'THE APPLICATION OF OPTIMAL CONTROL THEORY

TO A CLASS OF EXTREMUM CONTROL SYSTEMS'

A thesis submitted to

THE UNIVERSITY OF OXFORD

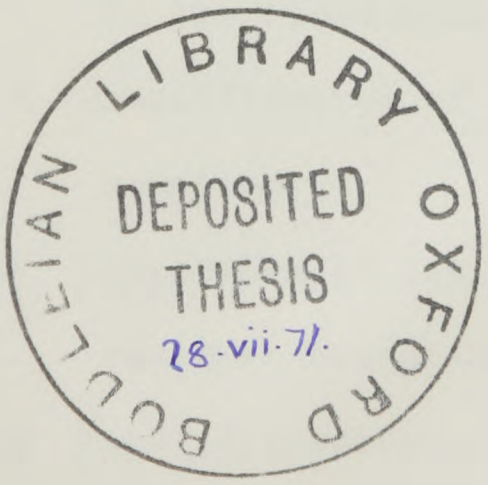
FACULTY OF PHYSICAL SCIENCES

by

STANLEY MICHAEL LANGDON

for the degree of

DOCTOR OF PHILOSOPHY



in

1970

FORWARD ON NOMENCLATURE

The phrases 'extremum control' and 'extremal control' are used interchangeably in the literature. The phrase 'extremum control' appears in the title of the thesis as this corresponds to the official title of the thesis. The phrase'extremal control' is used in the remainder of the thesis as it then corresponds to the use of the established phrase 'optimal control'.

CONTENTS

Acknowledgements

Abstract

Principal Symbols

CHAPTER 1 Introduction 1

1.1 Extremal control systems 1

1.2 Applications and potential applications of extremal control 3

1.3 The development of extremal controllers 7

1.4 The incentive to use optimal control theory 10

1.5 Optimal control theory 11

1.6 The contents and contribution to automatic control of this thesis 13

CHAPTER 2 The Formulation and Formal Solution of the Extremal Control Problem as an Optimal Control Problem 15

2.1 A mathematical model for an extremal control process 15

2.2 The application of optimal control theory 22

CHAPTER 3 The Simplification and Numerical Solution of the Functional Recurrence Equation 29

3.1 Simplification of the functional recurrence equation 29

3.2 Properties of the iterates of the functional recurrence equation 36

3.3 Numerical solution of the functional recurrence equation 38

3.4 The optimal extremal controller 57

CHAPTER 4 Simulation of the Optimal Extremal Control System 58

4.1 Digital simulation 59

4.2 Hybrid simulation 61

CHAPTER 5 Comparison of the Performance of the Optimal Extremal Controller with the Performance of other Extremal Controllers 69

5.1 The effect of output noise on the performance of the optimal extremal controller 70

5.2 Comparison with the performance of heuristic controllers 71

5.3 Comparison with the performance of controllers based on optimal control theory 75

CHAPTER 6 The Application of Optimal Control Theory to more Complex Extremal Processes 79

6.1 Optimal control of an extremal process with an output lag 80

6.2 Optimal control of an extremal process with measurement noise 83

6.3 Optimal control of a multi-input extremal process 86

CHAPTER 7 Conclusions 89

APPENDIX 1 Proof of the Properties of the Iterates of the Functional Recurrence Equation 97

APPENDIX 2 The Description of Digital Computer Programs 105

APPENDIX 3 Digital Computer Programs Used in Solving the Functional Recurrence Equation 109

APPENDIX 4 Details of Computer Simulations of Extremal Control Systems 124

References 138

Figures

Tables

Acknowledgements

The author wishes to acknowledge the many hours of fruitful discussion with his supervisor, Dr. O.L.R. Jacobs, without whose guidance this thesis would not have been realised. Thanks are also due to the following: the Science Research Council for providing a grant for this work; St. John's College, Oxford, the Department of Engineering Science, Oxford, and particularly the Oxford University Systems Engineering Group for providing equipment and facilities; Professor W.E.J. Farvis of the University of Edinburgh, Department of Engineering Science where this work originated, and finally Miss H.F. Langdon for hours of patient typing.

Abstract

This thesis constitutes a contribution to the theory of extremal control systems which come within the automatic control field of learning in engineering science.

An extremal control system comprises an extremal process (which is to be controlled) and an extremal controller. The class of extremal control systems referred to in the title of the thesis refers to those systems in which the extremal process concerned can be represented by a particular form of mathematical model.

Investigation of the application of optimal control theory to this class of extremal control systems is motivated by two basic requirements. The first of these originates in the control of complex industrial processes, particularly in the direct digital control situation. Here there are a number of possible applications for extremal control, and since costs involved are generally high, the requirement is generally for extremal controllers maintaining as low a system operating cost as possible, rather than for controllers with a simple structure. Hence there is a need for optimal extremal controllers.

The second requirement for establishing optimal extremal controllers lies in the design of simpler extremal controllers. The performance of the optimal extremal controller is required as a basis on which to judge the simpler controllers and decide whether any more significant improvement can be made.

The major part of this investigation of the application of optimal control theory to extremal control systems is concerned with the simplest possible case, where no account is taken of input or output lags or noise, or of multiple inputs. Since no

previous optimal extremal controllers are known this is the natural starting point.

A rigorous analysis of state variables, probability distributions and sufficient statistics for this simplified case is presented, and this leads on to a new approach to the application of dynamic programming which in turn results in an original functional recurrence equation. Analysis of this functional recurrence equation leads to a numerical solution procedure including many checks. This eventually establishes the optimal extremal controller.

Simulation techniques are used to confirm the performance of the optimal extremal controller, and hybrid computing facilities used to show that it can be implemented on a small on-line computer and used in a direct digital control situation without suffering from interfacing effects. Thus for the first time there is now available an optimal extremal controller, and moreover it can be confidently expected to perform optimally in a practical situation.

Comparison of the performance of the optimal extremal controller with the performance of simpler controllers shows that, in the simplified case, there is still room for considerable improvement on the simple controllers, and a quantitative measure of just how much improvement might be possible is now available.

An initial investigation of the application of optimal control theory to more complex processes, involving noise, lags and multiple inputs, is presented and this shows that theoretical difficulties are likely to prevent further optimal extremal controllers from being established. In these cases there is therefore a requirement to establish closely optimal controllers. One approach would be just to use the optimal controller from the

simplified case in these more complex situations. Simulations are presented establishing the performance of the optimal extremal controller, and a suboptimal extremal controller, controlling not the process for which they are designed b t a similar process involving output noise. These simulations show that these controllers are certainly not optimal in the more complex situation. A different approach must therefore be taken to establishing closely optimal controllers in the more complex situation. The thesis finishes with a discussion of how this might be achieved.

List of Principal Symbols

This list contains only the principal symbols used in the thesis. Opposite each symbol are the most relevant page numbers. Some of the symbols are used locally (particularly in chapter 6) with different meanings. These cases are clearly indicated in the text.

Symbol	Meaning	Page
a	Fixed intentional error	76
A	Extremal process constant	18
c(i)	Stage cost	12,18
$\bar{c}$	Mean stage cost	17,21
c^*	Optimal mean cost per stage	34,55
c_N^*	Substitution variable	34
dq	Approximation parameter	41,47
dr	Approximation parameter	45,47
dv	Approximation parameter	42,50
dw	Approximation parameter	41,47
$e[w,q]$	Optimal intentional error function	77
exp	Exponential	—
Exp	Mathematical expectation	—
$f[w,q]$	Optimal transient cost function	34,56
$f_N[w,q]$	Substitution cost function	34
$F_N[u,w,q]$	Optimal cost function	26,31
$\bar{F}_N[\cdot]$	Substitution cost function	31,32
$I_N[w,q]$	Minimised iteration integral	38
$\bar{I}_N[w,q,v]$	Iteration integral	40
K	Integrator gain	62,66
L	Stages/group of simulation	61
M	Number of stages in a process	12,13

M	Number of groups/simulation	61
n(t)	Simulated white noise signal	62
N	Number of stages remaining	13
$p[\cdot]$	Gaussian probability distribution	23,31
$p_O[\cdot]$	A priori probability distribution	24
$p_i(\Theta)$	A posteriori probability distribution	88
P1 → P4	Probability variables	25,31,33
q,q(i)	Sufficient statistic	23,26
q^+,q^-	A priori stochastic variables	26,33
$Q[\cdot]$	Updating function	30,33
r,r(i)	Random sample, input	19,26,31
$\underline{r}(i)$	Vector random sample, input	12
R	Approximation parameter	45,48
T	Control period	12,62,64
u,u(i)	Control input, sufficient statistic	18,23,26,31
$\underline{u}(i)$	Vector control input	12
u1(i),u2(i), ...	Elements of $\underline{u}(i)$	16
$u_N[\cdot]$	Optimal control function	26,31
v,v(i)	Control increment	32,57
$v[w,q]$	Optimal control function	55
$v_N[w,q]$	Optimal control function	33
v_N	Optimal control constant	37
w,w(i)	Sufficient statistic	23,26,31
w^+,w^-	A priori stochastic variables	26,31,33
W	Approximation parameter	41,46
x(i)	Extremal process variable	19
$\underline{X}(i)$	State vector	12,20
X1(i),X2(i), ...	Elements of $\underline{X}(i)$	20

$y(i)$	Measurement	18
$\underline{y}(i)$	Vector measurement	12
$y1(i), y2(i), \ldots$	Elements of $\underline{y}(i)$	17
$z(i)$	Brownian motion disturbance	18
$\delta_N[\cdot]$	Iteration function	39
$\xi(i)$	Output noise	70
$\pi_1 \rightarrow \pi_3$	Dimensionless variables	70,72,77
σ	Extremal process constant	19
σ_1	Standard deviation (output noise)	70
σ_i	Standard deviation (integrated n(t))	65
σ_n	Standard deviation (n(t))	64
ϕ	Power spectral density	63
ψ	Autocorrelation function	63,65

CHAPTER 1

Introduction

The purpose of this thesis is to present some new results which, it is felt, extend in a significant manner the present state of knowledge about extremal control systems[1], known also as hill-climbing systems, optimalising systems or adaptive[2] systems. These systems are part of the automatic control[3,4] field of learning in engineering science. This introductory chapter is intended first to present a background of information about extremal control systems so as to show that further study is justified and that the direction taken in this work i.e. the application of optimal control theory[5] is a reasonable one. Given this approach to the further study of extremal control systems the chapter goes on to give the major results from the theory of optimal control which are necessary for the understanding of the remainder of the thesis. Having established this background the chapter finishes by describing the layout of the remaining chapters of the thesis and pointing out the new contributions which are made.

1.1 Extremal control systems[6]

An extremal control system, as with most control systems, may be considered to be composed of two elements - a controlled object, and a controller. General features of the controlled object are the adjustments which can be made to it - the inputs - and the measurements which can be made on it - the outputs - together with the fact that as it

operates it incurs a cost, or alternatively produces a return. It will be assumed that all objects considered incur a cost, though all arguments can be inverted to deal with the alternative case.

The inputs, outputs and cost are not independent features, but are related - though not necessarily in deterministic manner - according to the physical laws governing the controlled object. When this relation satisfies three basic conditions the controlled object may be termed an 'extremal process'. The first condition is that the rate at which the total operating cost of the object is increasing at any given time - the cost rate - is a function of some system parameters at that time. The form of the function is such that it has a unique minimum with respect to the parameters at all times; otherwise there are no restrictions. As time proceeds, random effects cause the exact form of the function to change, and in particular the parameter values which minimise the function - the optimum parameter values - change. Figure 1.1.1 shows how the form of the function might change as time t proceeds from t_1 through t_2, t_3, t_4, when there is only one parameter.

The second condition which must be met for a controlled object to be an extremal process is that the adjustments which can be made must be directly related to the parameters. By this is meant, in the simplest case, that they should be identical. However two effects may be considered to intervene between each adjustment and the corresponding parameter. The first is that an adjustment is not generally reflected in an immediate change of cost rate i.e. there may be an input lag.

The second is that adjustments can rarely be made precisely i.e. there may be input noise.

The final condition determining a controlled object as an extremal process is that all measurements should be derived from the cost rate. In the simplest case there will be only one measurement which will be the cost rate. However measurements are rarely made instantaneously or precisely, so that output lags and output noise may have to be allowed for. When this is taken into account the possibility of having several measurements to compensate for lags and noise arises.

Ah extremal control system is a control system in which the controlled object is an extremal process, and hence the controller is termed an 'extremal controller'. The object of the extremal controller is to carry out the adjustments to the extremal process on the basis of the measurements, so as to minimise the average operating cost of the system while satisfying any constraints and allowing the extremal process to carry out its function. Figure 1.1.2 shows a block diagram of the configuration of an extremal control system.

1.2 Applications and potential applications of extremal control

One of the main justifications for continuing to study extremal control systems lies in the wide variety of potential applications[7-14], but an extremal controller will generally only be used if it can be justified economically. This means that the saving in operating cost of the extremal process in question by use of extremal control must compare favourably with the capital cost of installing an extremal controller.

Unfortunately this consideration has led to many potential applications being missed.

One of the earliest investigations into extremal control[7] shows this well. The extremal process considered was an internal combustion engine operating with constant fuel supply rate, and at constant speed. Under these conditions the brake mean effective pressure can be regarded as a return rate - analagous to cost rate for a process giving a return to be maximised. The total return from the engine in a given time is the total mechanical energy produced, found by integrating with respect to time the brake mean effective pressure, multiplied by various constants. Analysis shows that air/fuel ratio and ignition timing can be regarded as inputs to an extremal process in which the output and return rate are the brake mean effective pressure. Although much work was done investigating extremal controllers in this situation, it has never been found economic to apply extremal control to internal combustion engines in anything other than this experimental scale. This is because conventional means of adjusting air/fuel ratio and ignition timing allow internal combustion engines to operate sufficiently close to maximum efficiency to prevent the economic addition of even the simplest extremal controller.

Greater success has been met more recently considering the focussing of a scanning microscope as an extremal control problem[8]. Basically the microscope is well focussed if the mean square value of the high frequency components of the brightness signal is kept as high as possible. This is

therefore taken as the return rate and measurement of an extremal process, where the adjustment to be made is the position of the objective lens. An extremal controller for this application has been designed and is in practical use The success of extremal control in this situation lies in the fact that without it the microscope does not operate satisfactorily because it must be refocussed manually at frequent intervals.

Experiments have been carried out with a theoretical background to investigate the use of extremal control techniques in servomechanism loops[9], by adjusting parameters in both feedback and feedforward paths to minimise mean-squared error. The change in optimum settings is here due to random variations of plant parameters, and it has been suggested that this might be useful in aircraft control loops[10] where large variations in atmospheric conditions as the flight proceeds cause large variations in the performance of the aircraft control loops. However no definite practical application is known in this direction.

Basically it seems that such simple structures as internal combustion engines and servomechanisms generally operate at cost levels such that it is difficult for extremal control to make an economic improvement. To try to find more suitable applications, it is therefore desirable to consider processes operating at higher cost levels where the capital cost of installing an extremal controller may be relatively less expensive, and hence there will be a greater chance of successful application. It now seems that such a possibility may be found with the advent

of relatively cheap on-line computers and the applications of direct digital control to industrial processes[11]. With the computer already provided for the direct digital control it seems quite possible that the addition of extremal control to conventional control may provide substantial savings at a small additional cost.

An investigation has been carried out into such a possibility recently, in the control of the air/fuel ratio of a power station boiler[12] so as to provide the required power with minimum average coal consumption. The optimum ratio varies with grade of coal, and so the possibility of extremal control arises. Results show encouraging prospects for the larger power stations which are now envisaged.

Consideration has also been given to general chemical processes[13] where the yield of desired product must be kept maximised by adjusting the relative quantities of raw materials to compensate for variation of quality. Prospects here are also encouraging.

The model reference control technique[14] is another possible approach to controlling these industrial processes which is being considered. Extremal control may prove useful within this concept, either to adjust process parameters to minimise mean-squared deviation from model performance, or for indentification purposes to adjust model parameters to minimise mean-squared deviation from process performance.

The conclusion to this investigation of potential applications of extremal control is that prospects are encouraging, although success is more likely to be found in the control of the higher

cost industrial processes rather than every-day devices. Further investigation of extremal control systems should therefore be made with these possible applications in mind. The possible use of extremal controllers in the lower cost area should not be forgotten about and it may still be profitable to try to find simple extremal controllers which will maintain a lower average system operating cost than is possible at present.

1.3 The development of extremal controllers

The variety of potential applications was given in the last section as one justification for continued study of extremal control systems. Certainly another justification can be given in the fact that to date the design of extremal controllers for various extremal processes has yielded a series of interesting theoretical problems. This is probably due to the combination of random effects with non-linearity which can take quite a simple form initially, and then become more complex as effects of lags, noise and multiple inputs are taken into account singly and together.

In order to design an extremal controller it is necessary to know something of the characteristics of the extremal process concerned, and also the restrictions of the hardware available from which the controller must be realised. Initially hardware restrictions were sufficiently great to limit the choice of control algorithm to a few simple forms, and these were optimised by direct experiment on the process concerned, but this can have a major drawback if disturbances to the process while experiments are carried out are significant. It has therefore come to be

almost standard practice to construct a mathematical model for the extremal process and test extremal control algorithms using computer simulation techniques before direct application to the process. This will certainly be desirable in future if expensive industrial processes are being controlled.

Once simulation became feasible, restrictions on the complexity of extremal control algorithms being tested became negligible and investigation of the performance of various extremal control algorithms became an interesting academic pursuit in complete abstraction from possible application. A considerable body of knowledge has therefore been built up of extremal control algorithms[7,15-26]. To follow their development it is convenient to distinguish between continuous-time algorithms and discrete-time algorithms. A continuous-time algorithm is one which adjusts the input to the process continuously, and makes a continuous measurement. A discrete-time algorithm is one which adjusts the inputs and makes measurements only at fixed points in time.

Taking continuous-time single-input algorithms first, one of the earliest and simplest is the peak-holding type[7]. This simply changes the input to the process at a constant rate until the measurement indicates that the cost rate is increasing. At this point the direction of input change is reversed, and the procedure repeats indefinitely. To improve on this system it is desirable to try to estimate the gradient of the cost rate v. parameter function at the current operating point, and hence by allowing the basic rate of change of the input to be variable, it can be related to this estimated gradient. The basic method generally employed to estimate the

radient is that of dual control[15], where the input is made up of two parts. The first is the basic control input already described, with the second, a perturbation superimposed, and intended specifically to help determine an estimate of the gradient by correlation with the corresponding perturbation in the output. Periodic perturbations, both sine wave[16] and square wave[17] were used initially, and this has since been extended to pseudo-random[18] and random[19] (sometimes naturally occurring) perturbations.

The simplest of the discrete-time, single-input algorithms is analagous to the peak-holding type - the input is changed in constant steps until the cost-rate appears to increase, when the direction of stepping is reversed[20-22]. Again an estimate of the gradient of the cost rate v. parameter function at the current operating point is desirable, this time so that the basic step size can be varied with the estimated gradient. The estimate can be made by making a short step prior to every control step, known as the derivative sensing system[21,23], or again by application of dual control techniques, a bias of fixed size but alternate direction can be added to each control step, known as the alternate biasing system[21,23]. Gradient estimation is again by correlation of the bias with the corresponding perturbation in the output.

Multiple input systems have been considered, and one of the simplest suggestions is to perform extremal control of each input sequentially[7,24]. An alternative for continuous-time systems is to use perturbations of different frequencies on each input. For discrete-time systems the method of steepest

descent[25] has been proposed, but suffers from dependence on the relative scales on which the inputs are measured. This has been overcome by the method of conjugate gradients[26] which retains results of adjustments made over a longer period. Further progress in investigating multiple-input systems is made difficult by such effects as ridges on the cost-rate v. parameter function.

The conclusion to this section is that considerable progress has been made in investigating extremal control algorithms, but much work remains to be done. Past experience suggests that to follow along this line will provide problems of considerable academic interest, but some attempt must be made to try to maintain investigations within sight of practical needs.

1.4 The incentive to use optimal control theory

The last two sections have shown that further study of extremal control systems seems worthwhile both for possible practical application and for academic interest. The question which arises at this point is what direction the further study should take. It has been seen that the design of extremal control algorithms has been departing from simple structures in order to find improved performance, i.e. controllers which will maintain a lower average system cost. Returning to applications it is seen that for the proposed area of interest in the direct digital control field, low average system cost is now much more important than simplicity. Thus from both points of view the direction to be taken is towards improved performance in terms of average system cost. Low cost systems would also benefit from such investigations because the performance of the best complex extremal controllers will provide a basis on which to judge simple extremal controllers

and decide when further variations can make only a neglgible improvement.

It would be possible to go on indefinitely proposing extremal control algorithms and perfecting them by analysis and simulation, for processes involving lags, noise and multiple inputs. However the question arises whether there is a limit to the possible performance of an extremal controller i.e. whether there is an optimal extremal control algorithm. This question has been raised before[27-32] and the appropriate analytic tool for deriving optimal control algorithms - optimal control theory[33] - has been used together with approximations to give approximately optimal extremal control algorithms[27,28], and combined with intuitive ideas to give suboptimal extremal control algorithms[29]. It is the purpose of this thesis to present the results of continued study of extremal control systems along these lines[32], but before defining the contribution more precisely it is desirable to digress to present the basic facts about optimal control theory which are used in the remainder of the thesis so that the background is completely established.

1.5 Optimal control theory

Optimal control theory[33] is a general name for a variety of philosophies for designing automatic control algorithms. They all require a mathematical model of the controlled object to be constructed, relating adjustments, measurements and costs, and give as a result that algorithm which is best in a precise mathematical sense. The particular philosophy used depends upon the form of the mathematical model. It will be seen that the models used to represent extremal processes in this thesis constitute discrete-time stochastic control processes[34], and are

chosen this way to make the application of optimal control theory feasible while remaining representative of extremal processes. The class of extremal control systems referred to in the title of the thesis consists of those extremal control systems which can be reasonably described by these models. The appropriate form of optimal control theory in this situation is dynamic programming[35,36].

A discrete-time stochastic control process which is sufficiently general for the purposes of this thesis can be defined as follows. It is a multi-stage process such that at time $t = iT$, where $i = 1,2, \ldots$, and T is a fixed period, there are:-

(1) a control input $\underline{u}(i)$ the elements of which are the chosen inputs or adjustments.

(2) a random input $\underline{r}(i)$ which is a sample from a fixed (vector) probability distribution, independent of samples at other stages.

(3) an output $\underline{y}(i)$ the elements of which are the measurements.

(4) a stage cost $c(i)$.

Both $\underline{y}(i)$ and $c(i)$ are determined by values of $\underline{u}$ and $\underline{r}$ at stages up to and including i.

A vector $\underline{X}(i)$ satisfying the conditions

$$\underline{X}(i+1) = \underline{f}\left[\underline{X}(i), \underline{u}(i+1), \underline{r}(i+1)\right] \qquad 1.5.1$$

$$\underline{y}(i+1) = \underline{Y}\left[\underline{X}(i+1), \underline{u}(i+1), \underline{r}(i+1)\right] \qquad 1.5.2$$

$$\text{and} \quad c(i+1) = C\left[\underline{X}(i+1), \underline{u}(i+1), \underline{r}(i+1)\right] \qquad 1.5.3$$

for $i = 0,1,2, \ldots$, where $\underline{f}$, $\underline{Y}$ and C are known functions is a state vector for the process. The elements of $\underline{X}$ are the state variables.

For a multi-stage process of M-stages, the optimal control

problem is to choose the series of control inputs $\underline{u}(i)$, $i = 1,2, \ldots , M$ so as to minimise the expected total cost $\sum_{i=1}^{M} c(i)$, where expectation is taken over the probability distribution for the initial state vector $\underline{X}(0)$ and the probability distribution for the series of random inputs $\underline{r}(i)$, $i = 1,2, \ldots , M$.

If after i stages of the process there are N stages to go then optimal control theory gives the result[34] that the optimal choice of $\underline{u}(i+1)$, termed $\underline{u}_N(i+1)$ should be based on the probability distribution for $\underline{X}(i)$. If for all i, $i = 1,2, \ldots , M$ this probability distribution can be defined by a finite set of numbers, these are called sufficient statistics, and $\underline{u}_N(i+1)$ is a function of the sufficient statistics for $\underline{X}(i)$.

The actual form of the function can be determined from the principle of optimality[34] which is the basis of dynamic programming. This principle states that if an M stage discrete-time stochastic control process has control inputs chosen optimally, then the final N control inputs constitute an optimal choice of control inputs for the final N stages considered as a discrete-time stochastic control process alone. This general statement is true for $N = 1,2, \ldots , M$ and can be interpreted as a recurrence relation which can in principle be iterated to solve the optimal control problem in this case for $M = 1,2,3, \ldots$

1.6 The contents and contribution to automatic control of the thesis

Chapter 2 starts by deriving a mathematical model for a very simple extremal process and posing the optimal control

problem as has been done before[27-31]. It then goes on to find a formal solution in terms of a functional recurrence equation which is original. In chapter 3 properties of this equation are derived and used to facilitate a numerical solution which is then described. The result is thought to be the first optimal extremal control algorithm reported, excluding two which are known to be erroneous[30,31]. (See chapter 5).

As has been said, it is important to test the performance of extremal control algorithms by simulation, and results of simulation of the optimal extremal control system are given in chapter 4. This is done first in a digital computer for the purely discrete-time case, and then using hybrid computing facilities for the analagous, but more realistic case when the process is continuous-time, but the controller is discrete-time. Chapter 5 compares and discusses the performance of the optimal extremal control algorithm in relation to other extremal control algorithms. In order to make the comparison wider results are given of the simulated performance of the optimal extremal controller arbitrarily applied to a process with output noise. Original results are also given of the simulated performance of a suboptimal extremal control algorithm applied to an extremal process with output noise.

The optimal extremal control algorithm derived is for an extremely simple extremal process, and so chapter 6 goes on to outline the possibilities, and difficulties of applying optimal control theory to more complex processes. Finally in chapter 7 the results of this work are summarised and used as a background to a further discussion of the prospects for extremal control systems and in particular the role of optimal control theory.

CHAPTER 2

The Formulation and Formal Solution of the Extremal Control Problem as an Optimal Control Problem

Continuing from the introduction which has already been given this chapter starts by deriving a mathematical model for an extremal control process which is sufficiently simple to allow optimal control theory to be applied, but at the same time remains representative of the main characteristics of extremal control processes. The limitations of the model are also discussed. The formal mathematical procedures required to apply optimal control theory are then followed through, so that the chapter ends with the optimal extremal control algorithm for the model of the extremal control process defined as the solution of a functional recurrence equation.

2.1 A mathematical model for an extremal control process

As has been stated in section 1.5 the first stage in the application of optimal control theory to the design of a controller is the construction of a mathematical model to represent the controlled object. When both non-linear and random effects are present simultaneously, as is the case when the controlled object is an extremal process, the optimal control technique most likely to lead to a solution is dynamic programming[35,36]. Other techniques are liable to lead to multi-dimensional two-point boundary value problems which are generally difficult if not impossible to solve.[34] The form of the mathematical model most suited to dynamic programming in this case is a discrete-time stochastic control process which

has already been defined in section 1.5.

The derivation of a mathematical model of an extremal process will be taken in two stages. First the three basic features of an extremal process outlined in continuous-time terminology in section 1.1 will be reformulated in discrete-time terms. Then the mathematical representation chosen for each feature will be presented, and taken together they will be shown to form a mathematical model of an extremal process which constitutes a discrete-time stochastic control process.

A discrete-time extremal control process which is sufficiently general for the purposes of this thesis can be defined as follows. At times $t = iT$, where $i = 1,2, \ldots$, and T is a fixed period

(1) adjustments $u1(i)$, $u2(i)$, ... , are made to the process. These adjustments are the elements of the input or adjustment vector $\underline{u}(i)$

$$\underline{u}(i) = \begin{bmatrix} u1(i) \\ u2(i) \\ \vdots \end{bmatrix} \qquad 2.1.1$$

(2) input lags and noise act on the elements $u1(j)$, $u2(j)$, ... , of $\underline{u}(j)$, $j = 1,2, \ldots$, i so as to determine the parameters $m1(i)$, $m2(i)$, ... , respectively. These parameters are the elements of the parameter vector $\underline{m}(i)$

$$\underline{m}(i) \quad \begin{bmatrix} m1(i) \\ m2(i) \\ \vdots \end{bmatrix} \qquad 2.1.2$$

(3) a cost $c(i)$ - the stage cost- is incurred which is a function of $\underline{m}(i)$. The form of this function is similar to that of the cost rate v. parameter function discussed in section 1.1. It has a unique minimum for all values of

i, i = 1,2, ... , but random effects cause the exact form of the function to vary with i; otherwise there are no restrictions.

(4) output lags and noise act on the costs c(j), j = 1,2, ... , i so as to determine the measurements y1(i), y2(i), ... , which are the elements of the output or measurement vector $\underline{y}(i)$.

$$\underline{y}(i) \equiv \begin{bmatrix} y1(i) \\ y2(i) \\ \vdots \end{bmatrix} \qquad 2.1.3$$

For a finite M- stage process the extremal control problem is to design a controller which acting on the basis of the measurements will make the adjustments so as to try to minimise the total cost C_M,

$$C_M \equiv \sum_{i=1}^{M} c(i) \qquad 2.1.4$$

This is seen to be equivalent to minimising the average stage cost $\bar{c}_M$,

$$\bar{c}_M \equiv \frac{1}{M} \sum_{i=1}^{M} c(i) \qquad 2.1.5$$

For an indefinitely long process an extremal controller must be designed to minimise the average stage cost $\bar{c}$,

$$\bar{c} \equiv \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{\infty} c(i) \qquad 2.1.6$$

Since neither the total cost nor the average cost can be predicted due to the random effects, some form of estimate must be used. How this is incorporated into the extremal controller depends on the philosophy behind the design of the controller.

In formulating a mathematical model to represent an

extremal process, it is now necessary to choose a mathematical representation for each of three basic relationships.

(1) the relationship between the adjustments $\underline{u}(i)$, $i = 1,2, \ldots$, and the parameters $\underline{m}(i)$, $i = 1,2, \ldots$,

(2) the relationship between the parameters $\underline{m}(i)$, $i = 1,2, \ldots$, and the costs $c(i)$, $i = 1,2, \ldots$

(3) the relationship between the costs $c(i)$, $i = 1,2, \ldots$, and the measurements $\underline{y}(i)$, $i = 1,2, \ldots$

Although both input and output lags and noise are generally present when an extremal process is being considered, they are not an essential feature, and in many cases some or even all of these effects may be negligible. In order therefore to start with the simplest case effects of lags and noise will not be considered initially, and it will be assumed that there is only one adjustment to be made at each stage. Omission of output lags and noise means that there can be only one significant measurement at each stage. Thus

$$\underline{u}(i) \equiv u(i) \equiv \underline{m}(i) \equiv m(i), \quad i = 1,2, \ldots \qquad 2.1.7$$

and

$$c(i) \equiv \underline{y}(i) \equiv y(i), \quad i = 1,2, \ldots \qquad 2.1.8$$

The most difficult part of the problem is to find a realistic representation of the stage cost v. parameter function which will not lead to insuperable difficulties in the subsequent mathematics of the optimal control theory. To this end it is assumed throughout the remainder of the thesis that the function has a fixed quadratic shape, and that random changes cause the function to change only by movement in the direction of the abscissa i.e. the function is of the form

$$c(i) = A\left\{u(i) + z(i)\right\}^2 \quad i = 1,2, \ldots \qquad 2.1.9$$

where A is a known constant, and random changes to the function take the form of changes to z(i), i = 1,2, ... The assumption of quadratic shape may be quite realistic in many practical cases in the regions close to optimum where extremal controllers generally operate. However further useful study may be directed towards the results of allowing random effects to move the function also along the direction of the ordinate.

The sequence z(i), i = 1,2, ... , is most conveniently represented by a random walk or Brownian motion defined by

$$z(i+1) = z(i) + r(i+1), \quad i = 0,1,2, \ldots \qquad 2.1.10$$

$$\text{and } z(0) = 0 \qquad 2.1.11$$

where the sequence r(i), i = 1,2, ... , is a sequence of independent samples from a Gaussian probability distribution[37] having zero mean value and standard deviation σ, another known constant.

Figure 2.1.1 shows a block diagram of this mathematical model of a simplified extremal control process. The variable x(i) defined by

$$x(i) \equiv m(i) + z(i) \equiv u(i) + z(i) \qquad 2.1.12$$

is defined for later mathematical convenience.

Equations 2.1.8 ⟶ 2.1.11 show that

$$y(i) = A\left\{u(i) + \sum_{j=1}^{i} r(j)\right\}^2, \quad i = 1,2, \ldots \qquad 2.1.13$$

$$\text{and } c(i) = A\left\{u(i) + \sum_{j=1}^{i} r(j)\right\}^2, \quad i = 1,2, \ldots \qquad 2.1.14$$

i.e. y(i) and c(i) are determined by values of u(j) and r(j), j = 1,2, ... , i. Thus if r(i) is considered as a random input

to the process at stage i, i = 1,2, ... , then the mathematical model derived for the extremal process satisfies all the conditions set out in section 1.5 for it to be regarded as a discrete-time stochastic control process.

Because in general a state vector is not unique, there cannot be a logical sequence by which the state vector is found, although there might be a sequence which would determine a state vector. In the present case a state vector has been found by trial and error to find a vector to fit the conditions set out in equations 1.5.1, 1.5.2 and 1.5.3. The state vector found, $\underline{X}(i)$, is defined by

$$\underline{X}(i) \equiv \begin{bmatrix} X1(i) \\ X2(i) \end{bmatrix} \equiv \begin{bmatrix} x(i) \\ u(i) \end{bmatrix}, \quad i = 1,2, \ldots \qquad 2.1.15$$

$$\text{and } \underline{X}(0) \equiv \begin{bmatrix} X1(0) \\ X2(0) \end{bmatrix} \equiv \begin{bmatrix} x(0) \\ u(0) \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad 2.1.16$$

It then follows that for i = 0,1,2, ... ,

$$\begin{aligned} X1(i+1) &= x(i+1) = u(i+1) + z(i+1) \\ &= u(i+1) + z(i) + r(i+1) \\ &= u(i+1) - u(i) + x(i) + r(i+1) \\ &= u(i+1) - X2(i) + X1(i) + r(i+1) \end{aligned} \qquad 2.1.17$$

$$\text{and } X2(i+1) = u(i+1) \qquad 2.1.18$$

which shows that $\underline{X}(i)$, i = 0,1,2, ... satisfies equation 1.5.1.

$$\text{Also } y(i) = A\{X1(i)\}^2, \quad i = 1,2, \ldots \qquad 2.1.19$$

$$\text{and } c(i) = A\{X1(i)\}^2, \quad i = 1,2, \ldots \qquad 2.1.20$$

which shows that $\underline{X}(i)$, i = 0,1,2, ... satisfies equations

1.5.2 and 1.5.3 respectively and hence can be interpreted as a state vector.

For a finite M- stage process the optimal extremal control problem is to design a controller which acting on the basis of the measurements will make the adjustments so as to minimise the expected total cost, c_M,

$$c_M \equiv \text{Exp} \sum_{i=1}^{M} c(i) \qquad 2.1.21$$

where expectation is taken over the probability distribution for the initial state vector and the probability distribution for the random inputs r(i), i = 1,2, ...

This is seen to be equivalent to minimising the expected average stage cost, $\bar{c}_M$,

$$\bar{c}_M \quad \text{Exp} \frac{1}{M} \sum_{i=1}^{M} c(i) \qquad 2.1.22$$

with expectation taken over the same two probability distributions.

For an indefinitely long process the optimal extremal controller must be designed to minimise the expected average stage cost $\bar{c}$.

$$\bar{c} = \text{Exp} \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} c(i) \qquad 2.1.23$$

again with expectation taken over the same two probability distributions.

So far then a discrete-time extremal control process has been defined; a mathematical model has been derived to represent it; the model has been shown to be a discrete-time stochastic control process; a state vector has been found for it;

and the mathematical requirement of the optimal extremal controller has been set down.

2.2 The application of optimal control theory

The position has now been reached where the formal mathematical procedures of optimal control theory can be applied to determine the optimal control algorithm for the mathematical model of the extremal process just defined. Calculation of the optimal control input $\underline{u}(i+1)$, $i = 0,1,2, \ldots$, of a discrete-time stochastic control process can be taken in two parts.

(1) The calculation of the a posteriori probability distribution for the state vector $\underline{X}(i)$ i.e. the probability distribution when $\underline{y}(i)$ has been measured.

(2) The calculation of $u(i+1)$ as a functional of this probability distribution and the number of stages remaining in the process.

In order to determine the optimal extremal control algorithm it is therefore necessary:-

(1) to determine the updating law by which the a posteriori probability distribution for the state vector $\underline{X}(i)$ is found in terms of the a posteriori probability distribution for $\underline{X}(i-1)$, $\underline{u}(i)$ and $\underline{y}(i)$, $i = 1,2, \ldots$ When $\underline{u}(1)$ is being calculated the a posteriori probability distribution for $\underline{X}(0)$ will be a given initial condition.

(2) to determine the series of functionals corresponding to the number of stages of the process to go, by which $\underline{u}(i+1)$ is found in terms of the a posteriori probability distribution for $\underline{X}(i)$, $i = 0,1,2, \ldots$

2.2.1 Probability distributions The calculation of the a posteriori probability distribution for $\underline{X}(i)$ is carried out using the laws of probability in two stages:-

(1) The a priori probability distribution for $\underline{X}(i)$ i.e. the probability distribution before $\underline{y}(i)$ is measured, is calculated from $\underline{u}(i)$ and the a posteriori probability distribution for $\underline{X}(i-1)$.

(2) The a posteriori probability distribution for $\underline{X}(i)$ is calculated from $\underline{y}(i)$ and the a priori probability distribution for $\underline{X}(i)$.

In the case of the extremal process, the a posteriori probability distribution for X1(i) is summed up by[29]

$$w(i) \equiv |X1(i)| \equiv \sqrt{y(i)/A} \qquad 2.2.1.1$$

and $q(i) \equiv$ probability that X1(i) is positive 2.2.1.2

and X2(i) is exactly known from the relation

$$X2(i) = u(i) \qquad 2.2.1.3$$

Thus u(i), w(i) and q(i) form a set of sufficient statistics for the a posteriori probability distribution for $\underline{X}(i)$.

Calculation of u(i) and w(i) is trivial, but calculation of q(i) in terms of u(i-1), w(i-1), q(i-1), u(i) and y(i) follows the two stages described above.

To calculate the a priori probability distribution for X1(i), it is convenient to define the function p[.] by

$$p[.] \equiv \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-[.]^2 / 2\sigma^2\right\} \qquad 2.2.1.4$$

which is the Gaussian probability distribution with zero mean and standard deviation σ.

Equations 2.1.17 and 2.1.18 show that X1(i) can be written as

$$X1(i) = u(i) - u(i-1) + X1(i-1) + r(i),\ i = 1,2, \ldots \qquad 2.2.1.5$$

Thus the a priori probability distributions for X1(i) given that X1(i-1) takes the possible values +w(i-1) and -w(i-1) respectively are

$$\left.\begin{aligned} & p\left[X1(i) - w(i-1) - u(i) + u(i-1)\right] \\ \text{and}\quad & p\left[X1(i) + w(i-1) - u(i) + u(i-1)\right] \end{aligned}\right\} \qquad 2.2.1.6$$

so that the a priori probability distribution for X1(i) is

$$\begin{aligned} & q(i-1).p\left[X1(i) - w(i-1) - u(i) + u(i-1)\right] \\ & + \overline{1-q(i-1)}.p\left[X1(i) + w(i-1) - u(i) + u(i-1)\right] \\ & \qquad \equiv p_o\left[X1(i)\right] \end{aligned} \qquad 2.2.1.7$$

In this case of an extremal process the second part of the calculation of the a posteriori probability distribution for $\underline{X}(i)$ reduces to calculating q(i) in terms of $p_o\left[X1(i)\right]$ and y(i). This is done by considering the probability P of the joint occurrence y(i) observed and $X1(i) = +w(i)$[29]. This probability may be expressed in two ways:-

P = Probability of observing y(i) given X1(i) = + w(i)

x Prior probability that X1(i) = + w(i).

or

P = Probability that X1(i) = + w(i) given y(i)

x Prior probability that y(i) be observed

Taken in order these probabilities may be expressed as

$$\{\text{Probability of observing } y(i) \text{ given } X1(i) = +w(i)\} = 1$$

$$\{\text{Prior probability that } X1(i) = +w(i)\} = p_0[+w(i)]$$

$$\{\text{Probability that } X1(i) = +w(i) \text{ given } y(i)\} = q(i)$$

$$\{\text{Prior probability of observing } y(i)\} = p_0[-w(i)] + p_0[+w(i)]$$

Equating the two expressions for P gives

$$q(i) = \frac{p_0[+ w(i)]}{p_0[- w(i)] + p_0[+ w(i)]} \qquad 2.2.1.8$$

Equations 2.2.1.4, 2.2.1.7 and 2.2.1.8 may now be combined to give the rule for determining q(i) as

$$q(i) = \frac{P1.\ q(i-1) + P2.\ \overline{1-q(i-1)}}{(P1+P3)\ .\ q(i-1) + (P2+P4)\ .\ \overline{1-q(i-1)}} \qquad 2.2.1.9$$

where

$$\begin{aligned} P1 &= \exp\left\{-\left[w(i) - w(i-1) - u(i) + u(i-1)\right]^2/2\sigma^2\right\} \\ P2 &= \exp\left\{-\left[w(i) + w(i-1) - u(i) + u(i-1)\right]^2/2\sigma^2\right\} \\ P3 &= \exp\left\{-\left[w(i) + w(i-1) + u(i) - u(i-1)\right]^2/2\sigma^2\right\} \\ P4 &= \exp\left\{-\left[w(i) - w(i-1) + u(i) - u(i-1)\right]^2/2\sigma^2\right\} \end{aligned} \qquad 2.2.1.10$$

2.2.2. The optimal control law To determine how the control input u(i+1) should depend on the a posteriori probability distribution for $\underline{X}(i)$ i.e. on u(i), w(i) and q(i), it is first necessary to define precisely the sense in which the extremal controller is to be optimal. Initially it is considered that an M- stage process is being controlled, and the optimal controller must choose the series of inputs u(i), i = 1,2, ... , M so as to minimise c_M as defined by equation 2.1.21.

It is a consequence of the principle of optimality, defined in section 1.5, that the optimal choice of u(i+1) where there are N = M-i stages of the process to go will not depend directly

on M, but on N, together with the sufficient statistics u(i), w(i) and q(i). It is therefore possible to make the following definitions:-

$u_N[u,w,q] \equiv$ Optimal choice of u(i+1) when there are N- stages to go and u(i) = u, w(i) = w and q(i) = q.

$F_N[u,w,q] \equiv \text{Exp} \sum_{j=i+1}^{i+N} c(j)$, the expected total remaining cost when there are N- stages to go and expectation is taken with respect to the a posteriori probability distribution for $\underline{X}(i)$ and the probability distribution for the sequence r(j), j = i+1, i+2, ...,i+N, with u(i) = u, w(i) = w and q(i) = q.

In order to use the principle of optimality to determine the optimal extremal control algorithm, it is necessary to determine a priori stochastic expressions for the possible sufficient statistics which may result at the end of stage (i+1). There are two possibilities depending on whether X1(i) = +w(i) or X1(i) = -w(i) The stochastic expressions for w(i+1) are respectively

$$w^+ = \left| w(i) + u(i+1) - u(i) + r(i+1) \right| \qquad 2.2.2.1$$

$$\text{and } w^- = \left| -w(i) + u(i+1) - u(i) + r(i+1) \right| \qquad 2.2.2.2$$

Corresponding to each of these are stochastic expressions for q(i+1), q^+ and q^- respectively, found by substituting w^+ and w^- in turn in equations 2.2.1.9 and 2.2.1.10. These alternatives occur with probability q(i) and $\overline{1 - q(i)}$ respectively. The expression for u(i+1) is trivial.

Dropping the time indices (i) from u, w and q and (i+1) from r, the expected total cost of the final N-1 stages of the process assuming w(i+1) = w^+ and q(i+1) = q^+, can be written

by virtue of the principle of optimality as

$$F_{N-1}\left[u(i+1),\ w^{+},\ q^{*}\right]$$

Similarly the expected total cost of the final N-1 stages of the process assuming $w(i+1) = w^{-}$ and $q(i+1) = q^{-}$ can be written as

$$F_{N-1}\left[u(i+1),\ w^{-},\ q^{-}\right]$$

Thus the expected total cost of the final N-1 stages of the process can be written as

$$\int_{-\infty}^{\infty}\left\{q.F_{N-1}\left[u(i+1),\ w^{+},\ q^{+}\right] + \overline{1-q}\ .\ F_{N-1}\left[u(i+1),\ w^{-},\ q^{-}\right]\right\} p[r]\ dr$$

The expected cost from the $(i+1)^{TH}$ stage is

$$\int_{-\infty}^{\infty}\left\{qA\ (w^{+})^{2} + \overline{1-q}\ A(w^{-})^{2}\right\}\ p[r]\ dr$$

Thus the expected total cost for all N- stages using optimal policy throughout is $F_N[u,w,q]$ now given by

$$F_N[u,w,q] = \min_{u(i+1)} \int_{-\infty}^{\infty}\left\{q.\left[A(w^{+})^{2} + F_{N-1}\left[u(i+1),\ w^{+},\ q^{+}\right]\right]\right.$$

$$\left. + \overline{1-q}\ .\left[A(w^{-})^{2} + F_{N-1}\left[u(i+1),\ w^{-},\ q^{-}\right]\right]\right\} p[r]\ dr \qquad 2.2.2.3$$

The value of u(i+1) which minimises the integral is a function of u,w,q and is $u_N[u,w,q]$.

When there is one stage of the process to go, the definitions given

$$F_1[u,w,q] = \min_{u(i+1)} \int_{-\infty}^{\infty} A\left\{q.\left[w + u(i+1) - u+r\right]^{2}\right.$$

$$\left. + \overline{1-q}\ .\left[-w+u(i+1) - u+r\right]^{2}\right\} p[r]\ dr$$

$$= A\left[4w^{2}\ q.\ \overline{1-q} + \sigma^{2}\right] \qquad 2.2.2.4$$

and the minimising value of u(i+1) is

$$u_1[u,w,q] = u + \overline{1-2q}\, w \qquad 2.2.2.5$$

In principle the optimal choice of u(i+1) can be determined as a function of u, w and q for any number N of stages to go by repeated iteration of the functional recurrence equation2.2.2.3 starting from equation 2.2.2.4.

CHAPTER 3

The Simplification and Numerical Solution of the Functional Recurrence Equation

This chapter is concerned with determining an explicit solution of the optimal extremal control problem which has been set and solved implicitly in the previous two chapters. To do this it is necessary to solve the functional recurrence equation derived in section 2.2 This is done in three stages. First the equation is simplified analytically as far as possible. Then properties of the iterates are found analytically. Finally these properties are used to facilitate the formulation of a numerical solution procedure which is described and then applied to solve the functional recurrence equation. The chapter ends by bringing together the results which collectively form the optimal extremal control algorithm for the simplified extremal control process which is being considered.

3.1 Simplification of the functional recurrence equation

In chapter 2 the control of an extremal process has been set as an optimal control problem, and the solution formally derived as the solution of a functional recurrence equation. The first step in solving such an equation explicitly is evidently to attempt to simplify it as far as possible analytically. To see how this can be done it is convenient to set down the defining equations together. These are

$$F_N[u,w,q] \equiv \min_{u(i+1)} \int_{-\infty}^{\infty} \left\{ q \cdot \left[A(w^+)^2 + F_{N-1}[u(i+1), w^+, q^+] \right] + \overline{1-q} \cdot \left[A(w^-)^2 + F_{N-1}[u(i+1), w^-, q^-] \right] \right\} p[r]\, dr, \qquad 3.1.1$$

$$N = 2,3, \ldots$$

and

$$F_1[u,w,q] \equiv A\left\{4w^2\, q \,.\, \overline{1-q} + q^2\right\} \qquad 3.1.2$$

where
$$\begin{aligned} w^+ &= |w + u(i+1) - u + r| \\ w^- &= |-w + u(i+1) - u + r| \\ q^+ &= Q[w^+, w, q, u(i+1), u] \\ q^- &= Q[w^-, w, q, u(i+1), u] \end{aligned} \qquad 3.1.3$$

where $Q[W, w, q, u(i+1), u]$ is defined by

$$Q[W, w, q, u(i+1), u] \equiv \frac{q \,.\, P1 + \overline{1-q} \,.\, P2}{q.[P1 + P3] + \overline{1-q} \,.\, [P2 + P4]} \qquad 3.1.4$$

and P1, P2, P3 and P4 are defined by

$$\left.\begin{aligned} P1 &= \exp\left\{-[W - w - u(i+1) + u]^2/2q^2\right\} \\ P2 &= \exp\left\{-[W + w - u(i+1) + u]^2/2q^2\right\} \\ P3 &= \exp\left\{-[W + w + u(i+1) - u]^2/2q^2\right\} \\ P4 &= \exp\left\{-[W - w + u(i+1) - u]^2/2q^2\right\} \end{aligned}\right\} \qquad 3.1.5$$

Also $p[r]$ is defined by

$$p[r] \equiv \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-r^2/2\sigma^2\right\} \qquad 3.1.6$$

The function $u_1[u,w,q]$ is defined by

$$u_1[u,w,q] \equiv u + \overline{1-2q} \,.\, w \qquad 3.1.7$$

For N = 2,3, ... , the function $u_N[u,w,q]$ is defined as the minimising value of u(i+1) in equation 3.1.1 expressed as a function of u,w, and q.

These then are the equations defining the sequences $F_N[u,w,q]$, N = 1,2, ... , and $u_N[u,w,q]$, N = 1,2, ... , which must be determined.

The first simplification which can be made is to eliminate A and σ from the equations by normalising $F_N[u,w,q]$, N = 1,2, ... , with respect to $A\sigma^2$, and normalising u,w,u(i+1), w^+,w^-,r,W, and $u_N[u,w,q]$, N = 1,2, ... , with respect to σ. No new notation will be introduced for normalised variables.

