*                                                                        *C
C*    THE FOLLOWING COMMANDS ARE AVAILABLE                                *C
C*                                                                        *C
C*    COMMAND   PARAMETERS                             PURPOSE            *C
C*                                                                        *C
C*    ADD       1. NAME OF 1ST POLY TO BE ADDED        TO ADD TWO POLYNOMIALS *C
C*              2. NAME OF 2ND POLY TO BE ADDED                           *C
C*              3. NAME OF POLY TO HOLD RESULT                            *C
C*                                                                        *C
C*    DIVIDE    1. NAME OF POLY TO BE DIVIDED          TO DIVIDE A POLYNOMIAL *C
C*              2. POWER OF Z**(-1) BY WHICH TO        BY Z**(-N)         *C
C*                 DIVIDE                                                 *C
C*              3. NAME OF POLY TO HOLD RESULT                            *C
C*                                                                        *C
C*    DRAW      1. NUMBER OF POLYS ON GRAPH (MAX 3)    TO DRAW A GRAPH OF (UP *C
C*              2.    )                                TO) THREE POLYNOMIALS *C
C*              .     ) NAMES OF POLYS TO BE DRAWN                        *C
C*              N+1.  )                                                   *C
C*              N+2. TITLE OF GRAPH (20 CHARACTERS)                       *C
C*                                                                        *C
C*    DUMP                                             TO DUMP A COPY OF ALL *C
C*                                                     POLYNOMIALS CURRENTLY *C
C*                                                     IN STORE TO FILE   *C
C*                                                                        *C
C*    END                                              TO END PROCESSING (CAN *C
C*                                                     BE USED AT ANY TIME) *C
C*                                                                        *C
C*    INPUT     1. NUMBER OF POLYS ABOUT TO BE INPUT   TO READ A BLOCK OF *C
C*              2. SOURCE OF POLYS (FILE OR STREAM)    POLYNOMIALS FROM FILE *C
C*              3.    ) NAME, HASH TOTAL               OR STREAM          *C
C*              .     ) AND COEFFICIENTS                                  *C
C*              N+2.  ) FOR EACH POLY                                     *C
C*                                                                        *C
C*    INVERT    1. NAME OF POLY TO BE INVERTED         TO INVERT A POLYNOMIAL *C
C*              2. NAME OF POLY TO HOLD RESULT                            *C
C*                                                                        *C
C*    MULTIPLY  1. NAME OF 1ST POLY TO BE MULTIPLIED   TO MULTIPLY TWO    *C
C*              2. NAME OF 2ND POLY TO BE MULTIPLIED   POLYNOMIALS        *C
C*              3. NAME OF POLY TO HOLD RESULT                            *C
C*                                                                        *C
C*    NSCALE    1. NAME OF NOISE POLY TO BE SCALED     TO SCALE THE RANDOM *C
C*              2. SCALAR WITH WHICH TO MULTIPLY       VARIABLE OF A NOISE *C
C*              3. NAME OF POLY TO HOLD RESULT         DISTRIBUTION POLYNOMAL *C
C*                                                                        *C
C*    NTRANS    1. NAME OF NOISE POLY TO BE            TO TRANSFORM A NOISE *C
C*                 TRANSFORMED                         NOISE DISTRIBUTION *C
C*              2. NAME OF Z-TRANSFER POLY             POLYNOMIAL BY A    *C
C*              3. NAME OF POLY TO HOLD RESULT         SYSTEM POLYNOMIAL  *C
C*                                                                        *C
C*    PRINT     1. NAME OF POLY TO BE PRINTED          TO LIST A PLOYNOMIAL *C
C*                                                                        *C
C*    SCALE     1. NAME OF POLY TO BE SCALED           TO MULTIPLY A      *C
C*              2. SCALAR WITH WHICH TO MULTIPLY       POLYNOMIAL BY A SCALAR *C
C*              3. NAME OF POLY TO HOLD RESULT                            *C
C*                                                                        *C
C*    TIDY      1. NUMBER OF POLYS TO BE DELETED       TO LIST NAMES OF ALL *C
C*              2.    )                                POLYNOMIALS IN STORE *C
C*              .     ) NAMES OF POLYS TO BE DELETED   AND DELETE THOSE NO *C
C*              N+1.  )                                LONGER REQUIRED    *C
C*                                                                        *C
C*************************************************************************C

C                                                                          C
C     SET INITIAL VALUES :                                                 C
C            ICHAN - STANDARD INPUT CHANNEL                                C
C            JCHAN - STANDARD OUTPUT CHANNEL                               C
C            KCHAN - POLYNOMIALS' INPUT CHANNEL (WHEN FROM FILE)           C
C            LCHAN - OUTPUT CHANNEL FOR DUMP OF POLYNOMIALS                C
C            IDIM  - MAXIMUM PERMISSIBLE NUMBER OF POLYNOMIALS             C
C            JDIM  - MAXIMUM PERMISSIBLE NUMBER OF COEFFICIENTS            C
C                                                                          C
C     SET DIMENSIONS OF POLYNOMIAL ARRAYS :                                C
C            NAMEPY(12,IDIM)                                               C
C            COEFPY(JDIM,IDIM+4)                                           C
C            HASHPY(IDIM+4)                                                C
C            X(JDIM+2)                                                     C
C            Y(JDIM+2)                                                     C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION NAMEPY(12,40),COEFPY(20,44),NASHPY(44),X(40),Y(40)
      COMMON / INFO / IBUF(30),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
      ICHAN=5
      JCHAN=6
      KCHAN=15
      LCHAN=16
      IDIM=40
      JDIM=20
      JDIM2=JDIM*2
      IDIM4=IDIM+4
      NUMBPY=0
      NEWREC=.TRUE.
      DRAW=.FALSE.
      DO 3 I=1,IDIM
      DO 1 K=1,12
      CALL COPY1(NAMEPY(K,I),ISPACE)
    1 CONTINUE
      DO 2 J=1,JDIM
      COEFPY(J,I)=0.0
    2 CONTINUE
      NASHPY(I)=0.0
    3 CONTINUE
      KOUNT=0
      ONLINE=.TRUE.
      IFORM=0
    4 NREPLY=0
C     INTERACTIVE RUNNING?
      CALL INSTIN(NREPLY)
      IF (NREPLY .GT. 0 .AND. NREPLY .LT. 3) GOTO 5
      KOUNT=KOUNT+1
      IF (KOUNT .EQ. 3) CALL ERSTOP(1)
      NEWREC=.TRUE.
      IFORM=5
      GOTO 4
    5 IF (NREPLY .EQ. 2) ONLINE = .FALSE.
      CALL GULP(COEFPY,NASHPY,NAMEPY,IDIM,IDIM4,JDIM,KCHAN)
    6 IFORM=19
    7 NREPLY=1
      CALL INSTIN(NREPLY)
      N1=NREPLY+1
      GOTO (20,20,20,20,8,9,10,11,12,13,14,15,20,20,16,17,18,19),N1

C ... COMMAND - SCALE
    8 IFORM=21
      NEWOLD=1
      CALL NAMEIN(M,NAMEPY,IDIM)
      IFORM=22
      SCAL=REALIN(N)
      IFORM=20
      NEWOLD=0
      CALL NAMEIN(N,NAMEPY,IDIM)
      IF (NEWOLD .EQ. (-2)) GOTO 6
      CALL SCAMLT(SCAL,M,N,COEFPY,NASHPY,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - MULTIPLY
    9 IFORM=23
      NEWOLD=1
      CALL NAMEIN(L,NAMEPY,IDIM)
      IFORM=24
      CALL NAMEIN(M,NAMEPY,IDIM)
      IFORM=20
      NEWOLD=0
      CALL NAMEIN(N,NAMEPY,IDIM)
      IF (NEWOLD .EQ. (-2)) GOTO 6
      CALL PLYMLT(L,M,N,COEFPY,NASHPY,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - ADD
   10 IFORM=25
      NEWOLD=1
      CALL NAMEIN(L,NAMEPY,IDIM)
      IFORM=26
      CALL NAMEIN(M,NAMEPY,IDIM)
      IFORM=20
      NEWOLD=0
      CALL NAMEIN(N,NAMEPY,IDIM)
      IF (NEWOLD .EQ. (-2)) GOTO 6
      CALL PLYADD(L,M,N,COEFPY,NASHPY,IDIM4,JDIM)
      GOTO 6
```

```
C ... COMMAND - INVERT
   11 IFORM=27
      NEWOLD=1
      CALL NAMEIN(M,NAMEPY,IDIM)
      IFORM=20
      NEWOLD=0
      CALL NAMEIN(N,NAMEPY,IDIM)
      IF (NEWOLD .EQ. (-2)) GOTO 6
      CALL PLYINV(N,M,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - PRINT
   12 IFORM=28
      NEWOLD=1
      CALL NAMEIN(M,NAMEPY,IDIM)
      CALL SCRAWL(M,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - DUMP
   13 CALL DUMP(LCHAN,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - INPUT
   14 CALL GULP(COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM,KCHAN)
      GOTO 6