Equation 3.1.1 now becomes

$$F_N[u,w,q] \equiv \min_{u(i+1)} \int_{-\infty}^{\infty} \left\{ q.\left[(w^+)^2 + F_{N-1}[u(i+1),\ w^+,\ q^+] \right] + 1-q.\left[(w^-)^2 + F_{N-1}[u(i+1),\ w^-,\ q^-] \right] \right\} p[r]\ dr \qquad 3.1.8$$

$$N = 2,3, \ldots$$

Equation 3.1.2 now becomes

$$F_1[u,w,q] \equiv \left\{ 4w^2\ q.\ \overline{1-q} + 1 \right\} \qquad 3.1.9$$

Equations 3.1.3 and 3.1.4 are not altered in form, but equations 3.1.5 now becomes

$$\left.\begin{aligned} P1 &= \exp\left\{ -[W - w - u(i+1) + u]^2/2 \right\} \\ P2 &= \exp\left\{ -[W + w - u(i+1) + u]^2/2 \right\} \\ P3 &= \exp\left\{ -[W + w + u(i+1) - u]^2/2 \right\} \\ P4 &= \exp\left\{ -[W - w + u(i+1) - u]^2/2 \right\} \end{aligned}\right\} \qquad 3.1.10$$

Equation 3.1.6 now becomes

$$p[r] \equiv \frac{1}{\sqrt{2\pi}} \exp\left\{ - r^2/2 \right\} \qquad 3.1.11$$

Equation 3.1.7 is not altered in form, but $u_N[u,w,q]$ is now defined as the minimising value of u(i+1) in equation 3.1.8 expressed as a function of u,w, and q.

Inspection of equation 3.1.8 shows that it can be simplified in form by the substitution

$$F_N[u,w,q] = \bar{F}_N[u,w,q] - w^2 \qquad 3.1.12$$

This substitution changes equation 3.1.8 to

$$\overline{F}_N\left[u,w,q\right] = w^2 = \underset{u(i+1)}{\text{Min}} \int_{-\infty}^{\infty} \left\{ q\overline{F}_{N-1}\left[u(i+1),\ w^+,\ q^+\right] + \overline{1-q}\ .\ \overline{F}_{N-1}\left[u(i+1),\ w^-,\ q^-\right] \right\} p[r]\ dr \qquad 3.1.13$$

and equation 3.1.9 to

$$\overline{F}_1\left[u,w,q\right] = \left\{ w^2 + 4w^2\ q\ .\ \overline{1-q} + 1 \right\} \qquad 3.1.14$$

No other defining equation is affected.

Examination of the expression to be minimised in equation 3.1.13 shows that for fixed $\overline{N-1}$, it is ultimately a function of the four variables u(i+1), u, w and q. However if $\overline{F}_{N-1}\left[u,w,q\right]$ is independent of its first argument, then the expression can be considered to be a function of the three variables $\overline{u(i+1) - u}$, w and q. Minimisation with respect to u(i+1) would therefore become minimisation with respect to $\overline{u(i+1) - u}$, and hence from equation 3.1.13 $\overline{F}_N\left[u,w,q\right]$ would also be independent of its first argument. Equation 3.1.14 shows that $\overline{F}_1\left[u,w,q\right]$ is independent of its first argument, and hence by induction $\overline{F}_N\left[u,w,q\right]$ is independent of its first argument for N = 1,2, ... , and can be written as $\overline{F}_N\left[w,q\right]$. Writing the dummy variable $\overline{u(i+1) - u}$ as v equation 3.1.13 now becomes

$$\overline{F}_N\left[w,q\right] \equiv w^2 + \underset{v}{\text{Min}} \int_{-\infty}^{\infty} \left\{ q\ .\ \overline{F}_{N-1}\left[w^+,\ q^+\right] + \overline{1-q}\ .\ \overline{F}_{N-1}\left[w^-,\ q^-\right] \right\} p[r]\ dr \qquad 3.1.15$$

$$N = 2,3, \ \ldots$$

Equation 3.1.14 becomes

$$\overline{F}_1\left[w,q\right] \equiv \left\{ w^2 + 4w^2\ q\ .\ \overline{1-q} + 1 \right\} \qquad 3.1.16$$

Equation 3.1.3 becomes

$$\begin{aligned} w^+ &= |w + v + r| \\ w^- &= |-w + v + r| \\ q^+ &= Q[w^+, w, q, v] \\ q^- &= Q[w^-, w, q, v] \end{aligned} \qquad 3.1.17$$

Equation 3.1.4 becomes

$$Q[W, w, q, v] \quad \frac{q \cdot P1 + \overline{1-q} \cdot P2}{q \cdot (P1 + P3) + \overline{1-q} \cdot (P2 + P4)} \qquad 3.1.18$$

Equation 3.1.10 becomes

$$\left.\begin{aligned} P1 &= \exp\{-[W - w - v]^2/2\} \\ P2 &= \exp\{-[W + w - v]^2/2\} \\ P3 &= \exp\{-[W + w + v]^2/2\} \\ P4 &= \exp\{-[W - w + v]^2/2\} \end{aligned}\right\} \qquad 3.1.19$$

Equation 3.1.11 remains unaltered.

The minimising value of v in equation 3.1.15 is a function of w and q and can be written as $v_N[w,q]$, N = 2,3, ... If $v_1[w,q]$ is defined as

$$v_1[w,q] \equiv \overline{1-2q} \cdot w \qquad 3.1.20$$

then $u_N[u,w,q]$ is now defined by the equation

$$u_N[u,w,q] \equiv u + v_N[w,q] \qquad 3.1.21$$

$$N = 1,2, \ldots$$

The significance of this result is that the optimal extremal controller includes an integrator in this case.

In controlling an extremal process the case of interest is not generally when N is small, but when there is an indefinite large number of stages to go. In this case it is

necessary to determine whether $u_N[u,w,q]$ converges for large N. This is equivalent to determining whether $v_N[w,q]$ converges for large N. To determine this it is necessary to iterate equation 3.1.15 indefinitely. Since $F_N[u,w,q]$ is the expected total remaining cost it will be expected to grow indefinitely with N, and hence so also will $\overline{F}_N[w,q]$. These increasingly large numbers would make successive iterations of equation 3.1.15 extremely difficult, particularly if done numerically. However it seems reasonable to expect that for large N

$$\overline{F}_N[w,q] \approx f[w,q] + Nc^* \qquad 3.1.22$$

where $f[w,q]$ is the expected transient contribution to the total cost, dependent on initial conditions, and c^* is the expected average cost per stage remaining.

The substitution

$$\left.\begin{aligned} \overline{F}_N[w,q] &\equiv f_N[w,q] + Nc_N^* \\ f_N[0,\ 0] &\equiv 0 \end{aligned}\right\} \qquad 3.1.23$$

$$N = 1,2, \ldots$$

can therefore be expected to lead to a functional recurrence equation for $f_N[w,q]$, $N = 1,2, \ldots$, c_N^* , $N = 1,2, \ldots$, and $v_N[w,q]$, $N = 1,2, \ldots$ which might converge.

Making this substitution it is now possible to gather together all the equations defining the normalised sequences $F_N[u,w,q]$ and $u_N[u,w,q]$, $N = 1,2, \ldots$

$$F_N[u,w,q] \equiv f_N[w,q] - w^2 + Nc_N^* \ , \ N = 1,2, \ldots \qquad 3.1.24$$

$$f_N[w,q] + Nc_N^* \equiv w^2 + \overline{N-1}\, c_{N-1}^* +$$

$$\underset{v}{\text{Min}} \int_{-\infty}^{\infty} \left\{ q f_{N-1}[w^+, q^+] + \overline{1-q} \cdot f_{N-1}[w^-, q^-] \right\} p[r]\, dr \qquad 3.1.25$$

$$N = 2,3, \ldots$$

$$f_N[0, 0] \equiv 0 \qquad N = 1,2, \ldots \qquad 3.1.26$$

$$f_1[w,q] = \left\{ w^2 + 4w^2\, q \cdot \overline{1-q} \right\} \qquad 3.1.27$$

$$c_1^* \equiv 1 \qquad 3.1.28$$

$$\begin{aligned} w^+ &= |w + v + r| \\ w^- &= |-w + v + r| \\ q^+ &= Q[w^+, w, q, v] \\ q^- &= Q[w^-, w, q, v] \end{aligned} \qquad 3.1.29$$

$$Q[W, w, q, v] \equiv \frac{q.P1 + \overline{1-q}\, P2}{q(P1 + P3) + \overline{1-q}\,(P2 + P4)} \qquad 3.1.30$$

$$\left.\begin{aligned} P1 &= \exp\left\{-[W - w - v]^2/2\right\} \\ P2 &= \exp\left\{-[W + w - v]^2/2\right\} \\ P3 &= \exp\left\{-[W + w + v]^2/2\right\} \\ P4 &= \exp\left\{-[W - w + v]^2/2\right\} \end{aligned}\right\} \qquad 3.1.31$$

$$p[r] \equiv \frac{1}{\sqrt{2\pi}} \exp\left\{-r^2/2\right\} \qquad 3.1.32$$

$$v_1[w,q] \equiv \overline{1-2q} \cdot w \qquad 3.1.33$$

$v_N[w,q]$, $N = 2,3, \ldots$ is defined as the minimising value of v in equation 3.1.25, expressed as a function of w and q.

$$u_N[u,w,q] \equiv u + v_N[w,q] \qquad 3.1.34$$

3.2 Properties of the iterates of the functional recurrence equation

Having reduced the functional recurrence equation to as simple a form as has been found possible analytically, the next step in finding a solution is to examine it to find properties of the iterates. Properties which are of particular interest are those which facilitate the numerical solution of the functional recurrence equation presented in the next section.

3.2.1 Domain of definition The definitions of w, w^+, w^-, q, q^+, and q^- show that $f_N[w,q]$ and $v_N[w,q]$, $N = 1,2, \ldots$, are defined only on the semi-infinite strip of the two-dimensional space of the sufficient statistics w and q, defined by

$$\left.\begin{aligned} w &\geqslant 0 \\ 0 \leqslant q &\leqslant 1 \end{aligned}\right\} \qquad 3.2.1.1$$

This domain of definition is shown in figure 3.2.1.1.

3.2.2 Symmetry about $q = 0.5$ Referring to figure 2.1.1, the extremal control problem as set up so far is seen to be symmetrical about the ordinate of the non-linearity. Since q and $\overline{1-q}$ are equivalent probabilities referred to the positive and negative sides of this ordinate respectively, this suggests that both $f_N[w,q]$ and $v_N[w,q]$, $N = 1,2, \ldots$, will have symmetry properties about the line $q = 0.5$, defined by

$$f_N[w,q] \equiv f_N[w, \overline{1-q}], \quad N = 1,2, \ldots \qquad 3.2.2.1$$

$$\text{and} \quad v_N[w,q] \equiv -v_N[w, \overline{1-q}], \quad N = 1,2, \ldots \qquad 3.2.2.2$$

These properties are shown to be true analytically in appendix A1.1.

3.2.3 <u>The special case w = 0</u> When w = 0, it follows that X1 = 0, and hence the definition q as the probability that X1 is positive becomes meaningless. It is therefore to be expected that

$$f_N[0, q] \equiv \text{constant} \equiv 0 \ , \ N = 1,2, \ldots \qquad 3.2.3.1$$

$$\text{and} \quad v_N[0, q] \equiv \text{constant} \equiv v_N, \ N = 1,2, \ldots \qquad 3.2.3.2$$

It should be noted that $v_N[w,q]$, N = 1,2, ... may not be unique, and in particular the symmetry condition of section 3.2.2 shows that $v_N[0,q]$ has at least two alternative values

$$v_N[0,q] \equiv \pm v_N \qquad 3.2.3.3$$

An analytic proof of these properties is given in appendix A1.2.

3.2.4 <u>The special cases q = 0, q = 1, q = 0.5</u> It is shown in appendix A1.3 that in the cases q = 0 and q = 1 the following relationships hold:-

$$f_N[w,0] \equiv f_N[w, 1] \equiv w^2 \ , \ N = 1,2, \ldots \qquad 3.2.4.1$$

$$v_N[w,0] \equiv w \pm v_N \ , \ N = 1,2, \ldots \qquad 3.2.4.2$$

$$v_N[w,1] \equiv - w \mp v_N \ , \ N = 1,2, \ldots \qquad 3.2.4.3$$

where v_N is defined in section 3.2.3. The physical significance of this result is that the transient cost when the initial state is known (q = 0 or q = 1) consists only of the initial cost w^2.

No analytic representation of $f_N[w, 0.5]$ or $v_N[w, 0.5]$, N = 1,2, ... , has been found.

3.2.5 The special case of large w It is shown in appendix A1.4 that for large values of w and $q \neq 0$, 0.5 and 1, the functions $f_N[w,q]$ and $v_N[w,q]$, N = 1,2, ..., tend towards asymptotic values

$$f_N[w,q] \longrightarrow w^2 (1 + 4q \cdot \overline{1-q}), \quad N = 1,2, \ldots \qquad 3.2.5.1$$

$$\text{and} \quad v_N[w,q] \longrightarrow (1 - 2q)w, \quad N = 1,2, \ldots \qquad 3.2.5.2$$

The physical significance of this result is that when the extremal process is far from optimum ($w \gg 1$) the controller should try to bring it back to optimum immediately ($v_N[w,q] \longrightarrow v_1[w,q]$, N = 1,2, ...), and the transient cost is contained in the first stage ($f_N[w,q] \longrightarrow f_1[w,q]$, N = 1,2, ...)

3.3. Numerical solution of the functional recurrence equation

The numerical solution is presented in two sections. In the first the various stages of the solution are defined and the principles behind the solution of the problems arising at each stage are discussed. The second section presents and discusses the detailed results as the equation is solved.

3.3.1 Principles behind the solution The iterative procedure for deriving the sequences $f_N[w,q]$, $v_N[w,q]$ and c_N^*, N = 1,2, ... , is essentially defined by equations 3.1.25 and 3.1.26, with equations 3.1.27 and 3.1.28 defining the starting point. Each iteration consists of the evaluation of $f_N[w,q]$, $v_N[w,q]$ and c_N^* from $f_{N-1}[w,q]$ and c_{N-1}^* for a fixed value of N, and iterations are carried out in turn for N = 2,3, ... To see exactly what calculations are necessary for each iteration, it is convenient to define $I_N[w,q]$, N = 2,3, ... , by

$$I_N[w,q] \equiv \min_v \int_{-\infty}^{\infty} \left\{ q \cdot f_{N-1}[w^+, q^+] + \overline{1-q} \cdot f_{N-1}[w^-, q^-] \right\} p[r] \, dr \qquad 3.3.1.1$$

$$N = 2,3, \ldots$$

Substitution of equation 3.1.26 in equation 3.1.25 then shows that

$$Nc_N^* = \overline{N-1}\, c_{N-1}^* + I_N[0,0] \quad , N = 2,3, \ldots \qquad 3.3.1.2$$

and $$c_N^* = \frac{1}{N}\left\{I_N[0,0] + \overline{N-1}\, c_{N-1}^*\right\} , N = 2,3, \ldots \qquad 3.3.1.3$$

Combining equations 3.1.26, 3.3.1.1 and 3.3.1.2 then gives

$$f_N[w,q] = w^2 + I_N[w,q] - I_N[0,0] \quad , N = 2,3, \ldots \qquad 3.3.1.4$$

and $v_N[w,q]$ is the minimising value of v in equation 3.3.1.1 expressed as a function of w and q.

Equations 3.3.1.1 and 3.3.1.4 now show that the sequences $f_N[w,q]$ and $v_N[w,q]$, N = 1,2, ... , can be derived independently of the sequence c_N^*, N = 1,2, ... Figure 3.3.1.1 is a flow diagram of the procedure for doing this. It shows iterations continuing until $f_N[w,q]$ and $v_N[w,q]$, N = 1,2, ... , have converged. If iterations are continued until $I_N[w,q]$, N = 2,3, ... has also converged, then equation 3.3.1.3 shows that the sequence c_N^*, N = 1,2, ... will also converge, to the limiting value of the sequence $I_N[0,0]$, N = 2,3, ... Figure 3.3.1.2 therefore shows the iteration procedure which is to be followed since only the limiting value of the sequence c_N^*, N = 1,2, ... , is really required.

The only stage in the flow diagram of figure 3.3.1.2 which presents difficulties is that labelled, "Calculate $I_N[w,q]$ from equation 3.3.1.1, generating $v_N[w,q]$." This procedure may be resolved into three parts represented by the equations

$$\delta_N[w,q,v,r] \equiv \left\{q \cdot f_{N-1}[w^+, q^+] + \overline{1-q} \cdot f_{N-1}[w^-, q^-]\right\} p[r] \quad N = 2,3, \ldots \qquad 3.3.1.5$$

$$\bar{I}_N[w,q,v] \equiv \int_{-\infty}^{\infty} \delta_N[w,q,v,r]\, dr \quad , N = 2,3, \ldots \qquad 3.3.1.6$$

$$\text{and } I_N[w,q] = \underset{v}{\text{Min}}\ \bar{I}_N[w,q,v] \quad , N = 2,3, \ldots \qquad 3.3.1.7$$

with equation 3.3.1.7 generating $v_N[w,q]$.

Since $\delta_N[w,q,v,r]$ is a complicated function of r for $N \geqslant 2$, it seems unlikely that the integration in equation 3.3.1.6 can be carried out analytically for general w,q,v. The alternative is to carry out a numerical integration which is possible only at fixed points in (w,q,v) space. Since only a finite number of such integrations can be carried out, it is necessary to decide at which finite set of points (w,q,v) integrations should be carried out at each iteration.

Choice of the (w,q,v) points can be resolved into

(1) a choice of (w,q) points at which $f_N[w,q]$ and $v_N[w,q]$, N = 2,3, ... must be determined, followed by

(2) a choice of values of v for each point (w,q) for which integrations must be carried out.

The points (w,q) must be chosen with the following considerations in mind:-

(1) It must be possible to approximate the functions $f_N[w,q]$, N = 2,3, ... , within the domain of definition from values determined at the points (w,q) together with the known properties, so that successive iterations can be carried out sufficiently accurately.

(2) It must be possible to approximate the functions $v_N[w,q]$, N = 2,3, ... , within the domain of definition from values determined at the points (w,q) together with the known properties so that no significant departure

from the optimal extremal control algorithm is made.

A discussion of how sufficient accuracy is determined is reserved until the end of this section.

Since $f_N[w,q]$ and $v_N[w,q]$, N = 2,3, ... , have known asymptotes for large w, there may be some W beyond which it is unnecessary to distinguish $f_N[w,q]$ from its asymptote, and so computations become unnecessary for $w > W$. For $w < W$ the known symmetry properties mean that computations need only be carried out in the region

$$0 \leqslant w$$
$$0 \leqslant q \leqslant 0.5 \qquad 3.3.1.8$$

and the known solutions make computations unnecessary along the line q = 0.

A rectangular grid of points can now be set in the region

$$0 \leqslant w \leqslant W$$
$$0 < q \leqslant 0.5 \qquad 3.3.1.9$$

having spacing dw in the w direction and d_q in the q direction. The intersection points of the grid indicated in figure 3.3..1.3 are the required points (w,q) at which computations must be made. Between grid points $f_N[w,q]$ and $v_N[w,q]$, N = 2,3, ... can be determined by linear interpolation. Discussion of the choice of values for the approximation parameters W, dw and d_q is left until the end of this section.

Having decided in principle at which (w,q) points integrations should be carried out at each iteration, the next step is to decide for which values of v integrations should be carried out at each (w,q) point. For each point (w,q)

at each iteration, integrations must be carried out for sufficient values of v to give sufficient information about $\bar{I}_N[w,q,v]$. N = 2,3, ... , to find an estimate of the minimum value i.e. $I_N[w,q]$, N = 2,3, ... , together with the minimising value of v, i.e. $v_N[w,q]$, N = 2,3, ...

The method to be used is to construct graphs of $\bar{I}_2[w,q,v]$ against v at a few representative (w,q) points in order to determine the general characteristics. From these suitable starting values of v from which to search for the minimum at all the (w,q) points chosen can be determined. Starting from these values a stepping procedure with fixed step size dv is used to locate the minimum. When three values of v surrounding the minimum have been determined, estimates of the minimum value of $\bar{I}_2[w,q,v]$, and the minimising value of v are made by fitting a parabola through the points. For the remaining iterations, the general characteristics of the functions $\bar{I}_N[w,q,v]$, N = 3,4, ... , are again confirmed by constructing graphs, but starting values of v from which to search for the minima are generally taken as the minimising value of v from the previous iteration, with some exceptions discussed later due to the occurence of multiple minima. Discussion of the choice of dv is also left until later.

Some insight into the expected shape of the graphs of $\bar{I}_N[w,q,v]$ against v , N = 2,3, ... , can be gained by considering in a more heuristic manner what the controller is trying to do when it is choosing v(i). Basically it is trying to carry out the two functions of dual control[15] which often lead to conflicting requirements. These are:-

(1) to ensure that the cost c(i+1) incurred at the next stage is as small as possible, and

(2) to ensure that the state vector $\underline{X}(i+1)$ will be known as far as possible at the end of stage (i+1).

If only the first of these requirements is considered then the choice of v would be

$$v = v_1[w,q] = \overline{1-2q} \cdot w \qquad 3.3.1.10$$

since choice of v(i) when there is only one stage to go is entirely concerned with minimising the cost of the final stage.

In this particular case if only the second requirement were considered, then v(i) would be chosen to make the expected value of q(i+1) close to 0 or 1. Equation 2.2.1.8 shows that q(i+1) can be expected to be close to 0 or 1 if the ratio $p_0[+ w(i+1)] \;/\; p_0[- w(i+1)]$ is either close to 0, or is very large i.e. if the function $p_0[.]$ is very asymmetric for the expected values of w(i+1).

Appropriate substitution in equation 2.2.1.7 shows that $p_0[x(i+1)]$ can be written as

$$p_0[x(i+1)] = q \cdot p[x(i+1) - w - v] + \overline{1-q} \cdot p[x(i+1) + w - v] \qquad 3.3.1.11$$

i.e. two superposed Gaussian probability distributions, spaced 2w apart, and with relative heights q and $\overline{1-q}$ respectively. A typical case is shown in figure 3.3.1.4. The shape of the distribution is fixed for fixed w and q, but the position of the ordinate varies directly with v. A positive value of v displaces the ordinate a distance v in the negative direction

from the mid-position shown and vice-versa. Thus if the only requirement in choosing v were that $p_0\left[x(i+1)\right]$ be asymmetric, it would be expected that v would be chosen to place the ordinate in one of the four general regions labelled (1) to (4) in figure 3.3.1.4.

Combining these considerations, the following predictions can be made about the shape of the $\overline{I}_N[w,q,v]$ aganst v graphs, N = 2,3, ...

(1) $\overline{I}_N[w,q,v]$, N = 2,3, ... , may have minima for values of v corresponding to the four regions labelled (1) to (4) in figure 3.3.1.4.

(2) For $0 \leq q < 0.5$, minima (1) and (2) will dominate (3) and (4) because the absolute minimum should not be unnecessarily far from $v = \overline{1-2q}\ w$. For q = 0.5 there are two absolute minima symmetric about the v = 0 axis.

(3) For smaller values of w, minimum (1) will dominate minimum (2) as the interference of the two superposed Gaussian probability distributions will make the overall distribution rather symmetric in region (2). For larger values of w, minimum (2) will dominate minimum (1) as it is closer to the value $v = \overline{1-2q}\ .\ w$.

These predictions show that it is particularly desirable to examine the $\overline{I}_N[w,q,v]$ against v graphs, N = 2,3, ... ,

(1) In general to confirm the expected shape, and

(2) In particular to see if the absolute minimum jumps sharply from position (1) to position (2) for increasing w and fixed q.

Having decided in principle at what points (w,q,v) the integrations to determine $\overline{I}_N[w,q,v]$ should be carried out, it is now necessary to discuss the numerical method by which the

integration is to be approximated. The method used is the trapezoidal rule because it is the simplest. To use this method, r is quantised in units of dr, and the infinite integration is truncated at $\pm$ R.

Putting

$$R = J \,.\, dr, \; J \text{ an integer} \qquad 3.3.1.12$$

equation 3.3.1.6 is now approximated by

$$\bar{I}_N[w,q,v] = \left\{ \delta_N[w,q,v,-J.dr] + \delta_N[w,q,v,J.dr] + 2 \cdot \sum_{j=-J+1}^{J-1} \delta_N[w,q,v,j.dr] \right\} .dr/2 \qquad N = 2,3, \ldots \qquad 3.3.1.13$$

In all approximation parameters W, dw, dq, dv, R and dr have been introduced. A discussion of the initial choice of these parameters is given in the next section. The last five are checked by improving each by a factor of 2 in turn and comparing results of the first iteration. Results given in the next section show that the functions $f_N[w,q]$, N = 1,2, ... , are of similar shape, and so this check is taken to be valid for all iterations. The parameter W is checked by carrying out the whole iteration procedure twice with different values of W, and showing that the more accurate value gives no significant improvement in results.

3.3.2 The numerical solution All the essential information necessary for the numerical solution of the functional recurrence equation has now been presented. The logical flow of the solution is shown in the flow diagram in figure 3.3.2.1. This section gives the results of following through the various stages of this flow diagram. In order to correlate between the flow diagram and the comments in this section, each block in the diagram

has been numbered, and all comments are headed by the appropriate block number.

The solution is largely dependent on the use of digital computer programs. The results given here were obtained using FORTRAN[38] on an IBM 1130 digital computer and ATLAS AUTOCODE[39] on an English Electric KDF9 digital computer. The method of description of programs is given in Appendix 2. The descriptions of programs used in this section are given in Appendix 3, and are arranged to be independent of the programming language used.

BLOCK 1 Before any calculations can be made, an initial choice of values for the approximation parameters W, dw, dq, dr and R (but not dv) must be made. It should be emphasised that although this initial choice is made on a reasoned basis, it is the check which is later made in blocks 5, 6 and 7 which really establishes the choice of values as satisfactory.

(1) Choice of W: A suboptimal extremal control algorithm for the process being considered is known[29] which maintains a normalised mean cost per stage of 2.2. Thus the values of w which can be expected to arise when the process is under optimal control can be expected to average about 1.5 ($\approx \sqrt{2.2}$) or less. It is therefore reasonable to expect that asymptotic behaviour may occur for $w > 3$ and a suitable initial choice of the approximation parameter W is

$$W = 3.0 \qquad 3.3.2.1$$

(2) Choice of dw and dq: Each iteration after the first involves the linear interpolation of the functions $f_N[w,q]$, $N = 2,3, \ldots$, for $w \leq W$. The accuracy of this interpolation is determined by the values of dw and dq. The optimal extremal control

algorithm must also be approximated by linear interpolation for $w \leq W$, and again the accuracy is determined by the values of dw and dq. The initial choice of values for dw and dq is based on the effect of interpolation on the known function $f_1[w,q]$. The values chosen

$$dw = 0.1$$
$$dq = 0.1 \qquad 3.3.2.2$$

give an error $< 1\%$ when the interpolated value of $f_1[1.55, 0.25]$ is compared with the known value.

This initial choice of W, dw and dq means that the set of points at which $f_N[w,q]$ and $v_N[w,q]$, N = 2,3, ... , must be evaluated is w = 0.00(0.1)3.0, q = 0.1(0.1)0.5, making a total of 155 points per iteration.

(3) Choice of dr: Equation 3.3.1.13 shows that the numerical integration involves the evaluation of $\delta_N[w,q,v,r]$, N = 2,3, ... , for values of r spaced dr apart. Equations 3.3.1.5 and 3.1.29 show that this in turn involves the evaluation of $f_{N-1}[w,q]$, N = 2,3, ... , for values of w spaced dr apart. Since $f_{N-1}[w,q]$ is generally found by interpolation between values of w spaced dw apart, a suitable initial choice of dr is to make it rather more precise than dw. This allows for the fact that interpolation of $f_{N-1}[w,q]$ occurs also in the q direction. The initial value chosen for dr is thus

$$dr = 0.05 \qquad 3.3.2.3$$

(4) Choice of R: The only basis on which R can be chosen is to make $\exp(-R^2/2)$ small compared to unity so that the extreme elements of the numerical integration will be insignificant.

The value chosen

$$R = 5.0 \qquad 3.3.2.4$$

gives $\exp(-R^2/2)$ the value of 3.7×10^{-6} which appears to be adequate.

As explained in Appendix A3.1 the digital computer programs do not work directly from these approximation parameters, but from related parameters, some of which can now be evaluated:-

$$\begin{aligned} IDRDW &\equiv dw/dr &&= 2 \\ IDWWL &\equiv 1 + W/dw &&= 31 \\ IDQQM &\equiv 0.5/dq &&= 5 \\ IDRRL &\equiv 1 + 2R/dr &&= 201 \end{aligned} \qquad 3.3.2.5$$

BLOCK 2 The digital computer program ITER 1 described in Appendix A3.8 can now be used to evaluate $\bar{I}_2[w,q,v]$ at any number of (w,q,v) triplets. (The data IDRDV used by ITER 1 has no effect on the results and can be set to any integer). Figures 3.3.2.2 - 3.3.2.4 show curves of $\bar{I}_2 \sim$ v drawn from values of $\bar{I}_2$ determined by ITER 1 for a representative set of (w,q) points, in the region $0 < q \leq 0.5$, $0 \leq w \leq W$.

BLOCK 3 The curves drawn in figures 3.3.2.2 - 3.3.2.4 confirm the features predicted in section 3.3.1 that

(1) the absolute minimum of the $\bar{I}_2 \sim$ v curves occurs for positive v when $0 \leq q < 0.5$

(2) there are two absolute minima of the $\bar{I}_2 \sim$ v curves symmetric about the v = 0 axis when q = 0.5.

(3) for small values of w the minimising value of v is greater than w, but for larger values of w, the minimising value of v is less than w.

However it is not possible to determine from these curves whether the absolute minimum of the $\bar{I}_2 \sim v$ curves jumps between two relative minima occuring for values of v respectively greater than and less than w, as w increases at constant q.

BLOCK 4 Figures 3.3.2.5 and 3.3.2.6 show further curves of $\bar{I}_2 \sim v$ drawn from values of $\bar{I}_2$ again evaluated by ITER 1. These curves are particularly chosen to show cases where the minimising value of v is approximately equal to w.

BLOCK 3 The additional $\bar{I}_2 \sim v$ curves show that there is no detectable jump in the minimising value of v as w increases at constant q.

An initial estimate vmino$[w,q]$ of the minimising value of v at each grid point can now be constructed as follows:-

(1) From figures 3.3.2.2 - 3.3.2.4

$$\begin{aligned} &\text{vmino}[0,q] = 0.48 && \text{vmino}[1.2,\ 0.1] = 1.46 \\ &\text{vmino}[1.2,\ 0.3] = 1.12 && \text{vmino}[1.2,\ 0.5] = 0.81 \\ &\text{vmino}[3.0,\ 0.1] = 2.30 && \text{vmino}[3.0,\ 0.3] = 1.92 \\ &\text{vmino}[3.0,\ 0.5] = 1.56 \end{aligned} \qquad 3.3.2.6$$

(2) By interpolation

$$\begin{aligned} &\text{vmino}\ [1.2,\ 0.2] = 1.29 && \text{vmino}\ [1.2,\ 0.4] = 0.96 \\ &\text{vmino}\ [3.0,\ 0.2] = 2.11 && \text{vmino}\ [3.0,\ 0.4] = 1.74 \end{aligned} \qquad 3.3.2.7$$

(3) By interpolation $0 \leq w \leq 1.2$

$$\text{vmino}[w,q] = \{(1.2 - w)/1.2\} \cdot \text{vmino}[0,q] + \{w/1.2\} \cdot \text{vmino}\ [1.2,q] \qquad 3.3.2.8$$

(4) By interpolation $1.2 < w \leq 3.0$

$$\text{vmino}[w,q] = \{(3.0 - w)/1.8\} \cdot \text{vmino}\,[1.2,q] + \{(w-1.2)/1.8\} \cdot \text{vmino}[3.0, q] \qquad 3.3.2.9$$

The function vmino $[w,q]$ is shown in figure 3.3.2.7.

The curves in figures 3.3.2.2 - 3.3.2.6 also show that the minima appear to be quite smooth and suited to approximation by parabolae. Also a suitable initial choice of the approximation parameter dv,

$$dv = 0.1 \qquad 3.3.2.10$$

is suggested from inspection of the curves. The choice of dv gives the data IDRDV from which the programs operate the value

$$IDRDV \equiv dv/dr = 2 \qquad 3.3.2.11$$

BLOCK 5 The initial values given to the approximation parameters dr and R must now be checked. This is done by evaluating $\bar{I}_2[w,q,v]$ at representative (w,q,v) triplets and comparing values of $\bar{I}_2$ obtained when dr and R are varied. The program ITER 1 is again used for this purpose. Table 3.3.2.1 is a comparison of values of $\bar{I}_2$ obtained for dr = 0.05, R = 5.0 (the initial values), with values of $\bar{I}_2$ obtained for dr = 0.02, R = 5.0 (dr improved by a factor of 2.5) and dr = 0.05, R = 10 (R improved by a factor of 2). This comparison shows that the change in the values of $\bar{I}_2$ obtained is less than .025%.

BLOCK 6 Equation 3.3.1.4 shows that the important values to be determined at each iteration are not the values of $I_N[w,q]$, N = 2,3, ... , but values of the difference $I_N[w,q] - I_N[0,0]$, N = 2,3, ... , together with values of $v_N[w,q]$, N = 2,3, ...

Tables 3.3.2.2 and 3.3.2.3 contain comparisons of values of $I_2[w,q] - I_2[0,0]$ and $v_2[w,q]$ respectively evaluated for $dq = 0.1$, $dv = 0.1$ (the initial values) with values obtained for $dq = 0.05$, $dv = 0.1$ (dq improved by a factor of 2) and $dq = 0.1$, $dv = 0.05$ (dv improved by a factor of 2). The values were obtained using the program ITER 2. Variation of the values of the difference $I_2[w,q] - I_2[0,0]$ is generally less than 3% and variation of the values of $v_2[w,q]$ is generally less than 1%.

BLOCK 7 Since $f_1[w,q]$ is not interpolated in the w direction in the first iteration, choice of dw does not normally have any effect on the results of the first iteration. However the program ITER 5 can be used to evaluate $I_2[w,q]$ and $v_2[w,q]$ at various (w,q) points, interpolating $f_1[w,q]$ instead of using precise values. Tables 3.3.2.2 and 3.3.2.3 contain comparisons of the values of the difference $I_2[w,q] - I_2[0,0]$ and $v_2[w,q]$ respectively evaluated in this way with values obtained from the program ITER 2. The value of dw used is the initial value $dw = 0.1$. Variation of the results obtained is negligible compared to the variation obtained when dq and dv are varied.

BLOCK 8 The results presented in tables 3.3.2.1 - 3.3.2.3 show that the initial values given to the approximation parameters R, dr, dq, dv and dw can reasonably be expected to lead to satisfactory results being obtained. It is therefore unnecessary to go to block 9 when any changes to the approximation parameters could have been made.

BLOCK 10 The first iteration can now be carried out using the program ITER 3. The data required are the approximation

parameters (or related parameters) which have already been determined and the array VMINO (IDWWL, IDQQM) which is set from the function vmino $[w,q]$ shown in figure 3.3.2.7. Tables 3.3.2.4 and 3.3.2.5 show respectively the values of $I_2[w,q]$ and $v_2[w,q]$ obtained in this way at all the required grid points. The time taken to run the program ITER 3 (and later ITER 6) on the IBM 1130 digital computer was approximately three hours.

BLOCK 11 N is set to 3 to indicate that the next iteration must evaluate $I_3[w,q]$ and $v_3[w,q]$.

BLOCK 12 Table 3.3.2.6 is a comparison of the values of $f_1[w,q]$ and $f_2[w,q]$ at representative (w,q) points. This shows that the values of $f_1[w,q]$ and $f_2[w,q]$ are of similar magnitude, and the functions $f_1[w,q]$ and $f_2[w,q]$ are of similar shape. This implies that it is unnecessary to change the values of any of the approximation parameters dw, d_q, dr and R for the next iteration.

BLOCK 13 The program ITER 4 can now be used to evaluate $\bar{I}_3[w,q,v]$ at (w,q,v) triplets of interest. Figures 3.3.2.8 - 3.3.2.10 show curves of $\bar{I}_3 \sim v$ drawn from values of $\bar{I}_3$ found in this way. These curves show that for values of q less than 0.3, there are two relative minima occuring for values of v respectively greater than and less than w. They also show that the absolute minimum transfers from the former to the latter as w increases at constant q, as predicted in section 3.3.1. The choice of the approximation parameter dv = 0.1 is also seen to be still valid.

BLOCK 14 It is now possible to proceed to use the program ITER 6 to evaluate $I_3[w,q]$ and $v_3[w,q]$. The only data which

presents difficulties is the array VMINO (IDWWL, IDQQM). Most elements of the array are taken equal to the corresponding value of $v_2[w,q]$. However in the region when the absolute minimum jumps between the two relative minima, it is necessary to set the relevant elements of VMINO from inspection of figures 3.3.2.8 - 3.3.2.10. Tables 3.3.2.4 and 3.3.2.5 show resp ectively the values of $I_3[w,q]$ and $v_3[w,q]$ thus obtained by use of the program ITER 6.

BLOCK 15 No question of convergence can be considered yet as insufficient iterations have been completed.

BLOCK 16 N is set to 4 as $I_4[w,q]$ and $v_4[w,q]$ are to be evaluated at the next iteration.

BLOCK 12 Comparison of values of $I_2[w,q]$ and $I_3[w,q]$ in table 3.3.2.4 shows that they are of very similar magnitude and hence $f_2[w,q]$ and $f_3[w,q]$ are of similar magnitude. Thus no change in the approximation parameters dw, d_q, dr and R is necessary for the next iteration.

BLOCK 13 Figures 3.3.2.11 - 3.3.2.13 show curves of $\bar{I}_4[w,q,v]$ drawn from values of $\bar{I}_4$ evaluated by the program ITER 4. These curves display similar features to the curves of $\bar{I}_3[w,q,v]$ previously discussed. The value of the approximation parameter dv = 0.1 is again shown to remain valid.

BLOCK 14 Tables 3.3.2.4 and 3.3.2.5 show respectively the values of $I_4[w,q]$ and $v_4[w,q]$ found using the program ITER 6. The elements of the data array VMINO (IDWWL, IDQQM) are generally set to the corresponding value of $v_3[w,q]$, but amended as necessary in the region when the absolute minimum jumps between relative minima, as sh wn in figures 3.3.2.11 - 3.3.2.13.

BLOCK 15 Sufficient iterations have still not been completed to

consider the question of convergence.

BLOCK 16 N is set to 5 as $I_5[w,q]$ and $v_5[w,q]$ must be evaluated at the next iteration.

BLOCK 12 Comparison of values of $I_4[w,q]$ and $I_3[w,q]$ in table 3.3.2.4 shows that they are of similar magnitude and hence as before there is no need to change the values of the approximation parameters dw, dq, dr and R.

BLOCK 13 Figures 3.3.2.14 - 3.3.2.16 show curves of $\bar{I}_5[w,q,v]$ drawn from values of $\bar{I}_5$ evaluated by the program ITER 4. These curves are almost identical to the curves of $\bar{I}_4[w,q,v]$ previously discussed, and in particular confirm that dv = 0.1 is still a suitable choice for the approximation parameter.

BLOCK 14 Tables 3.3.2.4 and 3.3.2.5 show respectively the values of $I_5[w,q]$ and $v_5[w,q]$ obtained using the program ITER 6. No amendment to the data array VMINO (IDWWL, IDQQM) is necessary in this case, all elements being set to the corresponding value of $v_4[w,q]$:

BLOCK 15 Comparison of values of $I_N[w,q]$ and $v_N[w,q]$, N = 2,3,4,5 in tables 3.3.2.4 and 3.3.2.5 respectively shows that convergence is almost complete in many cases, but further iteration is still necessary.

BLOCK 16 N is set to the value 6, indicating that $I_6[w,q]$ and $v_6[w,q]$ must be evaluated at the next iteration.

BLOCK 12 With convergence almost complete no question of changing the approximation parameters arises.

BLOCK 13 Since there is only a small difference between values of $I_4[w,q]$ and $I_5[w,q]$, it is considered unnecessary to evaluate $\bar{I}_6[w,q,v]$ at all.

BLOCK 14 Tables 3.3.2.4 and 3.3.2.5 show respectively values of $I_6[w,q]$ and $v_6[w,q]$ evaluated using the program ITER 6 with elements of the data array VMINO (IDWWL, IDQQM) all set to the corresponding value of $v_5[w,q]$.

BLOCK 15 Tables 3.3.2.4 and 3.3.2.5 show that convergence is complete to well within the accuracy expected from the approximations. However one further iteration to confirm the convergence is desirable.

BLOCK 16 N is set to 7 to indicate that $I_7[w,q]$ and $v_7[w,q]$ are to be calculated in the final iteration.

BLOCK 12 No adjustment of the approximation parameters need now be considered as convergence is virtually complete.

BLOCK 13 No evaluation of $\bar{I}_7[w,q,v]$ need be made in this final iteration, also because convergence is virtually complete.

BLOCK 14 Tables 3.3.2.4 and 3.3.2.5 show respectively values of $I_7[w,q]$ and $v_7[w,q]$ evaluating using the program ITER 6.

BLOCK 15 Tables 3.3.2.4 and 3.3.2.5 show that convergence is complete to an accuracy better than 0.05% i.e. considerably better than the accuracy of the approximations. No further iterations are therefore required.

BLOCK 17 The limiting value c* of the sequence c_N^*, N = 1,2, ... , can now be evaluated from the relationship

$$c^* = \lim_{N\to\infty} c_N^* = \lim_{N\to\infty} I_N[0,0] \approx I_7[0,0] \approx 2.17 \qquad 3.3.2.12$$

The limit, $v[w,q]$, of the sequence of functions $v_N[w,q]$, N = 1,2, ... , can be derived from the relationship

$$v[w,q] = \lim_{N\to\infty} v_N[w,q] \approx v_7[w,q] \qquad 3.3.2.13$$

The function $v[w,q]$ is shown in figure 3.3.2.17. The case $q = 0$ is drawn from the known property of equation 3.2.4.2. The break is arbitrarily placed to help interpolation.

The limit, $f[w,q]$ of the sequence of functions $f_N[w,q]$, $N = 1,2, \ldots$, can be drawn from the relation

$$f[w,q] = \lim_{N \to \infty} f_N[w,q] = \lim_{N \to \infty} \left\{ w^2 + I_N[w,q] - I_N[0,0] \right\}$$
$$\approx w^2 + I_7[w,q] - I_7[0,0] \qquad 3.3.2.14$$

The function $f[w,q]$ is shown in figure 3.3.2.18. The case $q = 0$ is drawn from the known relation of equation 3.2.4.1.

BLOCK 18 The whole iteration process has been repeated for a change of approximation parameters dw to 0.5 and W to 12. Detailed results are not presented, but the limiting values of the sequences $I_N[w,q]$ and $v_N[w,q]$, $N = 2,3, \ldots$ are given in table 3.3.2.7. Comparison of the results of the two iteration procedures in the overlapping region shows agreement to within $2\frac{1}{2}\%$. This confirms that the original choice of the approximation parameter $W = 3$ is satisfactory.

Figure 3.3.2.19 shows the function $v[w,q]$ using values obtained in the first iteration procedure, $w \leq 3$ and values obtained in the second iteration procedure for $3 < w \leq 12$. This figure shows that values of $v[w,q]$ can be reasonably interpolated from the known values for $w \leq 12$ i.e. for all values of w which can be reasonably expected to occur significantly often. Thus all numerical calculations necessary for the determination of the required optimal extremal control algorithm are now complete.

3.4 The optimal extremal controller

The results of this application of optimal control theory to the extremal control process defined in section 2.1 can now be summarised as follows:-

(1) It is assumed that the controller must minimise the expected average stage cost, assuming there is a large ($\geqslant 7$) number of stages to go.

(2) At each stage i, i = 1,2, ... , the controller must start from the sufficient statistics u(i-1), w(i-1) and q(i-1) to calculate the control input u(i). This is done in four steps:-

(a) The sufficient statistics u(i-1) and w(i-1) are normalised with respect to σ.

(b) The variable v(i) is calculated from

$$v(i) = v\left[w(i-1),\ q\ (i-1)\right] \qquad 3.4.1$$

where the function $v[w,q]$ is as shown in figures 3.3.2.17 and 3.3.2.19.

(c) The normalised control input u(i) is calculated from

$$u(i) = u(i-1) + v(i) \qquad 3.4.2$$

(d) The required control input u(i) (i.e. without normalisation) is found by multiplying the normalised control input u(i) by σ.

(3) After each stage i, i = 1,2, ... , i.e. when y(i) has been measured, the controller must calculate the sufficient statistics u(i), w(i) and q(i) using equations 2.2.1.1, 2.2.1.9 and 2.2.1.10.

(4) The expected average cost per stage, $\bar{c}$ (without normalisation), using the optimal extremal controller is

$$\bar{c} = c^{*} \cdot A\,\sigma^{2} \approx 2.17A\sigma^{2} \qquad 3.4.3$$

Chapter 4

Simulation of the Optimal Extremal Control System

It is important that any extremal controller be tested before its performance can be considered established. In the case of the optimal extremal controller derived in the previous chapter, testing is necessary to confirm that:-

(1) the analysis is free of errors.

(2) the approximations which are an essential part of the numerical solution are valid.

(3) the controller is not highly sensitive to effects such as quantisation, lags and noise which have not been considered, but which are bound to be present in any practical situation.

This chapter presents the results of testing the optimal extremal controller by simulation techniques. The first set of simulations are carried out entirely within a digital computer, simulating the system exactly as it is defined in chapters 1 and 2. The purpose of these simulations is to confirm the first two points noted above. The second set of simulations are carried out using a digital computer as the optimal controller, interfaced to an analogue computer and random signal generator, forming a hybrid system which simulates the optimal extremal control system. The primary purpose of these simulations is to confirm the third point made above.

The simulations described in this chapter were carried out using the equipment of the Oxford University Systems Engineering Group[40]. All digital computer programs were written in FORTRAN[38]. Details of all the computations, both digital and hybrid, are contained in appendix 4.

4.1 Digital simulation

This section describes the simulation of the optimal discrete-time extremal control system by digital computer. Figure 4.1.1 is a flow diagram of the simulation. The object of the simulation is to determine an estimate of the normalised mean stage cost, and an estimate of the standard deviation of this estimate.

The estimated mean stage cost and the estimate of the standard deviation of the estimate will in general depend on the process parameters A and σ. However since the system has two distinct types of unit (input units and output units) it follows from Buckingham's π theorem of dimensional analysis[41] that the estimated normalised mean stage cost and the estimate of the standard deviation of this estimate will be independent of A and σ. Thus simulation need be carried out only for one arbitrary set of values of A and σ. The simplest values to choose are

$$A = 1 \qquad \sigma = 1 \qquad 4.1.1$$

as this means that normalisation does not require any additional computation.

The previous chapter has shown that the expected mean stage cost over a long period is independent of the initial conditions. This means that the initial conditions for the simulation are immaterial, but the simulation must be sufficiently long to allow transient initial effects to decay. The method used to initialise the system is to make an initial control input

$$u(0) = 0 \qquad 4.1.2$$

An initial random sample r(0) is then generated (see later) from a Gaussian probability distribution with zero mean value and standard deviation, $\sigma = 1$. The initial values of z and y then become

$$z(0) = r(0) \qquad 4.1.3$$

$$\text{and} \quad y(0) = [z(0)]^2 \qquad 4.1.4$$

The sufficient statistics w(0) and q(0) are then set to

$$w(0) = \sqrt{y(0)}$$
$$q(0) = 0.5 \qquad 4.1.5$$

This completes the initialisation of the system.

In the main simulation calculation of the optimal control input follows the details set out in section 3.4. The function $v[w,q]$ is held as a table of values at the grid points. Linear interpolation is used within the table and the asymptotic function is used for values outside the table.

Calculation of the process response follows the equations set out in chapter 2. This involves the generation of a sequence of independent samples from a Gaussian probability distribution, simulated by a pseudo-random sequence[42]. Generation of the pseudo-ramdon sequence is based on the power residue method of generation of uniformly distributed random numbers, and the normal law which shows that the summation of ~ 12 (or more) such numbers will produce an approximately Gaussian distribution. The period of the sequence of pseudo-random numbers produced in this way is 2048 (in an IBM 1130 computer) which sets a limit on the number of stages of the extremal system which can be usefully simulated in this way.

Estimation of the mean stage cost and estimation of the standard deviation of this estimate can be carried out by considering the finite set of values of the stage cost obtained in the simulation as a small sample from the infinite set of values which would be obtained by allowing the process to continue indefinitely. The required results can then be obtained from the small sampling theory of statistics[43].

The estimate of the normalised mean stage cost, $\bar{c}$, is simply the average value of the stage costs obtained. The estimate of the standard deviation of this estimate is found by considering the values of the

stage costs obtained as M groups each containing L stages. For each group k, k = 1,2, ... , M, the mean value, $\bar{c}(k)$, is estimated as the average value of the L stage costs in the group. The estimate, SDOA, of the standard deviation of the estimate of the normalised mean stage cost is then given by

$$SDOA = \sqrt{\left\{ \frac{1}{M(M-1)} \sum_{k=1}^{M} \left[\bar{c}(k) - \bar{\bar{c}} \right]^2 \right\}} \qquad 4.1.6$$

Suitable values for L and M are

$$L = M = 45 \qquad 4.1.7$$

with $45^2 = 2025$ 2048.

Table 4.1.1 shows the results of several simulations carried out using the program SIM1. The average estimate of the mean stage cost is seen to be 2.26. The estimate of the standard deviation of this estimate is 0.1 i.e. about 5%. This indicates that the number of stages in the simulation is sufficient to have made initial transient effects negligible, and to have made a meaningful estimate of the normalised mean stage cost.

Comparison of the estimated normalised mean stage cost of 2.26 ± 0.1 with the predicted value of 2.17 is taken as confirmation that the optimal extremal controller has been derived without significant loss of accuracy, and that it performs as predicted.

4.2 Hybrid simulation

This section describes the simulation of the optimal extremal control system where the extremal process operates in continuous-time, but the controller is discrete-time. This situation is more like what might be expected in practice. The purpose of the simulation is

to show that the lags, noise and quantisation effects introduced by the interfacing between the process and the controller do not reduce performance significantly.

Description and discussion of the simulation is done in three parts. First it is shown how the continuous-time extremal process appears equivalent to a discrete-time process to the discrete-time controller. Then it is shown how the extremal process can be simulated using a random signal generator and an analogue computer. Finally the simulation of the hybrid extremal control system is discussed and the results presented and discussed.

4.2.1 Continuous extremal processes Figure 4.2.1.1 is a block diagram of a mathematical model which represents a continuous-time extremal process, where there are no lags or noise[6]. The mean operating cost of such a process can be maintained arbitrarily close to zero by use of a perturbation controller with perturbations of vanishingly small amplitude, and infinitely high frequency[6]. However if such a process is under computer control, then it will be possible to make measurements and adjust the input only at fixed points in time. In this case the control of the process is exactly equivalent to the discrete-time case, provided the object of the controller is to minimise the mean value of the cost rate at measurement times only.

This can be seen by considering operation of the process from time $t = iT$ to time $t = \overline{i+1}\,T$ when T is the time between measurement/adjustments. Letting (i) following a variable name denote the value at time $t = iT$, then referring to figure 4.2.1.1,

$$z(i+1) = z(i) + K \cdot \int_{iT}^{\overline{i+1}T} n(t)\,dt, \quad i = 0,1, \ldots \qquad 4.2.1.1$$

Defining $\phi_{nn}(w)$, the power spectral density of n(t) as

$$\phi_{nn}(w) = \frac{1}{\pi} \text{ Fourier Transform}\left\{\psi_{nn}(\tau)\right\} \qquad 4.2.1.2$$

where ψ_{nn} is the autocorrelation function of n(t), then n(t) is defined as a signal with Gaussian amplitude distribution of zero mean value, being white noise with power spectral density B^2. It is then a standard result[6] that $K \int_{iT}^{i+1T} n(t)dt$ is a sample r(i+1) from a Gaussian probability distribution with zero mean value and standard deviation $KB\sqrt{\pi T}$; also the sequence of values r(i), i = 1,2, ... are statistically independent.

Comparison with the equations defining the purely discrete-time process in chapter 2 now shows that control of the continuous and discrete-time processes is equivalent, with

$$\sigma \equiv KB\sqrt{\pi T} \qquad 4.2.1.3$$

Since the process involves no dynamics, and only values of the cost-rate at measurement times are considered, the value of the control input between measurement/adjustment times is of no consequence.

4.2.2 Simulation of the extremal process The extremal process shown in figure 4.2.1.1 can be simulated by representing all variables by voltages, using a random signal generator and an analogue computer. Choice of suitable parameters for the extremal process, and a suitable measurement/adjustment period, T, is governed by the bandwidth of the random signal generator and the saturation level of the amplifiers in the analogue computer. This section describes the simulation of the extremal process using a Servomex Controls Ltd. RG77 random signal generator with a nominal bandwidth of 10Hz ($\equiv$ 62.8 rad/sec) and an Electronic Associates Ltd. TR48 analogue computer which operates in the range $\pm$ 10 Volts.

Figure 4.2.2.1 shows the configuration used to simulate the white noise signal n(t). The object of the high-pass filter, cut off 0.016 Hz(≡ 0.1 rad/sec), is to remove any d.c. components of the signal. The object of the low-pass filter, cut-off 10Hz (≡ 62.8 rad/sec) is to ensure that any irregularities in the frequency response beyond the nominal bandwidth are suppressed. This leaves over two decades in which the simulated noise signal is approximately white i.e. has constant power spectral density.

Besides being white, the other requirement of the noise signal n(t) is that it should have a Gaussian amplitude distribution, with zero mean value. The latter is assured by the high-pass filter. The hybrid computation CAL1 has been used to estimate the amplitude distribution. A histogram of 2500 samples of n(t) is shown in figure 4.2.2.2, and compared with a Gaussian distribution. They are seen to agree within the accuracy of the histogram. The standard deviation, σ_n, of the distribution is 41.3 units ≡ 2.01 Volts.

The measurement/adjustment frequency of the simulated extremal process can now be chosen as 1 Hz (i.e. T = 1 sec), well between the cut-off frequencies of the filters on the random signal generator. Thus the simulated white noise source may be considered white in the sense that it has approximately constant power spectral density for approximately one decade above and below the measurement/adjustment frequency. It has also been shown to have a Gaussian amplitude distribution. It can therefore be expected to provide a good simulation of the signal n(t) in figure 4.2.1.1.

The real check on the suitability of the simulated noise signal is found by considering the discrete/continuous correspondence as discussed in section 4.2.1. The sequence r(i), i = 1,2, ... , where

$$r(i) \equiv \int_{t=i-1}^{t=1} n(t)\, dt \quad , i = 1,2, \ldots \qquad 4.2.2.1$$

t in seconds

should be a sequence of independent samples from a Gaussian probability distribution with zero mean value. The latter condition is assured by the high-pass filter.

The hybrid computation CAL2 has been used to estimate the probability distribution. A histogram of 1000 samples of the sequence $r(i)$, $i = 1,2, \ldots$ is shown in figure 4.2.2.3 and compared with a Gaussian distribution. They are seen to agree within the accuracy of the histogram. The standard deviation, σ_i, of the distribution is 32.0 units $\equiv$ 0.40 Volts.

The hybrid computation CAL3 has been used to estimate the normalised autocorrelation function $\psi_{rr}(j)$, $j = 0,1, \ldots$ of the sequence $r(i)$, $i = 1,2, \ldots$, from 1000 successive samples. The estimated normalised autocorrelation function is shown in fugure 4.2.2.4. $\psi_{rr}(j)$, $j = 0,1, \ldots$, represents the correlation between samples j periods apart[44]. The figure shows that $|\psi_{rr}(j)| \leq 0.11$, $j = 1,2, \ldots$, i.e. the sequence $r(i)$, $i = 1,2, \ldots$, is a sequence of approximately independent samples.

Thus the simulated variable $n(t)$ has been shown to be a suitable representation of the ideal variable $n(t)$, provided the measurement/adjustment frequency of the simulated extremal process is 1 Hz.

Figure 4.2.2.5 shows the configuration used to simulate the extremal process. The parameter A of the equivalent discrete-time process is seen to be

$$A = 0.1 \qquad 4.2.2.2$$

The parameter σ of the equivalent discrete-time process is determined by the constant K,

$$\sigma = 10\,K\,\sigma_i = 4.0K \qquad 4.2.2.3$$

This must be chosen to maintain the simulated signal x(t) within the region $\pm$ 10V. Experience from the discrete-time simulations shows that the maximum value of x/σ normally encountered is 6, so that

$$\text{Max}(x/\sigma) = 10/\sigma = 6 \qquad 4.2.2.4$$

$$\sigma = 1.67 \qquad 4.2.2.5$$

$$\text{and} \quad K = 0.417 \qquad 4.2.2.6$$

The only signal which is still likely to attain a value beyond 10 Volts is the simulated z(t) signal, and this problem is left until the next section.

4.2.3 <u>The complete simulation</u> The simulated extremal process derived in the previous section shows the optimal control input u(t) derived directly from the digital computer output. In this situation there is no guarantee that the simulated signals u(t) and z(t) will not drift off to values larger than the system can simulate (i.e. $\pm$ 10 Volts). An alternative, shown in figure 4.2.3.1 is to derive the optimal control increment v(t) from the digital computer output, and to generate the optimal control input u(t) by analogue integration over the period T = 1 sec. Since the value of u(t) between sampling times is irrelevant in this situation, this approach is valid. The integrators producing z(t) and u(t) can now be combined for simulation purposes to avoid explicit simulation of u(t) and z(t). This final configuration for the simulated extremal process is shown in figure 4.2.3.2.

Figure 4.2.3.3 is a block diagram of the configuration used to

simulate the hybrid extremal control system. Details of the interface units are given in the introduction to appendix 4. Details of the translation and scaling necessary on the digital and analogue computer input and output to compensate for interface effects are given in section A4.7. SIM2 is the hybrid computation used to carry out the simulation. The digital computer acts as the optimal controller. Every second the interupt clock causes it to:-

(1) Make a measurement and carry out the necessary scaling.

(2) Calculate the optimal control increment v, exactly as in the discrete-time case.

(3) Translate and scale the control increment and output it as an adjustment.

The time required for these three calculations is $\sim$ 0.1 sec. This continues until 45 groups each of 45 stages has passed, when simulation stops and the digital computer estimates the normalised mean stage cost and the standard deviation of this estimate exactly as for the discrete-time case.

Results show an estimated mean cost per stage of 2.08 with a standard deviation of 0.1 in the estimate. This result again confirms that the optimal extremal control system performs as predicted. It also shows that the effects of noise, lags and quantisation introduced by the interfacing equipment are negligible. Thus practical implementation of the optimal extremal controller can be expected to yield performance close to that predicted.

Core store requirements for the controller were found to be $\sim$ 4,000 words (16 bit). This could be reduced to about 3,400 words by storing the values of $v[w,q]$ in a backing store. Thus the optimal extremal control algorithm could be implemented on most small computers,

and still leave room for other requirements, particularly if some effort were put into efficient encoding of the control algorithm in machine code.

The conclusion to this chapter is that the optimal extremal control system has been successfully tested according to the requirements set out at the beginning of the chapter, and that the optimal extremal control algorithm has been shown to be capable of practical implementation.

Chapter 5

Comparison of the Performance of the Optimal Extremal Controller with the Performance of other Extremal Controllers

A meaningful comparison between the performances of the different controllers can only be made if they are controlling the same process, and the same performance criterion is used. A number of heuristic (i.e. not based on optimal control theory) discrete-time extremal control algorithms have been proposed[21-23] to control the extremal process for which the optimal extremal controller has been derived. The performance of these controllers has generally been evaluated not only for the simplified case, but also for the case where there is output noise. This chapter therefore starts by presenting the results of hybrid simulations carried out to evaluate the performance of the optimal extremal controller when it is used to control, not the process for which it is designed, but a similar process in which output noise is considered. These results then form a basis for the comparison of the performance of the optimal extremal controller with the performances of the heuristic controllers.

The chapter continues by comparing the performance of the optimal extremal controller with that of other extremal controllers based on optimal control theory [27-30]. This includes the presentation of the results of hybrid computations carried out to establish the performance of a suboptimal controller[29] where output noise is taken into account.

Finally the performances of heuristic extremal controllers and extremal controllers based on optimal control theory are compared.

The simulations described in this chapter were carried out

using the equipment of the Oxford University Systems Engineering Group[40]. All digital computer programs were written in FORTRAN[38]. Details of all the computations are contained in appendix 4.

5.1 The effect of output noise on the performance of the optimal extremal controller

In order to take the effect of output noise into account it is necessary to modify the mathematical model of the extremal process derived in chapter 2. This is done by replacing equation 2.1.8 by

$$y(i) = c(i) + \xi(i), \quad i = 1,2, \ldots \qquad 5.1.1$$

where $\xi(i)$, $i = 1,2, \ldots$, is a sequence of independent samples from a Gaussian probability distribution with zero mean value and standard deviation σ_1.

For any given controller the mean cost per stage, $\bar{c}$, of the process is a function of the three parameters A, σ and σ_1. Application of Buckingham's π theorem[41] of dimensional analysis shows that this function may be expressed as

$$\frac{\bar{c}}{A\sigma^2} = \bar{c}\left[\frac{\sigma_1}{A\sigma^2}\right] \qquad 5.1.2$$

$$\text{or} \quad \pi_1 = \bar{c}[\pi_2] \qquad 5.1.3$$

$$\text{where} \quad \pi_1 = \frac{\bar{c}}{A\sigma^2} \qquad 5.1.4$$

$$\text{and} \quad \pi_2 \quad \frac{\sigma_1}{A\sigma^2} \qquad 5.1.5$$

The hybrid computation SIM3 (with the subprogram CONT1) has been used to determine the $\pi_1 \sim \pi_2$ relationship using the optimal controller and simulation methods similar to those used in chapter 4. Points to note are:-

(1) the sequence $\xi(i)$, i = 1,2, ... is simulated from a pseudo-random sequence (see section 4.1) in the digital computer.

(2) if y(i), i = 1,2, ... , is negative then it is set to zero as the true stage costs are known to be positive semi-definite.

The results of the simulations are shown in figure 5.1.1. The accuracies of the estimated mean stage costs range from a standard deviation of $\sim 5\%$ for $\pi_2 = 0$ to $\sim 8\%$ for $\pi_2 = 10$. The figure shows that for $\pi_2 < 3.3$, $\pi_1 > \pi_2$ i.e. the output noise is dominated by the stage cost on average. In this region measurements are generally close to the true stage cost and the controller maintains good control. For $\pi_2 > 3.3$, the output noise takes over as the major contribution to the measurements, and since the controller has no inbuilt features to deal with the noise, the performance rapidly degenerates.

5.2 Comparison with the performance of heuristic controllers

In this section the performances of various heuristic controllers[21-23] as established by previous authors are set down in the notation of this thesis. The performances are then compared with each other and with the performance of the optimal extremal controller, and the differences discussed.

5.2.1 Fixed step size controller This type of control law has been simulated by both Jacobs[22] and Woodside[21]. The control input u(i+1) is chosen as

$$u(i+1) = u(i) + v(i+1), \quad i = 1,2, \ldots \qquad 5.2.1.1$$

For a given extremal process, i.e. with fixed A, σ and σ_1, v(i+1) is a step of fixed size L, defined by

$$v(i+1) = - L \text{ sign}\left[\left[y(i) - y(i-1)\right] / \left[u(i) - u(i-1)\right] \right], \quad i = 1,2, \ldots \qquad 5.2.1.2$$

so that u is effectively moved in constant steps until the stage cost appears to increase, then the direction of stepping is reversed.

The mean cost per stage, $\bar{c}$, is now a function of the four parameters A, σ , σ_1 and L, but use of Buckingham's π theorem[41] shows that this can be expressed as

$$\pi_1 = \bar{c}\left[\pi_2, \pi_3 \right] \qquad 5.2.1.3$$

with π_1 and π_2 defined as in section 5.1, and

$$\pi_3 \equiv L/\sigma \qquad 5.2.1.4$$

For each fixed π_2 there is a value of π_3 which minimises π_1. Both Jacobs and Woodside carried out simulations to find the optimising value of π_3 as a function of π_2, and then constructed the optimum $\pi_1 \sim \pi_2$ curve for this type of controller. The results differ slightly and are shown in figure 5.2.1.1. The reason for the difference is not clear but may lie in the different approaches taken to estimating when sufficient stages of the process have been simulated.

5.2.2 Derivative estimating gradient controller Chang[23] originally suggested the derivative estimating gradient controller, but in a form which is not directly comparable with these results because measurements are not made at equal time intervals. Woodside[21] modified this method to make it comparable. The

control input u(i+1) is again chosen as

$$u(i+1) = u(i) + v(i+1),\ i = 1,2,\ \ldots \qquad 5.2.2.1$$

and v(i+1) is given by

$$v(i+1) = \begin{cases} b(i) & i \text{ odd} \\ b(i) - a\,D(i) & i \text{ even} \end{cases},\ i = 1,2,\ \ldots \qquad 5.2.2.2$$

$$\text{where } b(i) = \begin{cases} 1/2 & ,\ i = 1 \\ l(-1)^{\overline{i+1}} & ,\ i > 1 \end{cases},\ i = 1,2,\ \ldots \qquad 5.2.2.3$$

$$\text{and } D(i) = \left[y(i) - y(i-1)\right]/l\ ,\ i = 1,2,\ \ldots \qquad 5.2.2.4$$

Both l and a are parameters of the controller, fixed for fixed A, σ , and σ_1. The effect of the controller is seen to be to estimate the gradient of the curve of stage cost c(i) against control input u(i), and make a step proportional to this gradient. Making use of dimensional analysis[41] as before, simulations were carried out by Woodside to find the optimum values of l and a in terms of the process parameters A, σ and σ_1. From this an optimum $\pi_1 \sim \pi_2$ curve was constructed with π_1 and π_2 defined as previously, and this is shown in figure 5.2.2.1.

5.2.3 Alternate biasing gradient controller This type of controller was first suggested and analysed by Chang[23]. First an estimate $u'(i+1)$, of the value of u(i+1) which would minimise c(i+1) is made as

$$u'(i+1) = u'(i) + \sum_{j=0}^{i} w(i-j).(-1)^j\, y(j),\ i = 1,2,\ \ldots \qquad 5.2.3.1$$

where w(i), i = 0,1,2, ... , is a specified weighting sequence found by linearisation of the problem and use of sampled data

theory. The control input u(i+1) is then chosen as

$$u(i+1) = u'(i+1) + (-1)^{i+1}l \qquad 5.2.3.2$$

where l is a parameter of the controller. Chang calculated the optimum l as a function of A, σ and σ_1 and hence the optimum $\pi_1 \sim \pi_2$ curve. This is shown in figure 5.2.3.1. The assumptions in the linearisation are not valid for small π_2, so this end of the curve is invalid.

Woodside simplified the algorithm and used

$$u(i+1) = u(i) + v(i+1),\ i = 1,2,\ \ldots \qquad 5.2.3.3$$

with $$v(i+1) = -aD(i) + 2b(i),\ i = 1,2,\ \ldots \qquad 5.2.3.4$$

$$D(i) = \left[y(i) - y(i-1)\right] / \left[u(i) - u(i-1)\right] \qquad 5.2.3.5$$

and $$b(i) = \begin{cases} 1/2 & i = 1 \\ l(-1)^{\overline{i+1}} & i > 1 \end{cases},\ i = 1,2,\ \ldots \qquad 5.2.3.6$$

Both a and l are constant controller parameters which Woodside optimised as in previous cases. The resultant optimum $\pi_1 \sim \pi_2$ curve is shown in figure 5.2.3.1. The approximate theoretical results of Chang are seen to agree within $\sim$ 15% with the experimental results of Woodside which is acceptable since the control algorithms are not identical.

5.2.4 Comparison of performances Figure 5.2.4.1 shows curves of $\pi_1 \sim \pi_2$ for the various types of heuristic controller. Avera e values have been taken between Woodside and Jacobs for the fixed step size and between Woodside and Chang for the alternate biasing systems. The following conclusions can be drawn:-

(1) The fixed step size controller which is the simplest of the

algorithms gives the worst performance of all the heuristic controllers, for the values of π_2 considered.

(2) The derivative sensing controller which is performing control effectively only every second stage generally gives slightly worse performance than the alternate biasing controller which performs control at every stage.

(3) For values of $\pi_2 < 3$ the optimal controller gives performance significantly better than the heuristic controllers, and the slope of the $\pi_1 \sim \pi_2$ curve for the optimal controller is similar to that of the heuristic controllers. However for $\pi_2 > 3$, the optimal controller is considerably affected by the fact that output noise is not taken into account in the design, and the performance deteriorates rapidly to become the worst.

(4) For values of $\pi_2 < 4$ performance improves consistently with the complexity of the controller. A decision as to which controller to use in any application would therefore be based on the individual trade-off between controller complexity and performance.

5.3 Comparison with the performance of controllers based on optimal control theory

5.3.1 Optimal controllers A previous attempt by MOZGOVAYA and SAMOYLENKO[30] has been made to find the optimal control algorithm for the extremal process defined in section 2.1. However this paper has an error in the calculation of the updating equations for the probability distributions which is not serious, and an error in the formulation of the cost functions ($\equiv f_N[w,q]$ in this thesis) which oversimplifies the problem and leads to completely erroneous results. A simulation has been carried out using a digital computation similar to SIM1 and control algorithm taken from the diagrams in the Russian

paper. The results show a normalised mean stage cost of 7 compared to the value 1.7 predicted in the paper, and this result verifies the existence of serious errors in the original analysis.

Roberts[28] has considered optimal control of continuous-time extremal processes, using linearisation techniques, but the continuous-time controller derived is not directly comparable with the optimal discrete-time controller.

5.3.2 Suboptimal controllers Florentin[27] has considered approximately optimal control of a discrete-time system with a lag in the control input, and noise. The principle involves approximating the probability distribution for the state by parameters specifying its main features, then basing control on these parameters. Simulation results are given for specific cases, but as the control input lag is a fixed feature, no direct comparison is possible.

Jacobs[29] has suggested a suboptimal algorithm for the extremal process defined in section 2.1, based on the same sufficient statistics w and q as are used by the optimal controller. The algorithm is defined by

$$u(i+1) = u(i) + v(i+1), \quad i = 1,2, \ldots \qquad 5.3.2.1$$

$$\text{where } v(i+1) = \overline{1-2q(i)}.w(i) + a \operatorname{sign}\left[1-2q(i)\right] \qquad 5.3.2.2$$

$$i = 1,2, \ldots$$

and a is a parameter of the controller. The first term in equation 5.3.2.2 is $-\mathrm{Exp}\{x(i)\}$ (Exp = mathematical expectation; x = input to extremal non-linearity), and the second term is called the intentional error. The mean cost per stage, $\bar{c}$, is then a function of A, σ and a, and can be expressed by virtue

of Buckingham's π theorem[41] in the form

$$\pi_1 = \bar{c}\left[\pi_3\right] \qquad 5.3.2.3$$

$$\text{where } \pi_3 = a/\sigma \qquad 5.3.2.4$$

and π_1 is as defined previously. Jacobs simulated the system and found the optimum value of π_3 to be 0.8, giving π_1 the value 2.2, indistinguishable from the performance of the optimal algorithm.

The hybrid computation SIM3 (with the subprogram CONT2) has been used to extend this algorithm to control the extremal process with output noise defined in section 5.1. In this case π_1 ($\equiv \bar{c}/A\sigma^2$) is a function of both $\pi_2 (\equiv \sigma_1/A\sigma^2)$ and $\pi_3 (\equiv a/\sigma)$. Figure 5.3.2.1 shows curves of $\pi_1 \sim \pi_3$ drawn from the results of the simulations, for fixed values of π_2. Figure 5.3.2.2 shows curves of $\pi_1 \sim \pi_2$ drawn for the optimum values of π_3 as established in figure 5.3.2.1. Comparison with the performance of the optimal controller shows that the optimal controller gives better performance for higher values of π_2, but the performances are not distinguishable for values of $\pi_2 < 2$.

To compare the form of the optimal and suboptimal control algorithms it is convenient to define the optimal intentional error $e[w,q]$ as

$$e\left[w,q\right] \equiv v\left[w,q\right] - \overline{1-2q}.w \qquad 5.3.2.5$$

Curves of $e[w,q]$ are shown in figure 5.3.2.3. They show that for values of $w < 1.5$ (i.e. the values normally encountered when there is no output noise) the optimal intentional error is approximately constant, and the average value is ~ 0.8 which

is the optimum value of the fixed intentional error used in the suboptimal algorithm. This explains the closely optimal performance of the suboptimal algorithm for small values of π_2. When π_2 is larger, values of w encountered become larger, and the optimal intentional error for $w > 1.5$ is far from constant. This explains the difference in performance found between optimal and suboptimal algorithms for larger values of π_2.

Comparing these results with the results found in section 5.2, the general result that for small values of π_2 (i.e. $\pi_2 < 4$) performance improves with complexity still holds true. Algorithms based on optimal control theory are significantly better than the heuristic controllers, but the optimal and suboptimal controllers are indistinguishable in performance. For larger values of π_2, the heuristic algorithms, which are designed taking output noise into account become significantly better than the algorithms based on optimal control theory.

A discussion of the significance of these results for applications and for further study of extremal control systems is left to chapter 7.

Chapter 6

The Application of Optimal Control Theory to more Complex Extremal Processes

The last chapter has shown that extremal controllers based on optimal control theory show a significantly better performance than heuristic controllers when the simplified extremal process defined in section 2.1 is being controlled. It has also shown that when the extremal controllers based on optimal control theory (and designed for the simplified process) are used to control an extremal process with output noise, they lose out to heuristic controllers as soon as the output noise becomes considerable. It therefore follows that it may be worth while to use optimal control theory to design extremal controllers, taking some of the more complex effects such as lags, noise and multiple inputs int account in the design. This chapter investigates some of these possibilities, indic ting in each case how further study might pr ceed.

The cases considered are

(1) the case when there is a time lag between the stage cost and the measurement.

(2) the case where the measurement is not precise i.e. there is output noise.

(3) the case where there are two control inputs.

It can reasonably be expected that the possibilities and difficulties found while investigating these three cases will be indicative of the possibilities and difficulties of using optimal control theory to design extremal control ers for other more complex extremal processes. The results of this chapter are

therefore intended to provide a basis for a discussion in chapter 7 of the future role of optimal control theory in designing extremal controllers.

6.1 Optimal control of an extremal process with an output lag

6.1.1 The mathematical model An extremal process with an output lag can be represented by the mathematical model shown in figure 6.1.1.1. The lag is assumed to be one control period. Equations defining the process are

$$z(i+1) = z(i) + r(i+1) \ , \ i = 0,1, \ldots \qquad 6.1.1.1$$

with $z(0) = 0$ and $r(i)$, $i = 1,2, \ldots$, is a sequence of independent samples from a Gaussian probability distribution with zero mean value and standard deviation σ .

$$x(i) = u(i) + z(i) \ , \ i = 0,1, \ldots \qquad 6.1.1.2$$

where $u(i)$, $i = 1,2, \ldots$ is the sequence of control inputs, and $u(0) = 0$.

$$y(i+1) = c(i) = Ax^2(i) \ , \ i = 0,1, \ldots \qquad 6.1.1.3$$

where $y(i)$, $i = 1,2, \ldots$ is the sequences of measurements, $c(i)$, $i = 1,2, \ldots$ is the sequence of stage costs, and A is a positive constant.

The optimal extremal controller must choose the sequence of control inputs on the basis of the measurements, so as to minimise

$$C_N = \mathrm{Exp} \sum_{i=1}^{N} c(i) \qquad 6.1.1.4$$

at each stage, where N is the number of stages remaining.

6.1.2 State variables Combining equations 6.1.1.1 and 6.1.1.2 gives

therefore intended to provide a basis for a discussion in chapter 7 of the future role of optimal control theory in designing extremal controllers.

6.1 Optimal control of an extremal process with an output lag

6.1.1 The mathematical model An extremal process with an output lag can be represented by the mathematical model shown in figure 6.1.1.1. The lag is assumed to be one control period. Equations defining the process are

$$z(i+1) = z(i) + r(i+1) \ , \ i = 0,1, \ldots \qquad 6.1.1.1$$

with $z(0) = 0$ and $r(i)$, $i = 1,2, \ldots$, is a sequence of independent samples from a Gaussian probability distribution with zero mean value and standard deviation σ .

$$x(i) = u(i) + z(i) \ , \ i = 0,1, \ldots \qquad 6.1.1.2$$

where $u(i)$, $i = 1,2, \ldots$ is the sequence of control inputs, and $u(0) = 0$.

$$y(i+1) = c(i) = Ax^2(i) \ , \ i = 0,1, \ldots \qquad 6.1.1.3$$

where $y(i)$, $i = 1,2, \ldots$ is the sequences of measurements, $c(i)$, $i = 1,2, \ldots$ is the sequence of stage costs, and A is a positive constant.

The optimal extremal controller must choose the sequence of control inputs on the basis of the measurements, so as to minimise

$$C_N = \mathrm{Exp} \sum_{i=1}^{N} c(i) \qquad 6.1.1.4$$

at each stage, where N is the number of stages remaining.

6.1.2 State variables Combining equations 6.1.1.1 and 6.1.1.2 gives

$$x(i+1) = x(i) + u(i+1) - u(i) + r(i+1) \qquad 6.1.2.1$$

Consider now the vector $\underline{X}$ defined by

$$\underline{X}(i) \quad \begin{bmatrix} X1(i) \\ X2(i) \\ X3(i) \end{bmatrix} \equiv \begin{bmatrix} x(i) \\ x(i-1) \\ u(i) \end{bmatrix} \qquad i = 1,2, \ldots \qquad 6.1.2.2$$

Equations 6.1.1.3 and 6.1.2.1 combined with equation 6.1.2.2 show that $\underline{X}$ satisfies all the conditions set out in chapter 1 to be a state vector.

6.1.3 Sufficient statistics It is now necessary to consider the probability distribution for $\underline{X}$ at stage i, i = 1,2, ... when y(i) has been measured. X3(i) is evidently completely known and represented by the sufficient statistic U1(i) = u(i).

Equation 6.1.1.3 shows that measurement y(i) determines the absolute value of X2(i), so that the probability distribution for X2(i) can be represented by the sufficient statistics

$$w(i) \equiv |x(i-1)| = \sqrt{y(i)/A} \qquad i = 1,2, \ldots \qquad 6.1.3.1$$

$$\text{and } q(i) \equiv \text{probability that } x(i-1) \text{ was positive} \qquad 6.1.3.2$$

These sufficient statistics are directly equivalent to the sufficient statistics used in the simplified case. The updating law for q(i) follows as before from the laws of probability, with a slight change of time suffices.

$$q(i+1) = \frac{P1.q(i) + P2.\overline{1-q(i)}}{(P1+P3).q(i) + (P2+P4).\overline{1-q(i)}}$$

$$\text{where } P1 = \exp\left\{-\left[w(i+1) - w(i) - u(i) + u(i-1)\right]^2/2\sigma^2\right\}$$
$$P2 = \exp\left\{-\left[w(i+1) + w(i) - u(i) + u(i-1)\right]^2/2\sigma^2\right\}$$
$$P3 = \exp\left\{-\left[w(i+1) + w(i) + u(i) - u(i-1)\right]^2/2\sigma^2\right\} \qquad 6.1.3.3$$
$$P4 = \exp\left\{-\left[w(i+1) - w(i) + u(i) - u(i-1)\right]^2/2\sigma^2\right\}$$

The updating equations for q(i+1) have introduced a new sufficient statistic U2(i) = u(i-1). This comes from the probability distribution for X1(i) given X2(i), $\bar{p}\left[X1 \mid X2\right]$. Equation 6.1.2.1 shows that

$$X1(i) = X2(i) + u(i) - u(i-1) + r(i) \qquad 6.1.3.4$$

Since y(i) yields no information about r(i), the probability distribution for X1(i) given X2(i) becomes a Gaussian distribution with mean value (X2(i) + u(i) - u(i-1)) and standard deviation σ. Thus it can be represented by the additional sufficient statistic U2(i) = u(i-1).

The probability distribution for $\underline{X}$ i.e. the joint probability distribution for the elements of $\underline{X}$ can thus be represented by the four sufficient statistics w,q, U1 and U2.

6.1.4 Optimal control Since sufficient statistics have now been established, it is now possible to proceed to use dynamic programming to formulate a functional recurrence equation for the cost functions and optimal control algorithm. This is a subject for further research. With four sufficient statistics to consider, it is unlikely that the equation could be solved without an impossible amount of computer time. It might therefore be valuable to examine the equations to see whether any suboptimal algorithms can be determined.

6.2 Optimal control of an extremal process with measurement noise

6.2.1 The mathematical model An extremal process with measurement noise can be represented by the mathematical model shown in figure 6.2.1.1. Equations definin the process are

$$z(i+1) = z(i) + r(i+1) \ , \ i = 0,1, \ldots \qquad 6.2.1.1$$

with $z(0) = 0$ and $r(i)$, $i = 1,2, \ldots$, is a sequence of independent samples from a Gaussian probability distribution with zero mean value and standard deviation σ.

$$x(i) = u(i) + z(i) \ , \ i = 0,1, \ldots \qquad 6.2.1.2$$

where $u(i)$, $i = 1,2, \ldots$ is the sequence of control inputs, and $u(0) = 0$.

$$c(i) = Ax^2(i) \ , \ i = 1,2, \ldots \qquad 6.2.1.3$$

where $c(i)$, $i = 1,2, \ldots$ is the sequence of stage costs and A is a positive constant.

$$y(i) = c(i) + \xi(i), \ i = 1,2, \ldots \qquad 6.2.1.4$$

where $y(i)$, $i = 1,2, \ldots$ is the sequence of measurements and $\xi(i)$, $i = 1,2, \ldots$, is another sequence of independent samples from a Gaussian probability distribution with zero mean value and standard deviation σ_1.

The optimal extremal controller must choose the sequence of control inputs on the basis of the measurements so as to minimise

$$C_N = \text{Exp} \sum_{i=1}^{N} c(i) \qquad 6.2.1.5$$

at each stage, where N is the number of stages remaining.

6.2.2 State variables Combining equations 6.2.1.1 and 6.2.1.2

gives

$$x(i+1) = x(i) + u(i+1) - u(i) + r(i+1) \qquad 6.2.2.1$$

Consider now the vector $\underline{X}$ defined by

$$\underline{X}(i) \equiv \begin{bmatrix} X1(i) \\ \\ X2(i) \end{bmatrix} \equiv \begin{bmatrix} x(i) \\ \\ u(i) \end{bmatrix}, \quad i = 1,2, \ldots \qquad 6.2.2.2$$

Equations 6.2.1.3, 6.2.1.4 and 6.2.2.1 combined with equation 6.2.2.2 show that $\underline{X}$ satisfies all the conditions set out in chapter 1 to be a state vector.

6.2.3 Sufficient statistics It is now necessary to consider the probability distribution for $\underline{X}$ at stage i, i = 1,2, ... when y(i) has been measured. X2(i) is evidently completely known and represented by the sufficient statistic u1(i) = u(i).

The probability distribution for X1(i) is $p_i[X1]$, and there is no obvious set of sufficient statistics to describe $p_i[X1]$ for i = 1,2, ... However it is still possible to use the laws of probability to determine updating equations giving $p_{i+1}[X1]$ from $p_i[X1]$, u(i), u(i+1) and y(i+1). This is done by considering the probability P of occurrence of the joint event $x \leq x(i+1) \leq x + \delta x$ and $y \leq y(i+1) \leq y + \delta y$, which can be written in two ways.

$$P = \text{Prob}\{y \leq y(i+1) \leq y + \delta y \mid x \leq x(i+1) \leq x + \delta x\} \times \text{Prior prob}\{x \leq x(i+1) \leq x + \delta x\} \qquad 6.2.3.1$$

$$\text{or } P = \text{Prob}\{x \leq x(i+1) \leq x + \delta x \mid y \leq y(i+1) \leq y + \delta y\} \times \text{Prior prob}\{y \leq y(i+1) \leq y + \delta y\} \qquad 6.2.3.2$$

It is convenient to define P1(r) and P2(ξ) as the respective Gaussian probability distributions for r and ξ respectively. Then the probabilities referred to in equations 6.2.3.1 and 6.2.3.2 can be determined as follows

$$\begin{aligned} PR1(x,y) &\equiv \text{Prob}\left\{ y \leq y(i+1) \leq y + \delta y \mid x \leq x(i+1) \leq x + \delta x \right\} \\ &\approx \text{Prob}\left\{ y \leq y(i+1) \leq y + \delta y \mid x(i+1) = x \right\} \text{ for small } \delta x \\ &= \text{Prob}\left\{ y \leq y(i+1) \leq y + \delta y \mid c = Ax^2 \right\} \\ &\approx \delta y \,.\, P2(y - Ax^2) \text{ for small } \delta y \end{aligned} \qquad 6.2.3.3$$

PR2(x) = Prior prob $\left\{ x \leq x(i+1) \leq x + \delta x \right\}$ is obtained from equation 6.2.2.1 by convolution of $p_i\left[x(i+1) \right]$ with P1(r) to give

$$PR2(x) \approx \delta x \int_{-\infty}^{\infty} P1(r)\; p_i\left[x - u(i+1) + u(i) - r \right] dr \qquad 6.2.3.4$$

for small δx

$$PR3(x,y) = \text{Prob}\left\{ x \leq x(i+1) \leq x + \delta x \mid y \leq y(i+1) \leq y + \delta y \right\}$$

$$\approx \delta x \,.\, p_{i+1}\left[x \right] \text{ for small } \delta x, \delta y \qquad 6.2.3.5$$

$$PR4(y) = \text{Prior prob}\left\{ y \leq y(i+1) \leq y + \delta y \right\}$$

$$\approx \int_{-\infty}^{\infty} PR1(x,y)\; \frac{PR2(x)}{\delta x}\; dx \text{ for small } \delta x, \delta y \qquad 6.2.3.6$$

Equating the two expressions for P gives

$$p_{i+1}\left[x \right] = \frac{P2(y-Ax^2) \int_{-\infty}^{\infty} P1(r)\; p_i\left[x - u(i+1) + u(i) - r \right] dr}{\int_{-\infty}^{\infty} P2(y-Ax^2) \int_{-\infty}^{\infty} P1(r) p_i\left[x - u(i+1) + u(i) - r \right] dr.dx} \qquad 6.2.3.7$$

as the updating equation for $p_i\left[X1 \right]$.

6.2.4 <u>Optimal control</u> Since no sufficient statistics have been found it is not possible to proceed to optimal control via dynamic programming. Further research is therefore required

in two possible directions. One possibility is to try to reformulate the state vector so that sufficient statistics can be found. The other possibility is to study the properties of equation 6.2.3.7 to see whether any definite characteristics can be found for the functions $p_i[X1]$ which might suggest approximate sufficient statistics and suboptimal control.

6.3 Optimal control of a multi-input extremal process

6.3.1 The mathematical model The simplest case of a multi-input extremal process is the two-dimensional case where the non-linearity is an elliptic paraboloid with each input affecting movement along one of the principal axes. This may be represented as shown in figure 6.3.1.1. Equations defining the process are

$$z1(i+1) = z1(i) + r1(i+1) \quad , i = 0,1, \ldots \qquad 6.3.1.1$$

$$z2(i+1) = z2(i) + r2(i+1) \quad , i = 0,1, \ldots \qquad 6.3.1.2$$

with $z1(0) = z2(0) = 0$, and $r1(i)$, $r2(i)$, $i = 1,2, \ldots$ are independent sequences of independent samples from Gaussian probability distributions with standard deviations σ_1 and σ_2 respectively.

$$x1(i) = u1(i) + z1(i) \quad , i = 0,1, \ldots \qquad 6.3.1.3$$

$$x2(i) = u2(i) + z2(i) \quad , i = 0,1, \ldots \qquad 6.3.1.4$$

where $u1(i)$ and $u2(i)$, $i = 1,2, \ldots$ are the sequences of control inputs, and $u1(0) = u2(0) = 0$.

$$y(i) = c(i) = A1.x1^2(i) + A2.x2^2(i) \quad , i = 1,2, \ldots \qquad 6.3.1.5$$

where $y(i)$, $i = 1,2, \ldots$, is the sequence of measurements,

c(i), i = 1,2, ... , is the sequence of stage costs and A1 and A2 are positive constants.

The optimal extremal controller must choose the sequences of control inputs on the basis of the measurements, so as to minimise

$$C_N = \text{Exp} \sum_{i=1}^{N} c(i) \qquad 6.3.1.6$$

at each stage, where N is the number of stages remaining.

6.3.2 State variables Combining equations 6.3.1.1 with 6.3.1.3 and 6.3.1.2 with 6.3.1.4 gives

$$x1(i+1) = x1(i) + u1(i+1) - u1(i) + r1(i+1) \qquad 6.3.2.1$$

$$x2(i+1) = x2(i) + u2(i+1) - u2(i) + r2(i+1) \qquad 6.3.2.2$$

Consider now the vector $\underline{X}$ defined by

$$\underline{X}(i) \equiv \begin{bmatrix} X1(i) \\ X2(i) \\ X3(i) \\ X4(i) \end{bmatrix} \equiv \begin{bmatrix} x1(i) \\ u1(i) \\ x2(i) \\ u2(i) \end{bmatrix} \qquad 6.3.2.3$$

Equations 6.3.1.5, 6.3.2.1 and 6.3.2.2 combined with equation 6.3.2.3 shows that $\underline{X}$ satisfies all the conditions set out in chapter 1 to be a state vector.

6.3.3 Sufficient statistics It is now necessary to consider the probability distribution for $\underline{X}$ at stage i, i = 1,2, ... when y(i) has been measured. X2(i) and X4(i) are evidently completely known and represented by the sufficient statistics u1(i) and u2(i) respectively.

Equation 6.3.1.5 shows that the measurement y(i) confines the possible values of X1(i) and X3(i) to points on the ellipse

$$A1.X1^2(i) + A2.X3^2(i) = y(i) \qquad 6.3.3.1$$

This is shown in figure 6.3.3.1. The joint probability distribution for X1 and X3 may thus be characterised by y(i) as a sufficient statistic together with $p_i(\Theta)$, the probability distribution for the angle Θ defined in figure 6.3.3.1. Updating equations could be found for the probability distributions $p_i(\Theta)$, using the laws of probability, but this has not been done.

6.3.4 Optimal control Since a complete set of sufficient statistics have not been found it is not possible to proceed to optimal control via dynamic programming. Again it might be possible to reformulate the state vector in a form which does yield sufficient statistics. Another more hopeful possibility depends on the fact that $p_i(\Theta)$ need be defined only in the range $0 \leq \Theta < 2\pi$ This means that a suboptimal controller of some value might be determined by approximating $p_i(\Theta)$ by the first few terms of a Fourier expansion.

Chapter 7

Conclusions

The point has now been reached where all the material known to be relevant to the subject of this thesis has been presented. This final chapter is intended to draw together the conclusions which have been reached, to comment on their significance, and to indicate how further study should proceed.

The most concise way to gather together all the conclusions which have been reached is to follow through the thesis and list the conclusions individually. The main conclusions therefore are as follows:-

(1) The arrival of direct digital control of complex industrial processes has given rise to situations where it may be possible to use extremal control to advantage. Because the costs involved are generally high, it may often be more important that the extremal controller should maintain as low an average system operating cost as possible, than that the controller should have a simple structure. There is therefore a requirement to establish optimal or nearly optimal extremal controllers.

(2) There is a problem in the design of simple extremal controllers, knowing when further changes are unlikely to produce significant improvements. There is therefore a need to establish the performance of optimal extremal controllers as a basis on which to judge the simple controllers.

(3) A restricted set of extremal processes may be represented by the simplified mathematical model derived in section 2.1.

(4) This model does not deal with multi-input processes, or

processes involving lags and/or noise. Also it does not represent random changes to the extremal function in the direction of the ordinate. Nevertheless it does represent the basic features of an extremal process in a way which may often be quite realistic, and therefore provides a suitable starting point for an investigation of the application of optimal control theory to extremal control systems.

(5) Derivation of a state vector for the simplified extremal process has been put on a rigorous mathematical basis.

(6) Sufficient statistics for the a posteriori probability distribution for the state vector together with the relevant updating equations have been established.

(7) A new approach has been taken to the application of dynamic programming to the control of the simplified extremal process. This has led to a new functional recurrence equation for the optimal control algorithm and cost functions ($f_N[w,q]$, N=1,2, ...).

(8) Analysis of this equation, without actually solving it, has shown that the optimal extremal controller, in this case, includes an integrator as the final part of the controller.

(9) Further analysis of the equation has established a numerical solution procedure including many checks. This solution procedure has been used to determine the optimal extremal controller and its performance.

(10) The optimal extremal controller has been shown to include discontinuities which have been explained in section 3.3.1.

(11) Simulations of the optimal extremal controller controlling the simplified discrete-time extremal process for which it is designed have confirmed that it performs as predicted.

(12) Simulation of the optimal extremal controller controlling a continuous-time extremal process, equivalent to the process for which the controller is designed, has shown that practical implementation of the optimal extremal controller in a direct digital control situation is possible without significant degradation of performance. The optimal extremal controller could be implemented on most small on-line computers.

(13) Comparison of the performance of the optimal extremal controller with the performance of simpler (heuristic) extremal controllers has shown that there is still room for considerable improvement on even the best of the simpler controllers.

(14) Simulations of the optimal extremal controller controlling, not the process for which it is designed, but a similar process in which output noise is considered, have shown that when the standard deviation of the output noise becomes similar to the mean value of the measurement, the performance of the optimal controller rapidly becomes worse than the simpler controllers.

(15) Simulations have established the performance of a suboptimal extremal controller when controlling the simplified extremal process, extended to include output noise.

(16) Comparison of the performances of the optimal and suboptimal extremal controllers has shown that when the output noise is small they are indistinguishable, but when the output noise becomes more significant, the optimal controller is rather better. This effect has been explained.

(17) Mathematical models have been set up for extremal processes including output noise, output lag and double input, and initial investigations of the application of optimal control theory in these

cases has been made. These investigations have shown some areas where difficulties are likely to arise - in the formulation of sufficient statistics, and in the solution of functional recurrence equations.

The next question which arises is how far these conclusions are new. The first two conclusions are really the basic motivation for studying the application of optimal control theory to extremal control. While such study is not original[27-32], little progress has really been made hitherto in this field.

The various mathematical models used throughout the thesis are not original, but have been used for some considerable time in the development of simple extremal controllers. Nevertheless no previous rigorous derivation of state vectors for these models is known.

Apart from the introduction of u as a third sufficient statistic to maintain a rigorous approach, the derivation of sufficient statistics and updating equations for the simplified extremal process follows the development of the suboptimal controller by Jacobs[29]. Sufficient statistics and updating equations determined for more complex extremal processes in chapter 6 are original.

Originality is certainly claimed for the approach taken to the application of dynamic programming to the simplified extremal process. Thus the functional recurrence equation derived is entirely new, and all the analysis leading to its solution is original.

The result of this work is the first genuine optimal extremal controller, and thus all the work involved in testing it

by simulation techniques and comparing it with other controllers is also original. This includes original work in establishing the performance of the suboptimal controller when controlling the simplified extremal process extended to include output noise.

Finally the results derived in investigating the application of optimal control theory to more complex extremal processes are also new.

The significance of these results can be split in two directions. As has been stated previously, the major motivation for this investigation is summarised in the first two conclusions. The first significance of these results therefore lies in the extent to which these two aims are satisfied. The other significance lies in the pointers the results give to future investigations.

First then in situations such as the control of complex industrial processes where optimal or closely optimal performance of an extremal controller is often highly desirable, the following achievements can be reasonably claimed. Provided that the extremal process can be represented by the basic elements included in the simplified mathematical model, then for the first time the optimal control algorithm is known, and moreover it can be confidently expected to overcome interfacing effects, and provide a known performance, at least as good as any other controller. Alternatively the suboptimal control algorithm of Jacobs[29] can be used in place of the optimal control algorithm, and again can be confidently expected to give a performance indistinguishable from optimal. Either of these algorithms could be implemented on most small on-line computers.

Turning to the design of simpler extremal controllers, it can

now be claimed that for the first time there is a basis established on which to judge their performance. This basis is the performance of the optimal extremal controller. Although the performance of the suboptimal controller of Jacobs[29] has shown that there is still room for considerable improvement in the design of simpler extremal controllers, in the simplified case, there is now a definite quantitative measure of how much improvement might be made.

Although these achievements are considered to constitute a significant advance in the application of optimal control theory to extremal control systems, there is still a very considerable amount still to be done. The other significance of the results of this thesis lies in the guide they give to future study. Optimal control of the most basic extremal process has been established, but many processes are more complex. The same requirements therefore exist to determine optimal or closely optimal extremal controllers for these more complex processes. Important complexities which are likely to arise are

(1) Input and/or output noise

(2) Input and/or output lags

(3) Multiple inputs

(4) Random changes to the extremal function in the direction of the ordinate.

One of the first decisions to be made is a priority list determining which complexities should be considered first. This must be based jointly on practical requirements, and the particular difficulties found in each case.

The results of chapter 6 have shown two major theoretical difficulties which are likely to arise - the difficulty of determining sufficient statistics, and the difficulty of solving the functional recurrence equations. At present it would appear that these difficulties may prevent any further optimal extremal controllers for more complex processes being determined. If this is the case then attention must be turned to closely optimal - suboptimal - controllers giving performance close to optimal. The value of such suboptimal controllers will vary depending on how confidently their performance is thought to be close to optimal.

The simplest approach which might be taken would be just to use the optimal controller for the simplified process in the more complex situations. However simulations in chapter 5 where output noise is taken into account have shown that at least in this case, this is not a good approach.

When optimal control theory is applied to more complex processes, the first difficulty which is likely to arise is if sufficient statistics cannot be formed. In this case it might be possible to approximate the a posteriori probability distribution for the state vector by approximate sufficient statistics, as done by Florentin[27]. An example of this situation is given in section 6.3. In this case the first few terms of a Fourier expansion of $p_i(\ )$ might be satisfactory approximate sufficient statistics on which to base a suboptimal controller.

If sufficient statistics can be found, then it may still prove impossible to solve the functional recurrence equation. A suboptimal controller can therefore be based on the true sufficient statistics as done by Jacobs for the simplified case[29].

The success of Jacobs' suboptimal controller may be taken as encouragement for future approaches along this same line.

Before solving the functional recurrence equation in this thesis a number of properties were established analytically, including in particular the fact that the final part of the optimal controller is an integrator. It may be possible to determine similar properties in the more complex cases without actually solving the functional recurrence equations. If such properties are incorporated into the suboptimal controllers, this should enhance their chance of giving performances close to optimal.

To sum up there are good reasons for studying the application of optimal control theory to extremal control systems. This thesis has contributed a considerable advance to the class of extremal control systems which can be described by the mathematical models used. However there is still much research to be done, and this can be expected to continue to provide problems of considerable academic interest, with a good chance of practical application.