C ... COMMAND - DRAW
   15 IF (.NOT. DRAW) CALL PIC1ST(DRAW)
      CALL SQUIGL(COEFPY,NAMEPY,X,Y,IDIM,IDIM4,JDIM,JDIM2)
      GOTO 6
C ... COMMAND - TIDY
   16 CALL TIDY(COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - DIVIDE
   17 IFORM=29
      NEWOLD=1
      CALL NAMEIN(M,NAMEPY,IDIM)
      MDIV=INTIN(IDUMMY)
      IFORM=20
      NEWOLD=0
      CALL NAMEIN(N,NAMEPY,IDIM)
      IF (NEWOLD .EQ. (-2)) GOTO 6
      CALL PLYDIV(M,MDIV,N,COEFPY,HASHPY,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - NSCALE
   18 IFORM=38
      NEWOLD=1
      CALL NAMEIN(M,NAMEPY,IDIM)
      IFORM=20
      SCAL=REALIN(N)
      IFORM=20
      NEWOLD=0
      CALL NAMEIN(N,NAMEPY,IDIM)
      IF (NEWOLD .EQ. (-2)) GOTO 6
      CALL NSCALE(SCAL,M,N,COEFPY,HASHPY,IDIM4,JDIM)
      GOTO 6
C ... COMMAND - NTRANS
   19 IFORM=39
      NEWOLD=1
      CALL NAMEIN(L,NAMEPY,IDIM)
      IFORM=40
      NEWOLD=1
      CALL NAMEIN(M,NAMEPY,IDIM)
      IFORM=20
      NEWOLD=0
      CALL NAMEIN(N,NAMEPY,IDIM)
      IF (NEWOLD .EQ. (-2)) GOTO 6
      CALL NTRANS(L,M,N,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      GOTO 6
   20 IF (.NOT. ONLINE) CALL ERSTOP(7)
      NEWREC=.TRUE.
      IFORM=2
      GOTO 7
      END
```

```
      BLOCK DATA
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / NUMBER / NOS(10),IPLUS,MINUS,IPOINT
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                 IFORM,NUMBPY,NEWOLD,DRAW
      COMMON / CMANDS / INSTRU(3,17)
      COMMON / STRNGS / MESS(7,4),ISTR(31)
      DATA NOS(1),NOS(2),NOS(3),NOS(4),NOS(5),NOS(6),NOS(7),NOS(8),
     +     NOS(9),NOS(10) / 1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9 /
      DATA IPLUS / 1H+ /, MINUS / 1H- /, IPOINT / 1H. /, ISPACE / 1H  /
      DATA INSTRU(1, 1),INSTRU(2, 1),INSTRU(3, 1) / 1HY,1HE,1HS /,
     +     INSTRU(1, 2),INSTRU(2, 2),INSTRU(3, 2) / 1HN,1HO,1H  /,
     +     INSTRU(1, 3),INSTRU(2, 3),INSTRU(3, 3) / 1HE,1HN,1HD /,
     +     INSTRU(1, 4),INSTRU(2, 4),INSTRU(3, 4) / 1HS,1HC,1HA /,
     +     INSTRU(1, 5),INSTRU(2, 5),INSTRU(3, 5) / 1HM,1HU,1HL /,
     +     INSTRU(1, 6),INSTRU(2, 6),INSTRU(3, 6) / 1HA,1HD,1HD /,
     +     INSTRU(1, 7),INSTRU(2, 7),INSTRU(3, 7) / 1HI,1HN,1HV /,
     +     INSTRU(1, 8),INSTRU(2, 8),INSTRU(3, 8) / 1HP,1HR,1HI /,
     +     INSTRU(1, 9),INSTRU(2, 9),INSTRU(3, 9) / 1HD,1HU,1HM /,
     +     INSTRU(1,10),INSTRU(2,10),INSTRU(3,10) / 1HI,1HN,1HP /,
     +     INSTRU(1,11),INSTRU(2,11),INSTRU(3,11) / 1HD,1HR,1HA /,
     +     INSTRU(1,12),INSTRU(2,12),INSTRU(3,12) / 1HF,1HI,1HL /,
     +     INSTRU(1,13),INSTRU(2,13),INSTRU(3,13) / 1HS,1HT,1HR /,
     +     INSTRU(1,14),INSTRU(2,14),INSTRU(3,14) / 1HT,1HI,1HD /,
     +     INSTRU(1,15),INSTRU(2,15),INSTRU(3,15) / 1HD,1HI,1HV /,
     +     INSTRU(1,16),INSTRU(2,16),INSTRU(3,16) / 1HN,1HS,1HC /,
     +     INSTRU(1,17),INSTRU(2,17),INSTRU(3,17) / 1HN,1HT,1HP /
      DATA MESS(1,1),MESS(2,1),MESS(3,1),MESS(4,1),MESS(5,1),MESS(6,1),
     +     MESS(7,1) / 1HA,1HN,1HS,1HW,1HE,1HR,1H  /,
     +     MESS(1,2),MESS(2,2),MESS(3,2),MESS(4,2),MESS(5,2),MESS(6,2),
     +     MESS(7,2) / 1HC,1HO,1HM,1HM,1HA,1HN,1HD /,
     +     MESS(1,3),MESS(2,3),MESS(3,3),MESS(4,3),MESS(5,3),MESS(6,3),
     +     MESS(7,3) / 1HN,1HA,1HM,1HE,1H ,1H ,1H  /,
     +     MESS(1,4),MESS(2,4),MESS(3,4),MESS(4,4),MESS(5,4),MESS(6,4),
     +     MESS(7,4) / 1HN,1HU,1HM,1HB,1HE,1HR,1H  /
      DATA ISTR(1),ISTR(2),ISTR(3),ISTR(4),ISTR(5),ISTR(6),ISTR(7),
     +     ISTR(8),ISTR(9),ISTR(10),ISTR(11),ISTR(12),ISTR(13),ISTR(14),
     +     ISTR(15),ISTR(16),ISTR(17),ISTR(18),ISTR(19),ISTR(20),
     +     ISTR(21),ISTR(22),ISTR(23),ISTR(24),ISTR(25),ISTR(26),
     +     ISTR(27),ISTR(28),ISTR(29),ISTR(30),ISTR(31)
     +     / 1HE,1HN,1HT,1HE,1HR,1H ,1HN,1HA,1HM,1HE,1H ,1HO,1HF,1H ,
     +       1HP,1HO,1HL,1HY,1HN,1HO,1HM,1HI,1HA,1HL,1H ,1HT,1HO,1H ,
     +       1HB,1HE,1H   /
      END

      SUBROUTINE SCAMLT(SCAL,M,N,COEFPY,HASHPY,IDIM4,JDIM)
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4)
C
C     ROUTINE TO MULTIPLY A POLYNOMIAL BY A SCALAR
C
      HASHPY(N)=0.0
      DO 1 J=1,JDIM
      COEFPY(J,N)=SCAL*COEFPY(J,M)
      HASHPY(N)=HASHPY(N)+COEFPY(J,N)
    1 CONTINUE
      RETURN
      END

      SUBROUTINE PLYMLT(L,M,N,COEFPY,HASHPY,IDIM4,JDIM)
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4)
C
C     ROUTINE TO MULTIPLY TWO POLYNOMIALS
C
      HASHPY(N)=0.0
      DO 2 J=1,JDIM
      COEFPY(J,IDIM4)=0.0
      DO 1 I=1,J
      JI1=J-I+1
      COEFPY(J,IDIM4)=COEFPY(J,IDIM4)+COEFPY(I,L)*COEFPY(JI1,M)
    1 CONTINUE
    2 CONTINUE
      DO 3 J=1,JDIM
      COEFPY(J,N)=COEFPY(J,IDIM4)
      HASHPY(N)=HASHPY(N)+COEFPY(J,IDIM4)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLYADD(L,M,N,COEFPY,HASHPY,IDIM4,JDIM)
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4)
C
C     ROUTINE TO ADD TWO POLYNOMIALS
C
      HASHPY(N)=0.0
      DO 1 J=1,JDIM
      COEFPY(J,N)=COEFPY(J,L)+COEFPY(J,M)
    1 HASHPY(N)=HASHPY(N)+COEFPY(J,N)
      RETURN
      END



      SUBROUTINE PLYINV(M,N,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4),NAMEPY(12,IDIM)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO INVERT A POLYNOMIAL
C
      DO 1 J=1,JDIM
      IF (ABS(COEFPY(1,M)) .GT. 0.001) GOTO 3
      CALL PLYDIV(M,1,M,COEFPY,HASHPY,IDIM4,JDIM)
    1 CONTINUE
      WRITE(JCHAN,2001) (NAMEPY(K,M),K=1,12)
      WRITE(JCHAN,2002) (NAMEPY(K,N),K=1,12)
      DO 2 J=1,JDIM
      COEFPY(J,N)=0.0
    2 CONTINUE
      HASHPY(N)=0.0
      RETURN
    3 IF (J .EQ. 1) GOTO 4
      J1=-(J-1)
      WRITE(JCHAN,2003) (NAMEPY(K,N),K=1,12),J1
    4 I1=IDIM4-2
      I2=IDIM4-1
      I3=IDIM4
      SCAL=COEFPY(1,M)
      DO 5 J=1,JDIM
      COEFPY(J,I1)=-COEFPY(J,M)/SCAL
      COEFPY(J,I2)=0.0
      COEFPY(J,N)=0.0
    5 CONTINUE
      COEFPY(1,I1)=0.0
      COEFPY(1,I2)=1.0
      COEFPY(1,N)=1.0
      DO 7 J=1,JDIM
      CALL PLYMLT(I1,I2,I3,COEFPY,HASHPY,IDIM4,JDIM)
      CALL PLYADD(N,I3,N,COEFPY,HASHPY,IDIM4,JDIM)
      DO 6 JJ=1,JDIM
      COEFPY(JJ,I2)=COEFPY(JJ,I3)
    6 CONTINUE
    7 CONTINUE
      CALL SCAMLT(SCAL,N,N,COEFPY,HASHPY,IDIM4,JDIM)
      RETURN
 2001 FORMAT(15H CANNOT INVERT ,12A1,18H - ZERO POLYNOMIAL)
 2002 FORMAT(1H ,12A1,12H SET TO ZERO)
 2003 FORMAT(27H DIVIDE INVERSE POLYNOMIAL ,12A1,8H BY Z**(,I3,23H) BEFO
     +RE INTERPRETATION)
      END
```