Appendix 1

Proof of the Properties of the Iterates of the Functional Recurrence Equation

A1.1 Symmetry about q = 0.5

Consider the function $\bar{I}_N[w,q,v]$ defined by

$$\bar{I}_N[w,q,v] \equiv \int_{-\infty}^{\infty} \left\{ q \cdot f_{N-1}[w^+, q^+] + \overline{1-q} \cdot f_{N-1}[w^-, q^-] \right\} p[r]\, dr \qquad \text{A1.1.1}$$

$$N = 2,3, \ldots$$

Then

$$\bar{I}_N[w, \overline{1-q}, -v] = \int_{-\infty}^{\infty} \left\{ \overline{1-q} \cdot f_{N-1}[w_A^+, q_A^+] + q \cdot f_{N-1}[w_A^-, q_A^-] \right\} p[r]\, dr \qquad \text{A1.1.2}$$

$$N = 2,3, \ldots$$

where

$$\left.\begin{aligned} w_A^+ &\equiv | \; w - v + r \; | \\ w_A^- &\equiv | \; - w - v + r \; | \end{aligned}\right\} \qquad \text{A1.1.3}$$

$$q_A^{\pm} \equiv \frac{\overline{1-q} \cdot P1_A^{\pm} + q \cdot P2_A^{\pm}}{\overline{1-q} \cdot (P1_A^{\pm} + P3_A^{\pm}) + q \cdot (P2_A^{\pm} + P4_A^{\pm})} \qquad \text{A1.1.4}$$

and

$$\left.\begin{aligned} P1_A^{\pm} &\equiv \exp\left\{ -\left[w_A^{\pm} - w + v \right]^2/2 \right\} \\ P2_A^{\pm} &\equiv \exp\left\{ -\left[w_A^{\pm} + w + v \right]^2/2 \right\} \\ P3_A^{\pm} &\equiv \exp\left\{ -\left[w_A^{\pm} + w - v \right]^2/2 \right\} \\ P4_A^{\pm} &\equiv \exp\left\{ -\left[w_A^{\pm} - w - v \right]^2/2 \right\} \end{aligned}\right\} \qquad \text{A1.1.5}$$

Since the integration in equation A1.1.2 is infinite with respect to r in both directions, the sign for r in the integrand can be changed without affecting the result (i.e. the direction of the integration can be reversed),

Thus

$$\bar{I}_N\left[w, \overline{1-q}, -v\right] = \int_{-\infty}^{\infty}\left\{\overline{1-q} \cdot f_{N-1}\left[w_B^+, q_B^+\right] + q \cdot f_{N-1}\left[w_B^-, q_B^-\right]\right\} p\left[-r\right] dr \quad N = 2,3, \ldots$$

$$= \int_{-\infty}^{\infty}\left\{\overline{1-q} \cdot f_{N-1}\left[w_B^+, q_B^+\right] + q \cdot f_{N-1}\left[w_B^-, q_B^-\right]\right\} p\left[r\right] dr \qquad \text{A1.1.6}$$

$$N = 2,3, \ldots$$

where
$$\left.\begin{aligned} w_B^+ &\equiv |w - v - r| \\ w_B^- &\equiv |-w - v - r| \end{aligned}\right\} \qquad \text{A1.1.7}$$

and
$$q_B^{\pm} \equiv \frac{\overline{1-q} \cdot P1_B^{\pm} + q \cdot P2_B^{\pm}}{\overline{1-q} \cdot (P1_B^{\pm} + P3_B^{\pm}) + q \cdot (P2_B^{\pm} + P4_B^{\pm})} \qquad \text{A1.1.8}$$

and
$$\left.\begin{aligned} P1_B^{\pm} &\equiv \exp\left\{-\left[w_B^{\pm} - w + v\right]^2/2\right\} \\ P2_B^{\pm} &\equiv \exp\left\{-\left[w_B^{\pm} + w + v\right]^2/2\right\} \\ P3_B^{\pm} &\equiv \exp\left\{-\left[w_B^{\pm} + w - v\right]^2/2\right\} \\ P4_B^{\pm} &\equiv \exp\left\{-\left[w_B^{\pm} - w - v\right]^2/2\right\} \end{aligned}\right\} \qquad \text{A1.1.9}$$

Comparison of equations 3.1.29 - 3.1.31 with equations A1.1.7 - A1.1.9 now shows that

$$\left.\begin{aligned} w_B^+ &= w^- \\ w_B^- &= w^+ \\ q_B^+ &= 1 - q^- \\ q_B^- &= 1 - q^+ \end{aligned}\right\} \qquad \text{A1.1.10}$$

Substituting this in equation A1.1.6 gives

$$\bar{I}_N\left[w, \overline{1-q}, -v\right] = \int_{-\infty}^{\infty} \left\{ q\, f_{N-1}\left[w^+, \overline{1-q}^+\right] + \overline{1-q}\, .\, f_{N-1}\left[w^-, \overline{1-q}^-\right] \right\} p[r]\ dr \qquad \text{A1.1.11}$$

$$N = 2,3, \ldots$$

Comparison of equations A1.1.1 and A1.1.11 now shows that if for some N,

$$f_{N-1}\left[w,q\right] \equiv f_{N-1}\left[w, \overline{1-q}\right] \qquad \text{A1.1.12}$$

then $$\bar{I}_N\left[w,q,v\right] \equiv \bar{I}_N\left[w, \overline{1-q}, -v\right] \qquad \text{A1.1.13}$$

Combining equations 3.1.25, 3.1.26 and A1.1.1 shows that

$$f_N\left[w,q\right] \equiv w^2 + \operatorname*{Min}_v \left\{ \bar{I}_N\left[w,q,v\right] - \bar{I}_N\left[0,0,v\right] \right\} \qquad \text{A1.1.14}$$

$$N = 2,3, \ldots$$

and $$f_N\left[w, \overline{1-q}\right] \equiv w^2 + \operatorname*{Min}_v \left\{ \bar{I}_N\left[w, \overline{1-q}, v\right] - \bar{I}_N\left[0,0,v\right] \right\} \qquad \text{A1.1.15}$$

$$N = 2,3, \ldots$$

Comparison of equations A1.1.14 and A1.1.15 in the light of the property of equation A1.1.13 then shows that if for some N, equation A1.1.12 holds, then

$$f_N\left[w,q\right] \equiv f_N\left[w, \overline{1-q}\right] \qquad \text{A1.1.16}$$

$$\text{and } v_N\left[w,q\right] \equiv - v_N\left[w, \overline{1-q}\right] \qquad \text{A1.1.17}$$

The proof of these properties is completed by induction for all N, N = 1,2, ... , by noting from equation 3.1.27 that

$$f_1\left[w,q\right] \equiv f_1\left[w, \overline{1-q}\right] \qquad \text{A1.1.18}$$

A1.2 The special case w = 0

Putting w = 0 in equations 3.1.29 - 3.1.31 gives

$$w^+ = w^- = |v + r| \qquad \text{A1.2.1}$$

$$q^+ = q^- = \frac{P1}{P1 + P3} \qquad \text{A1.2.2}$$

where
$$\left.\begin{aligned} P1 &= \exp\left\{-\left[w^+ - v\right]^2/2\right\} \\ P3 &= \exp\left\{-\left[w^+ + v\right]^2/2\right\} \end{aligned}\right\} \qquad \text{A1.2.3}$$

Substituting these results in equation 3.1.25 then shows that

$$f_N\left[0,q\right] \equiv \overline{N-1}\; c^*_{N-1} - Nc^*_N + \min_v \int_{-\infty}^{\infty} f_{N-1}\left[w^+, q^+\right] p\left[r\right]\, dr \qquad \text{A1.2.4}$$

$$N = 2,3, \ldots$$

But equations A1.2.1 - A1.2.3 show that w^+ and q^+ are independent of q, and hence

$$f_N\left[0,q\right] = \text{constant}, \quad N = 2,3, \ldots \qquad \text{A1.2.5}$$

Equation 3.1.26 thus shows that

$$f_N\left[0,q\right] \equiv 0, \quad N = 2,3, \ldots \qquad \text{A1.2.6}$$

The minimising value of v in equation A1.2.4 is also independent of q, so that

$$v_N\left[0,q\right] = \text{constant}, \quad N = 2,3, \ldots \qquad \text{A1.2.7}$$

The symmetry property of appendix A1.1 shows that

$v_N[0,q]$ will have at least two alternative values,

$$v_N[0,q] = \pm v_N \ , \ N = 2,3, \ \ldots \qquad \text{A1.2.8}$$

Inspection of equations 3.1.27 and 3.1.33 shows that these properties hold also for the case N = 1 where

$$f_1[0,q] \equiv 0 \qquad \text{A1.2.9}$$

$$\text{and} \quad v_1[0,q] \equiv v_1 \equiv 0 \qquad \text{A1.2.10}$$

A1.3 <u>The special cases q = 0, q = 1</u>

Putting q = 0 in equation 3.1.25 gives

$$f_N[w,0] \equiv w^2 + (\overline{N-1}\ c^*_{N-1} - Nc^*_N) + \underset{v}{\text{Min}} \int_{-\infty}^{\infty} f_{N-1}[w^-, q^-]\ p[r]\ dr \qquad \text{A1.3.1}$$

$$N = 2,3, \ \ldots$$

$$\text{where } w^- = |-w + v + r| \qquad \text{A1.3.2}$$

$$\text{and } q^- = \frac{P2}{P2 + P4} \qquad \text{A1.3.3}$$

with P2 and P4 defined in equation 3.1.31.

Examination of equations A1.3.1 - A1.3.3 and 3.1.31 shows that the integral in equation A1.3.1 is a function of (- w + v) rather than w and v independently. Hence it has a constant minimum value with respect to v. Combining this result and equation 3.1.26 with equation A1.3.1 then gives

$$f_N[w,0] = w^2 \ , \ N = 2,3, \ \ldots \qquad \text{A1.3.4}$$

The minimising value of v, $v_N[w,0]$ must be of the form

$$-w + v_N[w,0] = \text{constant}, \ N = 2,3, \ \ldots \qquad \text{A1.3.5}$$

and to be consistent with the result in appendix A1.2

$$v_N\left[w,0\right] = w \pm v_N \ , \ N = 2,3, \ \ldots \qquad \text{A1.3.6}$$

Examination of equations 3.1.27 and 3.1.33 confirms that the results are true also for N = 1.

The symmetry properties of appendix A1.1 show that

$$f_N\left[w,1\right] \equiv w^2 \ , \ N = 1,2, \ \ldots \qquad \text{A1.3.7}$$

$$\text{and} \quad v_N\left[w,1\right] \equiv - w \mp v_N \ , \ N = 1,2, \ \ldots \qquad \text{A1.3.8}$$

A1.4 The special case of large w

Consider the control algorithm

$$v_N\left[w,q\right] = \overline{1-2q}\ w \ , \ N = 1,2, \ \ldots \qquad \text{A1.4.1}$$

It has been shown in section 3.1 that this algorithm is optimal when N = 1 i.e. it minimises the expected cost of the stage of the process at which it is used.

Substitution of $v = \overline{1-2q}\ w$ in equations 3.1.29 - 3.1.31 shows that for all significant values of r and for large w, $q \neq 0,\ 0.5,\ 1,$

$$P2^+,\ P3^+,\ P4^+,\ P1^-,\ P2^-\ \text{and}\ P3^- \rightarrow 0 \qquad \text{A1.4.2}$$

where the superscripts $\pm$ denotes that $w^{\mp}$ should be substituted for W. Thus

$$\left.\begin{aligned} q^+ &\rightarrow 1 \\ q^- &\rightarrow 0 \end{aligned}\right\} \qquad \text{A1.4.3}$$

So the control algorithm of equation A1.4.1 has the ideal

effect of perfect control and perfect determination of the state vector. Hence it is optimal.

The asymptotic expression for $f_N[w,q]$, N = 1,2, ... , can be found by noting that the total expected cost when there are N- stages to go can be written as

$F_N[u,w,q]$ = Expected cost of first stage +

Expected cost of final $\overline{N-1}$ stages using A1.4.4

the optimal control algorithm throughout.

Since the asymptotic algorithm is optimal for a 1-stage process, the expected cost of the first stage is $F_1[u,w,q]$ so that

$$F_N[u,w,q] = F_1[u,w,q] + \int_{-\infty}^{\infty} \left\{ q \cdot F_{N-1}[u+v,\ w^+,\ q^+] + \overline{1-q} \cdot F_{N-1}[u+v,\ w^-,\ q^-] \right\} p[r]\ dr \qquad A1.4.5$$

$$N = 2,3, \ldots$$

Substituting from equation 3.1.24 gives

$$f_N[w,q] - w^2 + Nc_N^* = f_1[w,q] - w^2 + c_1^* + \int_{-\infty}^{\infty} \left\{ q \left[f_{N-1}[w^+,\ q^+] - (w^+)^2 - \overline{N-1}\ c_{N-1}^* \right] + \overline{1-q} \cdot \left[f_{N-1}[w^-,\ q^-] - (w^-)^2 - \overline{N-1}\ c_{N-1}^* \right] \right\} p[r]\ dr \qquad A1.4.6$$

$$N = 2,3, \ldots$$

But for large w, $q \neq 0, 0.5, 1$,

$$\left.\begin{aligned} q^+ &\to 1,\ f_{N-1}[w^+,\ q^+] \to f_{N-1}[w^+,\ 1] = (w^+)^2 \\ q^- &\to 0,\ f_{N-1}[w^-,\ q^-] \to f_{N-1}[w^-,\ 0] = (w^-)^2 \end{aligned}\right\} \qquad A1.4.7$$

Substituting from equations 3.1.27, 3.1.28 and A1.4.7 in A1.4.6 gives for large w, q ≠ 0, 0.5, 1

$$f_N[w,q] \rightarrow w^2\left[1 + 4q \cdot \overline{1-q}\right] + 1 + \overline{N-1}\, c^*_{N-1} - Nc^*_N \qquad \text{A1.4.8}$$

$$\approx w^2\left[1 + 4q \cdot \overline{1-q}\right] \quad , N = 2,3, \ldots \qquad \text{A1.4.9}$$

since the last three terms in equation A1.4.8 are collectively negligible for large w.

Equation 3.1.27 shows that the result holds exactly for N = 1.

Appendix 2

The Description of Digital Computer Programs

Appendices 3 and 4 contain descriptions of the digital computer programs used in determining the results presented in this thesis. Each description is broken down into the following headings:-

1. NAME This is the name by which the program is known. After the program name all the dummy names which must be replaced before the program is executed are listed in order. Dummy names are broken down into three groups, separated by semi-colons. These groups are (1) retained data, (2) preset variables and (3) postset variables, and are described in more detail later.

2. FUNCTION This is a brief general description of what the program is intended to do.

3. SUBPROGRAMS USED Some programs include within them the execution of other programs which are then known as subprograms. All subprograms executed within the program being described are listed under this heading.

4. DATA Before a program (or subprogram) is executed it is essential to ensure that all input data used is present and in the correct order. To this end, names given to the input data by the program are listed in order under this heading. This includes any data required for subprograms used within the program being described. Data input to a subprogram may be retained for use after the subprogram has been executed. Before such a subprogram is executed names must be given to all such

retained data. These names replace the dummy names used in their place in the program description. The dummy data names are underlined in the subprogram descriptions. Details of how to specify data names before execution of a subprogram are given later in this section.

5. OUTPUT Some programs output results either in typewritten form, or to some form of data store (e.g. punched cards or magnetic disc), from which it can be retrieved as data for a subsequent program. Output from the program being described, including any subprograms within it, is listed under this heading in the order in which it appears.

6. PRESET VARIABLES It is often necessary for certain variables to be assigned values before a subprogram is executed. Such variables are given dummy names in the subprogram description, and must be replaced as described later before the subprogram is executed. The value of these preset variables is never altered during the execution of a subprogram.

7. DUMMY VARIABLES It is often convenient to have names set aside for variables to work with purely within a program. Such dummy variable names are listed in this section.

8. POSTSET VARIABLES The end result of many subprograms is to determine values for certain variables. Such variables are given dummy names in the subprogram description, and must be replaced as described later before the subprogram is executed.

9. COMMENTS Any comments relevant to the program are included under this heading.

10. FLOW DIAGRAM This diagram shows the logical flow of the

program when it is being executed. Reference to the comments section is made as necessary.

The manner in which retained data, preset variables and postset variables are specified before a subprogram is executed is best illustrated by a fictitious example. Consider the execution within a program ALPHA of a subprogram BETA which has

DATA a

b

c

PRESET VARIABLES d

e

POSTSET VARIABLES f

g

Then the execution of BETA, specified as

'Execute BETA [U,V; W,X; Y,Z]'

would cause BETA to be executed with

(1) U and V as the names of the variables in ALPHA which have set into them by BETA the data named b and c respectively in BETA.

(2) W and X as the names of the variables in ALPHA to which values have been assigned before the execution of BETA such that the variables named d and e in BETA should have the values W and X respectively.

(3) Y and Z as the names of the variables in ALPHA which have set into them by BETA the final value of the postset variables named f and g respectively.

Variables are of four types - real numbers, integer numbers, real arrays and integer arrays. The specification of an array

is best described by an example. A(m,n) specifices a 2-dimensional array named A. The elements of the array in the order implied by the specification are

A(1,1) , A(1,2) , ... , A(1,n) ,
A(2,1) , A(2,2) , ... , A(2,n),

A(m,1) , A(m,2) , ... , A(m,n) .

Thus if A(2,3) were specified in data, then the six data values would be placed in order in A(1,1) , A(1,2) , A(1,3) , A(2,1) , A(2,2) and A(2,3) respectively.

Appendix 3

Digital Computer Programs Used in Solving the Functional Recurrence Equation

A3.1 NAME APPROXPARASET [dr, IDRDW, IDRDV, IDWWL, IDQQM, IDRRL ; - ; dw, dv, W, dq , R, IDRWL, IDQQL]

FUNCTION The function of this subprogram is to set all the approximation parameters, and related parameters necessary for the iterations.

PROGRAMS USED None

DATA dr approximation parameter.

IDRDW the ratio dw/dr, where dw is an approximation parameter.

IDRDV the ratio dv/dr, where dv is an approximation parameter.

IDWWL (1 + W/dw), where W is an approximation parameter.

IDQQM (0.5/dq), where dq is an approximation parameter.

IDRRL (1 +2.R/dr), where R is an approximation parameter.

OUTPUT None

PRESET VARIABLES None

DUMMY VARIABLES None

POSTSET VARIABLES dw approximation parameter

dv approximation parameter

W approximation parameter

dq approximation parameter

R approximation parameter

IDRWL (1 + W/dr)

IDQQL (1 + 1/dq)

COMMENTS When carrying out an iteration to determine $I_N[w,q]$ from $f_{N-1}[w,q]$, N = 2,3, ... , it is necessary to evaluate $f_{N-1}[w,q]$ for w equal to various values of w^+ and w^-. Definitions of w^+ and w^- show that provided w, r and v are always integral multiples of dr, it is only necessary to evaluate $f_{N-1}[w,q]$ for values of w which are integral multiples of dr. For this reason dr is provided as direct data, but dw and dv are calculated indirectly from the ratios dw/dr and dv/dr which are provided as the integer data IDRDW and IDRDV respectively.

It is inherent in the numerical method employed that W should be an integral multiple of dw, 0.5 should be an integral multiple of dq and R should be an integral multiple of dr. For this reason W and dq are calculated indirectly from the integer data IDWWL and IDQQM which are respectively (1 + W/dw) and (0.5/dq). These integers are chosen to be the dimensions of the two dimensional arrays necessary to hold the values of $I_N[w,q]$ and $v_N[w,q]$ which are calculated during an iteration. R is calculated from the integer data IDRRL which is (1 + 2R/dr), the total number of elements in each numerical integration.

Before carrying out an iteration it is useful to calculate and store values of the function $f_{N-1}[w,q]$ at intervals O(dr)W in the w direction, and O(dq)1 in the q direction. The dimensions of the array necessary to hold this information are (1 +W/dr) x (1 + 1/dq). These dimensions are calculated as the integer parameters IDRWL and IDQQL respectively.

FLOW DIAGRAM Figure A3.1.1

A3.2 NAME GRIDSET [MINO(IDWWL, IDQQM); IDWWL, IDQQM; -]

FUNCTION Before an iteration to calculate $I_N[w,q]$ and $v_N[w,q]$, N = 3,4, ... , this subprogram is used twice, first to store values of $I_{N-1}[w,q]$, and then to store starting values for

v at all the grid points at which $I_N[w,q]$ and $v_N[w,q]$ are to be determined.

PROGRAMS USED None

DATA MINO(IDWWL, IDQQM), the array in which the data is stored

OUTPUT None

PRESET VARIABLES IDWWL parameter set by APPROXPARASET

IDQQM parameter set by APPROXPARASET

DUMMY VARIABLES i,j, integers used for counting

POSTSET VARIABLES None

COMMENTS At the end of the iteration to calculate $I_N[w,q]$ and $v_N[w,q]$, N = 2,3, ... , values of $I_N[w,q]$ and $v_N[w,q]$, calculated at the grid points defined in section 3.3.1 are stored in some form of data store (magnetic tape, cards or disk). Before the next iteration is carried out the values of $v_N[w,q]$ at the grid points are modified to provide suitable starting values for v for the next iteration. When the next iteration, to calculate $I_{N+1}[w,q]$ and $v_{N+1}[w,q]$, N = 2,3, ... , at the grid points is started, one of the first things to be done is to take in the values of $I_N[w,q]$ and the modified values of $v_N[w,q]$ at the grid points, from the store, and hold these in arrays. GRIDSET is the subprogram to do this and MINO is the dummy name for the array in which the data is stored. This name is replaced by IMINO when values of $I_N[w,q]$ are being taken in, and VMINO when modified values of $v_N[w,q]$ are being taken in.

FLOW DIAGRAM Figure A3.2.1.

A3.3 NAME FOSET [-; dr, IDRDW, IDWWL, IDQQM, IDRWL, IDQQL, IMINO(IDWWL, IDQQM); F1(IDRWL, IDQQL)]

FUNCTION The function of this subprogram is to calculate values

of $f_{N-1}[w,q]$ from the values of $I_{N-1}[w,q]$ set into the array IMINO(IDWWL, IDQQM) by GRIDSET and store these values in an array F1(IDRWL, IDQQL). The grid of points at which values of $f_{N-1}[w,q]$ are calculated and stored is an extended version of the grid of points at which values of $I_N[w,q]$ and $v_N[w,q]$, N = 2,3, ... , are calculated.

SUBPROGRAMS USED None

DATA None

OUTPUT None

PRESET VARIABLES
- dr approximation parameter
- IDRDW parameter set by APPROXPARASET
- IDWWL parameter set by APPROXPARASET
- IDQQM parameter set by APPROXPARASET
- IDRWL parameter set by APPROXPARASET
- IDQQL parameter set by APPROXPARASET
- IMINO(IDWWL, IDQQM) array set by GRIDSET

DUMMY VARIABLES
- i,j integers used for counting
- SMINO(IDWWL, 1+IDQQM) array used as an intermediate between IMINO and F1.
- FL variable used in interpolation
- IF integer used in interpolation
- FR variable used in interpolation.

POSTSET VARIABLES
- F1(IDRWL, IDQQL) array used to hold values of $f_{N-1}[w,q]$ calculated by FOSET.

COMMENTS The subprogram FOSET is one of the subprograms used in each iteration to determine $f_N[w,q]$ from $f_{N-1}[w,q]$, N = 3,4, ... Figure A3.3.1(a) shows the points in the (w,q) plane at which values of $I_{N-1}[w,q]$ must be held in the array IMINO(IDWWL, IDQQM) before FOSET is executed. The first thing that FOSET does is to

extend this grid of points to include points along the w- axis, by making use of the known properties of the iterates of the functional recurrence equation. The values of $I_{N-1}[w,q]$ at the points on this extended grid, shown in figure A3.3.1(b) are held in the array SMINO(IDWWL, 1+IDQQM).

FigureA 3.3.1(c) shows the grid of points at which values of $f_{N-1}[w,q]$ are calculated and stored by FOSET, in the array F1(IDRWL, IDQQL). This is done by linear interpolation in the w direction only, for points below and on the q = 0.5 line, and by symmetry for points above this line. Values of $f_{N-1}[w,q]$ are calculated and stored for values of w spaced dr apart in this way, because all iterations are so arranged that values of $f_{N-1}[w,q]$ are required only for these values of w. Thus no further interpolation in the w direction is necessary during iterations. As values of $f_{N-1}[w,q]$ may be required for any value of q in the range $0 \leq q \leq 1$, interpolation in the q direction must be done as required during each iteration. Thus no interpolation is made at this stage in the q direction.

The significance of the dummy variables FL, IF and FR is shown in figure A3.3.1(c). The meaning of the term 'IFIX ( )' used in figure A3.3.2 is that the integer immediately below the number enclosed in the brackets should be used.

FLOW DIAGRAM Figure A3.3.2

A3.4 NAME FO[-; dr, W, dq, IDQQL, w, q, F1(IDRWL, IDQQL); fo]

FUNCTION This subprogram is used within iterations which calculate $f_N[w,q]$ from $f_{N-1}[w,q]$, N = 2,3, ... The function of this subprogram is to evaluate $f_{N-1}[w,q]$ using the grid of points preset by FOSET together with the known asymptotic solution for large w.

PROGRAMS USED None

DATA None

OUTPUT None

PRESET VARIABLES

dr	approximation parameter
W	approximation parameter
dq	approximation parameter
IDQQL	parameter set by APPROXPARASET
w	argument of $f_{N-1}[w,q]$ to be evaluated
q	argument of $f_{N-1}[w,q]$ to be evaluated
F1(IDRWL, IDQQL)	array as set by FOSET

DUMMY VARIABLES

JW	first index of the F1 array elements used
JQ	used in interpolation in the q direction
FQ	used in interpolation in the q direction

POSTSET VARIABLES fo the value of $f_{N-1}[w,q]$ calculated by FO

COMMENTS The subprogram starts by determining whether $w > W$ or not. For $w > W$, fo is set to the value determined by the asymptotic expression given in section 3.2.5. For $w \leq W$, the value of fo is determined from the array F1(IDRWL, IDQQL). No interpolation is necessary in the w direction, as explained in the subprogram FOSET. Linear interpolation is used in the q direction. The meaning of the term 'IFIX( )' used in figure A3.4.1 is as explained in the subprogram FOSET.

FLOW DIAGRAM Figure A3.4.1

A3.5 NAME QU[-; WU, w, q, v; qu]

FUNCTION This subprogram evaluates the function Q[W,w,q,v] defined by equations 3.1.30 and 3.1.31.

SUBPROGRAMS USED None

DATA None

OUTPUT None

PRESET VARIABLES WU the argument W of the function Q

w the argument w of the function Q

q the argument q of the function Q

v the argument v of the function Q.

DUMMY VARIABLES P1, P2, P3, P4 used as in equation 3.1.31

POSTSET VARIABLES qu the value of the function calculated by QU.

COMMENTS The meaning of the term 'EXP ( )' used in figure A3.5.1 is that the exponential of the number enclosed in brackets should be used.

FLOW DIAGRAM Figure A3.5.1

A3.6 NAME SUM [-; dr, IDRRL, W, dq, R, IDRWL; IDQQL,
F1(IDRWL, IDQQL), w,q,v; sum]

FUNCTION This subprogram is used in iterations which calculate $f_N[w,q]$ from $f_{N-1}[w,q]$, N = 2,3, ... It evaluates $\bar{I}_N[w,q,v]$ for preset values of w,q and v, using the numerical methods described in chapter 3.

SUBPROGRAMS USED FO, QU

DATA None

OUTPUT None

PRESET VARIABLES dr approximation parameter

IDRRL parameter set by APPROXPARASET

W approximation parameter

dq approximation parameter

R approximation parameter

IDRWL parameter set by APPROXPARASET

IDQQL parameter set by APPROXPARASET

F1(IDRWL, IDQQL) array as set by FOSET

w argument of $\bar{I}_N[w,q,v]$

q argument of $\bar{I}_N[w,q,v]$

v argument of $\bar{I}_N[w,q,v]$

DUMMY VARIABLES

TOT	running total as integral is evaluated
r	the variable 'r' in chapter 3
JR	used for counting
WP	the variable 'w^+' in chapter 3
WM	the variable 'w^-' in chapter 3
QP	the variable 'q^+' in chapter 3
QM	the variable 'q^-' in chapter 3
fp	the value of $f_{N-1}[w^+, q^+]$
fm	the value of $f_{N-1}[w^-, q^-]$
δ	the value of the function $\delta[w,q,v,r]$ in chapter 3

POSTSET VARIABLES sum the value of $\bar{I}_N[w,q,v]$ calculated by SUM

COMMENTS This subprogram is based on equations 3.3.1.13, 3.3.1.5, 3.1.32 and 3.1.29. The meaning of the term 'ABS( )' used in figure A3.6.1 is that the absolute value of the number enclosed in brackets should be used.

FLOW DIAGRAM Figure A3.6.1

A3.7 NAME IMIN [-; dr, IDRRL, dv, W, dq, R, IDRWL, IDQQL, F1(IDRWL, IDQQL), w, q, v; imin, vmin]

FUNCTION This subprogram is used within iterations which calculate $f_N[w,q]$ from $f_{N-1}[w,q]$, N = 2,3, ... The function of the subprogram is to find the minimum of $\bar{I}_N[w,q,v]$ with respect to v for preset values of w and q and a preset starting value of v.

SUBPROGRAMS USED SUM, QU, FO

DATA None

OUTPUT None

PRESET VARIABLES

dr	approximation parameter
IDRRL	parameter set by APPROXPARASET

dv — approximation parameter

W — approximation parameter

dq — approximation parameter

R — approximation parameter

IDRWL — parameter set by APPROXPARASET

IDQQL — parameter set by APPROXPARASET

F1(IDRWL, IDQQL) — array as set by FOSET

w — argument of $I_N[w,q]$

q — argument of $I_N[w,q]$

v — starting value of V for stepping procedure

DUMMY VARIABLES — V used to represent v in $\bar{I}_N[w,q,v]$

SAR(3) — array used to store values of $\bar{I}_N[w,q,v]$.

POSTSET VARIABLES — imin — the value of $I_N[w,q]$

vmin — the minimising value of V, i.e. $v_N[w,q]$.

COMMENTS — The subprogram starts by setting the starting value of V to an integral multiple of dr so that the values of w^+ and w^- used in SUM are integral multiples of dr, and $f_{N-1}[w,q]$ need only be evaluated for values of w which are integral multiples of dr, as explained in FOSET. The subprogram then follows a stepping procedure with step size dv, so that eventually the required minimum is located with

$$SAR[1] = \bar{I}_N[w,q,V - dv] \geqslant SAR[2] = \bar{I}_N[w,q,V] \leqslant SAR[3] = \bar{I}_N[w,q,V + dv] \qquad A3.7.1$$

The subprogram ends by determining the minimum of $\bar{I}_N[w,q,v]$ as the minimum of the parabola passing through these three points.

The sympol '*' used in figure A3.7.1 represents 'dr, IDRRL, W, dq, R, IDRWL, IDQQL, F1(IDRWL, IDQQL), w, q'. The meaning of the term 'IFIX( )' used in figure A3.7.1 is as explained in the subprogram FOSET.

FLOW DIAGRAM — Figure A3.7.1

A3.8 NAME ITER 1[-; -; -]

FUNCTION The function of this program is to evaluate $\bar{I}_2[w,q,v]$ for a number of data points (w,q,v)

SUBPROGRAMS USED APPROXPARASET, FO, QU, SUM

DATA dr approximation parameter

IDRDW parameter used by APPROXPARASET

IDRDV parameter used by APPROXPARASET

IDWWL parameter used by APPROXPARASET

IDQQM parameter used by APPROXPARASET

IDRRL parameter used by APPROXPARASET

POINTS the number of (w,q,v) triplets at which $\bar{I}_2[w,q,v]$ is to be evaluated

w argument of $\bar{I}_2[w,q,v]$

q argument of $\bar{I}_2[w,q,v]$ } Repeated for each triplet

v argument of $\bar{I}_2[w,q,v]$

OUTPUT sum the value of $\bar{I}_2[w,q,v]$ Repeated for each triplet.

PRESET VARIABLES None

DUMMY VARIABLES i,j used for counting

F1(IDRWL, IDQQL) array set by FOSET for iterations after the first.

IDQQL parameter set by APPROXPARASET

IDRWL parameter set by APPROXPARASET

POSTSET VARIABLES None

COMMENTS The program ITER 1 sets the points in the array F1(IDRWL, IDQQL) to values corresponding to the known function $f_1[w,q]$. In this way when $f_1[w,q]$ is evaluated no interpolation is used in the w- direction. Interpolation is maintained in the q direction, but results in section 3.3.2 show that this is

insignificant. The symbol '*' is used to represent dr, IDRDW, IDRDV, IDWWL, IDQQM, IDRRL; - ; dw, dv, W, dq, R, IDRWL, IDQQL. The symbol '†' is used to represent dr, IDRRL, W, dq, R, IDRWL, IDQQL, F1(IDRWL, IDQQL), w,q,v.

FLOW DIAGRAM Figure A3.8.1

A3.9 NAME ITER 2 [-; -; -]

FUNCTION The function of this program is to evaluate $I_2[w,q]$ and $v_2[w,q]$ for a number of data points (w,q).

SUBPROGRAMS USED APPROXPARASET, FO, QU, SUM, IMIN

DATA dr approximation parameter

IDRDW parameter used by APPROXPARASET

IDRDV parameter used by APPROXPARASET

IDWWL parameter used by APPROXPARASET

IDQQM parameter used by APPROXPARASET

IDRRL parameter used by APPROXPARASET

POINTS the number of (w,q) points at which $I_2[w,q]$ and $v_2[w,q]$ are to be evaluated.

w argument of $I_2[w,q]$, $v_2[w,q]$

q argument of $I_2[w,q]$, $v_2[w,q]$

v starting value required by IMIN

} Repeated for each point

OUTPUT imin the value of $I_2[w,q]$

vmin the value of $v_2[w,q]$

} Repeated for each point

PRESET VARIABLES None

DUMMY VARIABLES i,j used for counting

F1(IDRWL, IDQQL) array set by FOSET for iterations after the first

IDQQL parameter set by APPROXPARASET

IDRWL parameter set by APPROXPARASET

POSTSET VARIABLES None

COMMENTS The setting of the array F1(IDRWL, IDQQL) in ITER 2 is as done in ITER 1, and so the same comments concerning interpolation apply. The symbol '*' represents the same group of names as in ITER 1. The symbol '+' is used to represent -; dr, IDRRL, dv, W, dq, R, IDRWL, IDQQL, F1(IDRWL, IDQQL)

FLOW DIAGRAM Figure A3.9.1

A3.10 NAME ITER 3[-; -; -]

FUNCTION The function of this program is to evaluate $I_2[w,q]$ and $v_2[w,q]$ at all the required grid points.

SUBPROGRAMS USED APPROXPARASET, GRIDSET, FO, QU, SUM, IMIN

DATA dr approximation parameter

IDRDW parameter used by APPROXPARASET

IDRDV parameter used by APPROXPARASET

IDWWL parameter used by APPROXPARASET

IDQQM parameter used by APPROXPARASET

IDRRL parameter used by APPROXPARASET

VMINO(IDWWL, IDQQM) an array containing starting values for v, for the search for the minimising value of v at each grid point.

OUTPUT IMIN 1(IDWWL, IDQQM) an array containing the value of $I_2[w,q]$ at each grid point.

VMIN 1(IDWWL, IDQQM) an array containing the value of $v_2[w,q]$ at each grid point.

PRESET VARIABLES None

DUMMY VARIABLES i,j used for counting

F1(IDRWL, IDQQL) array set by FOSET for iterations after the first

dq approximation parameter

IDRWL parameter set by APPROXPARASET

dw approximation parameter

IDQQL parameter set by APPROXPARASET

POSTSET VARIABLES None

COMMENTS The setting of the array F1(IDRWL, IDQQL) in ITER 3 is as done in ITER 1, and so the same comments concerning interpolation apply. The symbol '*' represents the same group of names as in ITER 1. The symbol '†' is used to represent -; dr, IDRRL, dv, W, dq, R, IDRWL, IDQQL, F1(IDRWL, IDQQL),

FLOW DIAGRAM Figure A3,10.1

A3.11 NAME ITER 4 [-; -; -]

FUNCTION The function of this subprogram is to evaluate $\overline{I}_N[w,q,v]$, N = 3,4, ... for a number of data points (w,q,v).

SUBPROGRAMS USED APPROXPARASET, GRIDSET, FOSET, FO, QU, SUM.

DATA dr approximation parameter

IDRDW parameter used by APPROXPARASET

IDRDV parameter used by APPROXPARASET

IDWWL parameter used by APPROXPARASET

IDQQM parameter used by APPROXPARASET

IDRRL parameter used by APPROXPARASET

IMINO(IDWWL, IDQQM) an array used by GRIDSET

POINTS the number of (w,q,v) triplets at which $\overline{I}_N[w,q,v]$ is to be evaluated

w argument of $\overline{I}_N[w,q,v]$

q argument of $\overline{I}_N[w,q,v]$ } Repeated for each triplet

v argument of $\overline{I}_N[w,q,v]$

OUTPUT sum the value of $\overline{I}_N[w,q,v]$. Repeated for each triplet

PRESET VARIABLES None

DUMMY VARIABLES i used for counting

POSTSET VARIABLES None

COMMENTS The symbols '*' and '†' used have the same meaning as in ITER 1. The symbol '‡' is used to represent -; dr, IDRDW, IDWWL, IDQQM, IDRWL, IDQQL, IMINO(IDWWL, IDQQM); F1(IDRWL, IDQQL)

FLOW DIAGRAM Figure A3.11.1

A3.12 NAME ITER 5[-; -; -]

FUNCTION The function of this subprogram is to evaluate $I_N[w,q]$, and $v_N[w,q]$, N = 3,4, ... , for a number of data points (w,q,v). Suitable choice of the data IMINO(IDWWL, IDQQM) also allows the program to be used to demonstrate the accuracy of the approximation parameters.

SUBPROGRAMS USED APPROXPARASET, GRIDSET, FOSET, FO, QU, SUM, IMIN.

DATA dr approximation parameter

IDRDW parameter used by APPROXPARASET

IDRDV parameter used by APPROXPARASET

IDWWL parameter used by APPROXPARASET

IDQQM parameter used by APPROXPARASET

IDRRL parameter used by APPROXPARASET

IMINO(IDWWL, IDQQM) an array used by GRIDSET

POINTS the number of (w,q) points at which $I_N[w,q]$ and $v_N[w,q]$ are to be calculated.

w argument of $I_N[w,q]$

q argument of $I_N[w,q]$

v starting value required by IMIN

} Repeated for each triplet.

OUTPUT imin the value of $I_N[w,q]$

vmin the value of $v_N[w,q]$

} Repeated for each triplet

PRESET VARIABLES None

DUMMY VARIABLES i used for counting

POSTSET VARIABLES None

COMMENTS The symbols '*' and '‡' have the same significance as in ITER 4. The symbol '†' has the same significance as in ITER 3.

FLOW DIAGRAM Figure A3.12.1

A3.13 NAME ITER 6[_; _; _]

FUNCTION The function of this program is to evaluate $I_N[w,q]$ and $v_N[w,q]$, N = 3,4, ... , at all the required grid points.

SUBPROGRAMS USED APPROXPARASET, GRIDSET, FOSET, FO, QU, SUM, IMIN

DATA dr approximation parameter

IDRDW parameter used by APPROXPARASET

IDRDV parameter used by APPROXPARASET

IDWWL parameter used by APPROXPARASET

IDQQM parameter used by APPROXPARASET

IDRRL parameter used by APPROXPARASET

IMINO(IDWWL,IDQQM) an array used by GRIDSET

VMINO(IDWWL,IDQQM) an array used by GRIDSET

OUTPUT IMIN1(IDWWL,IDQQM) an array containing the value of $I_N[w,q]$ at each grid point

VMIN1(IDWWL,IDQQM) an array containing the value of $v_N[w,q]$ at each grid point

PRESET VARIABLES None

DUMMY VARIABLES i, j used for counting

POSTSET VARIABLES None

COMMENTS The symbols '*', '‡' and '†' have the same significance as in ITER 5.

FLOW DIAGRAM Figure A3.13.1

Appendix 4

Details of Computer Simulations of Extremal Control Systems

This appendix contains details of the computer simulations of extremal control systems referred to in chapters 4 and 5. Descriptions of the purely discrete-time digital computer simulations follow the pattern already set in appendix 2. The descriptions of hybrid computations start with the name and function. There then follows a section on the analogue configuration and a section on the interface configuration used. The remaining sections concerned mainly with the digital computer program are then presented as before.

As most of the analogue and interface units used are common to most of the hybrid computations, it is convenient to present relevant information about them before proceeding to the descriptions.

Random signal generator The random signal generator used was a Servomex Controls Ltd. RG77, with nominal bandwidth $0 \rightarrow 10$Hz.

Analogue computer The analogue computer used was an Electronic Associates Ltd. TR48, operating in the range $\pm$ 10 Volts. The two modes of operation of this computer which are important to these descriptions are called RS and OP. These modes may be selected either manually, or via the mode control unit (see later). An important feature of the computer is that each integrator may be individually patched so that it resets (i.e. holds its output at the initial condition) and operates (i.e. integrates) according to any of the four possible relationships to the computer RS and OP modes. Functioning of each integrator with respect to the RS and OP modes is indicated in the relevant analogue configuration figure.

Digital computer The digital computer used was an IBM 1130.

Oxford University Systems Engineering Group Interface (OUSEGI)

This interface was used as the means of communication between the digital

computer and other interface units.

Interupt clock Interupts to the digital computer were generated by an interupt clock acting through OUSEGI. The clock could be switched on and off either manually or by program control. All computations used the clock set to 1Hz.

Mode control unit The function of the mode control unit is to allow the analogue computer to be switched between RS and OP modes under program control.

D/A convertor Transfer of signals from the digital computer to the analogue computer was accomplished via OUSEGI and an 8 bit D/A convertor. The relationship between the analogue signal A (Volts) and the digital integer D was

$$A = \frac{12.8}{255} \quad D \approx 0.05\ D\ ,\ 0 \leq D \leq 255 \qquad \text{A4.1}$$

A/D convertor Transfer of signals from the analogue computer to the digital computer was accomplished via OUSEGI and either an 8- bit or a 12 bit A/D comvertor. The relationships between the digital integer D, the offset B (digital units) and the analogues signal A (Volts) were

$$\text{8 bit:}\quad D = B + \frac{255}{12.8} A \approx B + 20A,\ 0 \leq B,D \leq 255 \qquad \text{A4.2}$$

$$\text{12 bit:}\quad D = B + \frac{4095}{10} A \approx B + 410A,\ 0 \leq B,D \leq 4095 \qquad \text{A4.3}$$

The digital integer D was converted from the analogue signal and stored immediately after each read. Thus two reads were necessary to obtain a value of the current analogue signal.

Scanner In order to be able to transmit more than one analogue signal to the digital computer a scanner was used between the A/D convertor and the analogue computer. This was arranged so that the signal set into the A/D store after the first read was the signal on channel 0 of the scanner. Subsequently the scanner cycled round each channel in turn up

to the highest number used and back to channel 0 again.

When comparing the effects of different control algorithms on the same process, it is convenient to be able to refer only to one description, and specify the required control algorithm by specifying a subprogram name. This has been done by allowing a subprogram name to be considered as a preset variable. Thus for example ALPHA[_; BETA; _] refers to the use of hybrid computation ALPHA using the control algorithm specified in subprogram BETA.

A4.1 NAME UOPT[_ ; VAR1(31,6), VAR2(21,6), u, w, q; u1, v]

FUNCTION The function of this subprogram is to calculate the optimal value of the control input to the simplified extremal process, from the sufficient statistics.

PROGRAMS USED None

DATA None

OUTPUT None

PRESET VARIABLES VAR1(31,6) contains values of v[w,q] in the range

$0 \leq w \leq 3$; $0 \leq q \leq 0.5$

VAR2(21,6) contains values of v[w,q] in the range

$0 \leq w \leq 10$; $0 \leq q \leq 0.5$

u sufficient statistic

w sufficient statistic

q sufficient statistic

DUMMY VARIABLES q' represents $\overline{1-q}$ for $q > 0.5$ and q for $q \leq 0.5$

IW integer used in interpolation

FW variable used in interpolation

IQ integer used in interpolation

FQ variable used in interpolation

POSTSET VARIABLES u1 the optimal control input

v the optimal control increment

COMMENTS Calculation of the optimal control input is split into three regions. In the first region, $w \geqslant 10$, the asymptotic law for the control increment is used, except where this is known to break down for $q \sim 0.5$. This case is accounted for by ensuring that the absolute value of v is always greater than 3, an approximation suggested by figure 3.3.2.19.

In the second region, $3 \leqslant w < 10$, the control increment is calculated using the symmetry property and linear interpolation from values of $v[w,q]$ held in the array VAR2(21,6). This array is such that

$$VAR2[i,j] = v\left[\overline{i-1}\ /2,\ \overline{j-1}\ /10\right] \quad \text{for } i = 1,2, \ldots, 21,\ j = 1,2, \ldots, 6$$

A4.1.1

In the third region the control increment is similarly calculated using the array VAR1(31,6) for which

$$VAR1\ [i,j] = v\left[\overline{i-1}\ /10,\ \overline{j-1}\ /10\right] \quad \text{for } i = 1,2, \ldots, 31,\ j = 1,2, \ldots, 6$$

A4.1.2

The term IFIX ( ) indicates that the integer immediately below the number included in brackets should be used.

FLOW DIAGRAM Figure A4.1.1

A4.2 NAME RESP[_; IX, z, u1, u, w, q; z1, y, w1, q1]

FUNCTION The function of this subprogram is to calculate the response of the simplified extremal process to a control input, and calculate the new sufficient statistics.

PROGRAMS USED None

DATA None

OUTPUT None

PRESET VARIABLES IX see comments

z extremal process variable

u1 control input, new sufficient statistic

u sufficient statistic

w sufficient statistic

q sufficient statistic.

DUMMY VARIABLES r extremal process variable

x extremal process variable

P1, P2, P3, P4 used in calculating q1.

POSTSET VARIABLES z1 extremal process variable

y extremal process measurement

w1 new sufficient statistic

q1 new sufficient statistic

COMMENTS The subprogram GAUSS[45] used in this subprogram is a standard IBM 1130 program for the generation of sequences of Gaussian random numbers. The preset variable IX is initially set to an odd integer which specifies the particular sequence produced. The variable IX is changed (by GAUSS) each time the subprogram is executed. The sequence of values of the postset variable r produced by successive executions of GAUSS is a sequence of Gaussian random numbers with zero mean value and unit standard deviation.

The term EXP ( ) indicates that the exponential of the number included in brackets should be used.

FLOW DIAGRAM Figure A4.2.1

A4.3 NAME SIM1 [_; _; _]

FUNCTION The function of this subprogram is to simulate the performance of the optimal extremal control system so as to estimate the mean stage cost, and estimate the standard deviation of this estimate.

PROGRAMS USED UOPT, RESP

DATA L the number of stages per group of the simulation

M the number of groups in the simulation

IX see comments

VAR1(31,6) an array containing values of v[w,q] as required by UOPT

VAR2(21,6) an array containing values of v[w,q] as required by UOPT

OUTPUT YOVAV the estimated mean cost per stage

SDOA the estimated standard deviation of YOVAV

NSDOA SDOA normalised with respect to YOVAV

PRESET VARIABLES None

DUMMY VARIABLES YGRAV(M) an array used to build up the average stage cost of each group

u, z, r, x, y extremal process variables

w, q, u1, w1, q1 sufficient statistics

i, j used for counting

POSTSET VARIABLES None

COMMENTS The principles behind the simulation are discussed in chapter 4. Comments regarding the variable IX and the subprogram GAUSS are as in A4.2

FLOW DIAGRAM Figure A4.3.1

A4.4 NAME CAL1 [_; _; _]

FUNCTION The function of this hybrid computation is to sample the simulated white noise signal 2500 times, plot its amplitude distribution, and evaluate its r.m.s. value.

ANALOGUE CONFIGURATION The analogue configuration is shown in figure A4.4.1. The analogue computer must be set manually in OP mode before the digital computer is run.

INTERFACE CONFIGURATION The interface configuration is shown in figure A4.4.2. The interupt clock should not be switched on until the digital computer program first indicates it is awaiting an interupt.

PROGRAMS USED None

DATA None

OUTPUT IDIST(256) IDIST[j] contains the number of samples in the range $\overline{j - 129} \rightarrow \overline{j - 128}$ units, j = 2, ... , 255

RMS the r.m.s. value of the samples in units

PRESET VARIABLES None

DUMMY VARIABLES i used for counting

j used to hold incoming sample

POSTSET VARIABLES None

COMMENTS The program is sufficiently fast so that it is always awaiting an interupt when an interupt occurs. The r.m.s. value of the samples is calculated assuming zero mean value which is assured by the analogue configuration. All calculations are made in terms of the units of the A/D convertor. To convert to Volts in terms of the simulated n(t) signal it is necessary to multiply by 12.8/256 = .05 (due to the A/D convertor).

FLOW DIAGRAM Figure A4.4.3.

A4.5 NAME CAL2[_; _; _]

FUNCTION The function of this hybrid computation is to take 1000 samples of one second integrals of the simulated white noise signal, plot the amplitude distribution, and evaluate the r.m.s. value.

ANALOGUE CONFIGURATION The analogue configuration is shown in figure A4.5.1

INTERFACE CONFIGURATION The interface configuration is shown in figure A4.5.2. The interupt clock should not be switched on until the digital computer first indicates that it is awaiting an interupt.

PROGRAMS USED None

DATA None

OUTPUT IDIST(256) IDIST[j] contains the number of samples in the range $\overline{j - 129} \rightarrow \overline{j - 128}$ units, j = 2, ... , 255

RMS the r.m.s. value of the samples in units

PRESET VARIABLES None

DUMMY VARIABLES i used for counting

j used to hold incoming sample

POSTSET VARIABLES None

COMMENTS The program effectively splits the interupt clock into 2 phases. In the first the integration occurs. In the second a measurement is made, and resetting occurs. Computations are sufficiently fast so that the program is always awaiting an interupt when an interupt occurs. The r.m.s. value of the samples is calculated assuming zero mean value which is assured by the analogue configuration. All calculations are made in terms of units of the A/D convertor. To convert to Volts in terms of the simulated series $\int_{i}^{i+1} n(t)\,dt$, i = 1,3, ... it is necessary to multiply by $\frac{1}{4} \times \frac{12.8}{256} = .0125$ (due to scaling in the analogue configuration, and the A/D convertor).

FLOW DIAGRAM Figure A4.5.3.

A4.6 NAME CAL3[_; _; _]

FUNCTION The function of this hybrid computation is to take 1000 consecutive samples of one second integrals of the simulated white noise signal and plot the autocorrelation function.

ANALOGUE CONFIGURATION The analogue configuration is shown in figure A4.6.1. The configuration is more complex than in A4.5 because samples must be consecutive.

INTERFACE CONFIGURATION The interface configuration is shown in figure A4.6.2. The interupt clock should not be switched on until the digital computer first indicates that it is awaiting an interupt.

PROGRAMS USED None

DATA None

OUTPUT AF(10) AF[j] , j = 1,2, ... , 10 contains the correlation between samples spaced $\overline{j-1}$ periods apart. Correlation is in $(\text{unit})^2$ and is not normalised.

PRESET VARIABLES None

DUMMY VARIABLES i, j used for counting

IST(1000) IST[i] , i = 2,3, ... , 1000 contains the consecutive samples.

MV the mean value of the samples.

ST(999) ST[i] , i = 1,2, ... , 999 contains the consecutive values with zero mean value.

POSTSET VARIABLES None.

COMMENTS This computation is again two phase, similar to the previous one, but arranged to obtain consecutive samples. Again all computations are sufficiently fast so that the program is always awaiting an interupt when an interupt occurs. The computer handkey should be set to 1 before the program is run, and switched to 0 when it is apparent that the integrators are resetting properly. This avoids difficulties encountered if the integrator amplifiers have saturated before the computation has started. The autocorrelation function determined by the computation is not normalised. Normalisation is achieved by dividing each element of the array AF(10) by AF[1].

FLOW DIAGRAM Figure A4.6.3.

A4.7 NAME SIM2[_; _; _]

FUNCTION The function of this hybrid computation is to simulate the performance of the optimal extremal control system, using a continuous-time extremal process, so as to estimate the mean stage cost, and estimate the standard deviation of this estimate.

ANALOGUE CONFIGURATION The analogue configuration is shown in figure A4.7.1. Details of the translation and scaling of the input A2 from the interface are given in the comments section.

INTERFACE CONFIGURATION The interface configuration is shown in figure A4.7.2. The interupt clock should not be switched on until the digital computer first indicates that it is awaiting an interupt.

PROGRAMS USED UOPT

DATA L, M, VAR1(31,6), VAR2(21,6) as in SIM1

OUTPUT YOVAV, SDOA as in SIM1

PRESET VARIABLES None

DUMMY VARIABLES i, j used for counting

YGRAV(M) as in SIM1

IV holds outgoing integer to interface

IY holds incoming integer from interface

NY the normalised value of the measurement

u, w, q sufficient statistics

u1, W updated sufficient statistics

v optimal control increment, normalised.

POSTSET VARIABLES None

COMMENTS The digital section of this computation is similar to SIM1. The term 1F1X ( ) has the meaning defined in section A4.1

Translation and Scaling of the digital computer output and analogue computer input (i.e. the adjustment) is determined as follows. The digital computer program calculates the normalised control increment v. Translating and scaling this so that values v = +5 and v = -5 (maximum likely values found during the discrete-time simulations) just saturate the 8- bit D/A convertor gives the integer output from the digital computer as

$$IV = 128 + 1F1X\left(\frac{127.5}{5}\ v\right) = 128 + 25.5v \qquad A4.7.1$$

Scaling in the D/A convertor then gives the input to the analogue computer as

$$A2 = \frac{12.8}{256} IV = 6.4 + 1.28v \qquad A4.7.2$$

Before this can be applied to an amplifier it must be scaled by a factor $\frac{10}{12.8} = 0.781$ so as to avoid saturating the amplifier. This gives the signal E_5 (figure A4.7.1) as

$$E_5 = 5 + v \qquad A4.7.3$$

The remaining 5 Volt bias is removed and the normalised control increment multiplied by $\sigma/10$ (= 0.167) to produce v(t)/10 as shown in figure A4.7.1.