```
      SUBROUTINE PLYDIV(N,II,H,COEFPY,HASHPY,IDIM4,JDIM)
      LOGICAL NEWREC,ONLINE,DRAW,WRITE1
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO DIVIDE A POLYNOMIAL BY Z**(-II)
C
      WRITE1=.FALSE.
      HASHPY(N)=0.0
      DO 2 J=1,JDIM
      IF (WRITE1) GOTO 1
      IF (J .LE. II .AND. ABS(COEFPY(J,N)) .GT. 0.001) WRITE(JCHAN,2001)
      WRITE1=.TRUE.
    1 JI=J+II
      COEFPY(J,N)=0.0
      IF (JI .LE. JDIM) COEFPY(J,N)=COEFPY(JI,N)
      HASHPY(N)=HASHPY(N)+COEFPY(J,N)
    2 CONTINUE
      IF (II .GE. JDIM) WRITE(JCHAN,2002)
      RETURN
 2001 FORMAT(50H WARNING - DIVISION GENERATES POSITIVE POWERS OF Z/
     +35H CHECK MODEL FOR REVERSED CAUSALITY)
 2002 FORMAT(58H WARNING - POWER TOO BIG, RESULTANT POLYNOMIAL SET TO ZE
     +RO)
      END


      SUBROUTINE NSCALE(SCAL,H,N,COEFPY,HASHPY,IDIM4,JDIM)
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4)
C
C     ROUTINE TO SCALE THE RANDOM VARIABLE OF A NOISE DISTRIBUTION POLYNOMIAL
C
      IDIM1=IDIM4-3
      DO 1 J=1,JDIM
      COEFPY(J,N)=COEFPY(J,H)
      COEFPY(J,IDIM1)=0.0
    1 CONTINUE
      DO 4 J=1,JDIM
      IBOT=0
    2 TOP=FLOAT(J)/ABS(SCAL)
      IF (FLOAT(JDIM) .LT. TOP) TOP=FLOAT(JDIM)
      JJ=IBOT+1
      IF (TOP .LT. FLOAT(JJ)) GOTO 3
      COEFPY(J,IDIM1)=COEFPY(J,IDIM1)+COEFPY(JJ,N)
      IBOT=JJ
      GOTO 2
    3 IF (JDIM .LT. JJ) JJ=JDIM
      COEFPY(J,IDIM1)=COEFPY(J,IDIM1)+COEFPY(JJ,N)*(TOP-FLOAT(IBOT))
    4 CONTINUE
      J1=0
      ISIG=1
      IF (SCAL .GE. 0.0) GOTO 5
      J1=JDIM+1
      ISIG=-1
    5 JJ1=J1+ISIG*1
      COEFPY(JJ1,N)=COEFPY(1,IDIM1)
      HASHPY(N)=COEFPY(JJ1,N)
      DO 6 J=2,JDIM
      JJ=J-1
      JJ1=J1+ISIG*J
      COEFPY(JJ1,N)=COEFPY(J,IDIM1)-COEFPY(JJ,IDIM1)
      HASHPY(N)=HASHPY(N)+COEFPY(JJ1,N)
    6 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NTRANS(L,M,N,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4),NAMEPY(12,IDIM)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO TRANSFORM A NOISE DISTRIBUTION POLYNOMIAL BY A SYSTEM
C     Z-TRANSFER POLYNOMIAL
C
      IDIM2=IDIM4-2
      IDIM3=IDIM4-1
      DO 1 J=1,JDIM
      COEFPY(J,IDIM3)=COEFPY(J,L)
    1 CONTINUE
      RECIP=1.0/FLOAT(JDIM)
      JDIM1=JDIM+1
      DO 2 J=1,JDIM
      LDIM=JDIM1-J
      IF (COEFPY(LDIM,M) .GE. RECIP) GOTO 5
    2 CONTINUE
      WRITE(JCHAN,2001) (NAMEPY(K,M),K=1,12)
    3 WRITE(JCHAN,2002) (NAMEPY(K,L),K=1,12)
      DO 4 J=1,JDIM
      COEFPY(J,N)=0.0
    4 CONTINUE
      HASHPY(N)=0.0
      RETURN
    5 DO 6 J=1,JDIM
      J1=J+1
      IF (COEFPY(J,M) .GE. RECIP) GOTO 7
    6 CONTINUE
      GOTO 3
    7 SCAL=COEFPY(J,M)
      CALL MSCALE(SCAL,IDIM3,N,COEFPY,HASHPY,IDIM4,JDIM)
      IF (J1 .GT. LDIM) GOTO 9
      DO 8 J=J1,LDIM
      IF (COEFPY(J,M) .LT. RECIP) GOTO 8
      SCAL=COEFPY(J,M)
      CALL MSCALE(SCAL,IDIM3,IDIM2,COEFPY,HASHPY,IDIM4,JDIM)
      CALL PLYMLT(N,IDIM2,N,COEFPY,HASHPY,IDIM4,JDIM)
    8 CONTINUE
    9 HASHPY(N)=0.0
      DO 10 J=1,JDIM
      HASHPY(N)=HASHPY(N)+COEFPY(J,N)
   10 CONTINUE
      RETURN
 2001 FORMAT(21H Z-TRANSFER FUNCTION ,12A1,19H IS ZERO POLYNOMIAL)
 2002 FORMAT(11H NTRANS ON ,12A1,35H ABANDONED - POLYNOMIAL SET TO ZERO)
      END


      SUBROUTINE SCRAWL(N,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4),NAMEPY(12,IDIM)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO WRITE OUT A POLYNOMIAL
C
      WRITE(JCHAN,2001) (NAMEPY(K,N),K=1,12),HASHPY(N)
      WRITE(JCHAN,2002)
      JDIM2=(JDIM+1)/2
      DO 1 J=1,JDIM2
      J1=J-1
      J2=J+JDIM2
      J3=J2-1
      WRITE(JCHAN,2003) J1,COEFPY(J,N),J3,COEFPY(J2,N)
    1 CONTINUE
      RETURN
 2001 FORMAT(15H TIME SERIES : ,12A1,5X,13HHASH TOTAL : ,F10.2)
 2002 FORMAT(1H ,2(15X,4HTIME,3X,5HVALUE))
 2003 FORMAT(16X,I3,5X,F9.2,15X,I3,5X,F9.2)
      END
```

```
      SUBROUTINE DUMP(LCHAN,COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4),NAMEPY(12,IDIM)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO DUMP OUT ALL CURRENT POLYNOMIALS
C
      DO 2 I=1,NUMBPY
      WRITE(LCHAN,2001) (NAMEPY(K,I),K=1,12),HASHPY(I)
      WRITE(LCHAN,2002) (COEFPY(J,I),J=1,JDIM)
    2 CONTINUE
      RETURN
 2001 FORMAT(1H ,12A1,8X,F10.3)
 2002 FORMAT(1H ,8F9.2)
      END
```

```
      SUBROUTINE TIDY(COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM)
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4),NAMEPY(12,IDIM)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO LIST THE NAMES OF ALL POLYNOMIALS CURRENTLY IN STORE AND
C     TO DELETE A REQUESTED SET OF THEM
C
      WRITE(JCHAN,2001)
      WRITE(JCHAN,2002) ((NAMEPY(K,I),K=1,12),I=1,NUMBPY)
      IFORM=30
      NDEL=INTIN(IDUMMY)
      IF (NDEL .EQ. 0) RETURN
      DO 6 II=1,NDEL
      IFORM=31
      NEWOLD=1
      CALL NAMEIN(N,NAMEPY,IDIM)
      DO 3 I=N,NUMBPY
      I1=I+1
      DO 1 J=1,JDIM
      COEFPY(J,I)=COEFPY(J,I1)
    1 CONTINUE
      HASHPY(I)=HASHPY(I1)
      DO 2 K=1,12
      NAMEPY(K,I)=NAMEPY(K,I1)
    2 CONTINUE
    3 CONTINUE
      DO 4 J=1,JDIM
      COEFPY(J,NUMBPY)=0.0
    4 CONTINUE
      HASHPY(NUMBPY)=0.0
      DO 5 K=1,12
      CALL COPY1(NAMEPY(K,NUMBPY),ISPACE)
    5 CONTINUE
      NUMBPY=NUMBPY-1
    6 CONTINUE
      RETURN
 2001 FORMAT(31H POLYNOMIALS CURRENTLY IN STORE)
 2002 FORMAT(6X,12A1,3X,12A1,3X,12A1,3X,12A1,3X,12A1)
      END
```