Because the output A1 from the extremal process is essentially positive no translation is necessary. No analogue scaling is necessary because the saturation level of the 12 bit A/D convertor is +10 Volts. The integer IY input to the digital computer when a measurement is made is thus

$$IY = \frac{4095}{10} A1 \qquad A4.7.4$$

$$\text{i.e. } A1 = \frac{10}{4095} IY$$

and the normalised measurement NY is then given by

$$NY = \frac{A1}{A\sigma^2} = \frac{10}{4095} \cdot \frac{1}{(0.1)(1.67)^2} IY = .00878\ IY \qquad A4.7.5$$

FLOW DIAGRAM Figure A4.7.3

A4.8 NAME CONT1[VAR1(31,6), VAR2(21,6); i,j,u,w,q; u1,v]

FUNCTION The function of this subprogram is to provide the optimal extremal controller for SIM3.

PROGRAMS USED UOPT

DATA VAR1 (31,6), VAR2 (21,6) arrays used by UOPT.

OUTPUT None

PRESET VARIABLES i,j both set to 1 at the beginning of a simulation, showing that VAR1 and VAR2 must be read in.

u,w,q sufficient statistics.

DUMMY VARIABLES None

POSTSET VARIABLES u1, v variables set by UOPT

COMMENTS The arrays VAR1 and VAR2 are postset in the sense that they are used by CONT1 each time it is executed, but they are not used by SIM3.

FLOW DIAGRAM Figure A4.8.1

A4.9 NAME CONT2[a; i,j,u,w,q; u1,v]

FUNCTION The function of this subprogram is to provide the suboptimal extremal controller for SIM3.

PROGRAMS USED None

DATA a the normalised constant intentional error

OUTPUT None

PRESET VARIABLES i,j both set to 1 at the beginning of a simulation, showing that VAR1 and VAR2 must be read in

u,w,q sufficient statistics

DUMMY VARIABLES None

POSTSET VARIABLES u1 updated sufficient statistic

v suboptimal control increment

COMMENTS The variable a is postset in the sense that it is used by CONT2 each time it is executed, but it is not used by SIM3. The term 'SIGN( )' used in figure A4.9.1 is used to represent +1 if the argument is positive and -1 if the argument is negative.

FLOW DIAGRAM Figure A4.9.1

A4.10 NAME SIM3[_; CONT; _]

FUNCTION The function of this hybrid computation is to estimate the normalised mean stage cost of the extremal process with output noise. The controller used is specified by the preset subprogram name CONT.

ANALOGUE CONFIGURATION The analogue configuration used is identical to that used in SIM2.

INTERFACE CONFIGURATION The interface configuration used is identical to that used in SIM2.

PROGRAMS USED GAUSS A standard IBM subprogram described under subprogram RESP

CONT Either CONT1 or CONT2

UOPT Used only in CONT1.

DATA L, M determine the number of stages of the process simulated

IX , parameter used by GAUSS

SD the standard deviation of the sequence generated by GAUSS

VAR1, VAR2 arrays used only if CONT1 is used

a used only if CONT2 is used.

OUTPUT YOVAV the estimated normalised mean cost per stage

SDOA the standard deviation of the estimate.

PRESET VARIABLES None

DUMMY VARIABLES YGRAV(M) used to contain the mean cost per stage of the groups used in the estimation

IV interger used to contain output to the interface

IY integer used to contain input from the interface

NY the normalised stage cost; stage measurement

ξ the normalised output noise

u,w,q	sufficient statistics
i,j	used for counting
u1,W	updated sufficient statistics
v	control increment

POSTSET VARIABLES None

COMMENTS Input and output scaling and translation is as detailed in SIM2.

FLOW DIAGRAM Figure A4.10.1

References

1. JACOBS O., 'Hill Climbing', Control Vol.5, Feb. 1962.
2. JACOBS O., 'Two Uses of the Term 'Adaptive' in Automatic Control', I.E.E.E. Trans. Automatic Control, Vol AC-9, Oct. 1964.
3. DOUCE J., 'The Mathematics of Servomechanisms', E.U.P., 1963.
4. ELGERD O., 'Control Systems Theory', McGraw-Hill, 1967.
5. NOTON A., 'Variational Methods in Control Engineering' Pergammon Press, 1965.
6. SHERING G., 'The Analysis and Design of Linearised Single-Input Extremum Control Systems', Ph.D. Thesis, University of Edinburgh, 1966.
7. DRAPER C. and LI Y., 'The Principles of Optimalising Control Systems and an Application to the Internal Combustion Engine', A.S.M.E. (Special Publication), Sept. 1951.
8. DEELEY E. and ALLOS J., 'Automatic Focussing of an Optical System by Extremum Control', Proc. I.E.E., Vol 114, 1967.
9. TAYLOR W., 'An Experimental Control System with Continuous Automatic Optimisation', Proc. I.F.A.C. Congress, 1960.
10. DOUCE J., 'The Mathematics of Servomechanisms', Chapter 9, E.U.P., 1963.
11. THOMPSON A., 'Direct Digital Control - The State of the Art in 1967', Proc. I.E.E., Vol 115, 1968.
12. MORAN F., BERGER C. and XIROKOSTAS D., 'Development and Application of Self-Optimising Control to Coal-Fired Steam-Generating Plant', Proc. I.E.E., Vol 115, 1968.

13. IZAWA K. and OKAMOTO H., 'Optimising Control of Hydrogenation Process', Proc. I.F.A.C. Congress, 1969.

14. ELGERD O., 'Control Systems Theory', Chapter 10, McGraw-Hill, 1967.

15. FELDBAUM A., 'Dual Control Theory I - IV', Automation and Remote Control, Vols 21 and 22, 1960-61.

16. JACOBS O. and SHERING G., 'Design of a Single Input, Sinusoidal Perturbation, Extremum Control System', Proc. I.E.E., Vol 115, 1968.

17. DOUCE J. and BOND A. 'Development and Performance of a Self-Optimising system', Proc. I.E.E., Vol 110, 1963.

18. DOUCE J. and NG K. 'The Use of Pseudorandom Signals in Adaptive Control', I.F.A.C. Symposium on the Theory of Self-Adaptive Systems, 1965.

19. VAN DER GRINTEN P., 'The Application of Random Test Signals in Process Optimisation ', Proc. I.F.A.C. Congress, 1963.

20. JACOBS O. and WONHAM W., 'Extremum Control in the Presence of Noise', Journal of Electronics and Control, Vol. 11, 1961.

21. WOODSIDE C., 'The Design of Optimal Extremal Controllers', International Journal of Control, Vol 8, 1968.

22. JACOBS O., 'Simulation and Dimensional Analysis of a Stepping Extremum Control System', Proc. U.K.A.C. Convention, 1968.

23. CHANG S., 'Optimisation of the Adaptive Function by ≥ Transform Method' Trans. A.I.E.E., 1960.

24. NG K., 'High Frequency Perturbation in Hill Climbing Systems', Proc. I.E.E., 1964.

25. EVELEIGH V., 'Adaptive Control and Optimisation Techniques', Chapter 8, McGraw-Hill, 1967.

26. DAVIS R. and ROBERTS P., 'Method of Conjugate Gradients Applied to Self Adaptive Digital Control Systems' Proc. I.E.E., Vol 115, 1968.

27. FLORENTIN J., 'An Approximately Optimal Extremal Regulator', Journal of Electronics and Control, Vol. 17, 1964.

28. ROBERTS J., 'Extremum or Hill Climbing Regulation', Proc. I.E.E., Vol. 112, 1965.

29. JACOBS O. 'Extremum Control and Optimal Control Theory', Proc. I.F.A.C. Symposium on the Identification of Systems, 1967.

30. MOZGOVAYA E. and SAMOYLENKO Y., 'Construction of an Optimal Algorithm for Extremal Control Based on the Principle of Dynamic Programming', Engineering Cybernetics, 1964.

31. MOZGOVAYA E., 'Optimal Algorithm for Extremal Control in the Presence of Noise at the System Input and Output', Engineering Cybernetics, 1966.

32. JACOBS O. and LANGDON S., 'An Optimal Extremal Control System', Automatica, 1970.

33. ATHANS M. and FALB P., 'Optimal Control ', McGraw-Hill, 1966.

34. BELLMAN R., 'Adaptive Control Processes ', O.U.P., 1961.

35. BELLMAN R., 'Dynamic Programming', Princeton University Press, 1957.

36. JACOBS O., 'Dynamic Programming', Chapman and Hall, 1967.

37. COMRIE L., 'Mathematical Tables', W. & R. Chambers.

38. 'IBM 1130 FORTRAN Language', IBM Systems Reference Library.

39. SCHOFIELD P. and OSBORNE M., 'Programming in Atlas Autocode' University of Edinburgh Computer Unit Report Number One, 1966.

40.'IBM 1130 Hybrid and On-line Computing System' Report of the Oxford University Systems Engineering Group., 1968.

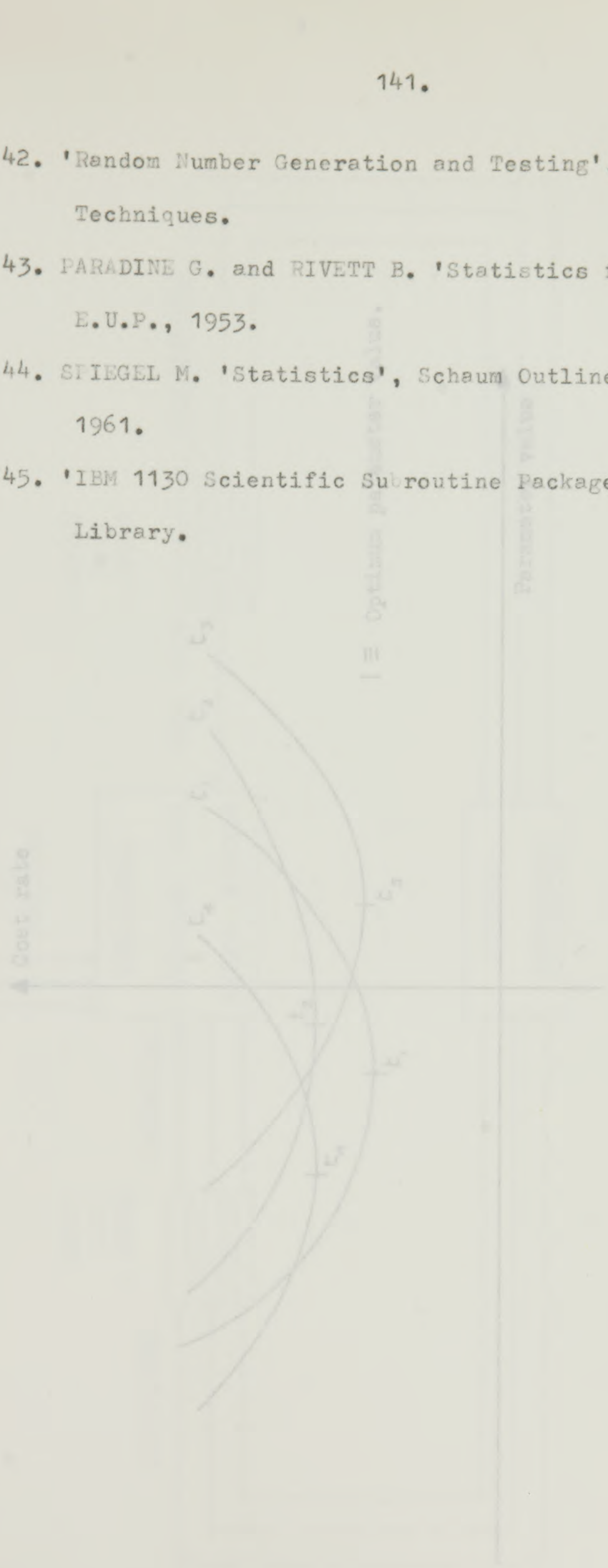
41. BUCKINGHAM E., 'Model experiments and the forms of empirical equations', Trans. A.S.M.E. Vol. 37, 1915.

42. 'Random Number Generation and Testing', IBM Data Processing Techniques.

43. PARADINE G. and RIVETT B. 'Statistics for technologists' E.U.P., 1953.

44. SPIEGEL M. 'Statistics', Schaum Outline Series, McGraw-Hill, 1961.

45. 'IBM 1130 Scientific Subroutine Package' IBM Systems Reference Library.



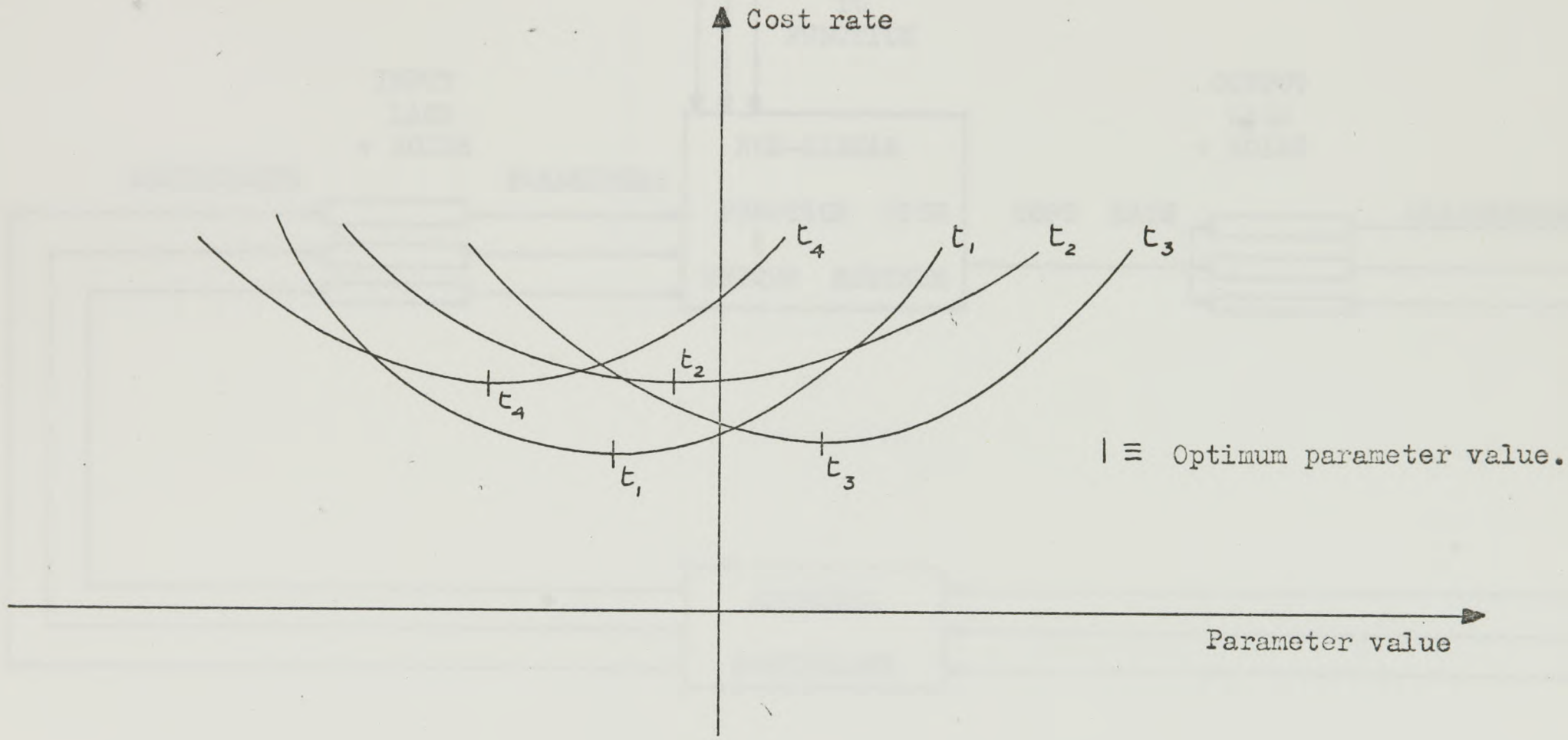

Cost rate
t_4
t_1
t_2
t_3
t_2
t_4
t_1
t_3
| ≡ Optimum parameter value.
Parameter value

Figure 1.1.1

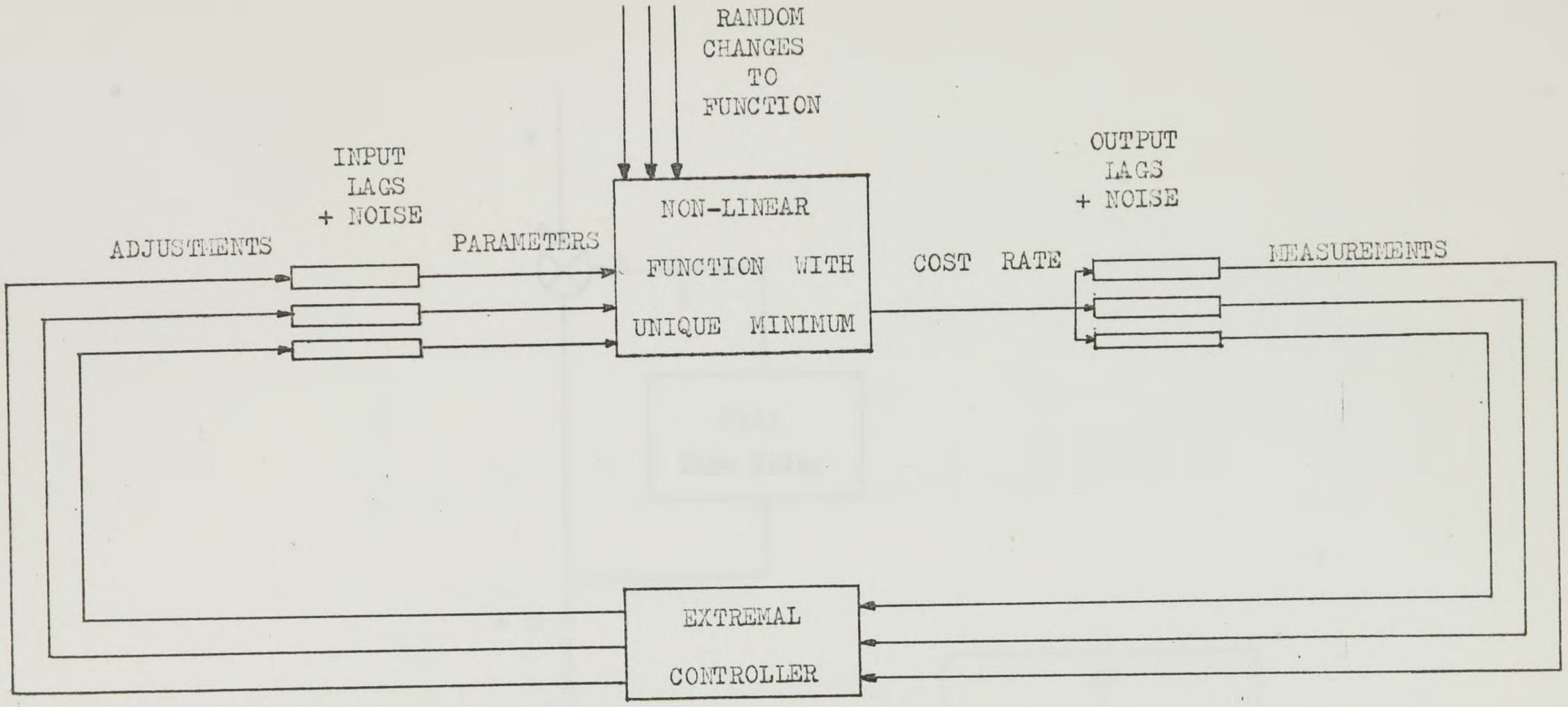

RANDOM
CHANGES
TO
FUNCTION
INPUT
LAGS
+ NOISE
ADJUSTMENTS
PARAMETERS
NON-LINEAR
FUNCTION WITH
UNIQUE MINIMUM
COST RATE
OUTPUT
LAGS
+ NOISE
MEASUREMENTS
EXTREMAL
CONTROLLER

Figure 1.1.2

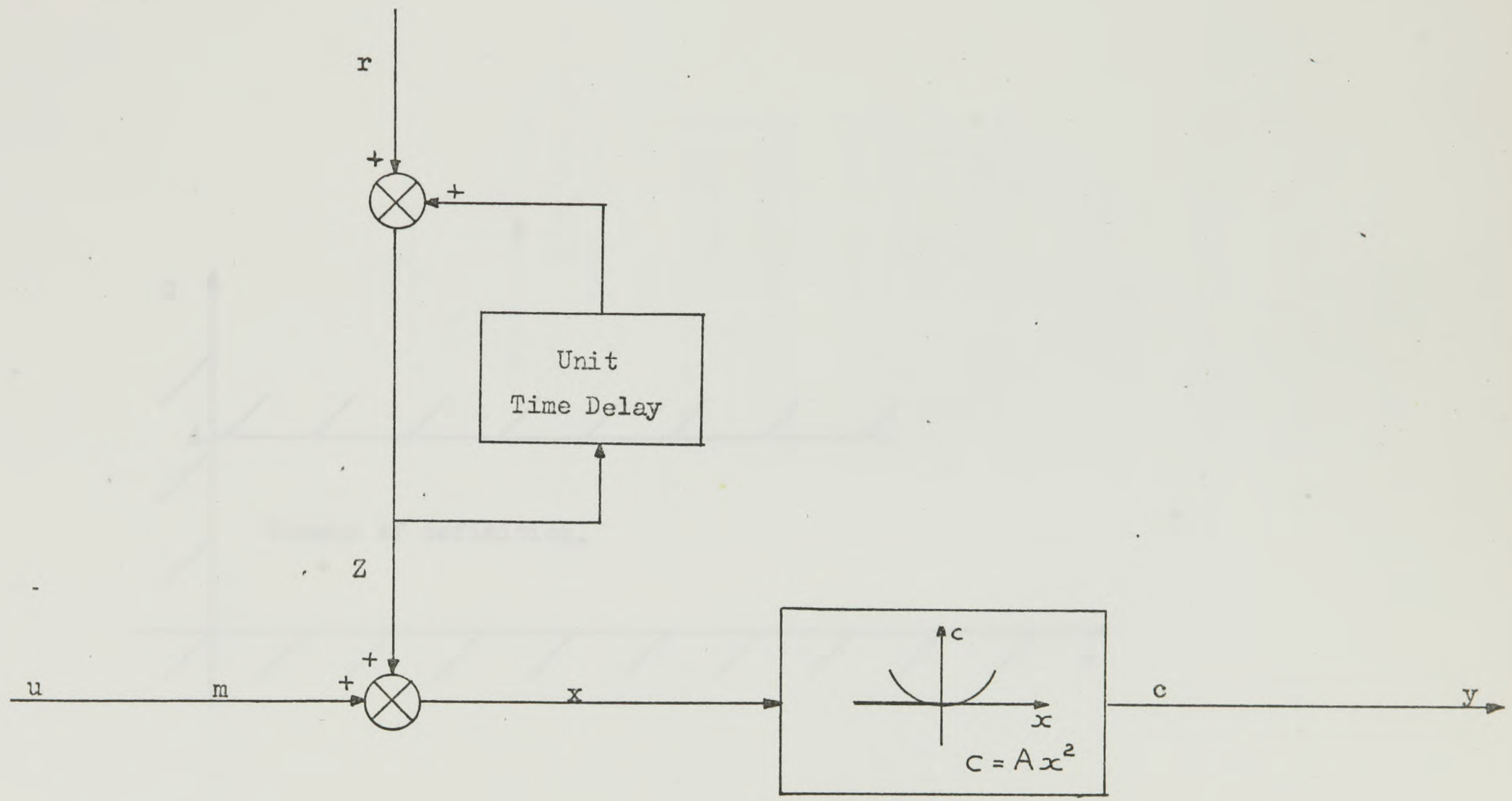


Figure 2.1.1

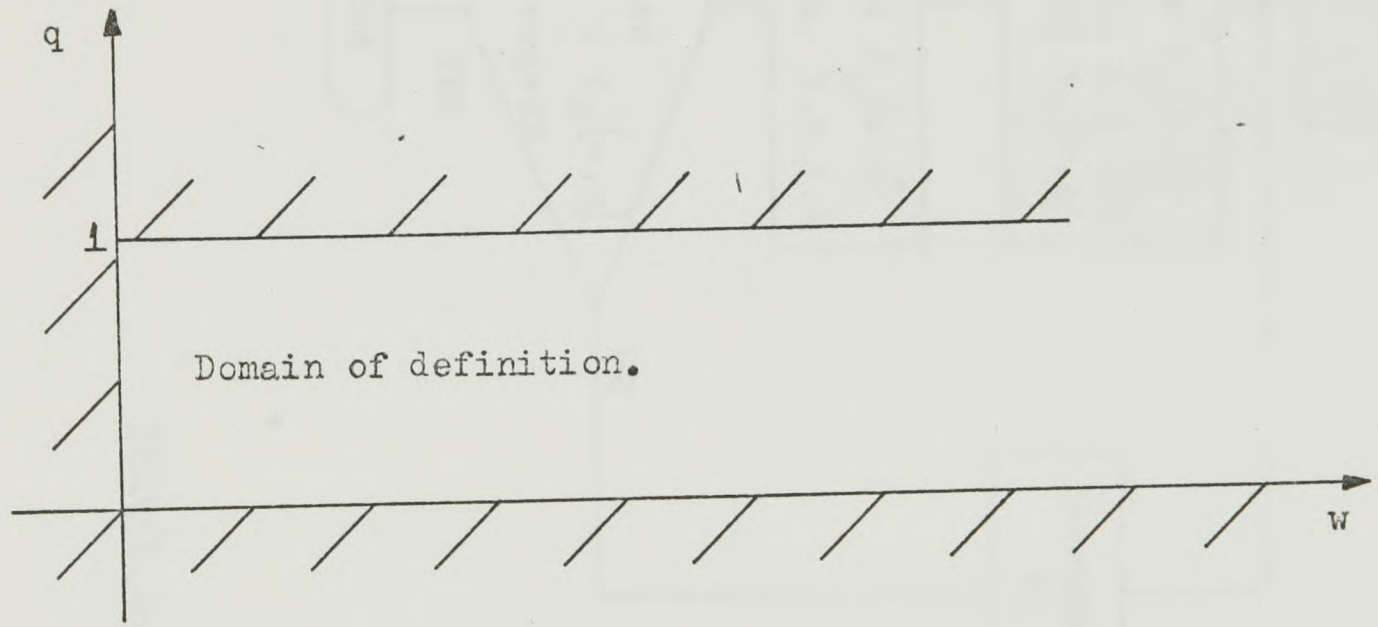


Figure 3.2.1.1

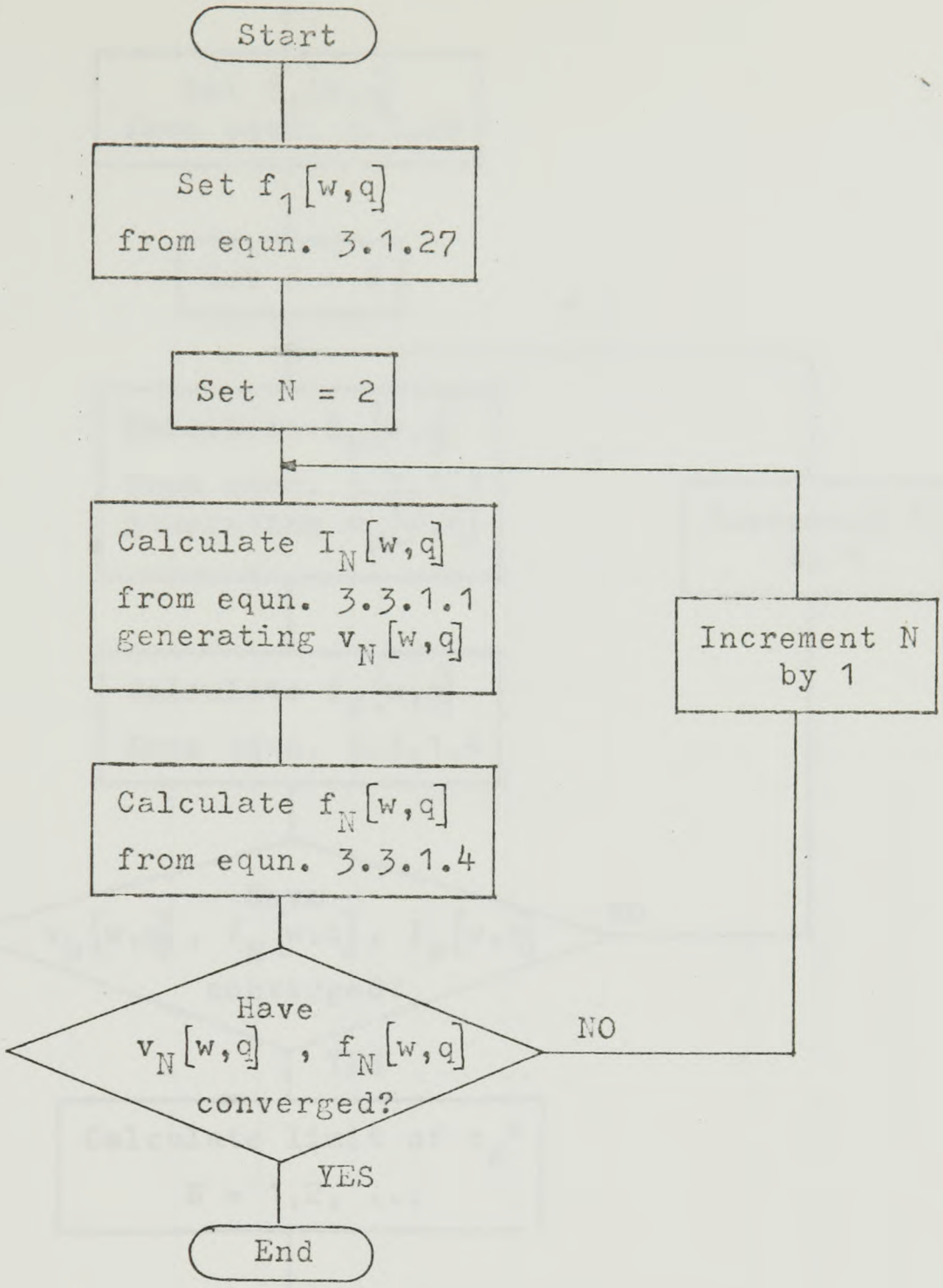

Start
Set $f_1[w,q]$ from equn. 3.1.27
Set N = 2
Calculate $I_N[w,q]$ from equn. 3.3.1.1 generating $v_N[w,q]$
Increment N by 1
Calculate $f_N[w,q]$ from equn. 3.3.1.4
Have $v_N[w,q]$, $f_N[w,q]$ converged?
NO
YES
End

Figure 3.3.1.1

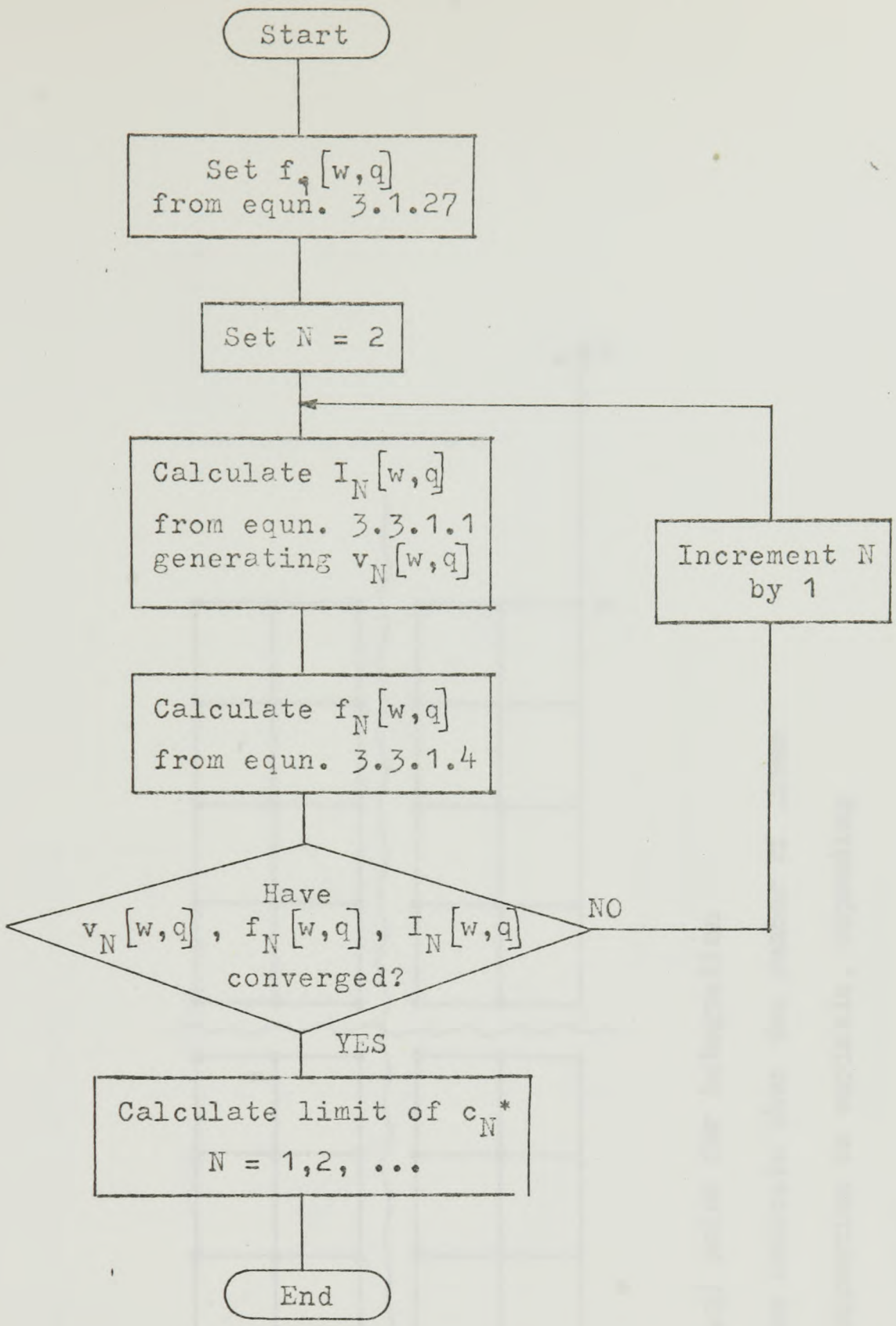
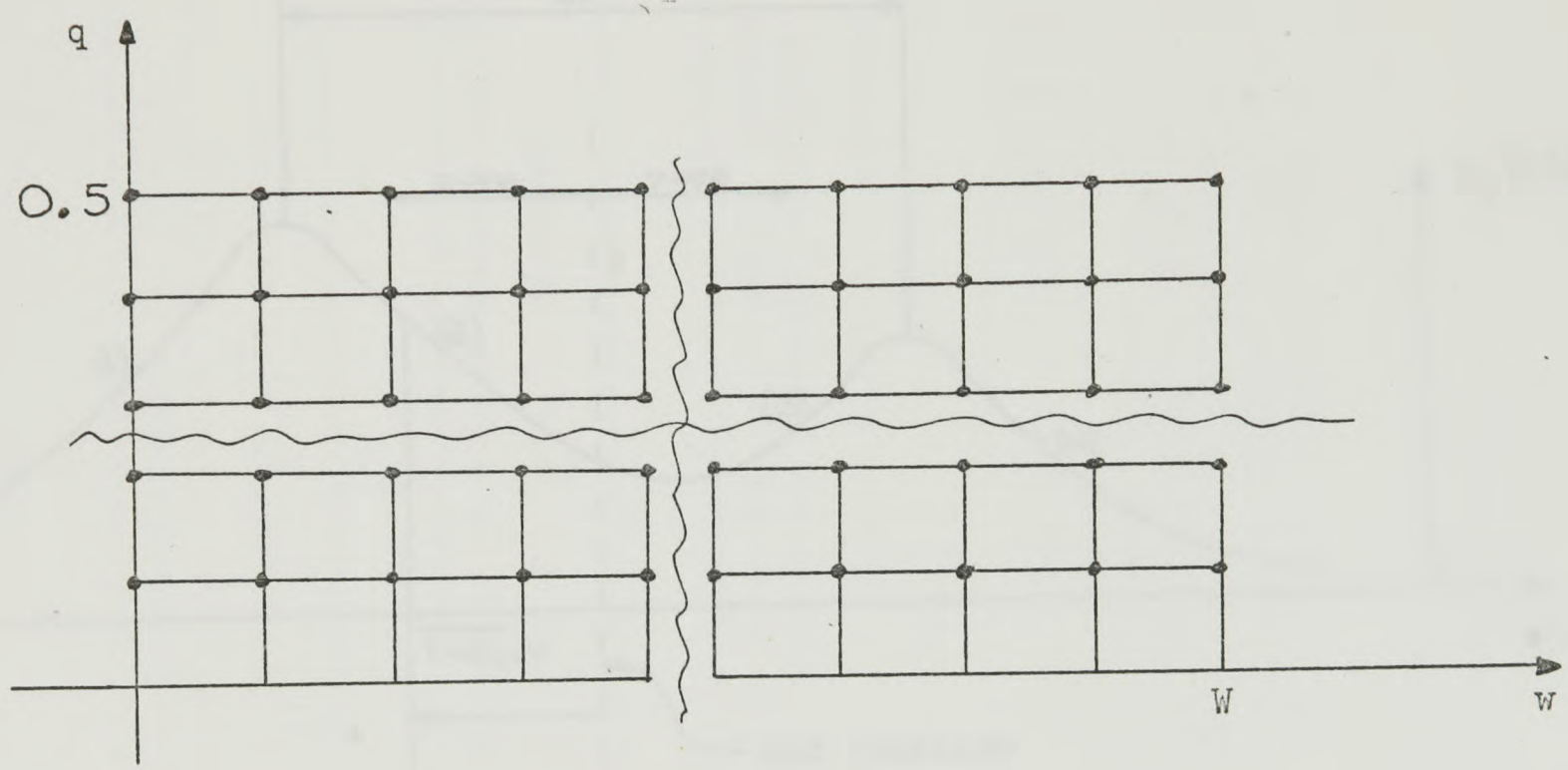

Start
Set $f_1[w,q]$ from equn. 3.1.27
Set N = 2
Calculate $I_N[w,q]$ from equn. 3.3.1.1 generating $v_N[w,q]$
Increment N by 1
Calculate $f_N[w,q]$ from equn. 3.3.1.4
Have $v_N[w,q]$, $f_N[w,q]$, $I_N[w,q]$ converged?
NO
YES
Calculate limit of c_N^* N = 1,2, ...
End

Figure 3.3.1.2



• ≡ (w,q) point for integration

wavy lines indicate that the number of lines in each direction is variable, depending on dw, dq and W.

Figure 3.3.1.3

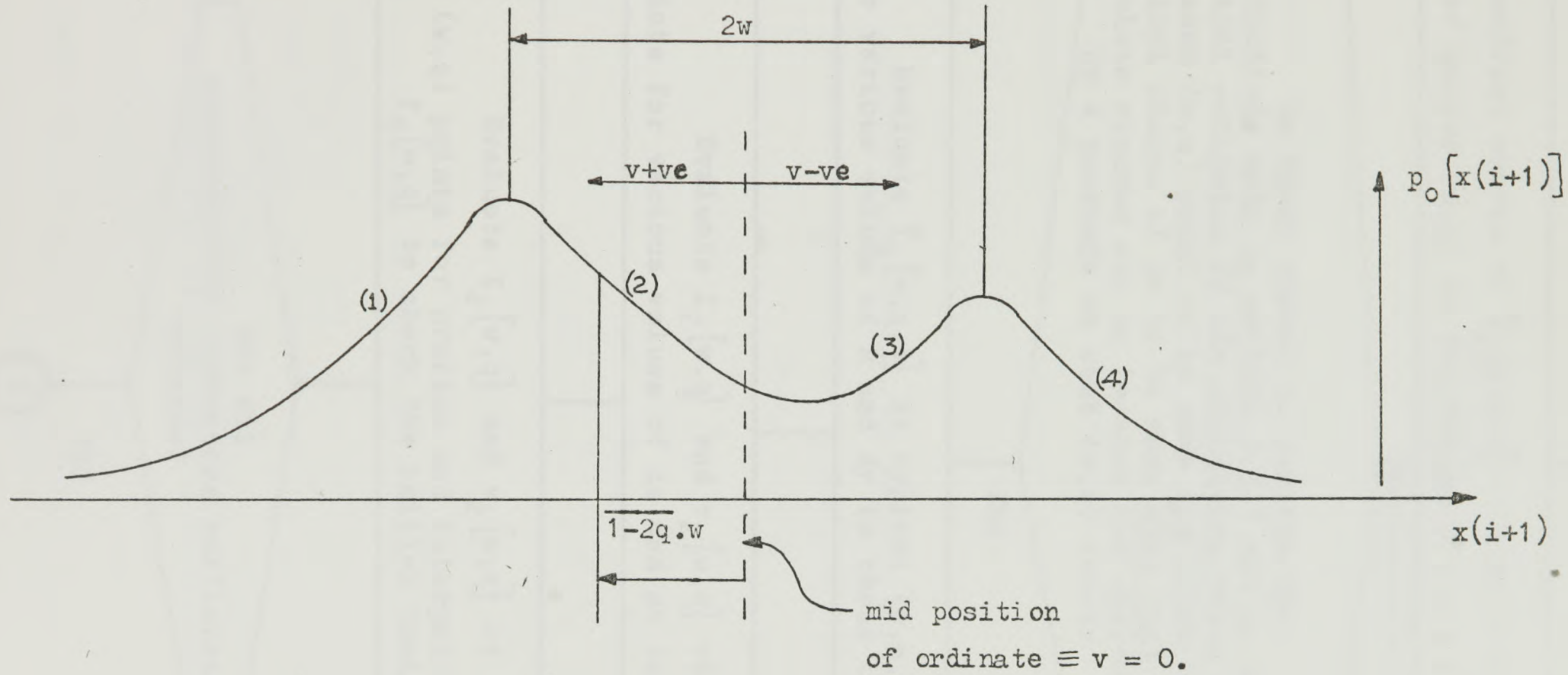
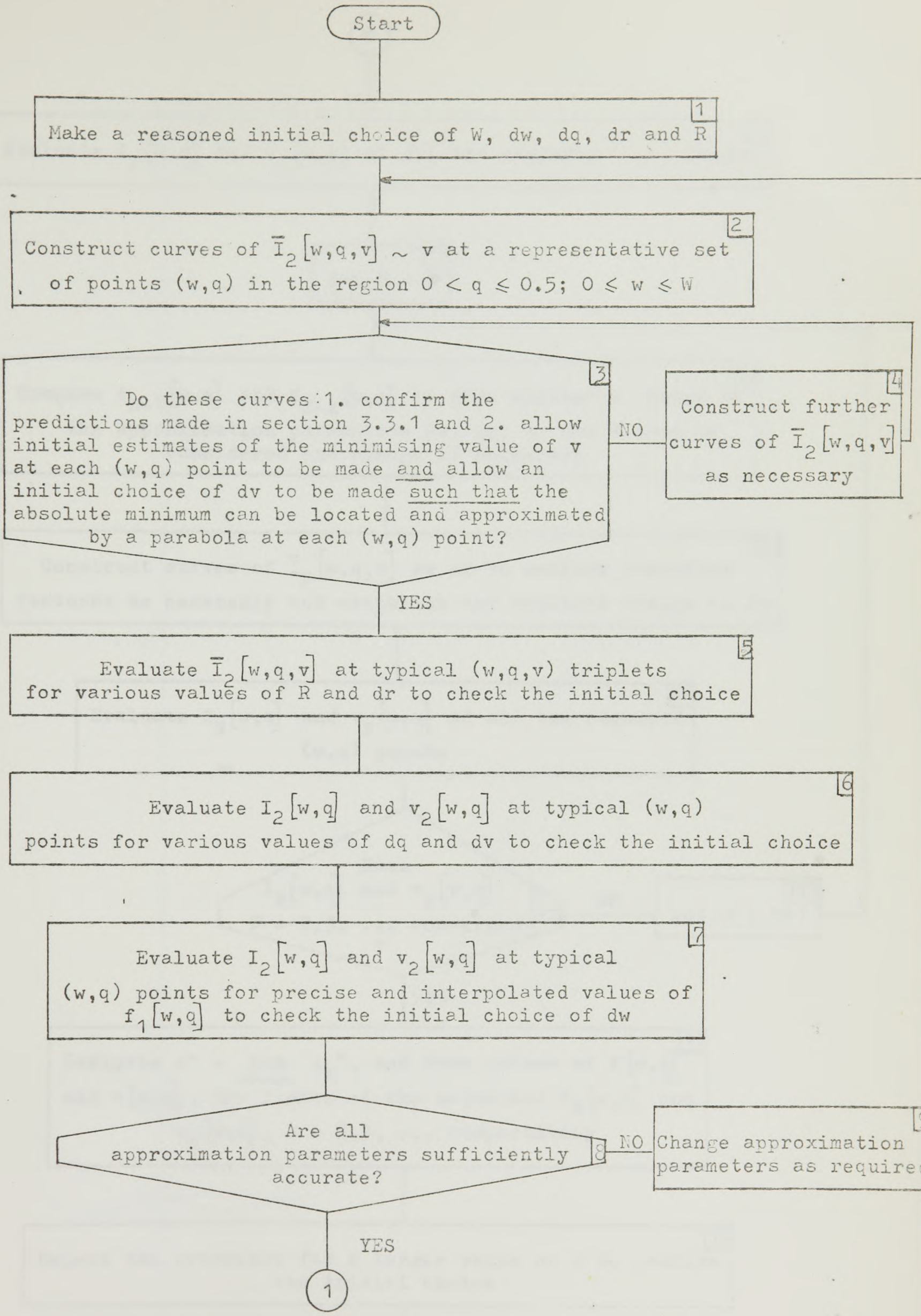

2w
v+ve
v-ve
(1)
(2)
(3)
(4)
$p_o[x(i+1)]$
$\overline{1-2q}.w$
x(i+1)
mid position
of ordinate ≡ v = 0.