```
      SUBROUTINE SQUIGL(COEFPY,NAMEPY,X,Y,IDIM,IDIM4,JDIM,JDIM2)
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION NAMEPY(12,IDIM),COEFPY(JDIM,IDIM4)
      DIMENSION LDRAW(3),ITITLE(20),X(JDIM2),Y(JDIM2)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO PLOT UP TO THREE POLYNOMIALS ON A GRAPH (USING GINO-F)
C
      AJDIM=FLOAT(JDIM)
      IFORM=32
    1 NDRAW=INTIN(IDUMMY)
      IF (NDRAW .EQ. 0) RETURN
      IF (NDRAW .LE. 3) GOTO 2
      IF (.NOT. ONLINE) CALL ERSTOP(9)
      IFORM=37
      CALL SCRIBE(IFORM,JCHAN,0)
      NEWREC=.TRUE.
      GOTO 1
    2 IFC=33
      DO 3 I=1,NDRAW
      IFORM=IFC
      NEWOLD=1
      CALL NAMFIN(N,NAMEPY,IDIM)
      IFC=IFC+1
      LDRAW(I)=N
    3 CONTINUE
      IF (.NOT. NEWREC .AND. IPT .LE. 80) GOTO 5
      IF (.NOT. ONLINE) GOTO 4
      IFORM=36
      CALL SCRIBE(IFORM,JCHAN,N)
    4 CALL INBUF
    5 CALL REPLY(NREPLY,LEN)
      IEND=IPT+19
      IF (IEND .GT. 80) IEND=80
      K=1
      DO 6 I=IPT,IEND
      CALL COPY1(ITITLE(K),IBUF(I))
      K=K+1
    6 CONTINUE
      IPT=IEND+1
      CALL LEFJUS
      CALL PICCLE
      TOP=0.0
      BOT=0.0
      DO 8 K=1,NDRAW
      L=LDRAW(K)
      DO 7 J=1,JDIM
      IF (COEFPY(J,L) .LT. BOT) BOT=COEFPY(J,L)
      IF (COEFPY(J,L) .GT. TOP) TOP=COEFPY(J,L)
    7 CONTINUE
    8 CONTINUE
      IF (BOT .LT. (-0.001)) GOTO 9
      CENT=10.0
      GOTO 11
    9 IF (TOP .GT. 0.01) GOTO 10
      CENT=120.0
      GOTO 11
   10 CENT=BOT*110.0/(BOT-TOP)+10.0

   11 CALL AXIPOS(0,20.0,CENT,150.0,1)
      CALL AXIPOS(0,20.0,CENT,110.0,2)
      CALL AXISCA(3,JDIM,0.0,AJDIM,1)
      CALL AXISCA(3,10,BOT,TOP,2)
      CALL AXIDRA(1,1,1)
      CALL AXIDRA(-1,-1,2)
      NDASH=0
      REPEAT=11.0
      DASH=10.0
      DO 13 K=1,NDRAW
      L=LDRAW(K)
      I=1
      DO 12 J=1,JDIM
      X(I)=J-1
      Y(I)=COEFPY(J,L)
      X(I+1)=J
      Y(I+1)=COEFPY(J,L)
      I=I+2
```

```
   12 CONTINUE
      CALL DASHED (NDASH,REPEAT,DASH,0,0)
      CALL GRAPOL(X,Y,JDIM2)
      NDASH=-1
      REPEAT=REPEAT-4.0
      DASH=DASH-4.0
   13 CONTINUE
      CALL DASHED(0,0.0,0.0,0.0)
      CALL MOVTO2(70.0,-5.0)
      CALL CHAARR(ITITLE,20,1)
C
C     MOVE CURSOR TO BOTTOM OF SCREEN
C
      CALL CHAPOS(0.0,0.0)
      RETURN
      END


      SUBROUTINE GULP(COEFPY,HASHPY,NAMEPY,IDIM,IDIM4,JDIM,KCHAN)
      LOGICAL NEWREC,ONLINE,TEMPOL,DRAW
      DIMENSION COEFPY(JDIM,IDIM4),HASHPY(IDIM4),NAMEPY(12,IDIM),
     +          NEWNAM(12)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO READ A BLOCK OF POLYNOMIALS, NPOLY IS THE NUMBER TO BE READ.
C     FOR EACH POLYNOMIAL THE ROUTINE REQUIRES NAME, HASH TOTAL AND COEFFICIENTS
C
    1 IFORM=10
      NOPOLY=INTIN(IDUMMY)
      IF (NOPOLY .EQ. 0) RETURN
      NOTOT=NOPOLY+NUMBPY
      IF (NOTOT .LE. IDIM) GOTO 2
      IF (.NOT. ONLINE) CALL ERSTOP(2)
      IFORM=11
      CALL SCRIBE(IFORM,JCHAN,0)
      NEWREC=.TRUE.
      GOTO 1
    2 IFORM=12
    3 NREPLY=0
      CALL INSTIN(NREPLY)
      IF (NREPLY .GT. 11 .AND. NREPLY .LE. 13) GOTO 4
      IF (.NOT. ONLINE) CALL ERSTOP(3)
      IFORM=5
      NEWREC=.TRUE.
      GOTO 3
    4 TEMPOL=ONLINE
      ONLINE=.FALSE.
      IF (TEMPOL .AND. NREPLY .EQ. 13) ONLINE=.TRUE.
      ITEMP=ICHAN
      IF (NREPLY .EQ. 12) ICHAN=KCHAN
C
C     READ AND CHECK POLYNOMIALS
C
      DO 8 I=1,NOPOLY
      IFORM=13
      NEWOLD=-1
      CALL NAMEIN(N,NAMEPY,IDIM)
      IFORM=15
      HASHPY(N)=REALIN(N)
    5 IFORM=16
      HSHTOT=0.0
      NOMORE=JDIM
      DO 6 J=1,JDIM
      COEFPY(J,N)=REALIN(NOMORE)
      HSHTOT=HSHTOT+COEFPY(J,N)
      NOMORE=NOMORE-1
      IFORM=17
    6 CONTINUE
      IF (ABS(HSHTOT-HASHPY(N)) .LT. 0.01) GOTO 8
      WRITE(JCHAN,2000) HSHTOT
 2000 FORMAT(51H WARNING : HASH-TOTAL ERROR, SUM OF COEFFICIENTS = ,
     +       F10.3)
      IF (.NOT. ONLINE) GOTO 8
      IFORM=18
      NEWREC=.TRUE.
```

```
    7 NREPLY=0
      CALL INSTIN(NREPLY)
      IF (NREPLY .EQ. 1) GOTO 5
      IF (NREPLY .EQ. 2) GOTO 8
      IFORM=5
      NEWREC=.TRUE.
      GOTO 7
    8 CONTINUE
      NASHPY(N)=NSHTOT
      ONLINE=TEMPOL
      ICHAN=ITEMP
      RETURN
      END


      SUBROUTINE REPLY(NO,LEN)
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
      COMMON / CHANDS / INSTRU(3,17)
C
C     ROUTINE TO CHECK FOR THREE CHARACTER COMMAND STRING IN IBUF STARTING
C     AT POSITION IPT.
C     COMMAND FOUND IS RETURNED IN NO
C        NO = 1 ... YES        NO = 7  ... INVERT    NO = 13 ... STREAM
C        NO = 2 ... NO         NO = 8  ... PRINT     NO = 14 ... TIDY
C        NO = 3 ... END        NO = 9  ... DUMP      NO = 15 ... DIVIDE
C        NO = 4 ... SCALE      NO = 10 ... INPUT     NO = 16 ... NSCALE
C        NO = 5 ... MULTIPLY   NO = 11 ... DRAW      NO = 17 ... NTRANS
C        NO = 6 ... ADD        NO = 12 ... FILE
C                NO = 0 ... COMMAND NOT RECOGNISED
C
      IF (IPT .GT. 80) GOTO 5
      IEND=IPT+12
      IF (IEND .GT. 80) IEND=80
      DO 1 I=IPT,IEND
      CALL COMP1(IBUF(I),ISPACE,IND)
      LEN=I-IPT
      IF (IND .EQ. 1) GOTO 2
    1 CONTINUE
    2 IF (LEN .EQ. 0 .OR. LEN .GT. 8) GOTO 5
      DO 4 NO=1,17
      DO 3 I=1,3
      II=IPT+I-1
      CALL COMP1(IBUF(II),INSTRU(I,NO),IND)
      IF (IND .EQ. 0) GOTO 4
    3 CONTINUE
      IF (NO .NE. 3 .OR. LEN .NE. 3) RETURN
      IF (DRAW) CALL DEVEND
      STOP
    4 CONTINUE
    5 NO=0
      RETURN
      END


      SUBROUTINE LEFJUS
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO MOVE POINTER PAST LEADING SPACES IN IBUF
C     IF NO NON-SPACE CHARACTER IS FOUND NEWREC IS SET TO .TRUE.
C
      DO 1 I=IPT,80
      CALL COMP1(IBUF(I),ISPACE,IND)
      IF (IND .EQ. 0) GOTO 2
    1 CONTINUE
      NEWREC=.TRUE.
      RETURN
    2 IPT=I
      NEWREC=.FALSE.
      RETURN
      END
```