Figure 3.3.1.4


Start
1
Make a reasoned initial choice of W, dw, dq, dr and R
2
Construct curves of $\bar{I}_2[w,q,v] \sim v$ at a representative set of points (w,q) in the region $0 < q \leq 0.5;\ 0 \leq w \leq W$
3
Do these curves: 1. confirm the predictions made in section 3.3.1 and 2. allow initial estimates of the minimising value of v at each (w,q) point to be made and allow an initial choice of dv to be made such that the absolute minimum can be located and approximated by a parabola at each (w,q) point?
NO
4
Construct further curves of $\bar{I}_2[w,q,v]$ as necessary
YES
5
Evaluate $\bar{I}_2[w,q,v]$ at typical (w,q,v) triplets for various values of R and dr to check the initial choice
6
Evaluate $I_2[w,q]$ and $v_2[w,q]$ at typical (w,q) points for various values of dq and dv to check the initial choice
7
Evaluate $I_2[w,q]$ and $v_2[w,q]$ at typical (w,q) points for precise and interpolated values of $f_1[w,q]$ to check the initial choice of dw
8
Are all approximation parameters sufficiently accurate?
NO
Change approximation parameters as required
YES
1

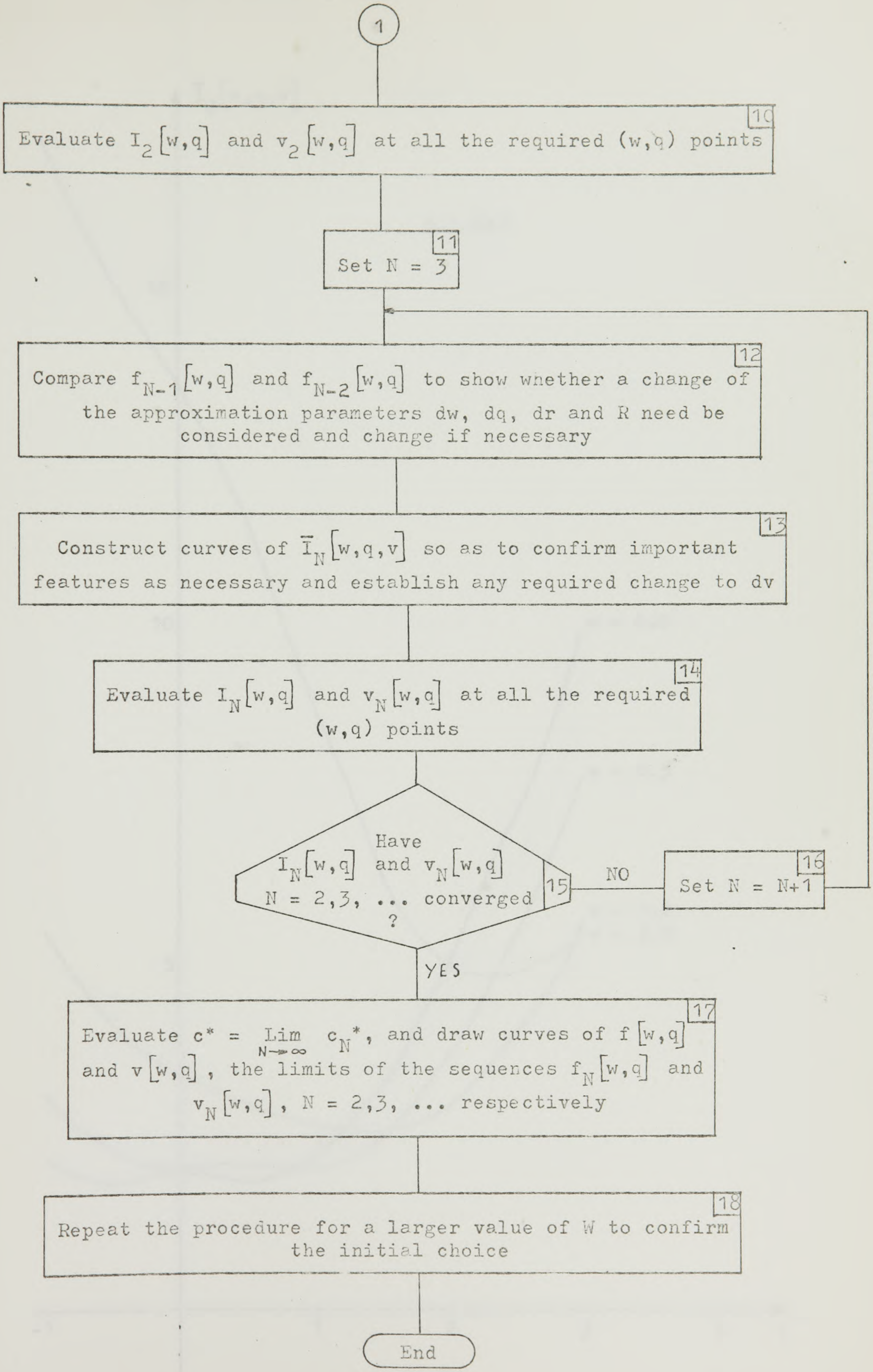

1
Evaluate $I_2[w,q]$ and $v_2[w,q]$ at all the required (w,q) points
10
Set N = 3
11
Compare $f_{N-1}[w,q]$ and $f_{N-2}[w,q]$ to show whether a change of the approximation parameters dw, dq, dr and R need be considered and change if necessary
12
Construct curves of $\bar{I}_N[w,q,v]$ so as to confirm important features as necessary and establish any required change to dv
13
Evaluate $I_N[w,q]$ and $v_N[w,q]$ at all the required (w,q) points
14
Have $I_N[w,q]$ and $v_N[w,q]$ N = 2,3, ... converged ?
15
NO
Set N = N+1
16
YES
Evaluate $c^* = \lim_{N\to\infty} c_N^*$, and draw curves of $f[w,q]$ and $v[w,q]$, the limits of the sequences $f_N[w,q]$ and $v_N[w,q]$, N = 2,3, ... respectively
17
Repeat the procedure for a larger value of W to confirm the initial choice
18
End

Figure 3.3.2.1

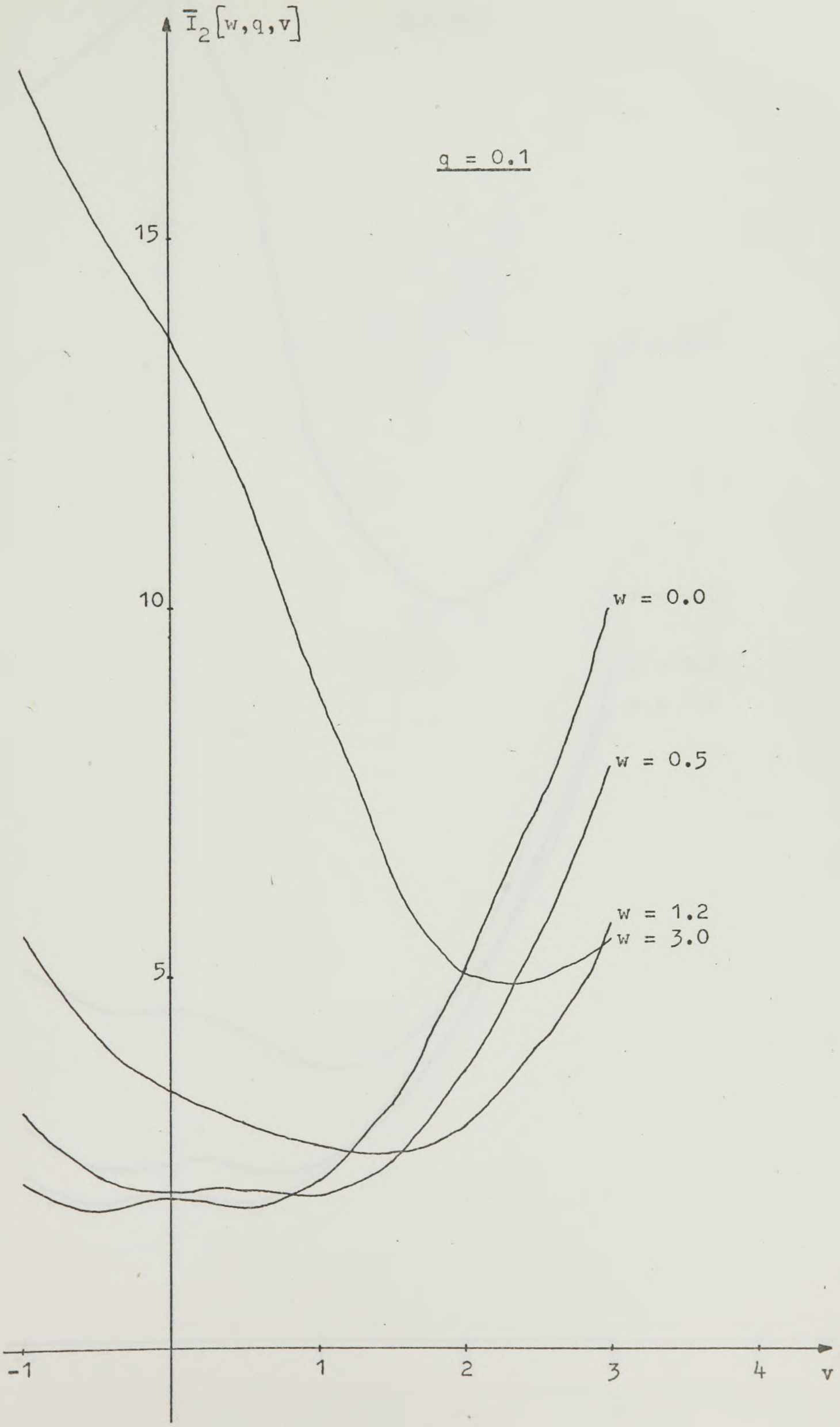


Figure 3.3.2.2

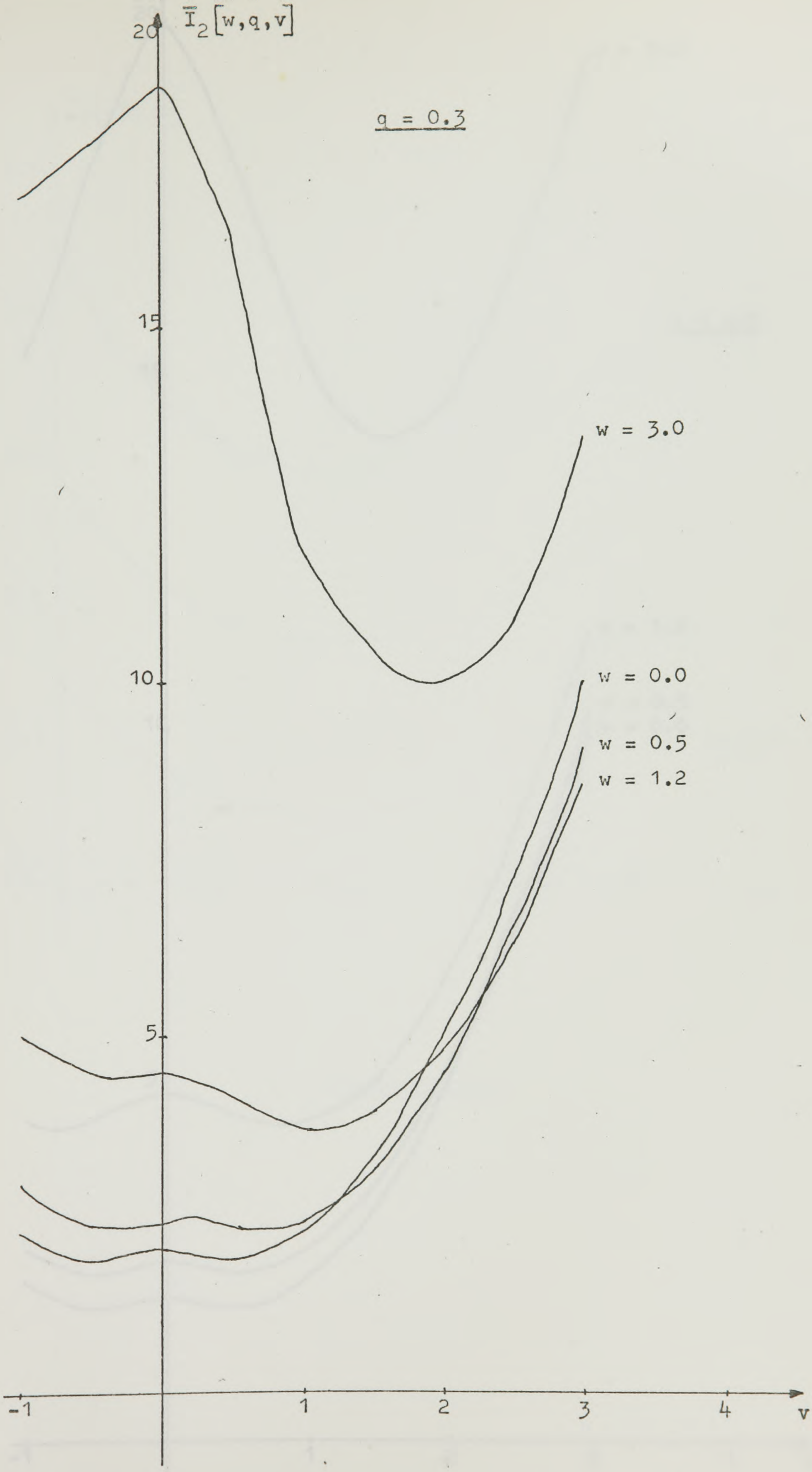


Figure 3.3.2.3

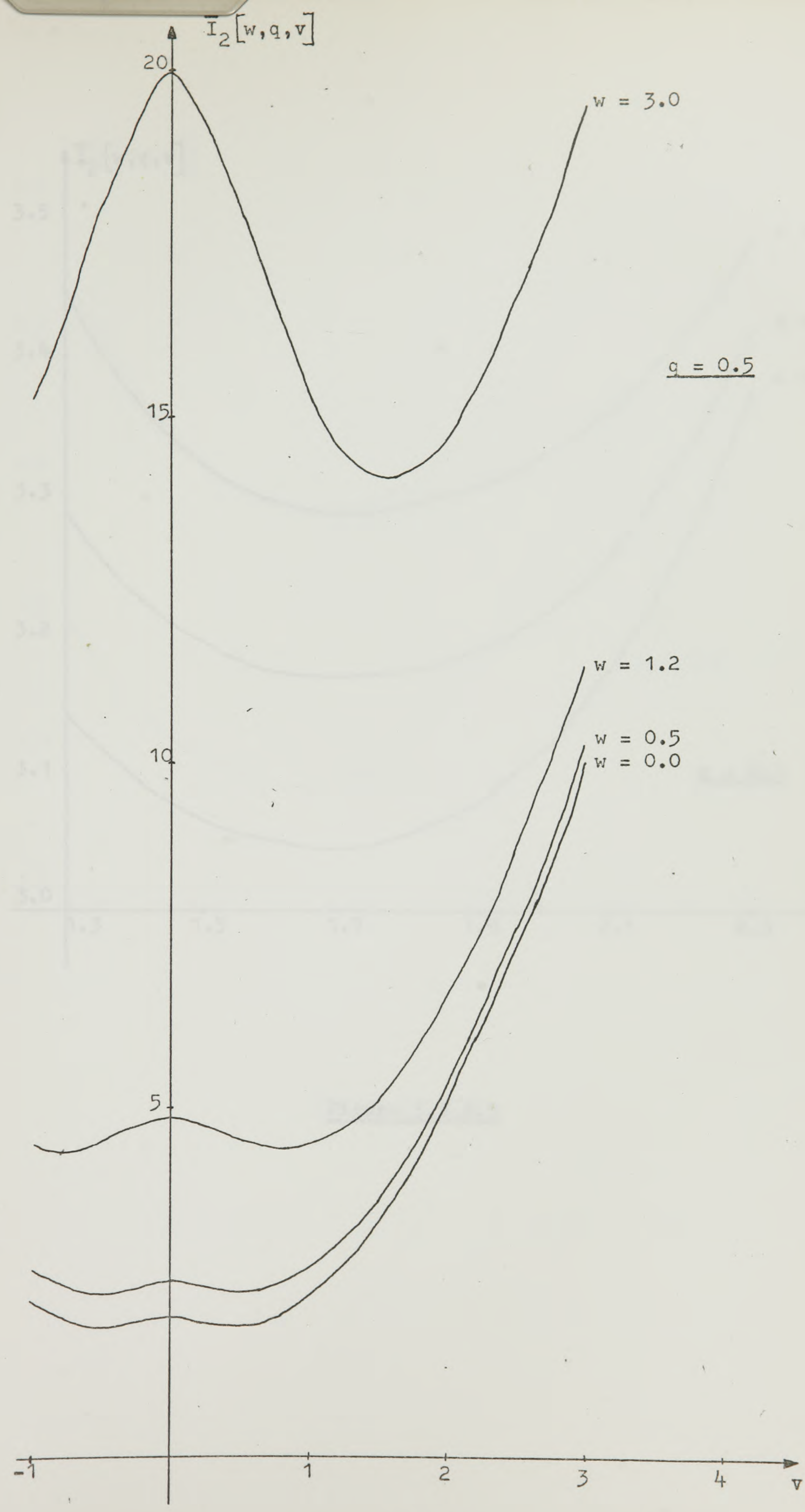


Figure 3.3.2.4

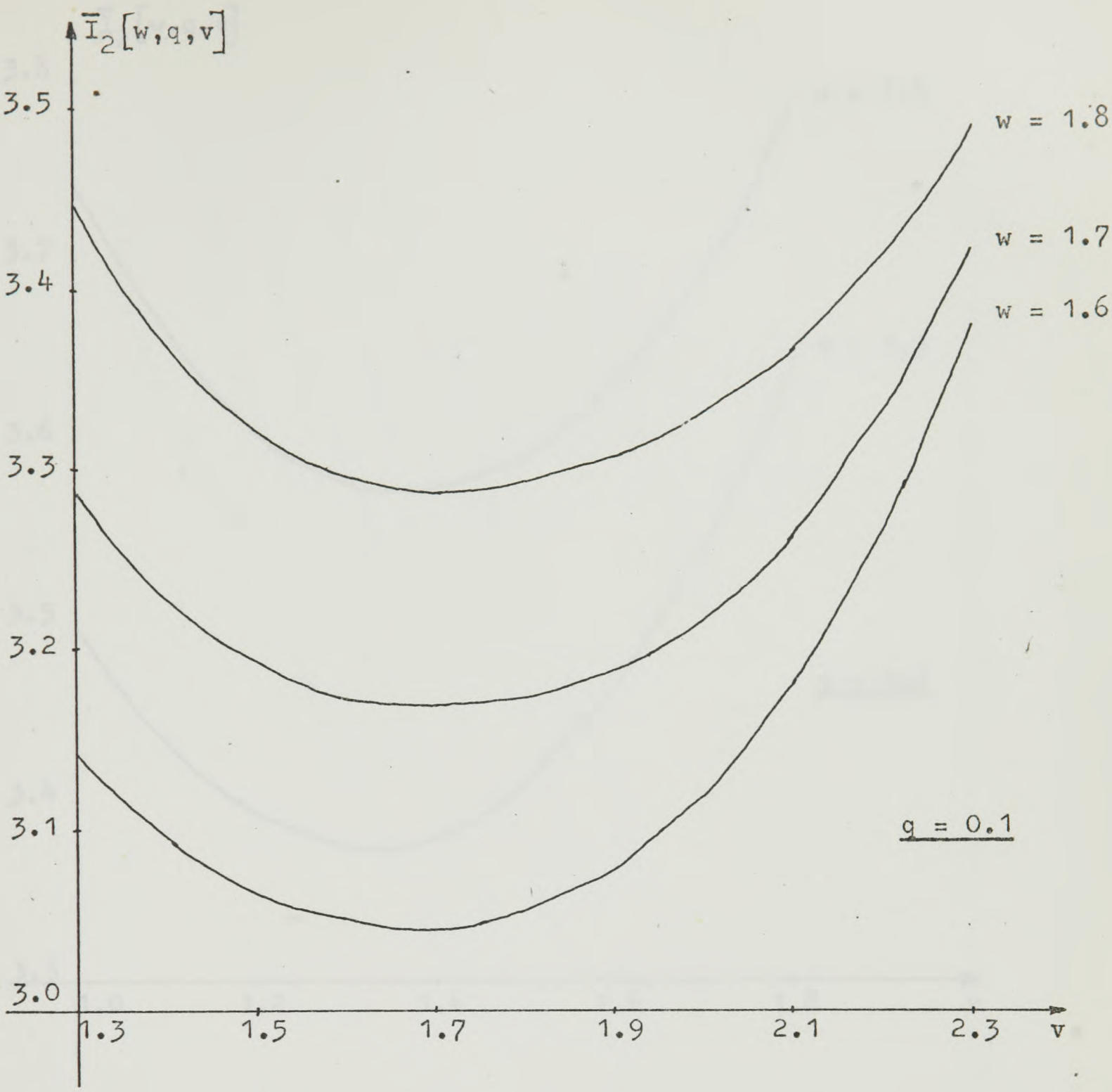


Figure 3.3.2.5

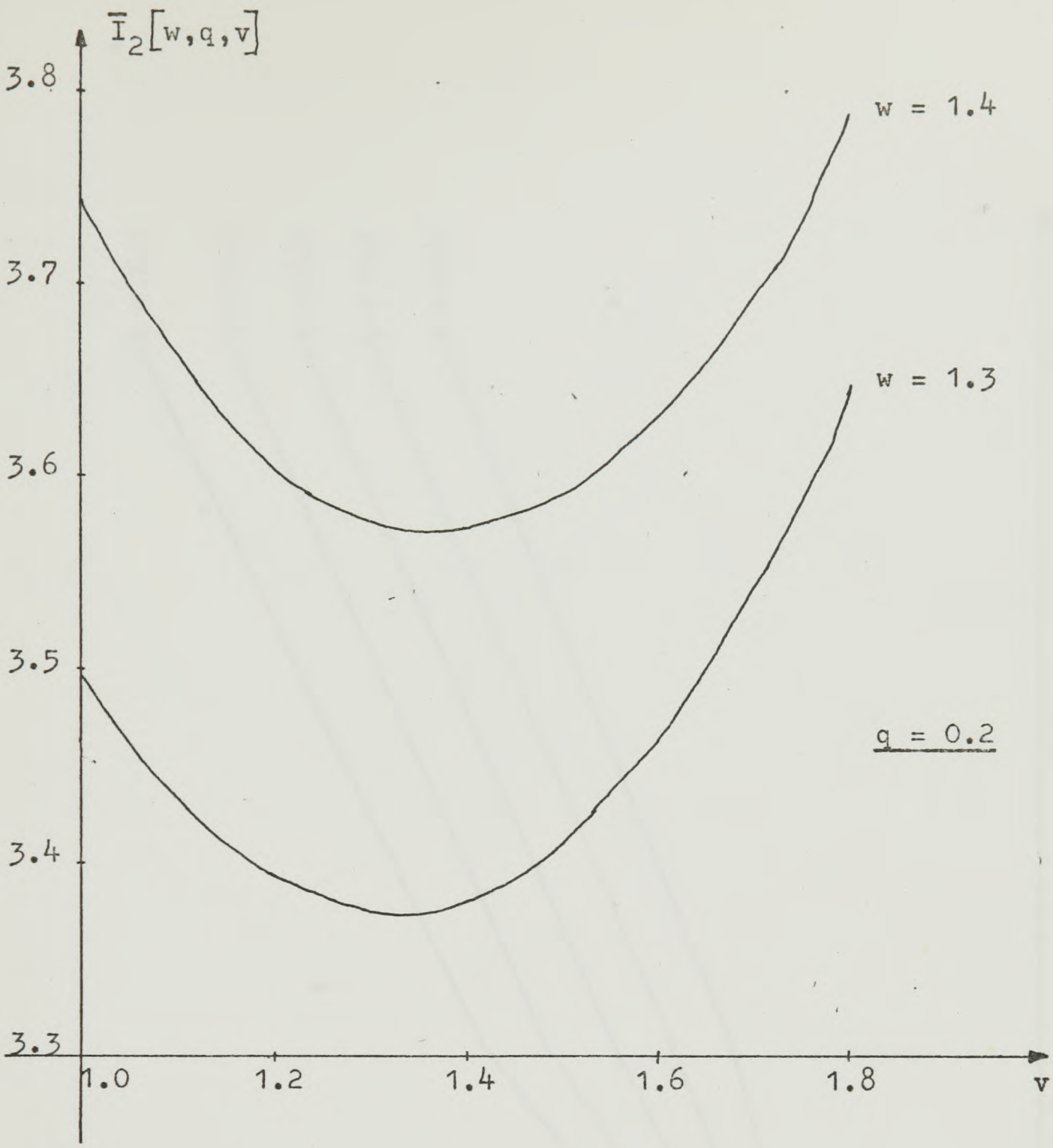


Figure 3.3.2.6

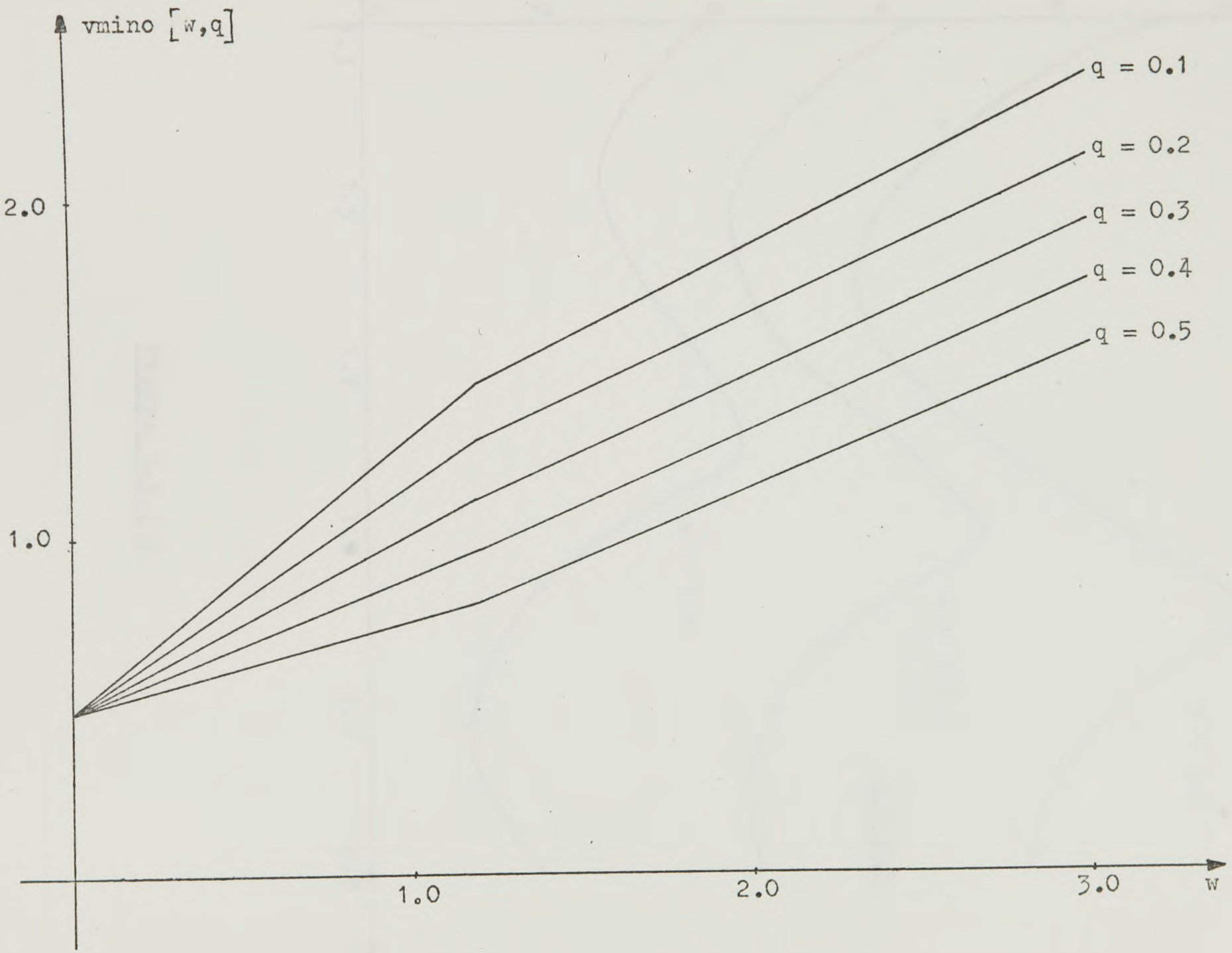


Figure 3.3.2.7

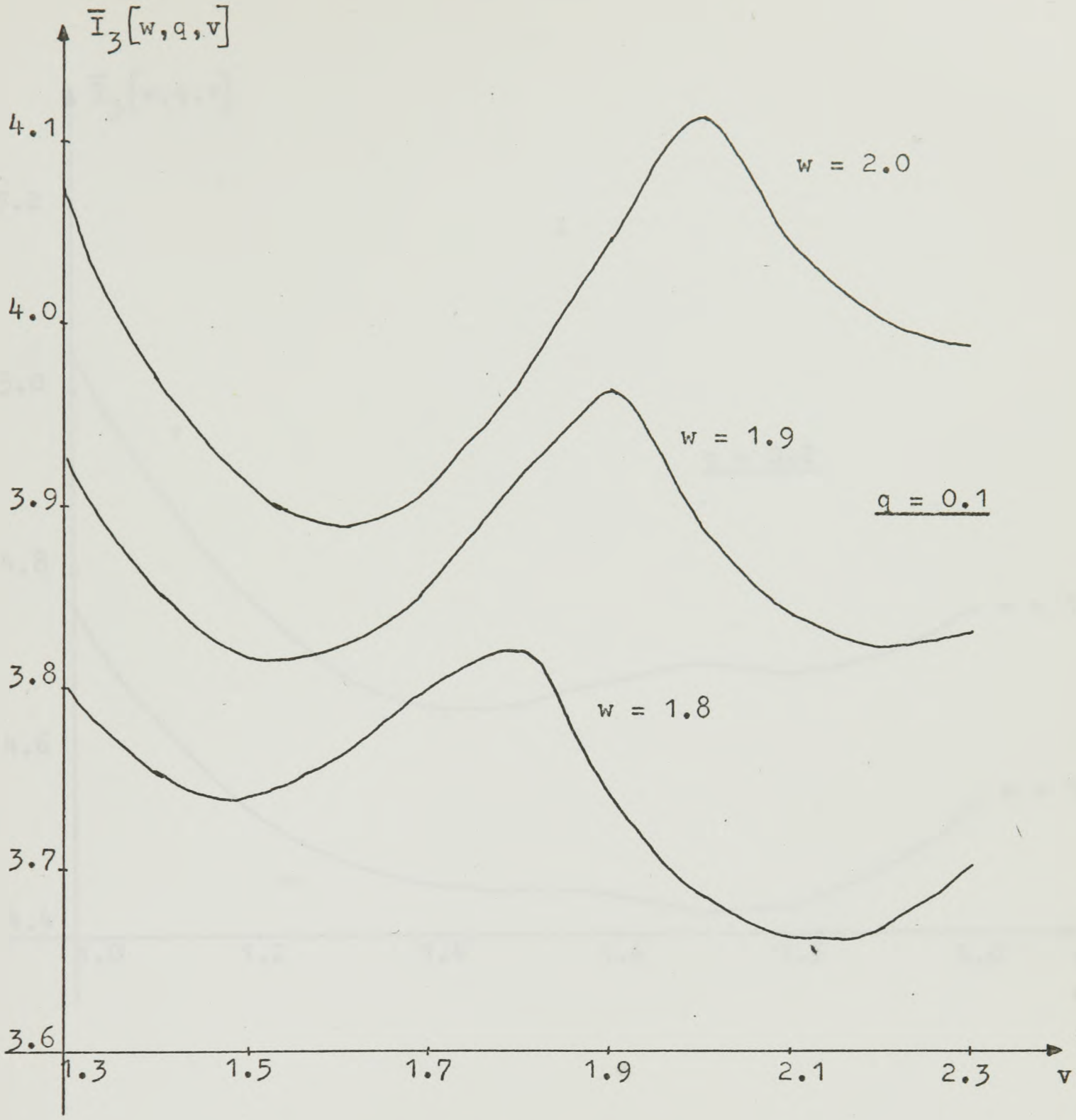


Figure 3.3.2.8

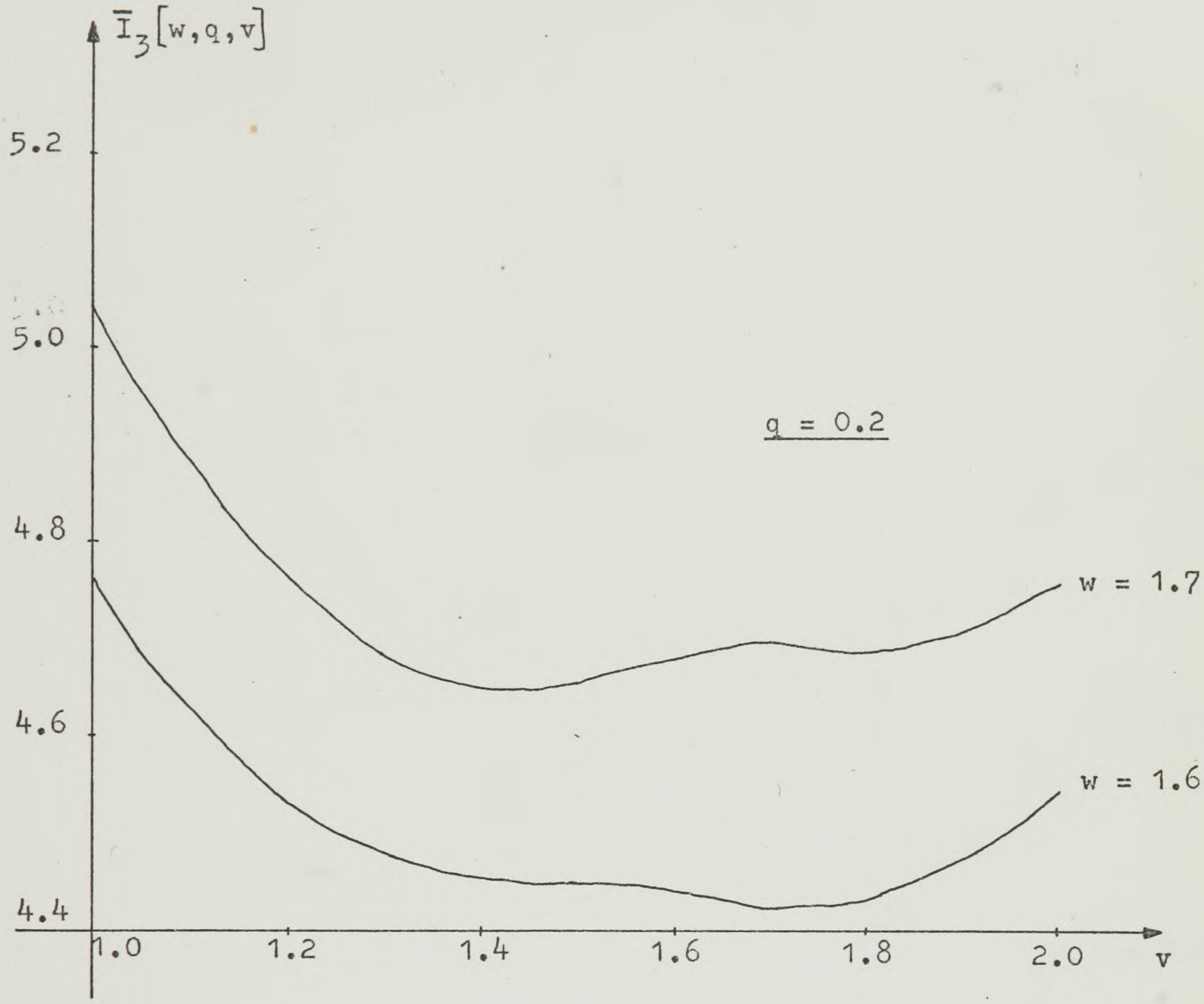


Figure 3.3.2.9

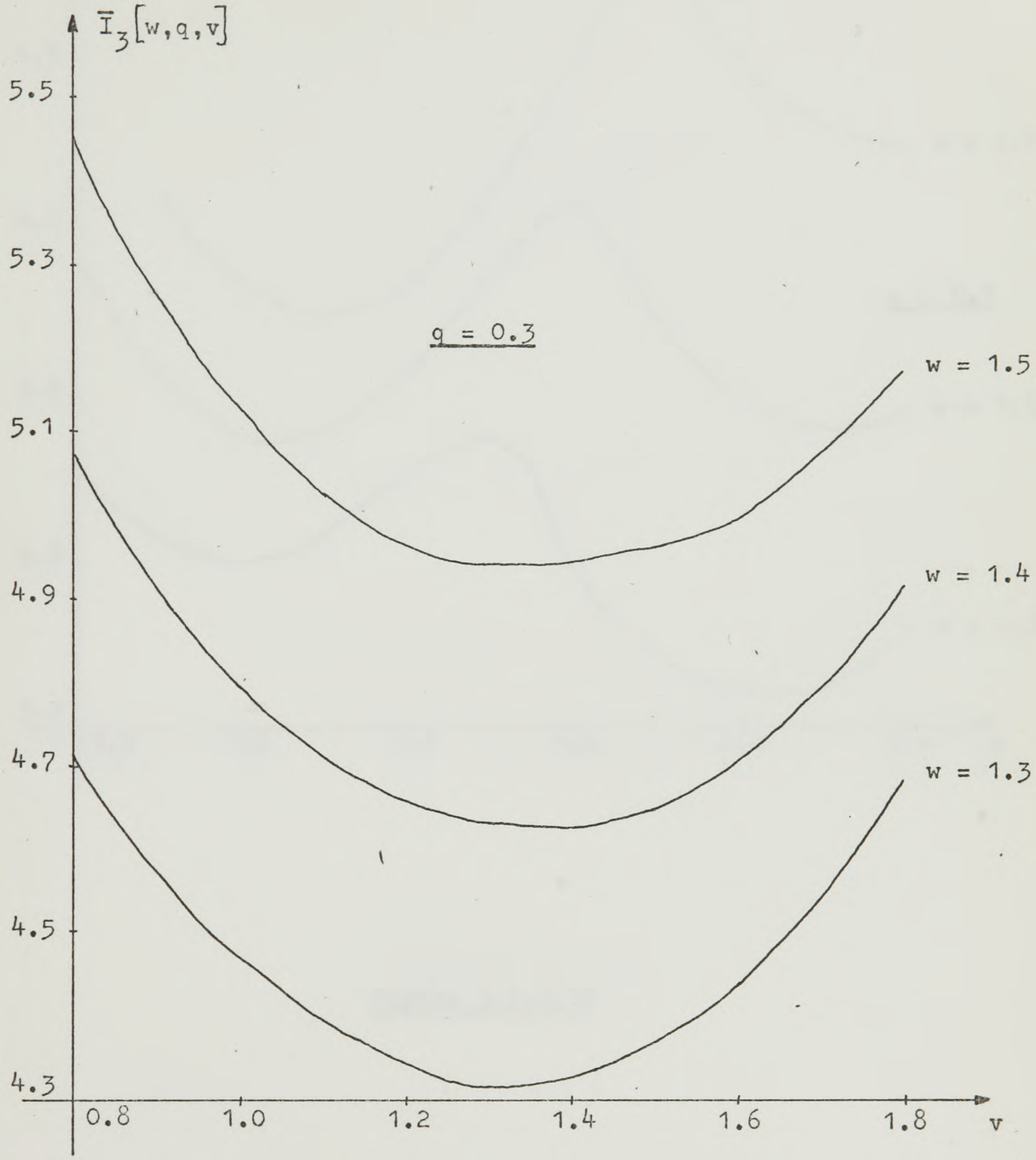


Figure 3.3.2.10

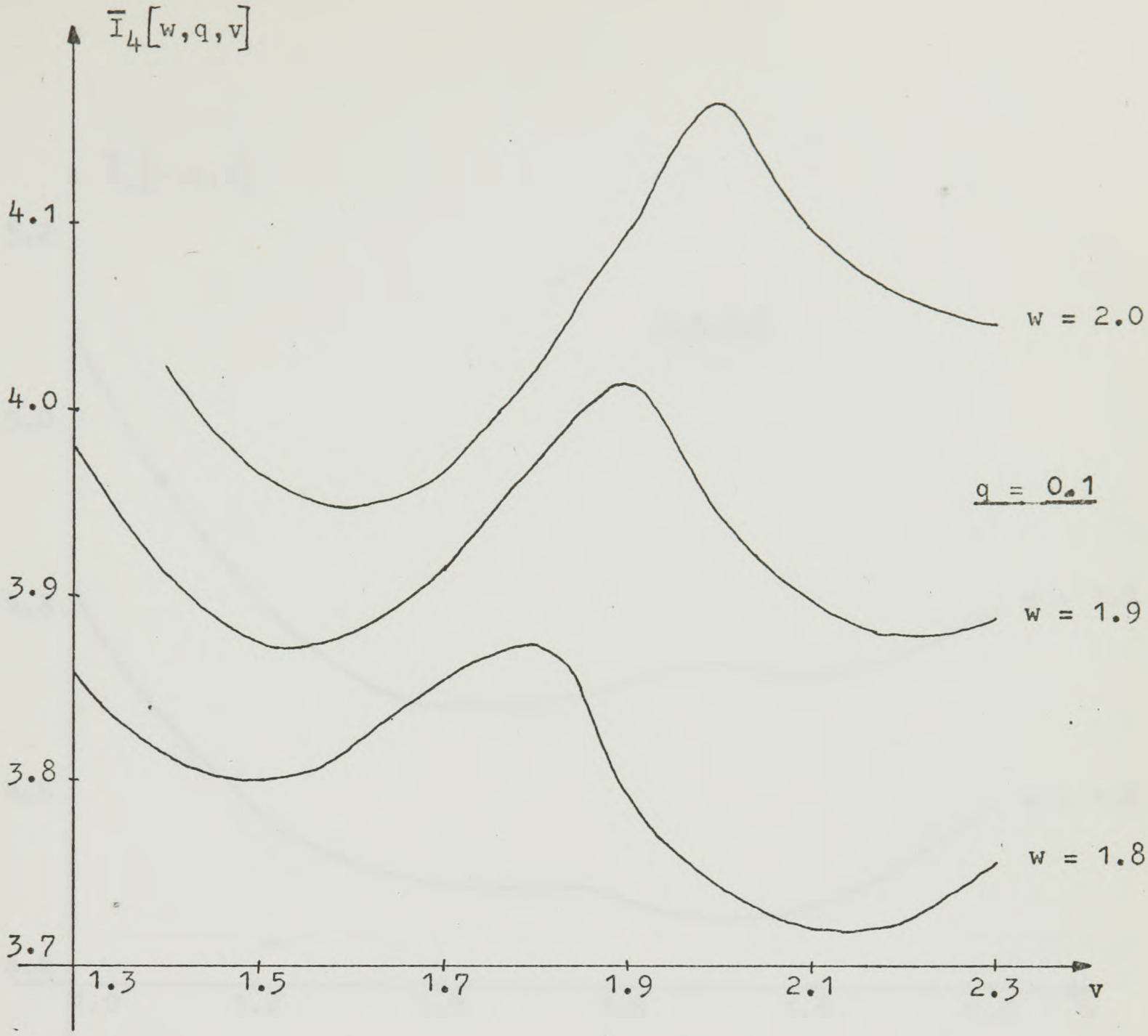


Figure 3.3.2.11

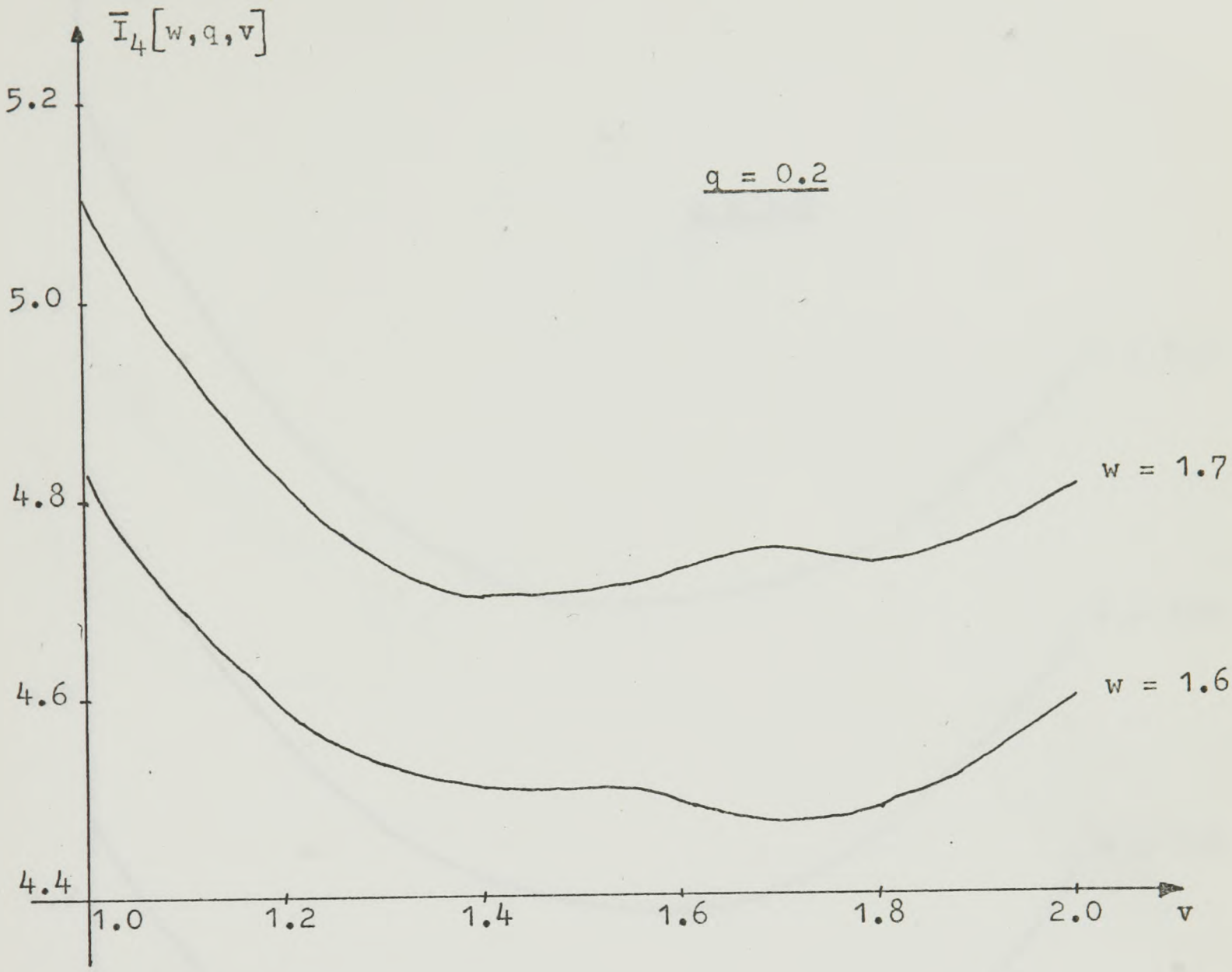


Figure 3.3.2.12

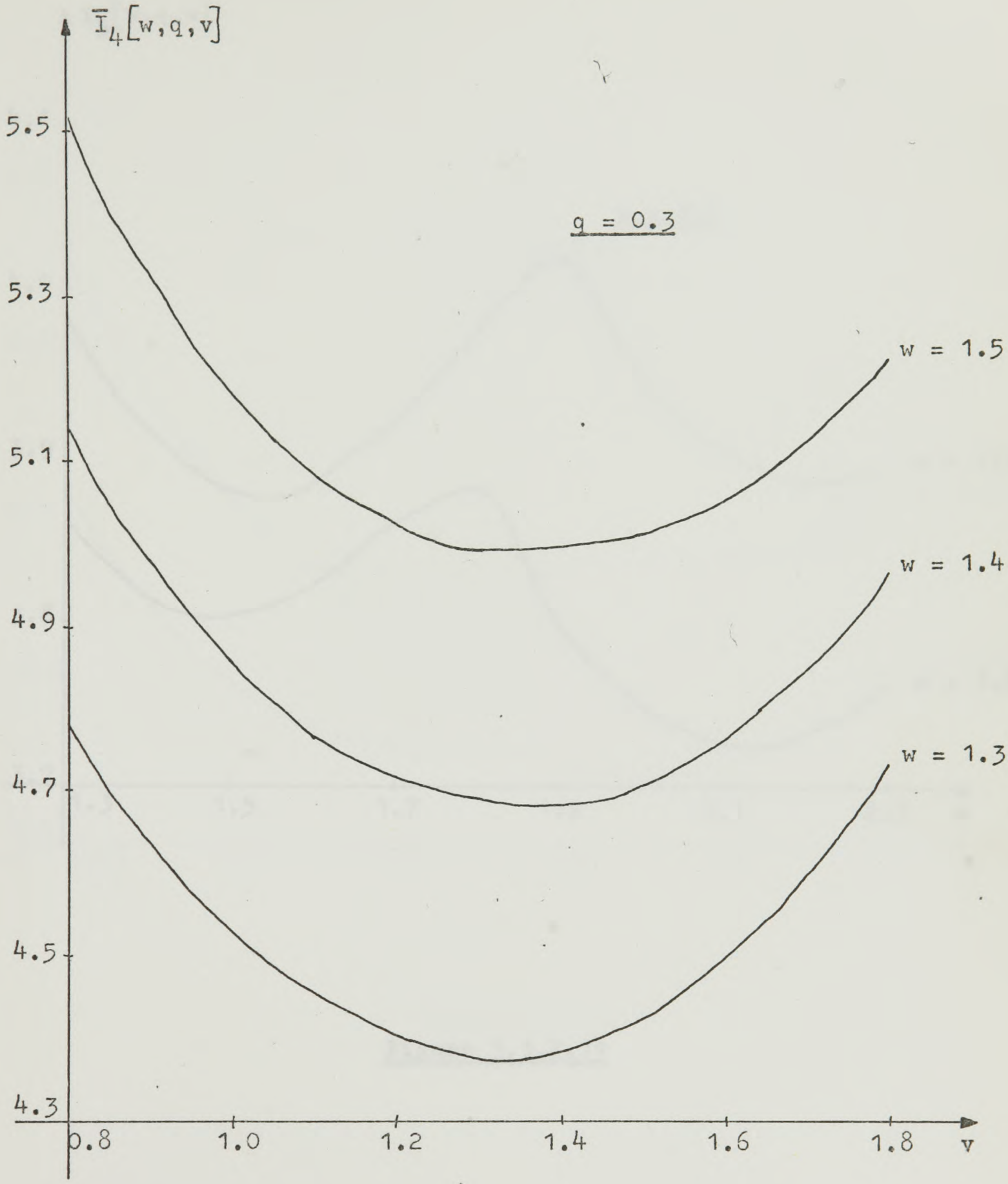


Figure 3.3.2.13

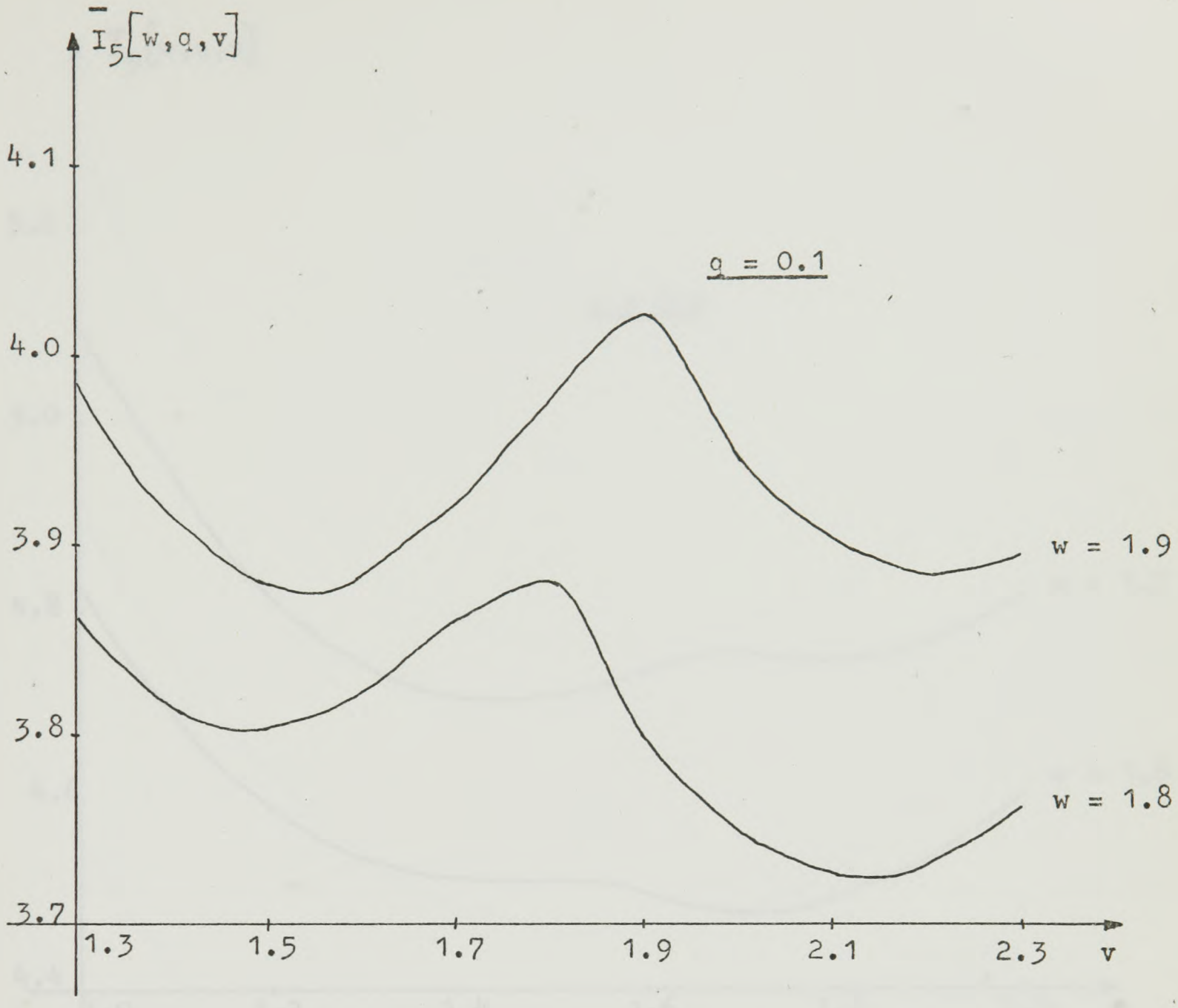


Figure 3.3.2.14

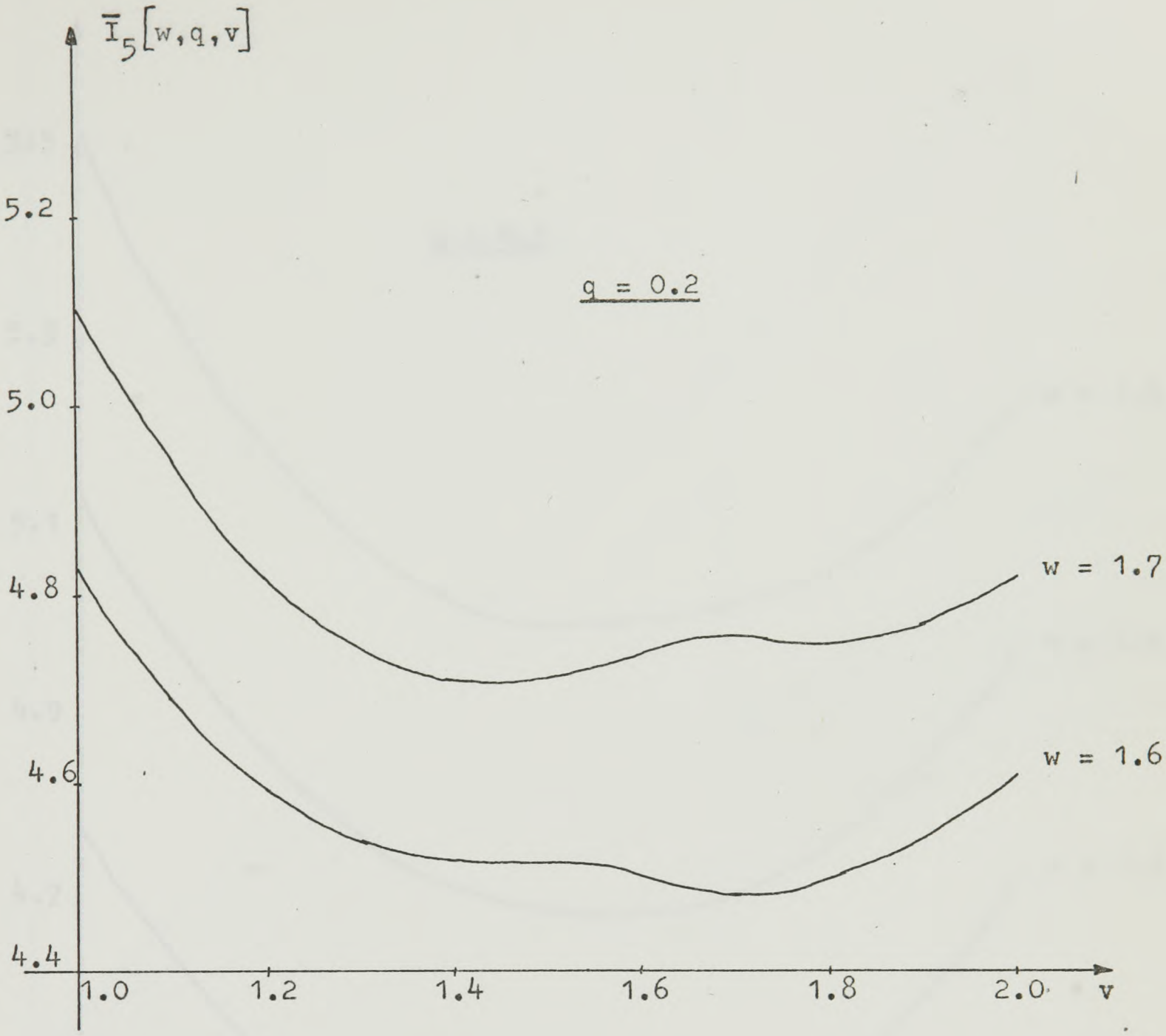


Figure 3.3.2.15

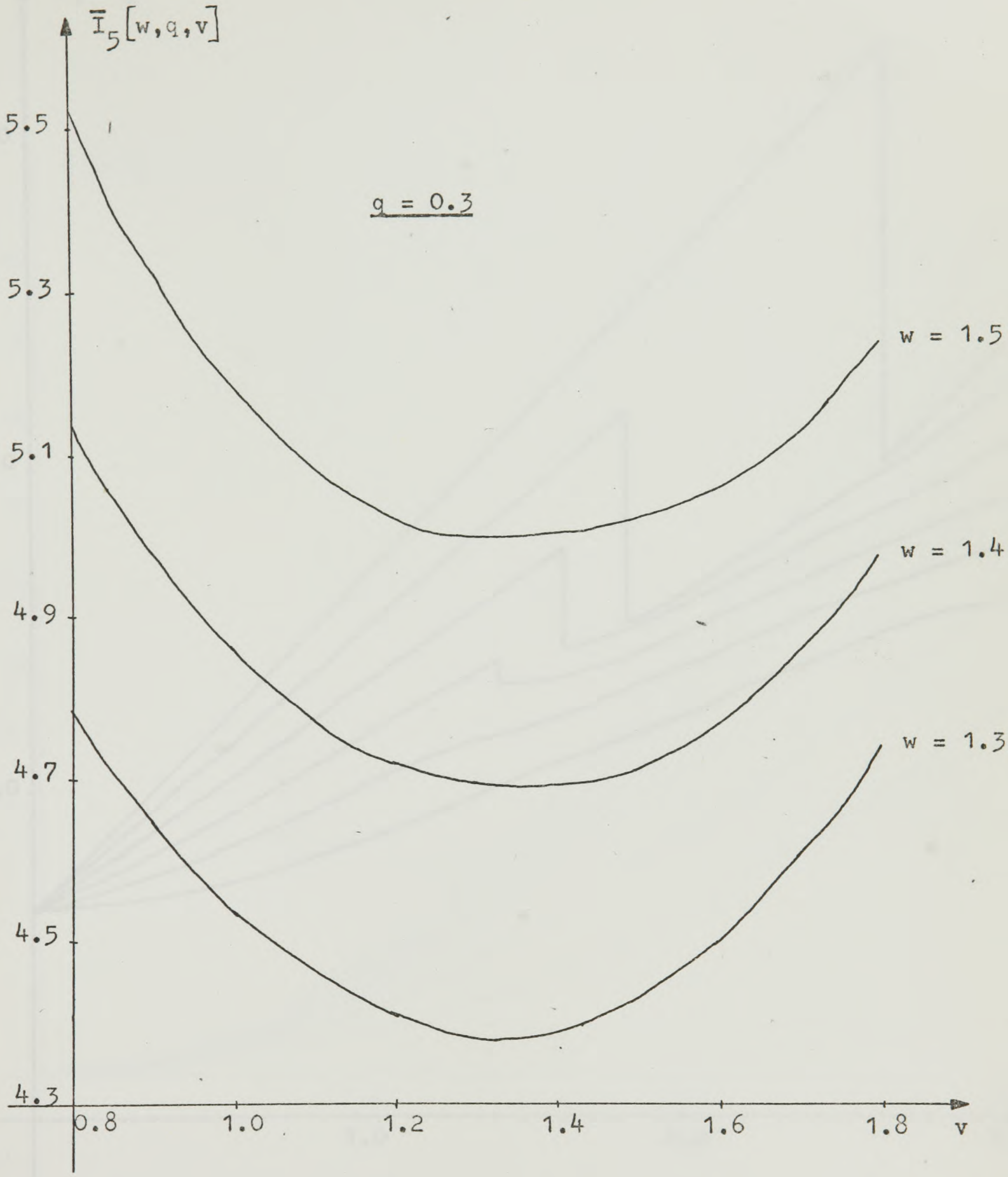


Figure 3.3.2.16

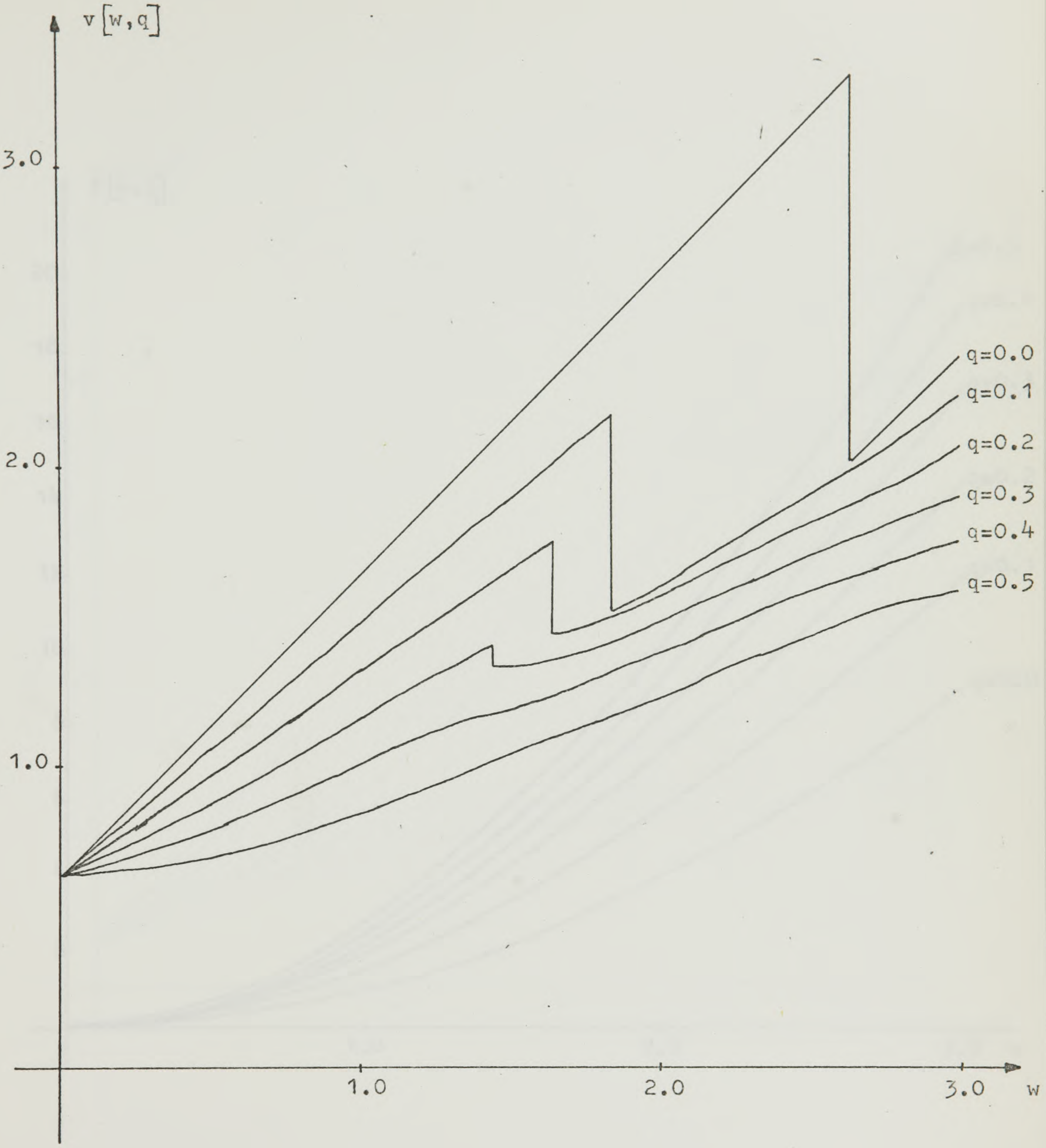


Figure 3.3.2.17

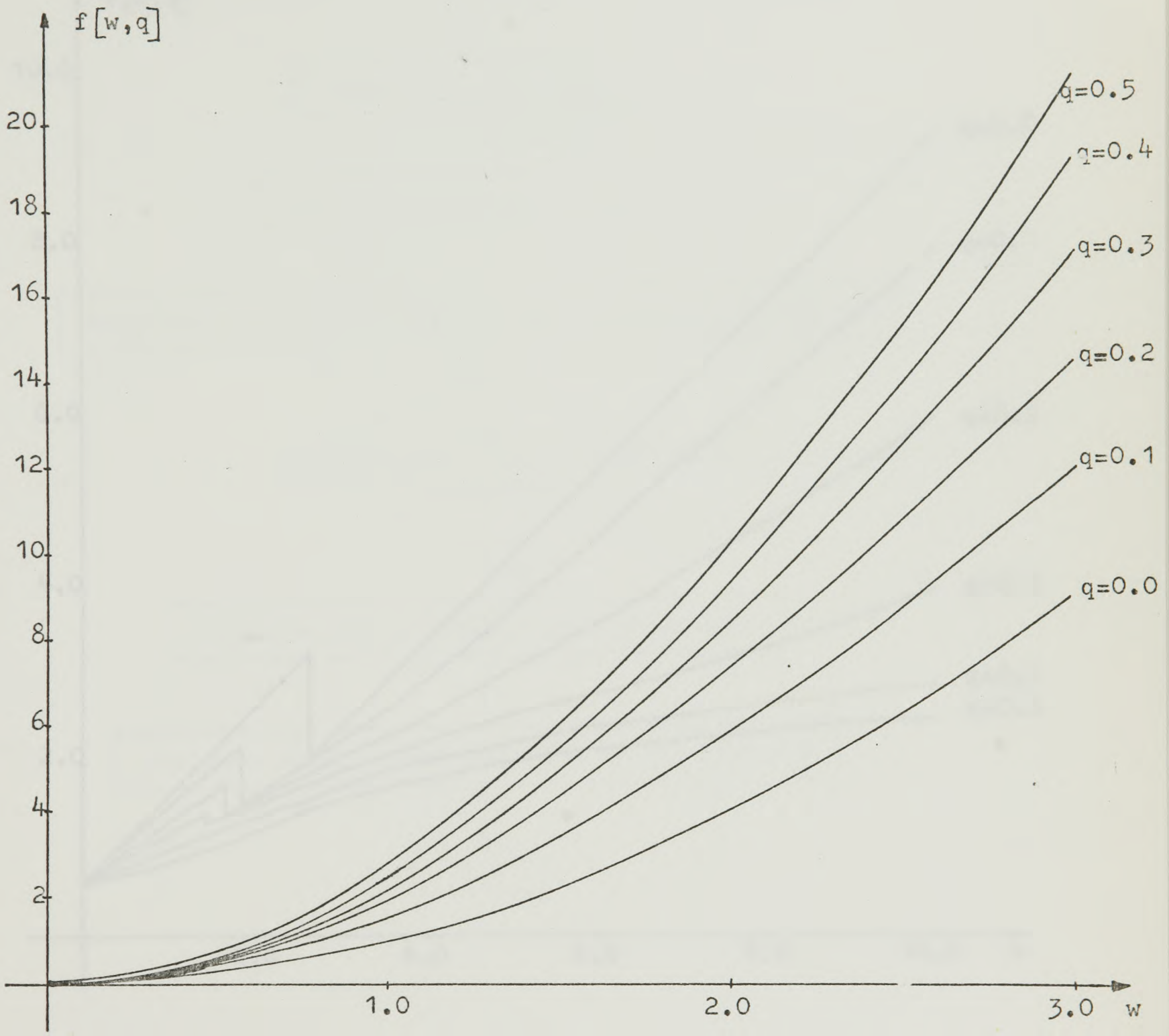


Figure 3.3.2.18

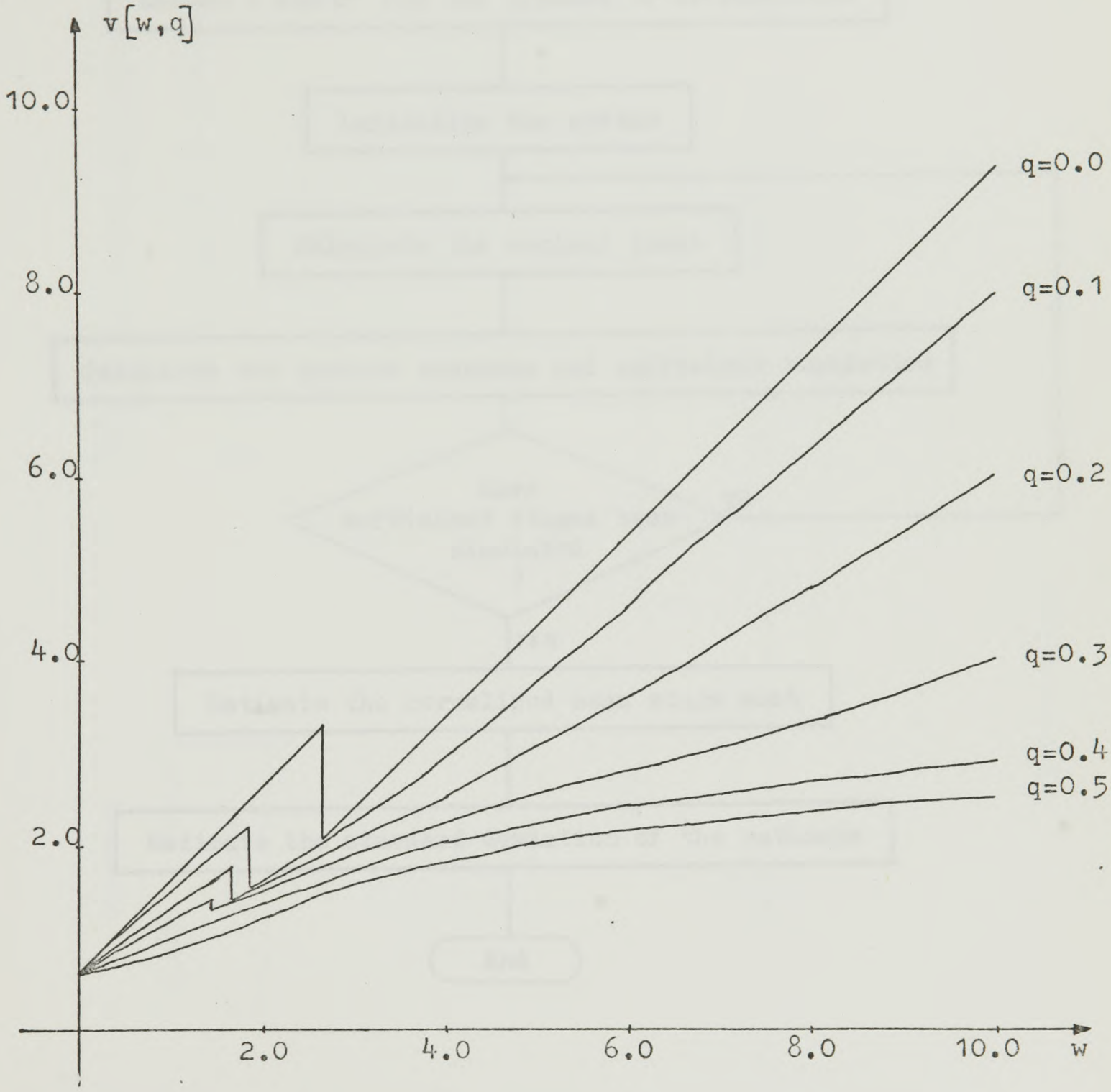


Figure 3.3.2.19

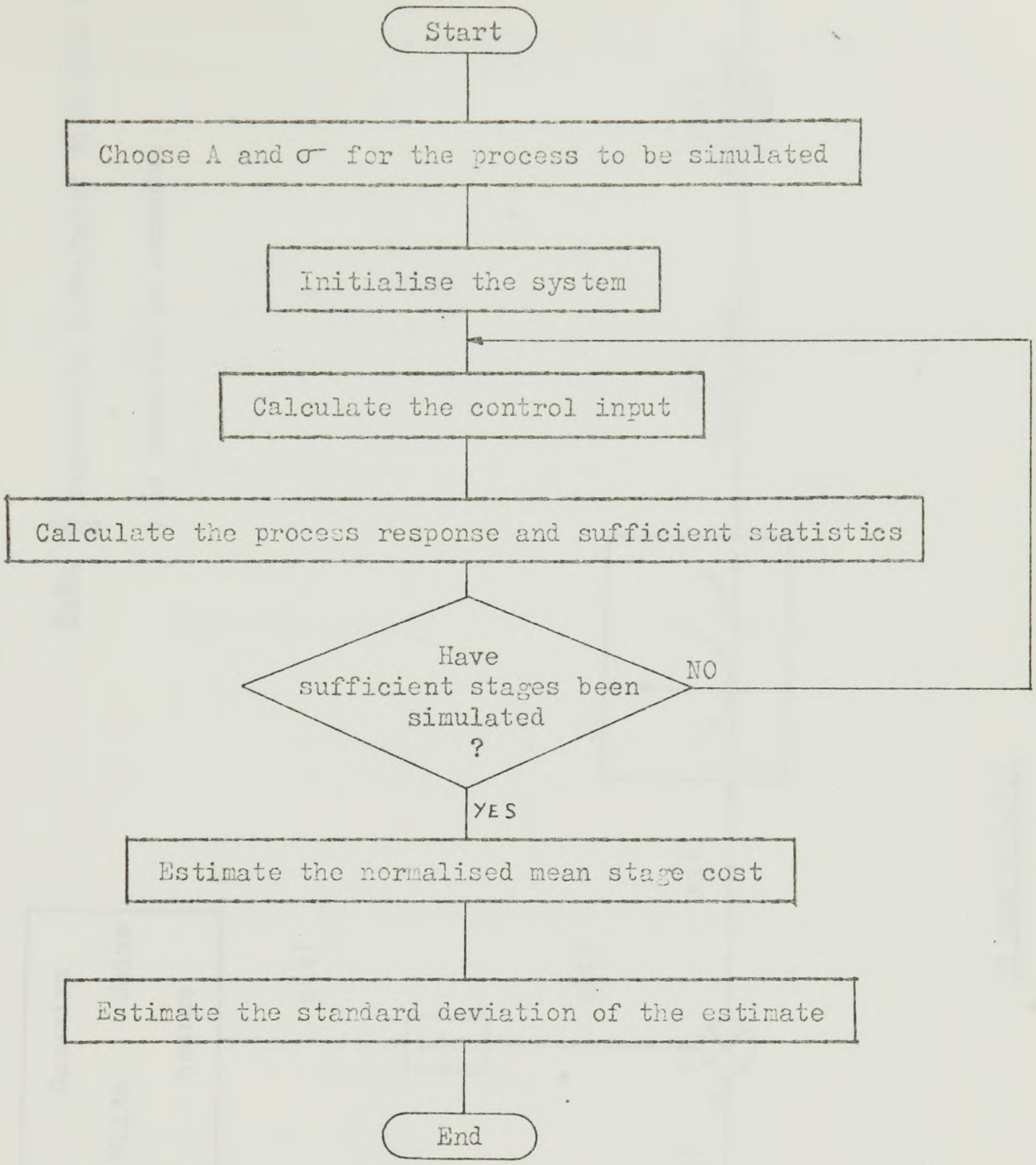
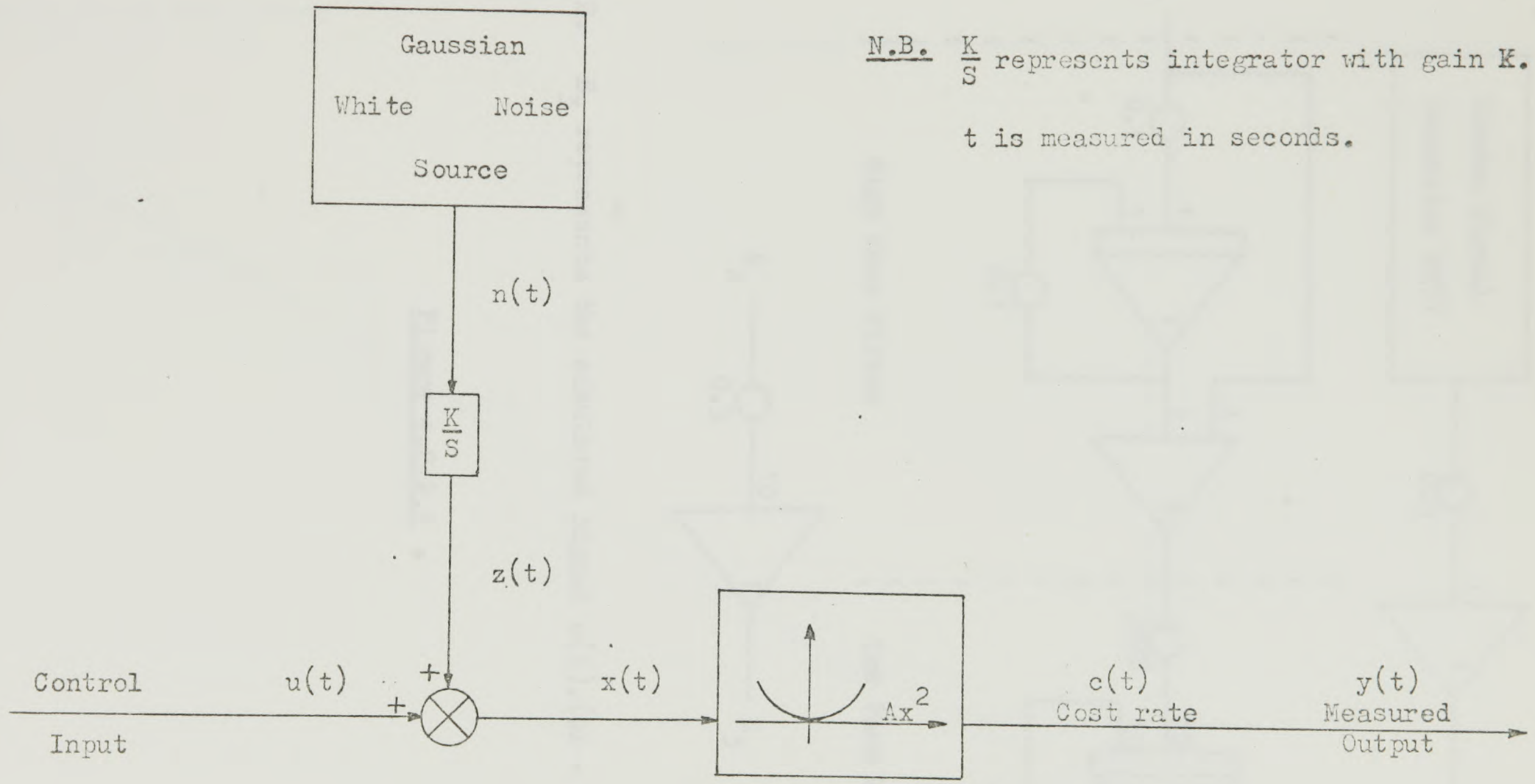

Start
Choose A and σ for the process to be simulated
Initialise the system
Calculate the control input
Calculate the process response and sufficient statistics
Have sufficient stages been simulated ?
NO
YES
Estimate the normalised mean stage cost
Estimate the standard deviation of the estimate
End

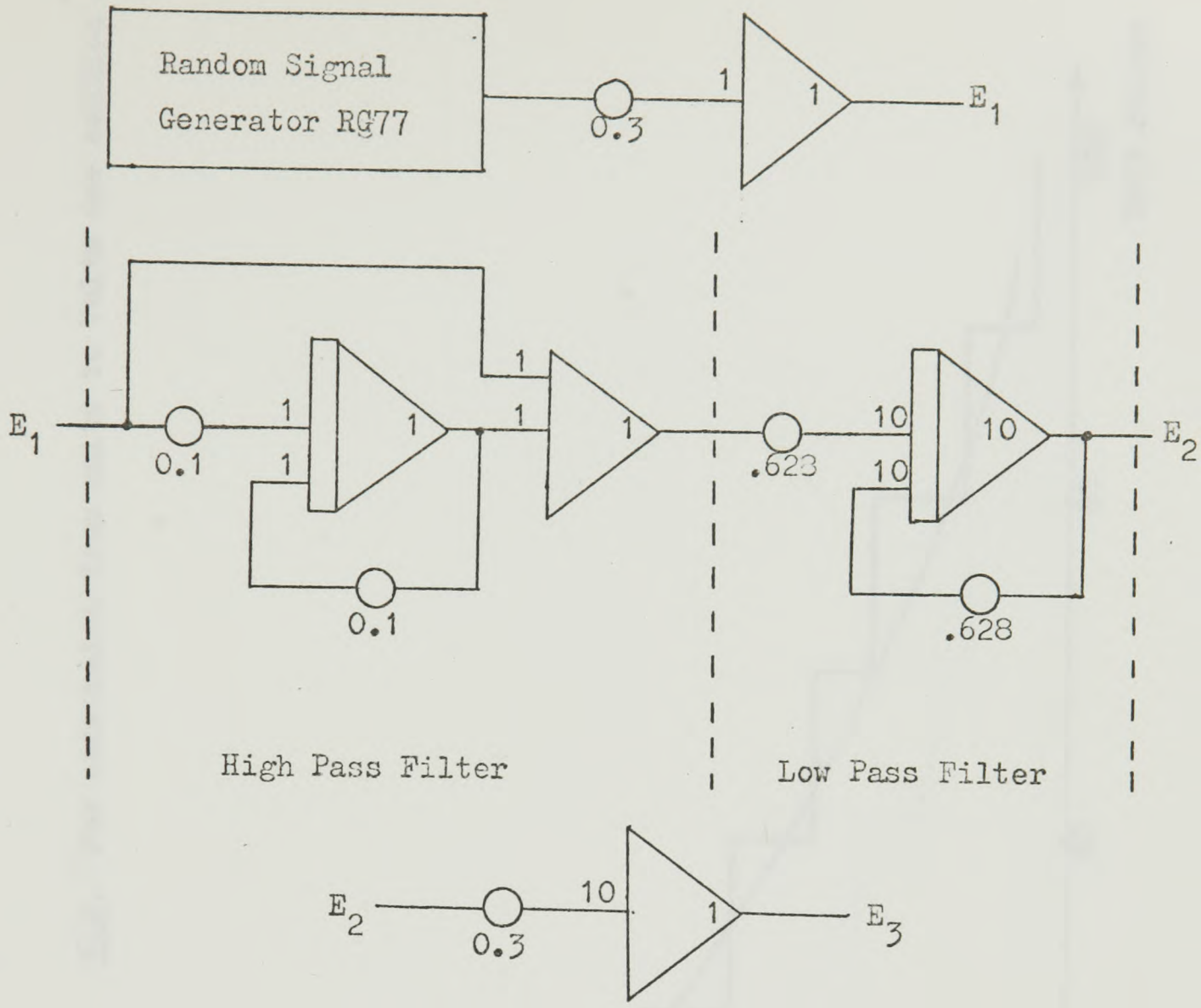
Figure 4.1.1



N.B. $\frac{K}{S}$ represents integrator with gain K.

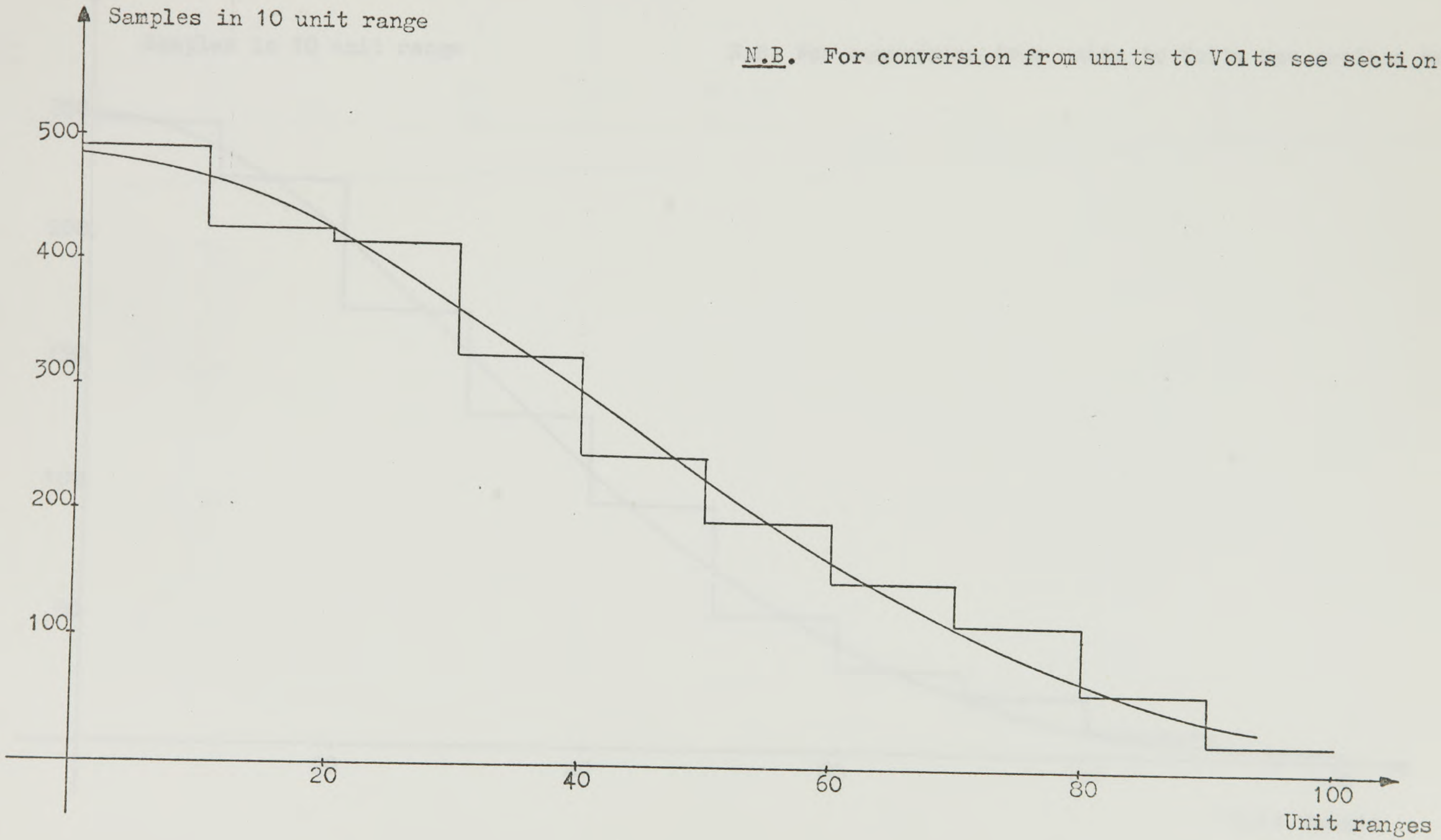
t is measured in seconds.

Figure 4.2.1.1



N.B. E_3 represents the simulated signal n(t).(or - n(t))

Figure 4.2.2.1



N.B. For conversion from units to Volts see section A4.4

Figure 4.2.2.2

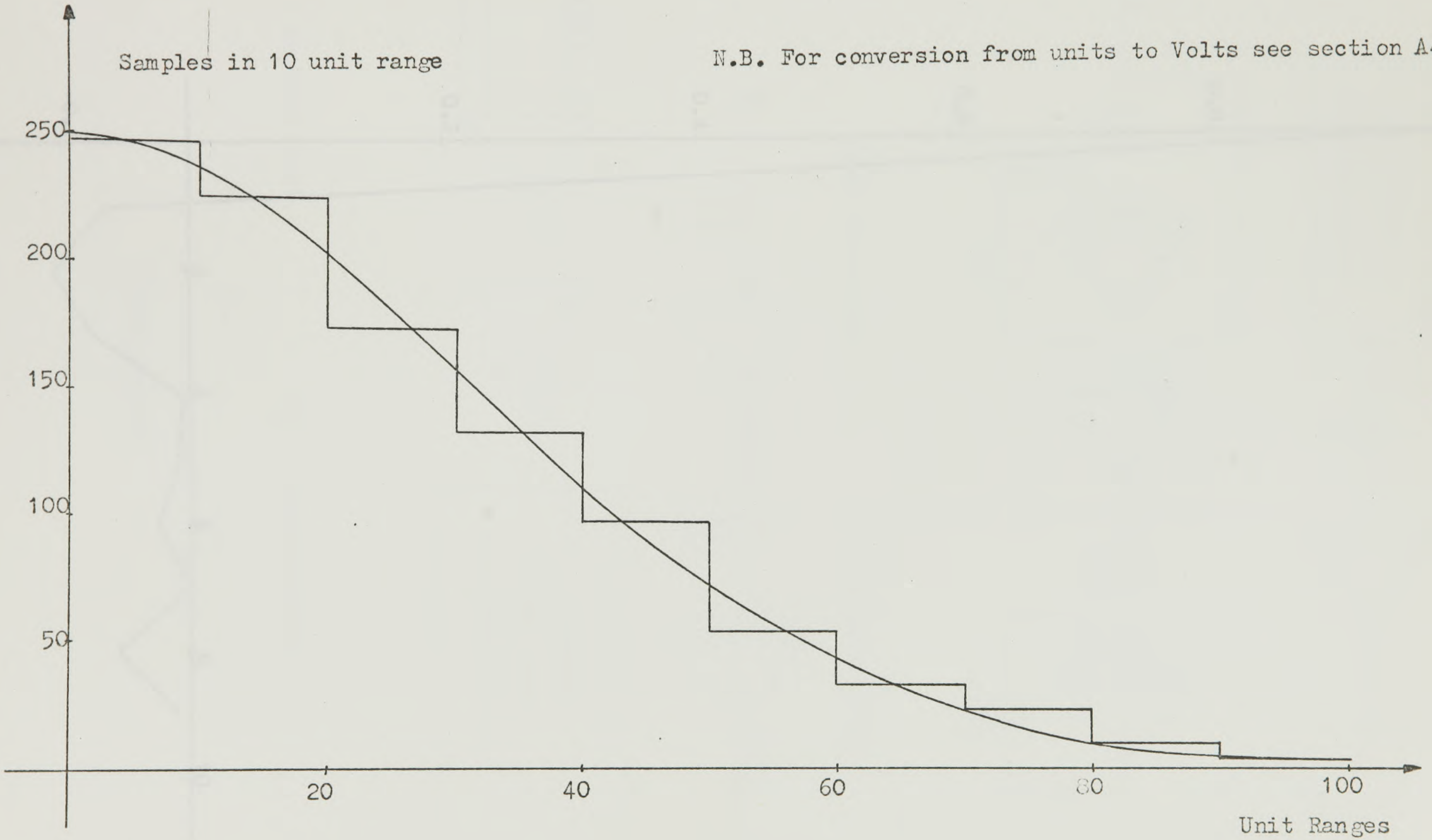
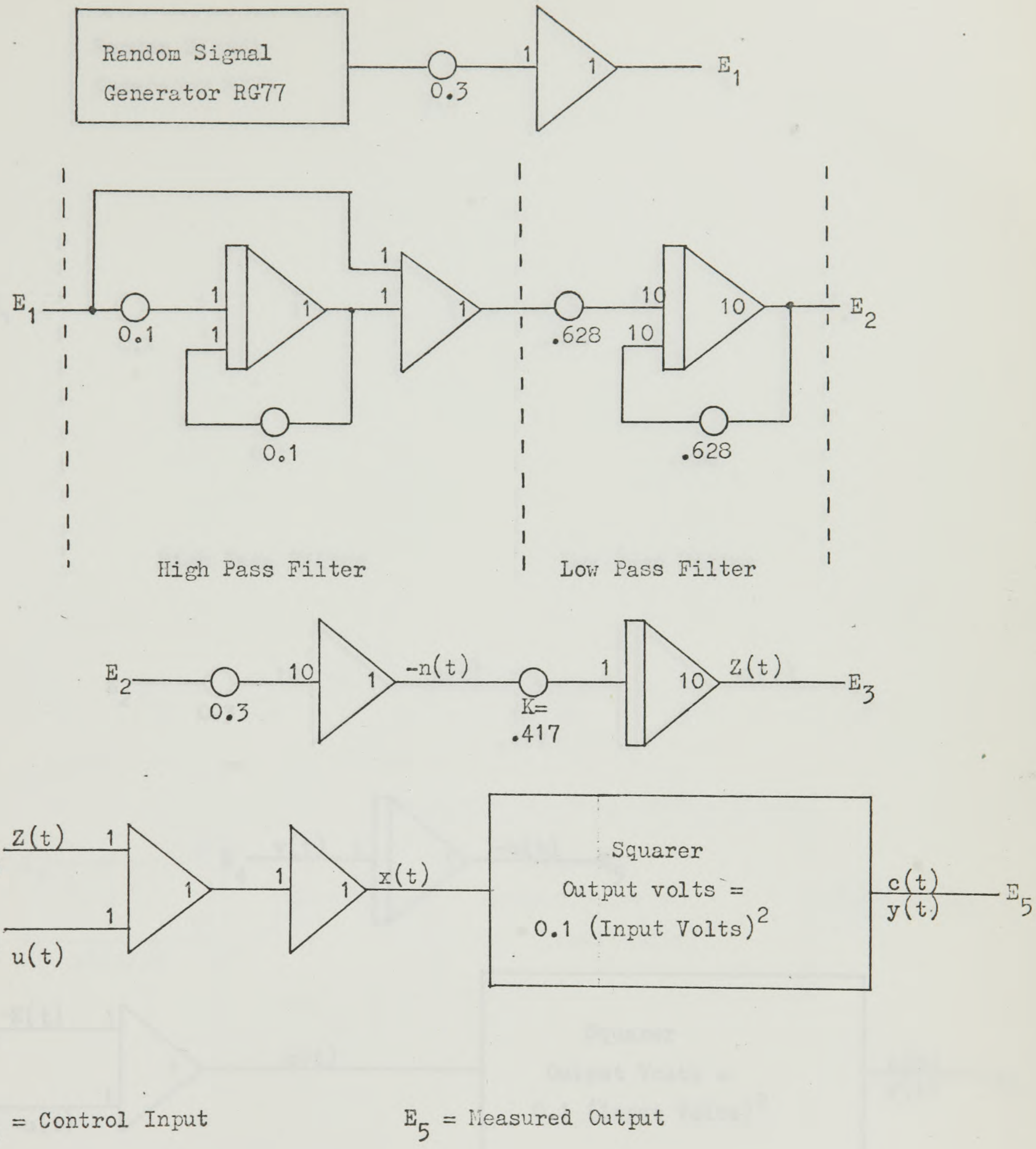

Samples in 10 unit range
N.B. For conversion from units to Volts see section A4.5
250
200
150
100
50
20
40
60
80
100
Unit Ranges

Figure 4.2.2.3



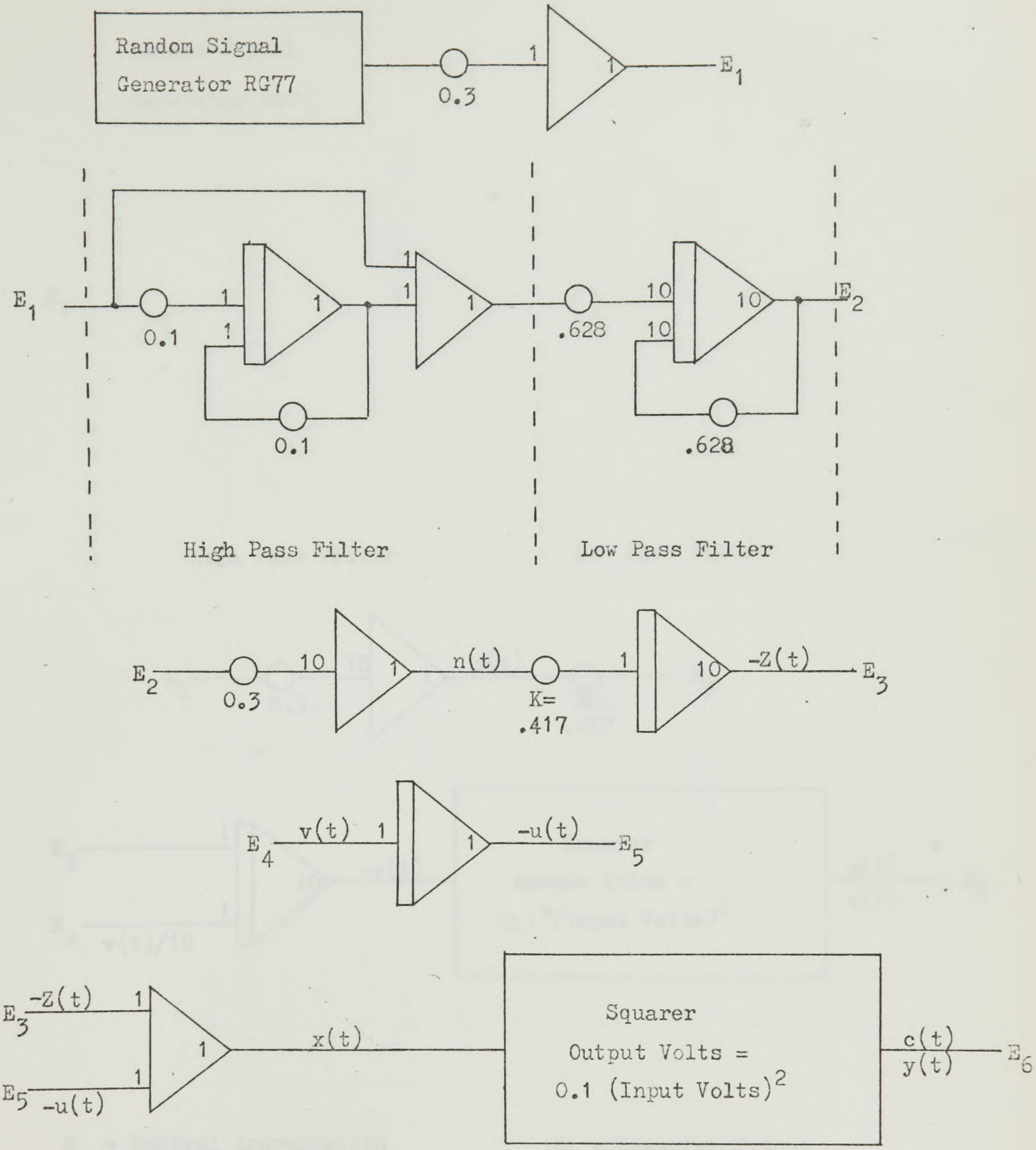
Figure 4.2.2.4



E_4 = Control Input

E_5 = Measured Output

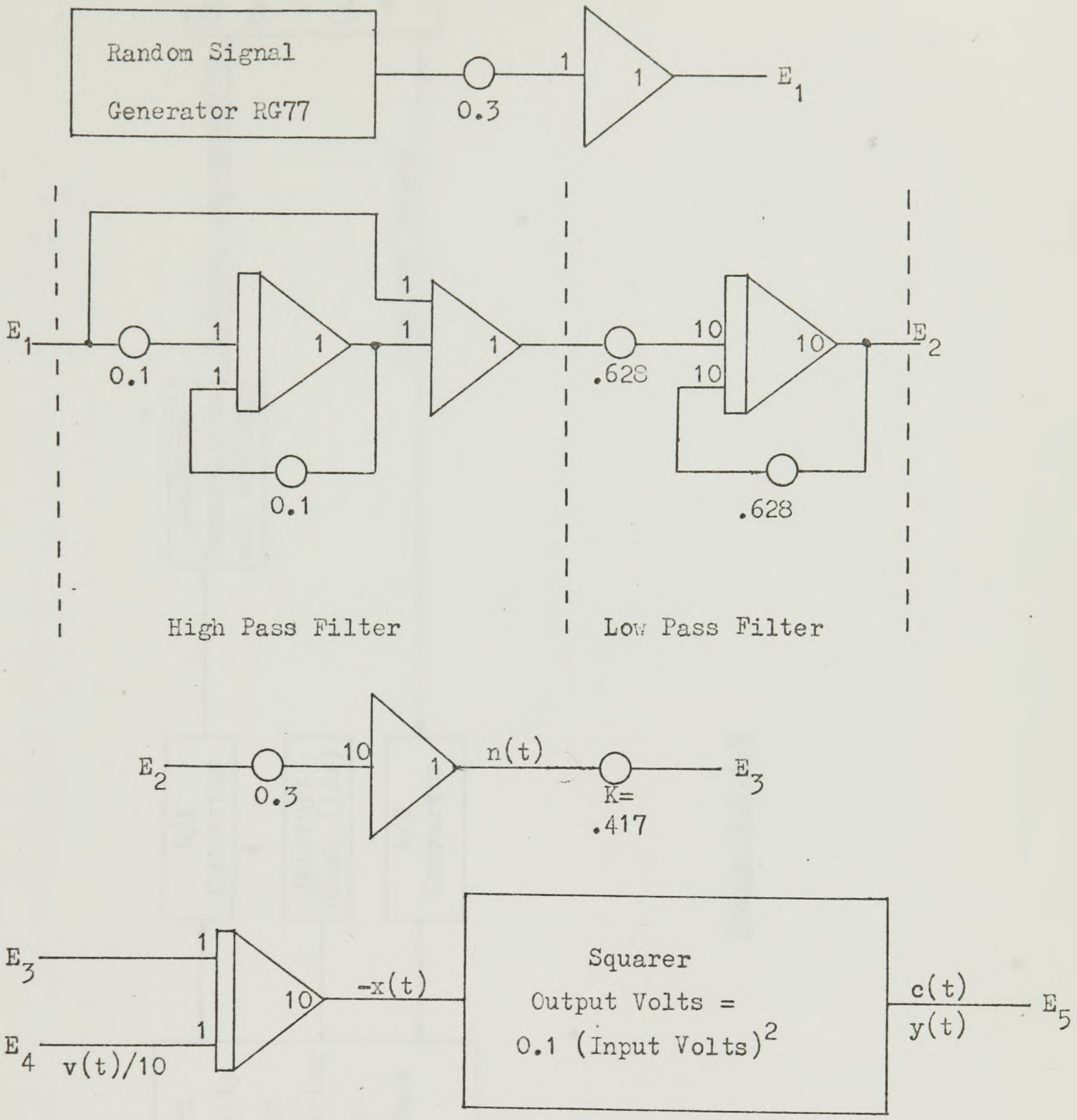
Figure 4.2.2.5



E_4 = Control Increment

E_6 = Measured Output

Figure 4.2.3.1



E_4 = Control Increment/10

E_5 = Measured Output

Figure 4.2.3.2

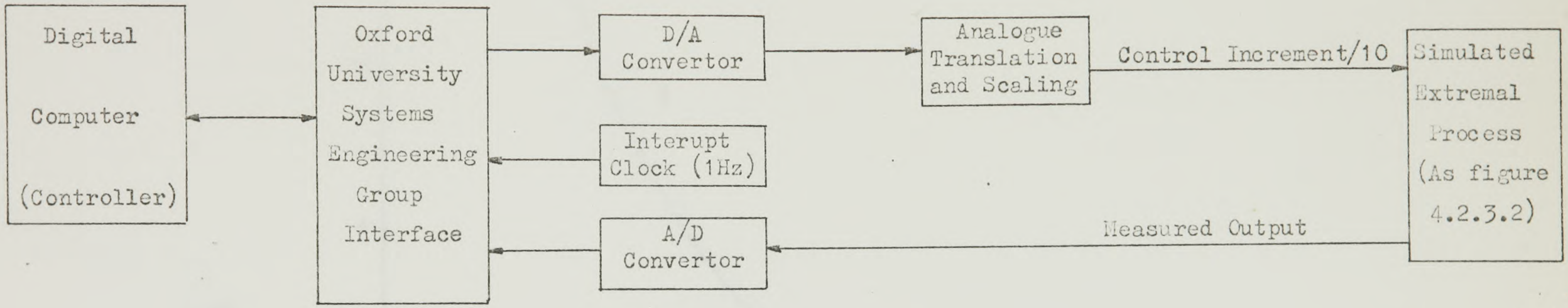

Digital
Computer
(Controller)
Oxford
University
Systems
Engineering
Group
Interface
D/A
Convertor
Interupt
Clock (1Hz)
A/D
Convertor
Analogue
Translation
and Scaling
Control Increment/10
Measured Output
Simulated
Extremal
Process
(As figure
4.2.3.2)