```
      SUBROUTINE INSTIN(NREPLY)
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO EXTRACT NEXT COMMAND/ANSWER FROM IBUF
C
    1 IF (.NOT. NEWREC .AND. IPT .LE. 78) GOTO 3
      IF (.NOT. ONLINE) GOTO 2
      CALL SCRIBE(IFORM,JCHAN,0)
      IFORM=1
      IF (NREPLY .GT. 0) IFORM=2
    2 CALL INBUF
      GOTO 1
    3 CALL REPLY(NREPLY,LEN)
      IPT=IPT+LEN
      CALL LEFJUS
      RETURN
      END


      SUBROUTINE NAMEIN(N,NAMEPY,IDIM)
      LOGICAL NEWREC,ONLINE,DRAW
      DIMENSION NAMEPY(12,IDIM),NEWNAM(12)
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO EXTRACT A STRING OF CHARACTERS FROM IBUF AND
C     COMPARE IT WITH THE NAMES OF THE POLYNOMIALS ALREADY KNOWN
C
    1 IF (.NOT. NEWREC .AND. IPT .LE. 80) GOTO 3
      IF (.NOT. ONLINE) GOTO 2
      CALL SCRIBE(IFORM,JCHAN,N)
      IF (NEWOLD .EQ. (-2)) RETURN
      IFORM=3
    2 CALL INBUF
      GOTO 1
    3 DO 4 K=1,12
      CALL COPY1(NEWNAM(K),ISPACE)
    4 CONTINUE
      CALL REPLY(NREPLY,LEN)
      IF (LEN .GT. 0 .AND. LEN .LT. 13) GOTO 5
      IERR=4
      IFORM=3
      GOTO 12
    5 I1=IPT-1
      DO 6 K=1,LEN
      I1K=I1+K
      CALL COPY1(NEWNAM(K),IBUF(I1K))
    6 CONTINUE
      IF (NUMBPY .EQ. 0) GOTO 9
      DO 8 I=1,NUMBPY
      DO 7 K=1,12
      CALL COMP1(NAMEPY(K,I),NEWNAM(K),IND)
      IF (IND .EQ. 0) GOTO 8
    7 CONTINUE
      N=I
      IF (NEWOLD .GE. 0) GOTO 11
      IFORM=14
      IERR=5
      GOTO 12
    8 CONTINUE
    9 N=NUMBPY+1
      IF (NEWOLD .GT. 0) GOTO 12
      IFORM=11
      IERR=2
      IF (N .GT. IDIM) GOTO 13
      DO 10 K=1,12
      CALL COPY1(NAMEPY(K,N),NEWNAM(K))
   10 CONTINUE
      NUMBPY=N
   11 IPT=IPT+LEN
      CALL LEFJUS
      RETURN
   12 IFORM=7
      IERR=6
   13 IF (.NOT. ONLINE) CALL ERSTOP(IERR)
      NEWREC=.TRUE.
      IF (IFORM .EQ. 11) NEWOLD=-2
      GOTO 1
      END
```

```
      FUNCTION INTIN(IDUMMY)
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
      COMMON / NUMBER / NOS(10),IPLUS,MINUS,IPOINT
C
C     ROUTINE TO EXTRACT A POSITIVE INTEGER FROM IBUF
C
    1 IF (.NOT. NEWREC .AND. IPT .LE. 80) GOTO 3
      IF (.NOT. ONLINE) GOTO 2
      CALL SCRIBE(IFORM,JCHAN,0)
      IFORM=4
    2 CALL INBUF
      GOTO 1
    3 CALL REPLY(NREPLY,LEN)
      NOOK=0
      INTIN=0
    4 IF (IPT .LE. 80) GOTO 5
      NEWREC=.TRUE.
      IF (NOOK .EQ. 1) GOTO 7
      GOTO 6
    5 DO 6 I=1,10
      CALL COMP1(IBUF(IPT),NOS(I),IND)
      IF (IND .EQ. 0) GOTO 6
      INTIN=INTIN*10+I-1
      NOOK=1
      IPT=IPT+1
      GOTO 4
    6 CONTINUE
      CALL COMP1(IBUF(IPT),ISPACE,IND)
      IF (IND .EQ. 0) GOTO 8
      CALL LEFJUS
    7 RETURN
    8 IF (.NOT. ONLINE) CALL ERSTOP(8)
      NEWREC=.TRUE.
      IFORM=4
      GOTO 1
      END


      FUNCTION REALIN(N)
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
      COMMON / NUMBER / NOS(10),IPLUS,MINUS,IPOINT
C
C     ROUTINE TO EXTRACT A REAL FROM IBUF
C
    1 IF (.NOT. NEWREC .AND. IPT .LE. 80) GOTO 3
      IF (.NOT. ONLINE) GOTO 2
      CALL SCRIBE(IFORM,JCHAN,N)
      IFORM=4
    2 CALL INBUF
      GOTO 1
    3 CALL REPLY(NREPLY,LEN)
      NOOK=0
      REALIN=0.0
      RNEG=1.0
      FACT=1.0
      CALL COMP1(IBUF(IPT),MINUS,IND)
      IF (IND .EQ. 0) GOTO 4
      IPT=IPT+1
      RNEG=-1.0
      GOTO 5
    4 CALL COMP1(IBUF(IPT),IPLUS,IND)
      IF (IND .EQ. 1) IPT=IPT+1
    5 IF(IPT .LE. 80) GOTO 6
      NEWREC=.TRUE.
      IF (NOOK .EQ. 1) GOTO 12
      GOTO 13
    6 DO 7 I=1,10
      CALL COMP1(IBUF(IPT),NOS(I),IND)
      IF (IND .EQ. 0) GOTO 7
      REALIN=REALIN*10.0+FLOAT(I-1)
      IPT=IPT+1
      NOOK=1
      GOTO 5
```

```
    7 CONTINUE
      CALL COMP1(IBUF(IPT),IPOINT,IND)
      IF (IND .EQ. 0) GOTO 11
    8 FACT=FACT*10.0
      IPT=IPT+1
      IF (IPT .LE. 80) GOTO 9
      NEWREC=.TRUE.
      IF (NOCK .EQ. 1) GOTO 12
      GOTO 13
    9 DO 10 I=1,10
      CALL COMP1(IBUF(IPT),NOS(I),IND)
      IF (IND .EQ. 0) GOTO 10
      REALIN=REALIN+FLOAT(I-1)/FACT
      NOCK=1
      GOTO 8
   10 CONTINUE
   11 CALL COMP1(IBUF(IPT),ISPACE,IND)
      IF (IND .EQ. 0) GOTO 13
   12 CALL LEFJUS
      REALIN=REALIN*NNEG
      RETURN

   13 IF (.NOT. ONLINE) CALL ERSTOP(8)
      NEWREC=.TRUE.
      IFORM=4
      GOTO 1
      END


      SUBROUTINE INBUF
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     ROUTINE TO READ A RECORD INTO IBUF
C
      READ(ICHAN,1000) IBUF
      IPT=1
      CALL LEFJUS
      RETURN
 1000 FORMAT(80A1)
      END


      SUBROUTINE ERSTOP(N)
      LOGICAL NEWREC,ONLINE,DRAW
      COMMON / INFO / IBUF(80),NEWREC,IPT,ISPACE,ICHAN,JCHAN,ONLINE,
     +                IFORM,NUMBPY,NEWOLD,DRAW
C
C     SUBROUTINE TO WRITE OUT ERROR MESSAGE AND ABANDON THE JOB
C
      GOTO (1,2,3,4,5,6,7,8,9,10),N
      GOTO 10
    1 WRITE(JCHAN,2001)
      GOTO 11
    2 WRITE(JCHAN,2002)
      GOTO 11
    3 WRITE(JCHAN,2003)
      GOTO 11
    4 WRITE(JCHAN,2004)
      GOTO 11
    5 WRITE(JCHAN,2005)
      GOTO 11
    6 WRITE(JCHAN,2006)
      GOTO 11
    7 WRITE(JCHAN,2007)
      GOTO 11
    8 WRITE(JCHAN,2008)
      GOTO 11
    9 WRITE(JCHAN,2009)
      GOTO 11
   10 WRITE(JCHAN,2010)
   11 WRITE(JCHAN,2011)
      IF (DRAW) CALL DEVEND
      STOP
```

```
2001 FORMAT(43H INVALID RESPONSE TO 'INTERACTIVE RUNNING?')
2002 FORMAT(38H MAXIMUM NUMBER OF POLYNOMIALS EXCEDED)
2003 FORMAT(45H INVALID RESPONSE TO 'SOURCE OF POLYNOMIALS?')
2004 FORMAT(13H INVALID NAME)
2005 FORMAT(15H DUPLICATE NAME)
2006 FORMAT(18H UNRECOGNISED NAME)
2007 FORMAT(16H INVALID COMMAND)
2008 FORMAT(15H INVALID NUMBER)
2009 FORMAT(51H MAXIMUM OF 3 POLYNOMIALS CAN BE DRAWN ON ONE GRAPH)
2010 FORMAT(19H UNRECOGNISED ERROR)
2011 FORMAT(14H JOB ABANDONED)
     END


     SUBROUTINE SCRIBE(IFORM,JCHAN,N)
     COMMON / STRNGS / MESS(7,4),ISTR(31)
C
C    ROUTINE TO WRITE OUT A MESSAGE
C
     GOTO (2,2,2,2,2,2,2,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
    +20,21,22,23,24,25,26,27,28,29,30,31,32,33,34),IFORM
   1 CALL ERSTOP(10)
   2 I=IFORM-(IFORM-1)/4*4
     IF (IFORM .LE. 4) WRITE(JCHAN,2001)(MESS(J,I),J=1,7)
     IF (IFORM .GE. 5) WRITE(JCHAN,2002)(MESS(J,I),J=1,7)
     RETURN
   3 WRITE(JCHAN,2003)
     RETURN
   4 WRITE(JCHAN,2004)
     RETURN
   5 WRITE(JCHAN,2005)
     RETURN
   6 WRITE(JCHAN,2006)
     RETURN
   7 WRITE(JCHAN,2007) (ISTR(I),I=1,25)
     RETURN
   8 WRITE(JCHAN,2008)
     RETURN
   9 WRITE(JCHAN,2009) (ISTR(I),I=1,6)
     RETURN
  10 WRITE(JCHAN,2010) (ISTR(I),I=1,6)
     RETURN
  11 WRITE(JCHAN,2011) N
     RETURN
  12 WRITE(JCHAN,2012)
     RETURN
  13 WRITE(JCHAN,2013) (ISTR(I),I=1,6)
     RETURN
  14 WRITE(JCHAN,2014) (ISTR(I),I=1,28)
     RETURN
  15 WRITE(JCHAN,2015) (ISTR(I),I=1,31)
     RETURN
  16 WRITE(JCHAN,2016)
     RETURN
  17 WRITE(JCHAN,2017) (ISTR(I),I=1,14)
     RETURN
  18 WRITE(JCHAN,2018) (ISTR(I),I=1,14)
     RETURN
  19 WRITE(JCHAN,2019) (ISTR(I),I=1,14)
     RETURN
  20 WRITE(JCHAN,2020) (ISTR(I),I=1,14)
     RETURN
  21 WRITE(JCHAN,2021) (ISTR(I),I=1,31)
     RETURN
  22 WRITE(JCHAN,2022) (ISTR(I),I=1,31)
     RETURN
  23 WRITE(JCHAN,2023) (ISTR(I),I=1,31)
     RETURN
  24 WRITE(JCHAN,2024)
     RETURN
  25 WRITE(JCHAN,2025) (ISTR(I),I=1,31)
     RETURN
  26 WRITE(JCHAN,2026)
     RETURN
```

```
27    WRITE(JCHAN,2027) (ISTR(I),I=1,25)
      RETURN
28    WRITE(JCHAN,2028) (ISTR(I),I=1,25)
      RETURN
29    WRITE(JCHAN,2029) (ISTR(I),I=1,25)
      RETURN
30    WRITE(JCHAN,2030) (ISTR(I),I=1,14)
      RETURN
31    WRITE(JCHAN,2031)
      RETURN
32    WRITE(JCHAN,2032) (ISTR(I),I=1,14)
      RETURN
33    WRITE(JCHAN,2033) (ISTR(I),I=1,14)
      RETURN
34    WRITE(JCHAN,2034) (ISTR(I),I=1,14)
      RETURN
2001  FORMAT(19H PLEASE TYPE VALID ,7A1)
2002  FORMAT(14H UNRECOGNISED ,7A1,12H - TRY AGAIN)
2003  FORMAT(42H INTERACTIVE RUNNING? (ANSWER - YES OR NO))
2004  FORMAT(46H HOW MANY POLYNOMIALS ARE ABOUT TO BE ENTERED?)
2005  FORMAT(38H MAXIMUM NUMBER OF POLYNOMIALS EXCEDED)
2006  FORMAT(49H SOURCE OF POLYNOMIALS? (ANSWER - FILE OR STREAM))
2007  FORMAT(1H ,25A1)
2008  FORMAT(27H DUPLICATE NAME - TRY AGAIN)
2009  FORMAT(1H ,6A1,10HHASH-TOTAL)
2010  FORMAT(1H ,6A1,12HCOEFFICIENTS)
2011  FORMAT(1H ,I2,25H MORE COEFFICIENTS NEEDED)
2012  FORMAT(22H RE-TYPE COEFFICIENTS?)
2013  FORMAT(1H ,6A1,9HA COMMAND)
2014  FORMAT(1H ,28A1,11HHOLD RESULT)
2015  FORMAT(1H ,31A1,6HSCALED)
2016  FORMAT(13H ENTER SCALAR)
2017  FORMAT(1H ,14A1,16H1ST MULTIPLICAND)
2018  FORMAT(1H ,14A1,16H2ND MULTIPLICAND)
2019  FORMAT(1H ,14A1,11H1ST SUMMAND)
2020  FORMAT(1H ,14A1,11H2ND SUMMAND)
2021  FORMAT(1H ,31A1,8HINVERTED)
2022  FORMAT(1H ,31A1,7HPRINTED)
2023  FORMAT(1H ,31A1,7HDIVIDED)
2024  FORMAT(36H HOW MANY POLYNOMIALS TO BE DELETED?)
2025  FORMAT(1H ,31A1,7HDELETED)
2026  FORMAT(34H HOW MANY POLYNOMIALS TO BE DRAWN?)
2027  FORMAT(1H ,25A1,17H(1ST TO BE DRAWN))
2028  FORMAT(1H ,25A1,17H(2ND TO BE DRAWN))
2029  FORMAT(1H ,25A1,17H(3RD TO BE DRAWN))
2030  FORMAT(1H ,14A1,5HGRAPH)
2031  FORMAT(51H MAXIMUM OF 3 POLYNOMIALS CAN BE DRAWN ON ONE GRAPH)
2032  FORMAT(1H ,14A1,29HNOISE POLYNOMIAL TO BE SCALED)
2033  FORMAT(1H ,14A1,34HNOISE POLYNOMIAL TO BE TRANSFORMED)
2034  FORMAT(1H ,14A1,21HZ-TRANSFER POLYNOMIAL)
      END


      SUBROUTINE COPY1(LOC1,LOC2)
C
C     ROUTINE TO COPY A CHARACTER (THIS ROUTINE IS MACHINE DEPENDENT)
C
      LOC1=LOC2
      RETURN
      END


      SUBROUTINE COMP1(LOC1,LOC2,IND)
C
C     ROUTINE TO COMPARE A CHARACTER  (THIS ROUTINE IS MACHINE DEPENDENT)
C
      IND=0
      IF (LOC1 .EQ. LOC2) IND=1
      RETURN
      END
```

```
      SUBROUTINE PIC1ST(DRAW)
C
C     ROUTINE TO SET UP BACKEND FOR GINO-F AND TO SHIFT ORIGIN.
C     THIS SHOULD BE ALTERED BY THE IMPLEMENTING SITE.
C     IF UNITS ARE SPECIFIED AS MM. THE CURRENT CALLS TO PLOTTING ROUTINES
C     PRODUCE A PICTURE WHICH FITS A DISPLAY TUBE MEASURING 134 X 140 MM.
C
      LOGICAL DRAW
      CALL T4014
      CALL UNITS(1.0)
C ... THE FOLLOWING CALL MUST BE PRESENT TO SHIFT THE ORIGIN
      CALL SHIFT2(0.0,16.0)
      DRAW=.TRUE.
      RETURN
      END
```

APPENDIX II

PROOFS

4.6.1 $$\zeta(d_{Om},d_{1m}) = \frac{F_{Om}D_{Om} + I_{Om}S_{Om}D_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

Proof

$$d_{1m} = (f_{Om} - s_{Om})D_{Om}$$

$$= d_{Om}F_{Om}D_{Om} - s_{Om}D_{Om}$$

$$= d_{Om}F_{Om}D_{Om} - (r_{Om} - d_{Om}I_{Om})S_{Om}D_{Om}$$

$$= d_{Om}F_{Om}D_{Om} + d_{Om}I_{Om}S_{Om}D_{Om} - r_{Om}S_{Om}D_{Om}$$

$$= d_{Om}(F_{Om}D_{Om} + I_{Om}S_{Om}D_{Om}) - d_{1m}L_{Om}S_{Om}D_{Om}$$

$$\therefore d_{1m}(1 + L_{Om}S_{Om}D_{Om}) = d_{Om}(F_{Om}D_{Om} + I_{Om}S_{Om}D_{Om})$$

$$\therefore \zeta(d_{Om},d_{1m}) = \frac{d_{1m}}{d_{Om}} = \frac{F_{Om}D_{Om} + I_{Om}S_{Om}D_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

4.6.2 $$\zeta(d_{Om},s_{Om}) = \frac{F_{Om}L_{Om}S_{Om}D_{Om} - I_{Om}S_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

Proof

$$s_{Om} = (r_{Om} - d_{Om}I_{Om})S_{Om}$$

$$= d_{1m}L_{Om}S_{Om} - d_{Om}I_{Om}S_{Om}$$

$$= (f_{Om} - s_{Om})D_{Om}L_{Om}S_{Om} - d_{Om}I_{Om}S_{Om}$$

$$= d_{Om}F_{Om}D_{Om}L_{Om}S_{Om} - s_{Om}D_{Om}L_{Om}S_{Om} - d_{Om}I_{Om}S_{Om}$$

$$\therefore\ s_{Om}(1 + L_{Om}S_{Om}D_{Om}) = d_{Om}(F_{Om}L_{Om}S_{Om}D_{Om} - I_{Om}S_{Om})$$

$$\therefore \zeta(d_{Om}, s_{Om}) = \frac{s_{Om}}{d_{Om}} = \frac{F_{Om}L_{Om}S_{Om}D_{Om} - I_{Om}S_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

<u>4.6.5</u> $$\zeta(\tilde{r}_{Om}, d_{1m}) = \frac{-S_{Om}D_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

<u>Proof</u>

$$d_{1m} = (f_{Om} - s_{Om})D_{Om}$$

$$= d_{Om}F_{Om}D_{Om} - (r_{Om} + \tilde{r}_{Om} - d_{Om}I_{Om})S_{Om}D_{Om}$$

$$= d_{Om}(F_{Om} - I_{Om}S_{Om})D_{Om} - r_{Om}S_{Om}D_{Om} - \tilde{r}_{Om}S_{Om}D_{Om}$$

$$= d_{Om}(F_{Om} - I_{Om}S_{Om})D_{Om} - d_{1m}L_{Om}S_{Om}D_{Om} - \tilde{r}_{Om}S_{Om}D_{Om}$$

Linearity lets us assume $d_{Om} = 0$ whilst considering $\zeta(\tilde{r}_{Om}, d_{1m})$ so we have

$$d_{1m} = - d_{1m}L_{Om}S_{Om}D_{Om} - \tilde{r}_{Om}S_{Om}D_{Om}$$

$$\therefore \zeta(\tilde{r}_{Om}, d_{1m}) = \frac{d_{1m}}{\tilde{r}_{Om}} \quad \frac{-S_{Om}D_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

<u>4.6.6</u> $$\zeta(\tilde{r}_{Om}, s_{Om}) = \frac{S_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

<u>Proof</u>

$$s_{Om} = (r_{Om} + \tilde{r}_{Om} - d_{Om}I_{Om})S_{Om}$$

$$= d_{1m}L_{Om}S_{Om} + \tilde{r}_{Om}S_{Om} - d_{Om}I_{Om}S_{Om}$$

$$= (d_{Om}F_{Om} - s_{Om})D_{Om}L_{Om}S_{Om} + \tilde{r}_{Om}S_{Om} - d_{Om}I_{Om}S_{Om}$$

Linearity lets us assume $d_{Om} = 0$ whilst considering $\zeta(\tilde{r}_{Om}, s_{Om})$ so we have

$$s_{Om} = - s_{Om}D_{Om}L_{Om}S_{Om} + \tilde{r}_{Om}S_{Om}$$

$$\therefore \zeta(\tilde{r}_{Om}, s_{Om}) = \frac{s_{Om}}{\tilde{r}_{Om}} = \frac{S_{Om}}{1 + L_{Om}S_{Om}D_{Om}}$$

4.7.2 $$s_{kp} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proof

$$s_{kp} = (r_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn})S_{kp}$$

$$= d_{(k+1)p}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}$$

$$= (\ddot{f}_{kp} - s_{kp})D_{kp}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}$$

$$= \sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp} - s_{kp}D_{kp}L_{kp}S_{kp}$$

$$\therefore s_{kp}(1 + L_{kp}S_{kp}D_{kp}) = \sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}$$

$$\therefore s_{kp} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

4.7.3 $$\zeta(\tilde{d}_{Om}, d_{(k+1)p}) = \frac{F_{kpm}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proof

From 4.7.1 we have:-

$$d_{(k+1)p} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp} + \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Using system linearity $\tilde{d}_{Om}$ can be assumed to be the only input so this becomes:-

$$d_{(k+1)p} = \frac{\tilde{d}_{Om}F_{kpm}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

$$\therefore \zeta(\tilde{d}_{Om}, d_{(k+1)p}) = \frac{d_{(k+1)p}}{\tilde{d}_{Om}} = \frac{F_{kpm}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

4.7.4 $$\zeta(\tilde{d}_{kn}, d_{(k+1)p}) = \frac{I_{kpn}S_{kp}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proof

From 4.7.1 we have:-

$$d_{(k+1)p} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp} + \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Using system linearity $\tilde{d}_{kn}$ can be assumed to be the only input so this becomes:-

$$d_{(k+1)p} = \frac{\tilde{d}_{kn} I_{kpn} S_{kp} D_{kp}}{1 + L_{kp} S_{kp} D_{kp}}$$

$$\therefore \zeta(\tilde{d}_{kn}, d_{(k+1)p}) = \frac{d_{(k+1)p}}{\tilde{d}_{kn}} = \frac{I_{kpn} S_{kp} D_{kp}}{1 + L_{kp} S_{kp} D_{kp}}$$

4.7.5 $$\zeta(\tilde{d}_{Om}, s_{kp}) = \frac{F_{kpm} D_{kp} L_{kp} S_{kp}}{1 + L_{kp} S_{kp} D_{kp}}$$

Proof

From 4.7.2 we have:-

$$s_{kp} = \frac{\sum_{m=1}^{M} d_{Om} F_{kpm} D_{kp} L_{kp} S_{kp} - \sum_{n=1}^{N} d_{kn} I_{kpn} S_{kp}}{1 + L_{kp} S_{kp} D_{kp}}$$

Using system linearity $\tilde{d}_{Om}$ can be assumed to be the only input so this becomes:-

$$s_{kp} = \frac{\tilde{d}_{Om} F_{kpm} D_{kp} L_{kp} S_{kp}}{1 + L_{kp} S_{kp} D_{kp}}$$

$$\therefore \zeta(\tilde{d}_{Om}, s_{kp}) = \frac{s_{kp}}{\tilde{d}_{Om}} = \frac{F_{kpm} D_{kp} L_{kp} S_{kp}}{1 + L_{kp} S_{kp} D_{kp}}$$

4.7.6 $$\zeta(\tilde{d}_{kn}, s_{kp}) = \frac{-I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proof

From 4.7.2 we have:-

$$s_{kp} = \frac{\sum_{m=1}^{M} d_{Om}F_{kpm}D_{kp}L_{kp}S_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Using system linearity $\tilde{d}_{kn}$ can be assumed to be the only input so this becomes:-

$$s_{kp} = \frac{-\tilde{d}_{kn}I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

$$\therefore \zeta(\tilde{d}_{kn}, s_{kp}) = \frac{-I_{kpn}S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

4.8.1 $$d_{(k+1)p} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proof

$$d_{(k+1)p} = (f_{kp} - s_{kp})D_{kp}$$

$$= \sum_{n=1}^{N} d_{kn}F_{kp}D_{kp} - (r_{kp} - \sum_{n=1}^{N} d_{kn}I_{kpn})S_{kp}D_{kp}$$

$$= \sum_{n=1}^{N} d_{kn}(F_{kp} + I_{kpn}S_{kp})D_{kp} - d_{(k+1)p}L_{kp}S_{kp}D_{kp}$$

$$d_{(k+1)p}(1 + L_{kp}S_{kp}D_{kp}) = \sum_{n=1}^{N} d_{kn}(F_{kp} + I_{kpn}S_{kp})D_{kp}$$

$$d_{(k+1)p} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

<u>4.8.2</u>

$$s_{kp} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

<u>Proof</u>

$$s_{kp} = (-\sum_{n=1}^{N} d_{kn}I_{kpn} + r_{kp})S_{kp}$$

$$= -\sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp} + d_{(k+1)p}L_{kp}S_{kp}$$

$$= -\sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp} + (f_{kp} - s_{kp})D_{kp}L_{kp}S_{kp}$$

$$= -\sum_{n=1}^{N} d_{kn}I_{kpn}S_{kp} + \sum_{n=1}^{N} d_{kn}F_{kp}D_{kp}L_{kp}S_{kp} - s_{kp}D_{kp}L_{kp}S_{kp}$$

$$= \sum_{n=1}^{N} d_{kn}(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp} - s_{kp}D_{kp}L_{kp}D_{kp}$$

$$\therefore s_{kp}(1 + L_{kp}S_{kp}D_{kp}) = \sum_{n=1}^{N} d_{kn}(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}$$

$$\therefore s_{kp} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

4.8.3 $$\zeta(\tilde{d}_{kn}, d_{(k+1)p}) = \frac{(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proof

From 4.8.1 we have:-

$$d_{(k+1)p} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Using system linearity $\tilde{d}_{kn}$ can be assumed to be the only input so this becomes:-

$$d_{(k+1)p} = \frac{\tilde{d}_{kn}(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

$$\therefore \zeta(\tilde{d}_{kn}, d_{(k+1)p}) = \frac{d_{(k+1)p}}{\tilde{d}_{kn}} = \frac{(F_{kp} + I_{kpn}S_{kp})D_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

4.8.4 $$\zeta(\tilde{d}_{kn}, s_{kp}) = \frac{(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Proof

From 4.8.2 we have:-

$$s_{kp} = \frac{\sum_{n=1}^{N} d_{kn}(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

Using system linearity $\tilde{d}_{kn}$ can be assumed to be the only input so this becomes:-

$$s_{kp} = \frac{\tilde{d}_{kn}(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

$$\therefore \zeta(\tilde{d}_{kn}, s_{kp}) = \frac{s_{kp}}{\tilde{d}_{kn}} = \frac{(F_{kp}D_{kp}L_{kp} - I_{kpn})S_{kp}}{1 + L_{kp}S_{kp}D_{kp}}$$

APPENDIX III

CONVOLUTION AND SCALING OF DISTRIBUTIONS

Convolution

If $\tilde{a}$ and $\tilde{b}$ are distributions of independent random variables a and b respectively, then the distribution $\tilde{c}$ of the random variable $c = a + b$ is the *convolution* of $\tilde{a}$ and $\tilde{b}$.

If $\tilde{a}$ and $\tilde{b}$ are expressed as histograms then $\tilde{c}$ is derived as:-

$$\tilde{c}(c) = \sum_{a+b=c} \tilde{a}(a)\tilde{b}(b) \qquad \text{for all } c$$

Scaling

Scaling of distributions is a multiplication of the random variable by a scalar.

If $b = s\,\tilde{a}$ then $\tilde{b}$ can be derived as:-

$$\tilde{b}(b) = \tilde{a}\left(\frac{b}{s}\right) \qquad \text{for all } b$$

APPENDIX IV

Brief Table of z-Transforms

TIME-SERIES		z-Transforms
Unit Impulse at time k	$f(t) = 1 \;, t = k$ $= 0 \;, t \neq k$	$f(z) = z^{-k}$
Unit Step at time k	$f(t) = 0 \;, t < k$ $= 1 \;, t \geq k$	$f(z) = \frac{z}{z-1} z^{-k}$
Unit Ramp at time k	$f(t) = 0 \;, t \leq k$ $= t-k \;, t \geq k$	$f(z) = \frac{z}{(z-1)^2} z^{-k}$
Unit sine frequency ω	$f(t) = \sin(\omega)$	$f(z) = \frac{z \sin \omega}{z^2 - 2 z \sin \omega + 1}$

BIBLIOGRAPHY

Ansoff, H.I. 1968.

An Appreciation of Industrial Dynamics.

Management Science, Vol 14, No. 7, pp. 383-397.

Axsater, S. 1978.

Balance of Integrated Production-Inventory Systems.

Advances in Operational Research, North Holland.

Beightler, C.S., Mitten, L.G & Nemhauser, G.L. 1961.

A Short Table of Z-Transforms and Generating Functions.

Operations Research, Vol. 9, No. 4, pp. 574-578.

Bensoussan, A. 1972.

A Control Theory Approach to Production Models of the HMMS Type.

European Institute for Advanced Studies in Management, Working Paper 72-19.

Bensoussan, A. 1972.

Some Simple Examples of the use of Control Theory in Production Systems.

European Institute for Advanced Studies in Management.

Bensoussan, A. & Hurst, G.E. Jr. 1972.

A General Approach to Inventory Modelling through Control Theory.

European Institute for Advanced Studies in Management, Working Paper 72-21.

Bishop, A.B. 1975.

Introduction to Discrete Linear Control - Theory and Applications.

Academic Press.

Box, G.E.P. & Jenkins, G.M. 1970.

Time-Series Analysis : Forecasting and Control.

Holden-Day.

Bradshaw, A., Mak, K.L. & Porter, B. 1976.

Modal Control of Production-Inventory Systems Incorporating unfilled Order Backlogs.

International Journal of Systems Science, Vol. 7, No. 1.

Bradshaw, A. & Porter, B. 1973.

Optimal Control of Production-Inventory Systems.

International Journal of Systems Science, Vol. 4, No. 5.

Bradshaw, A. & Porter, B. 1974.

Modal Control of Production-Inventory Systems using Piecewise-Constant Control Policies.

International Journal of Systems Science, Vol. 5, No. 8.

Bradshaw, A. & Porter, B. 1975.

Synthesis of Control Policies for a Production-Inventory Tracking System.

International Journal of Systems Science, Vol. 6, No. 3.

Brown, R.G. 1963.

Smoothing, Forecasting and Prediction of Discrete Time-Series.

Prentice-Hall.

Campbell, D.P. 1953.

Dynamic Behaviour of Linear Production Systems.

Mechanics Engineer 75, pp. 279-283.

Coyle, R.G. 1977.

Management Systems Dynamics.

John Wiley & Sons Ltd., London.

Crossley, T.R. & Porter, B. 1972.

Synthesis of Control Policies for Manufacturing Systems using Eigen-Value Assignment Techniques.

International Journal of Systems Science, Vol. 3, No. 2.

Elmagrabhy, S.A.E. 1959.

On the Feedback Approach to Industrial Systems Design.

Proceedings of the sixth International Meeting of the Institute of Managements Sciences.

Foo, K.L. 1978.

A Study of Industrial Dynamics Modelling and its Comparison with the Discrete Z-Transform Technique.

University of Nottingham, Department of Production Engineering and Production Management.

Forrester, J.W. 1961.

Industrial Dynamics.

John Wiley & Sons, New York.

Forrester, J.W. 1968.

Industrial Dynamics - A Response to Ansoff & Slevin.

Management Science, Vol 14, No. 9, pp. 600-618.

Holt, Modigliani, Muth & Simon. 1960.

Production, Inventories and Workforce.

Prentice-Hall, London.

Howard, R.A. 1963.

Systems Analysis of Linear Models.

Multistage Inventory Models and Techniques; EDS: H.E. Scarf, D.M. Gilford, M.W. Schelly, Stanford University Press.

Inderfurth, K & Schneeweis, C.H.

Suboptimal Policies for Stochastic Cash Balance Problems: A comparison of Linear Decision Rules and Rolling Horizon Deterministic Optimisations.

Jury, E.I. 1964.

Theory and Application of The z-Transform Method.

John Wiley & Sons.

Lewis, C.D. 1963.

Analogue Computer Applications to Production Control and Allied Fields.

International Journal of Production Research.

Lewis, C.D. 1967.

An Iterative Analogue Inventory Policy cost Method.

International Journal of Production Research, Vol. 6, No. 2.

Lewis, C.D. 1967.

Hybrid-Analogue Simulation of Stochastic Industrial Storage Problems.

Journal of Industrial Engineering, Vol. 18, No. 4.

Muth, E.J. 1977.

Transform Methods with Applications to Engineering and Operations Research.

Prentice-Hall, U.S.A.

Pinkham, R. 1958.

An Approach to Linear Inventory Production Rules.

Operations Research, Vol. 6, No. 2.

Porter, B. & Taylor, F. 1972.

Modal Control of Production-Inventory Systems.

International Journal of Systems Science, Vol. 3, No. 3.

Roberts, E.B. 1977.

Managerial Applications of Systems Dynamics.

John Wiley & Sons.

Sargent, R.G. 1966.

A Discrete Linear Control Theory Inventory Model.

University of Ann Arbor, Michigan.

Schneeweis, C.H. 1975.

Dynamic Certainty Equivalents in Production Smoothing Theory.

International Journal of Systems Science, Vol. 6, No. 4.

Simon, H.A. 1952.

On the Application of Servomechanism Theory in the Study of Production Control.

Econometrica 20, pp. 247-267.

Truxall, J.G. 1955.

Automatic Feedback Control System Synthesis.

McGraw-Hill

Vassian, H.J. 1955.

Application of Discrete Variable Servo-Theory to Inventory Control.

Operations Research 3, No. 4, pp. 272-282.

Wight, O.W. 1974.

Production and Inventory Management in the Computer Age.

CBI Publishing Co. Inc., Boston.

SYMBOLS IN GENERAL USE

d	demand for a product or part, or schedules for supply of a part.
f	forecast gross requirement for a product or part.
i	issues from stock of a product or part.
r	receipts into stock of a product or part.
D	supply scheduling for a product or part.
F	gross requirement forecasting for a product or part.
I	product delivery policy or part issue pattern.
K	number of levels controlled by system.
L	product or part supply by lead-time.
M	number of products controlled by system.
N	number of parts controlled by system.
S	Stock integration.
$T(x,y)$	system z-transfer function from time-series x to time-series y.
$\zeta(x,y)$	sub-system z-transfer function from time series x to time-series y.
$\tilde{}$	$\tilde{x}$ is the distribution of noise superimposed upon time series x.

GENERAL

1) Upper-case letters denote z-transfer functions.
2) Lower case letters denote time-series or their z-transforms.
3) T and ζ denote system and sub-system z-transfer functions, respectively.
4) Subscripts:-
   (i) first subscript denotes a level.
   (ii) second subscript denotes the controlled part.
   (iii) third subscript denotes a relevant part at a higher level.
5) The following symbols have local significance only, and are defined where they occur.

$a, b, g, h, j, k, m, n, p, q, w, x, y,$

$A, B, G, H, P, T, U, V, W, X, Y,$

$\alpha, \delta, \varepsilon, \zeta, \theta, \sigma, \rho, \mu, \psi, \lambda, \omega, \Delta, \nabla, \Phi, \Theta, \Psi,$

$\hat{}, s'$