Figure 4.2.3.3

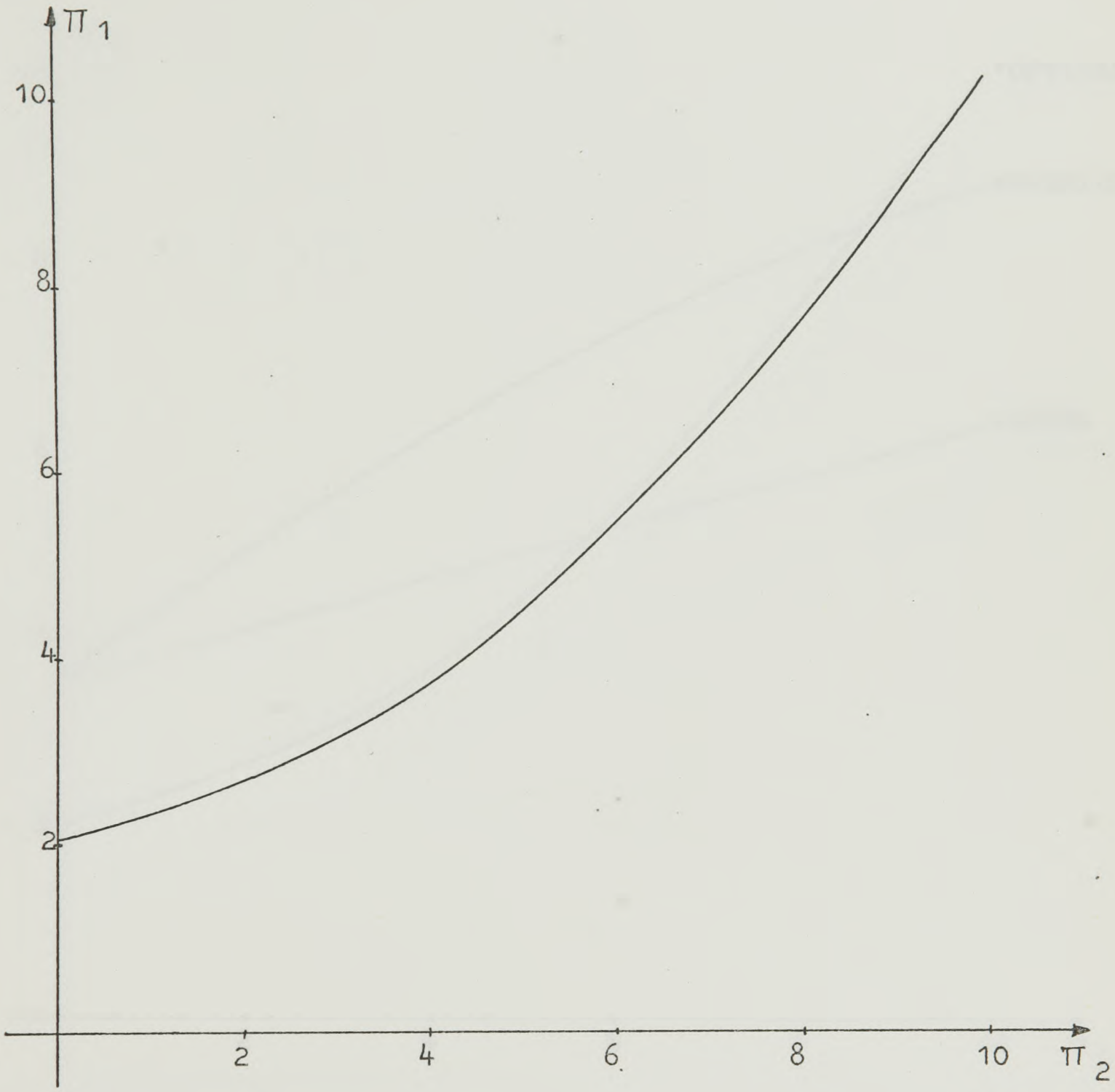


Figure 5.1.1

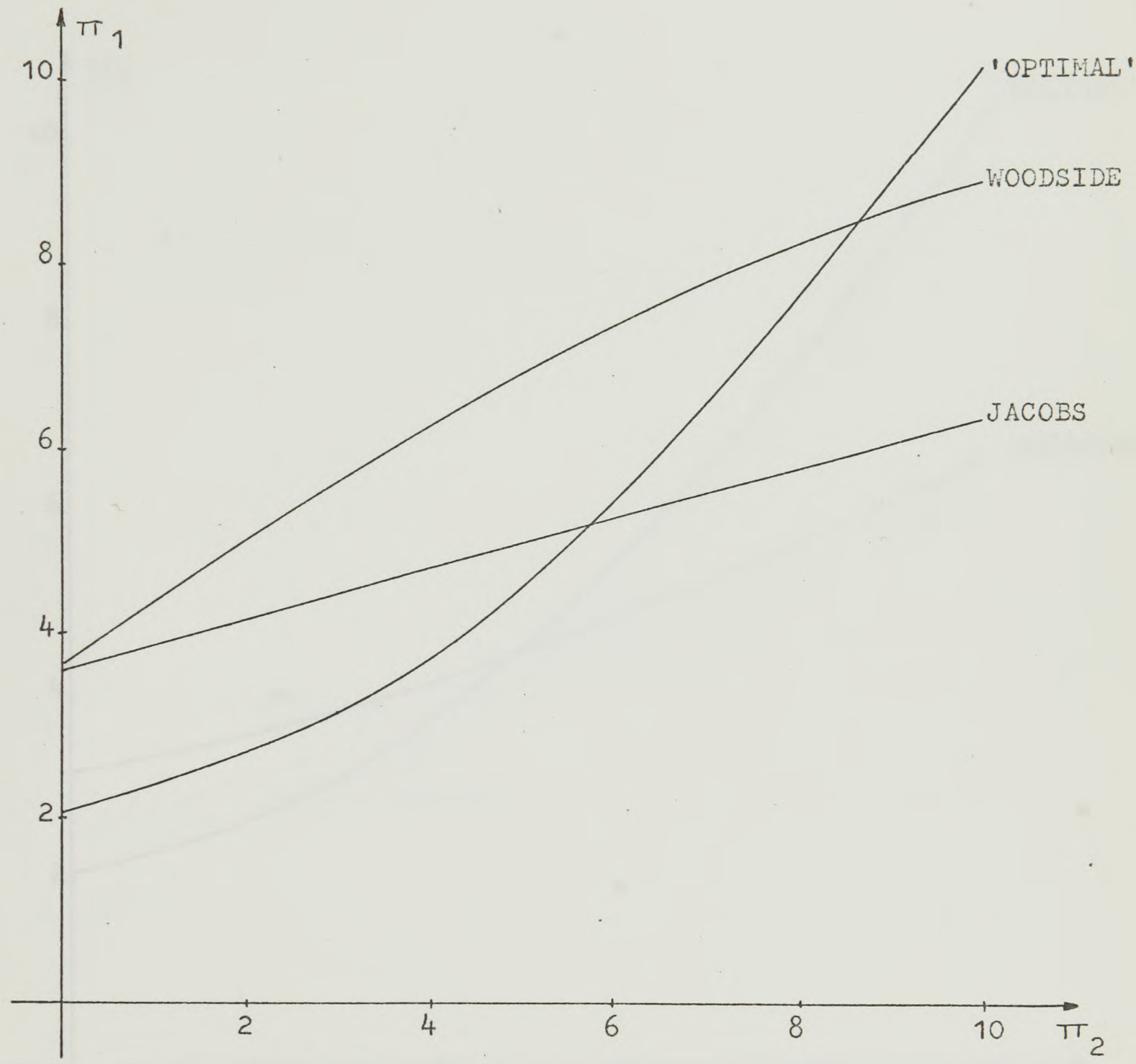


Figure 5.2.1.1

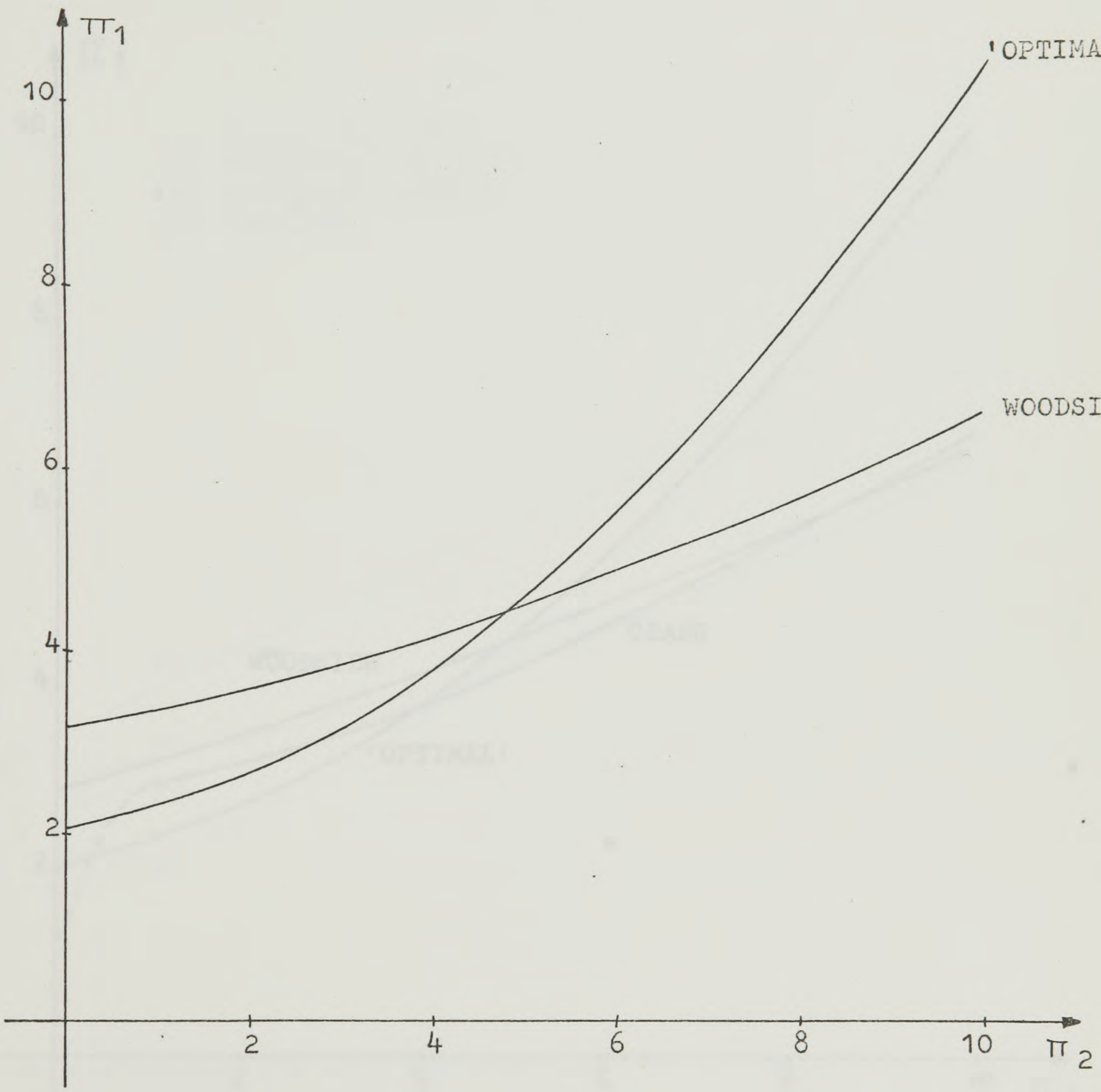


Figure 5.2.2.1

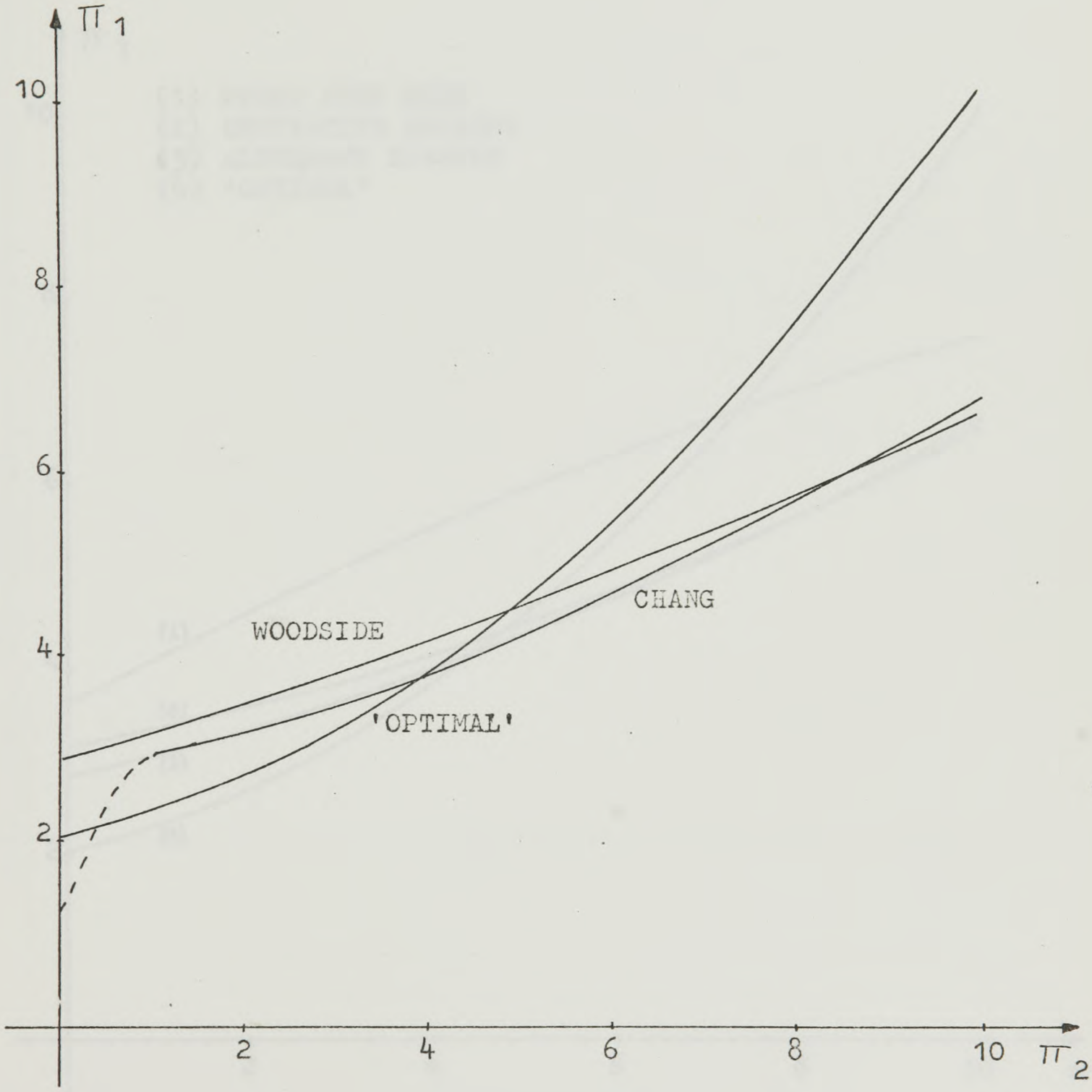


Figure 5.2.3.1

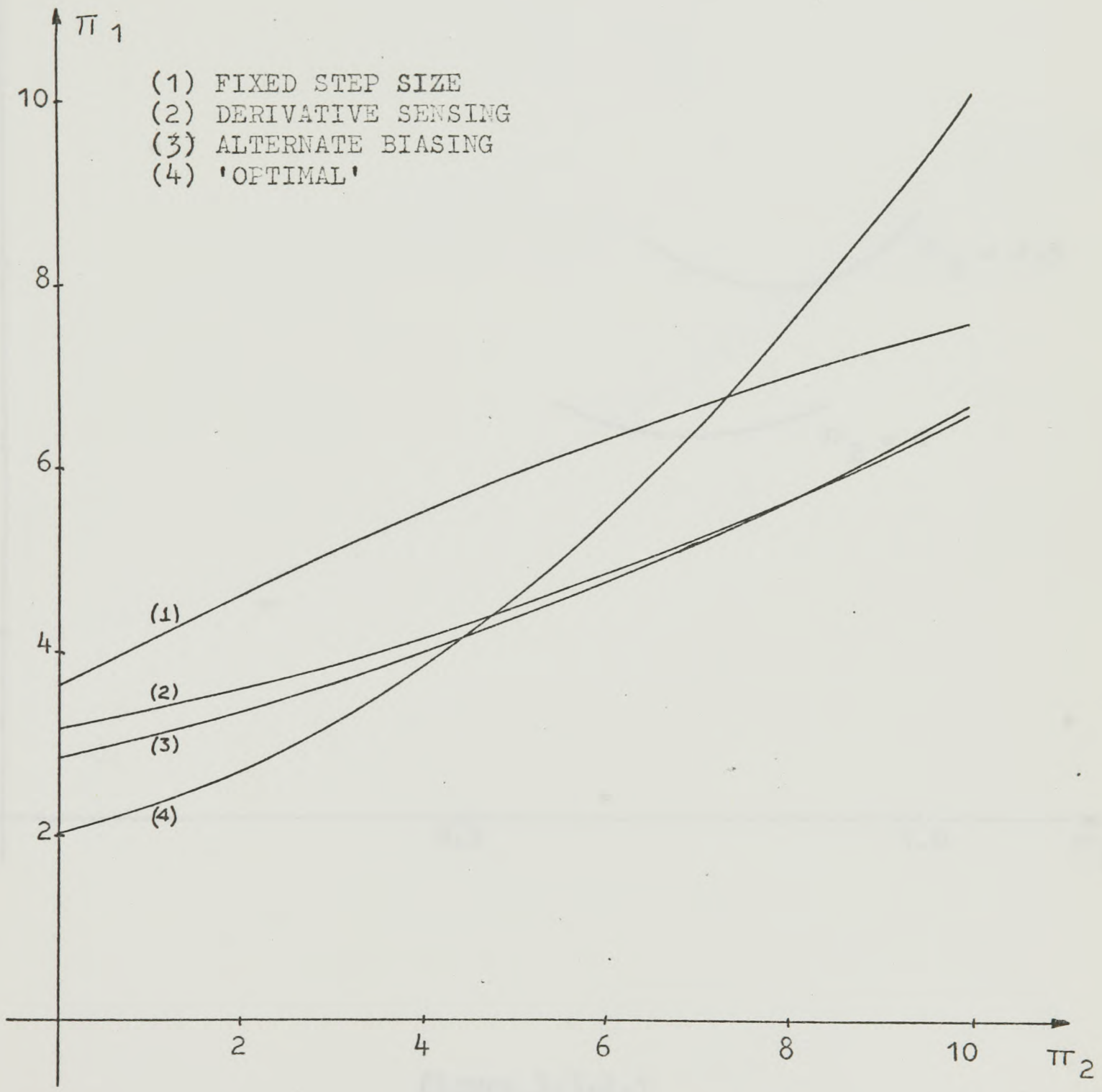

π_1
(1) FIXED STEP SIZE
(2) DERIVATIVE SENSING
(3) ALTERNATE BIASING
(4) 'OPTIMAL'
10
8
6
4
2
(1)
(2)
(3)
(4)
2
4
6
8
10
π_2

Figure 5.2.4.1

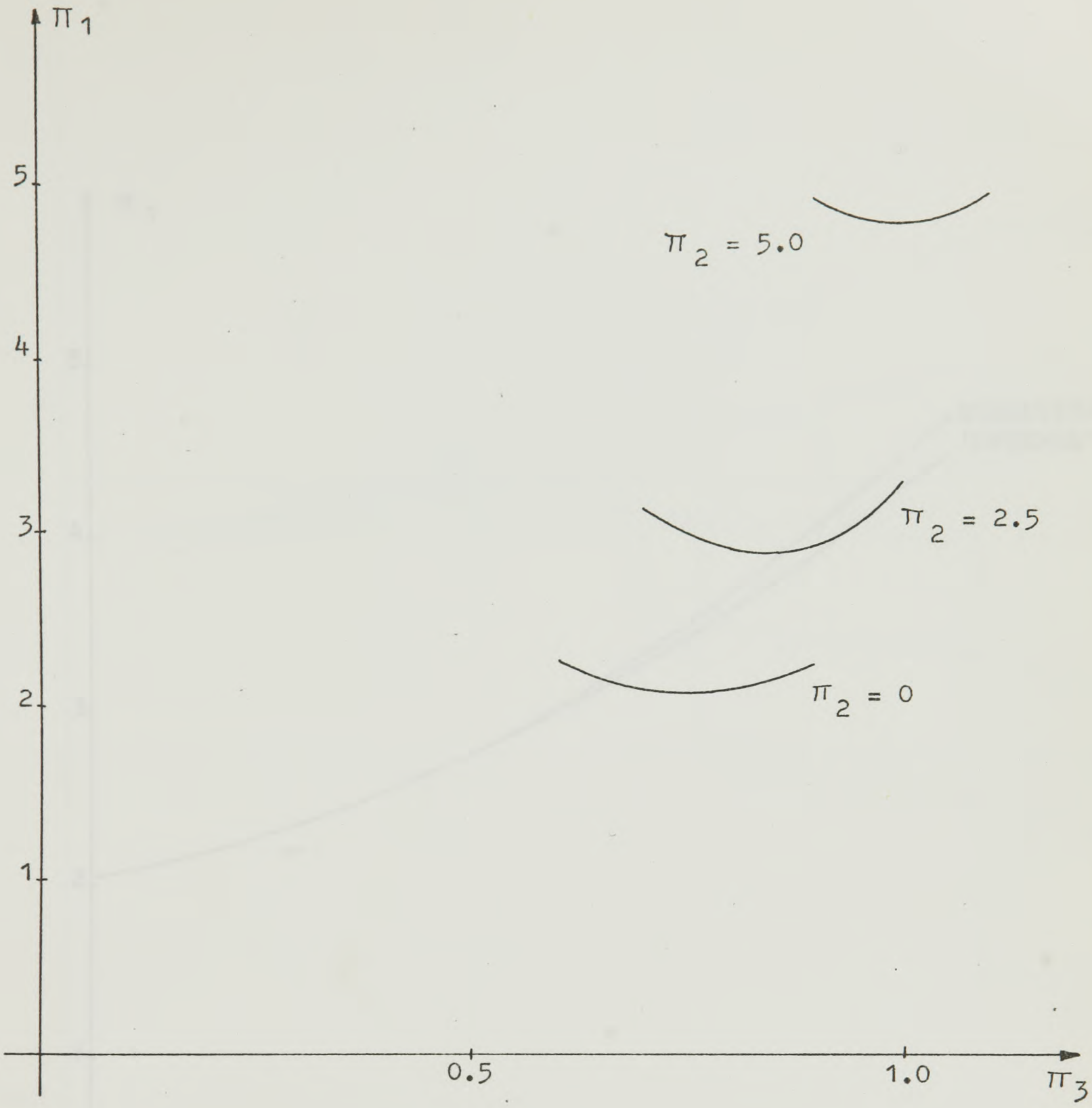


Figure 5.3.2.1

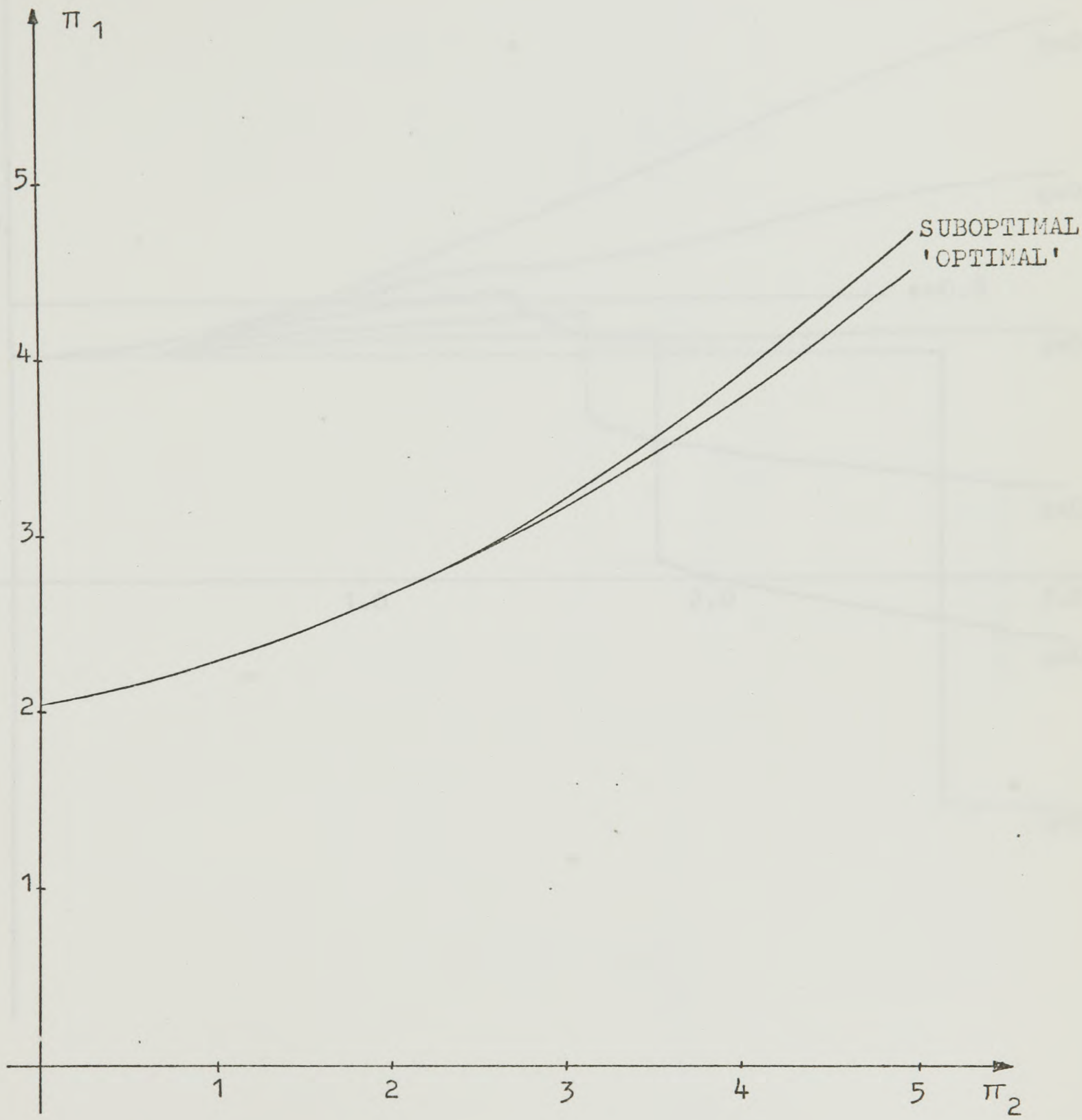


Figure 5.3.2.2

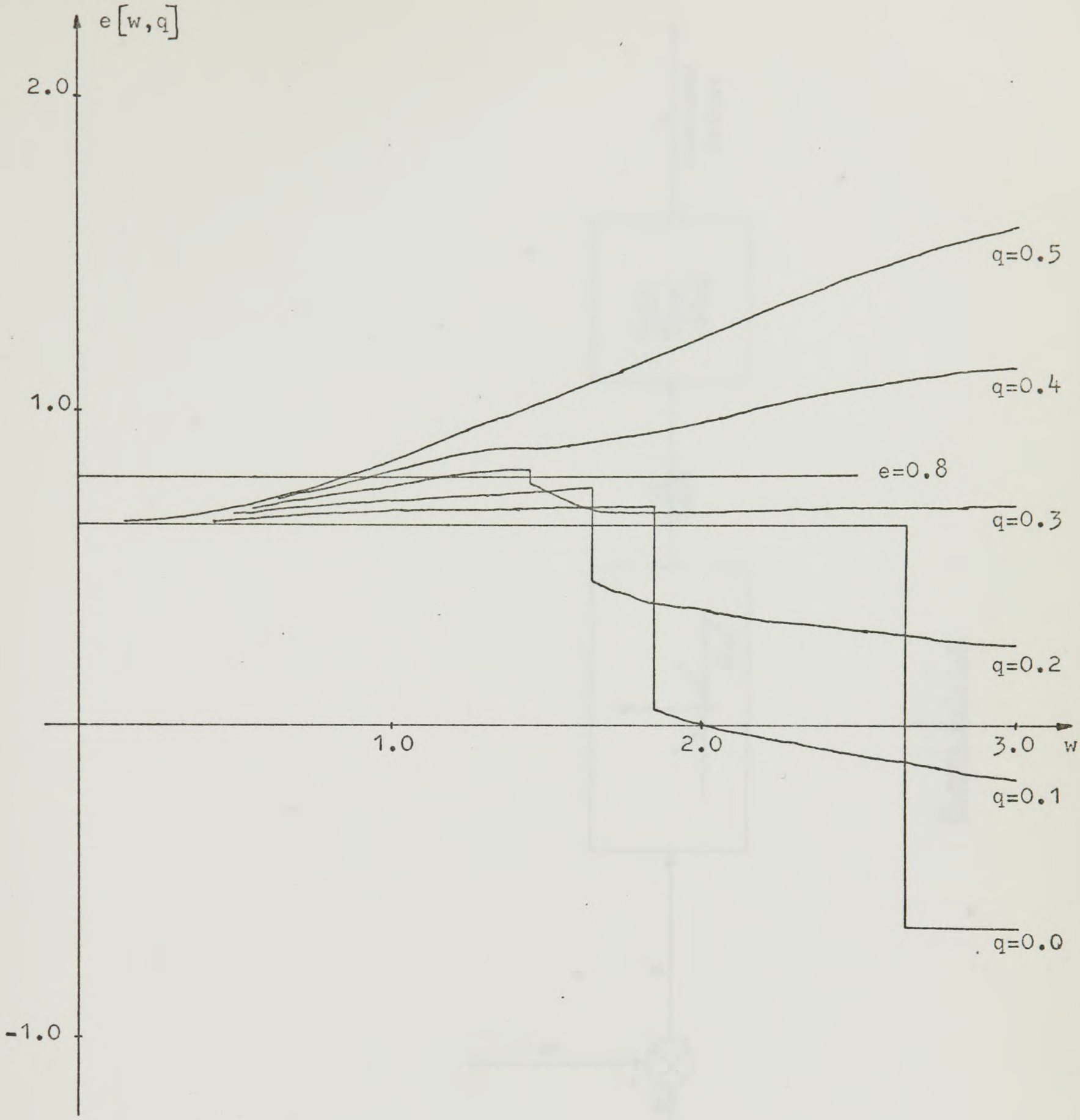


Figure 5.3.2.3

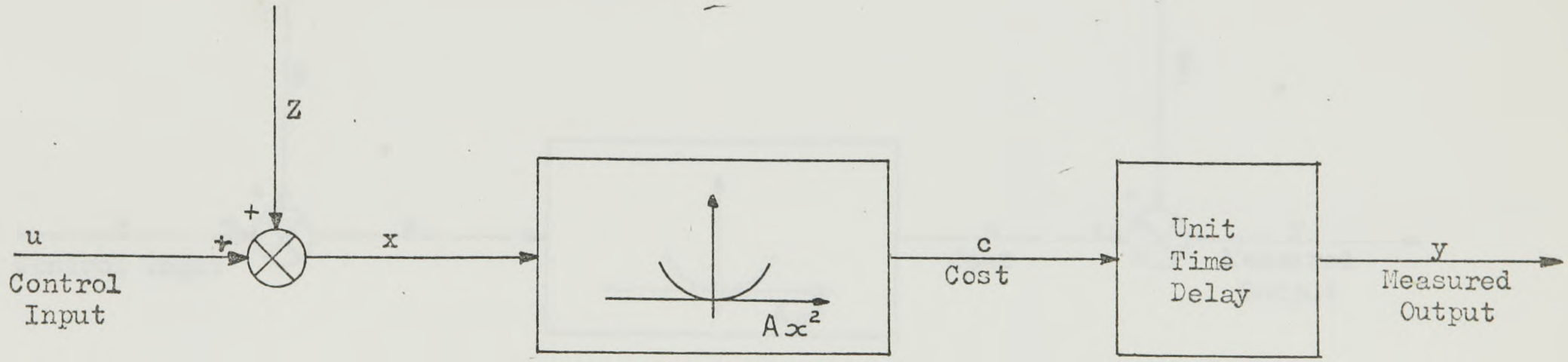

Z
+
+
u
Control
Input
x
Ax^2
c
Cost
Unit
Time
Delay
y
Measured
Output

Figure 6.1.1.1

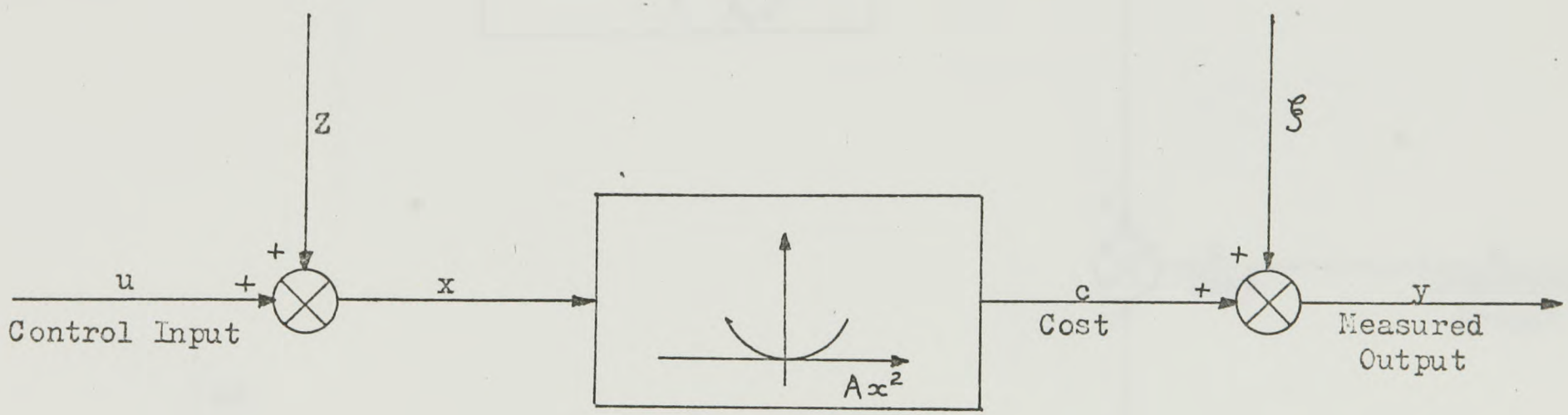


Figure 6.2.1.1

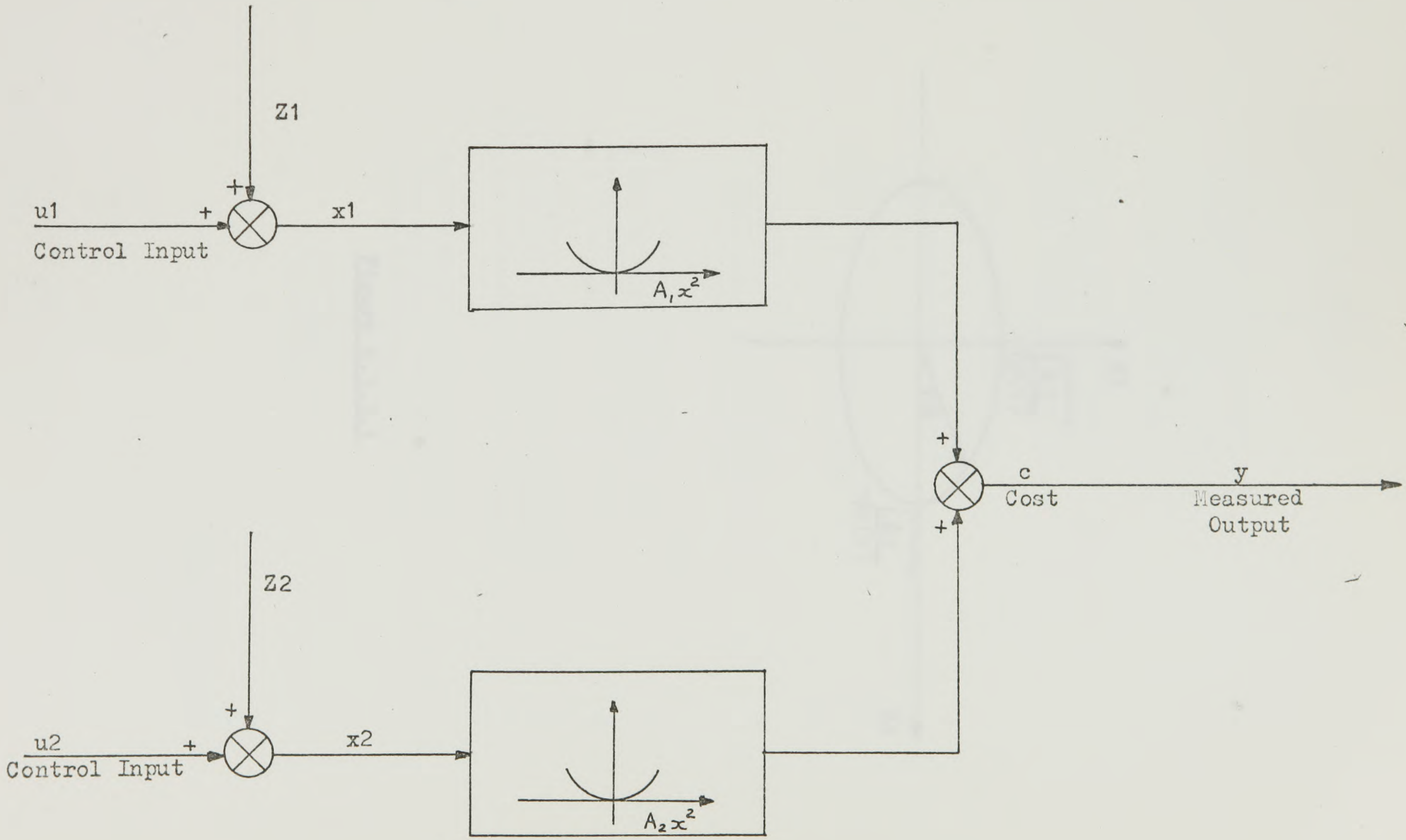

Z1
u1
Control Input
+
+
x1
$A_1 x^2$
+
+
c
Cost
y
Measured
Output
Z2
u2
Control Input
+
+
x2
$A_2 x^2$

Figure 6.3.1.1

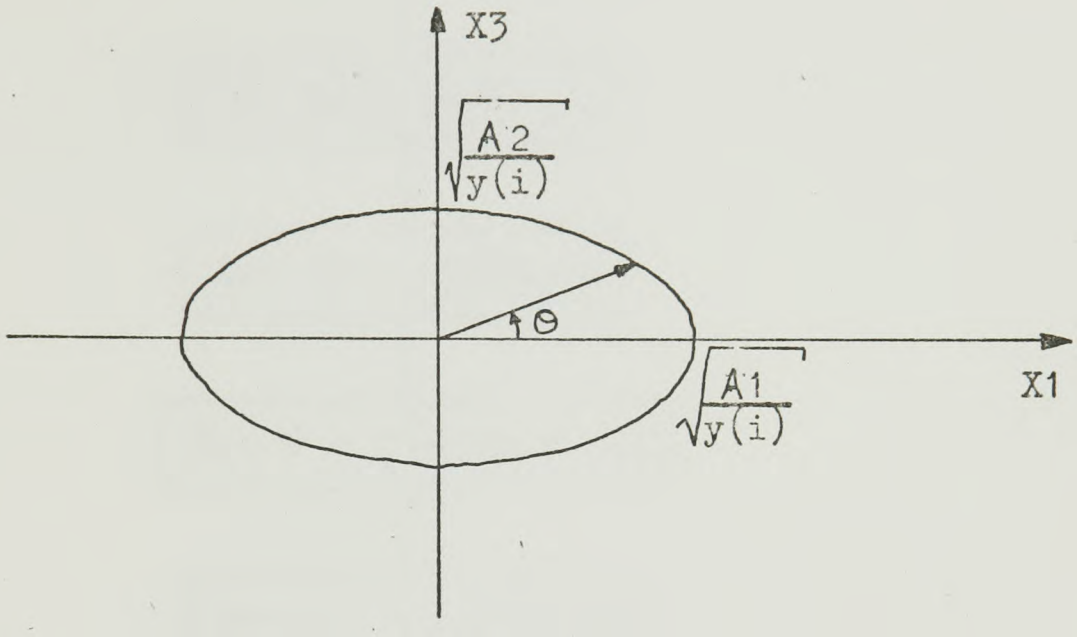


Figure 6.3.3.1

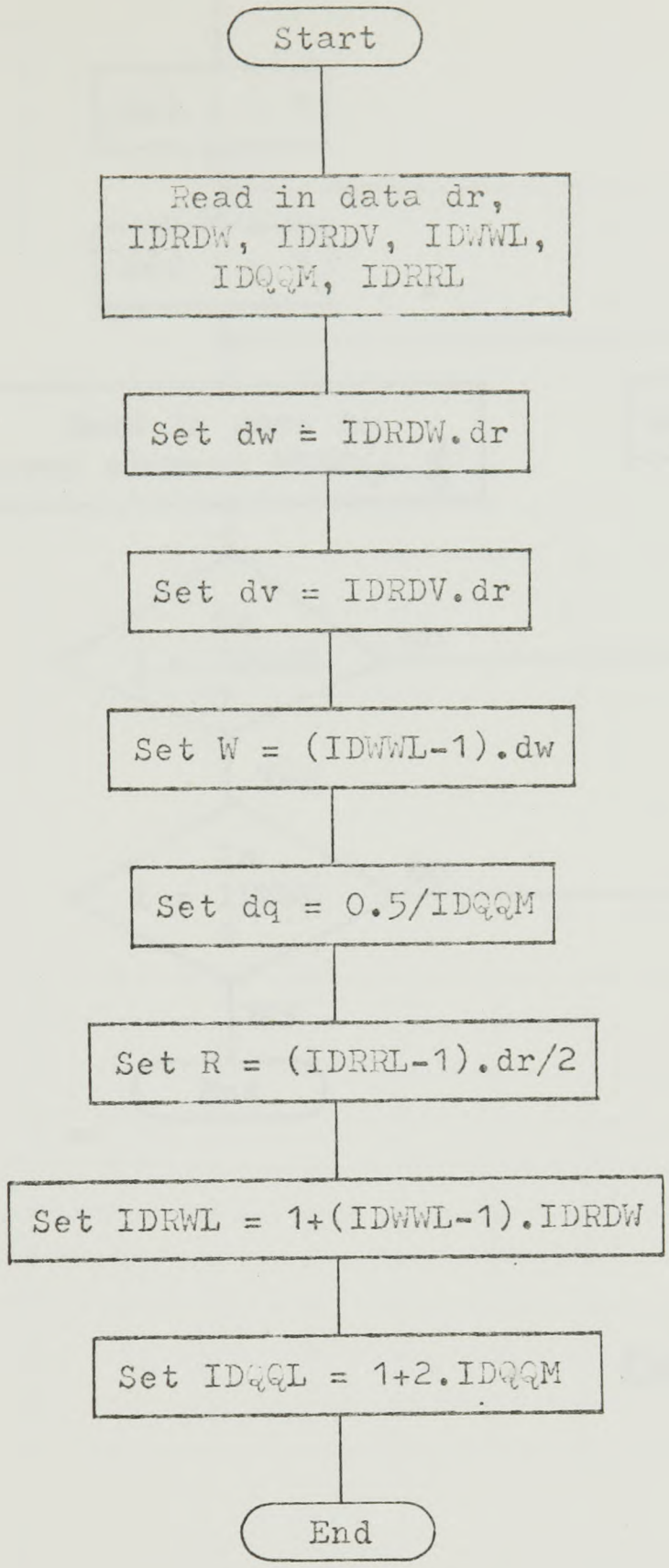

Start
Read in data dr, IDRDW, IDRDV, IDWWL, IDQQM, IDRRL
Set dw = IDRDW.dr
Set dv = IDRDV.dr
Set W = (IDWWL-1).dw
Set dq = 0.5/IDQQM
Set R = (IDRRL-1).dr/2
Set IDRWL = 1+(IDWWL-1).IDRDW
Set IDQQL = 1+2.IDQQM
End

Figure A3.1.1

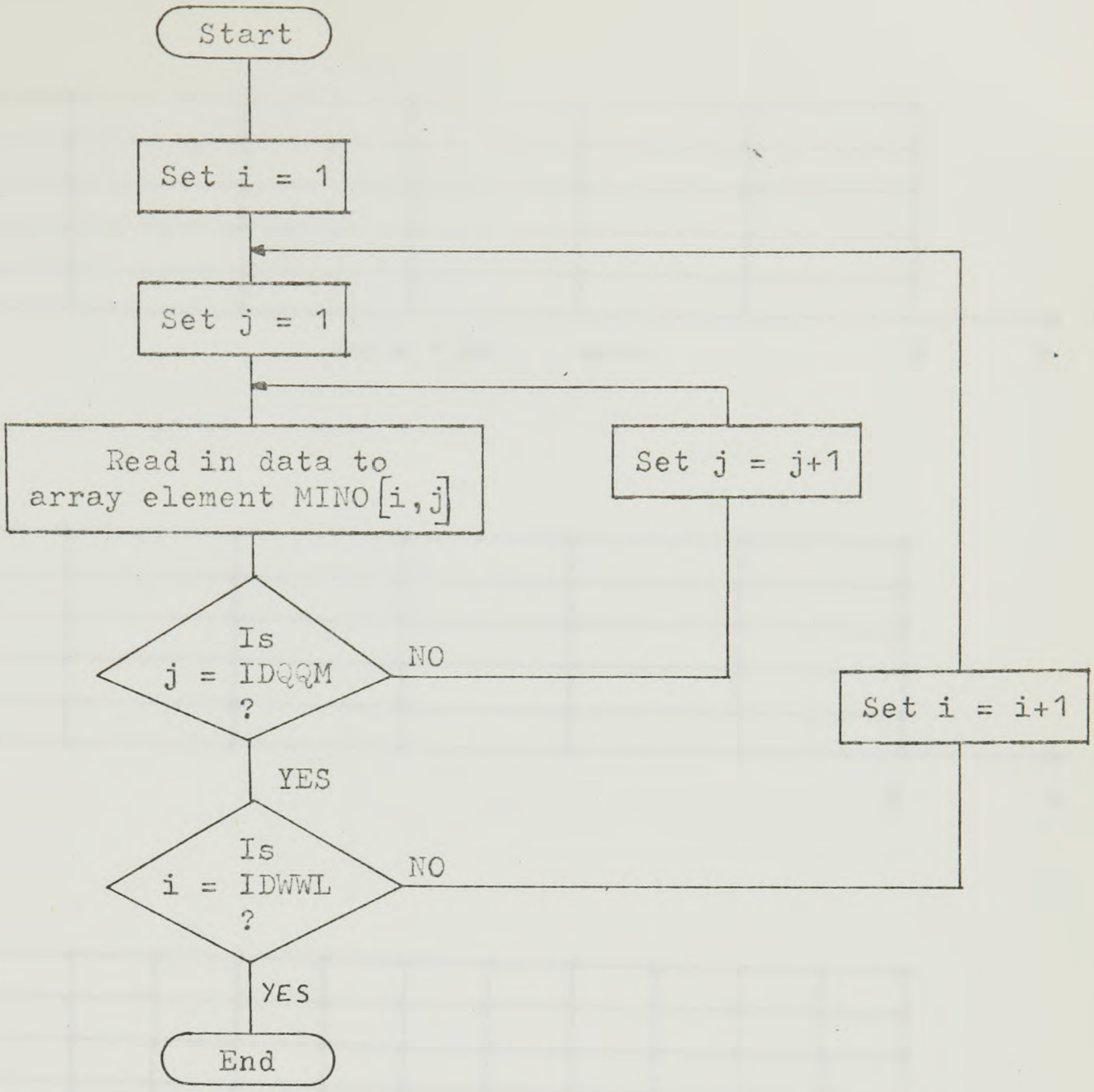
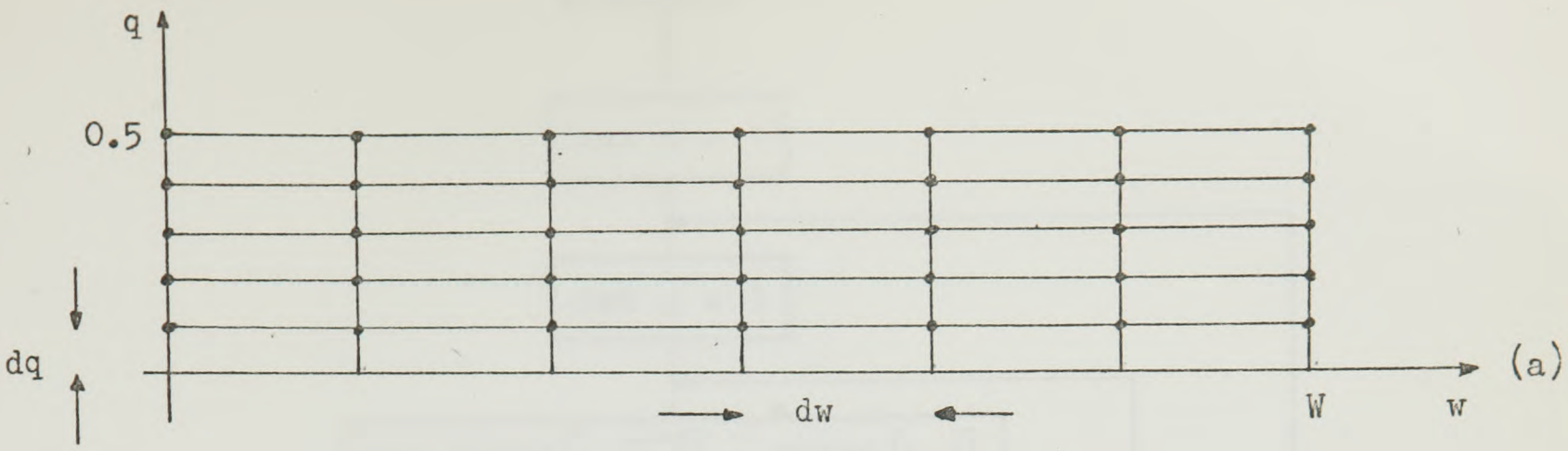
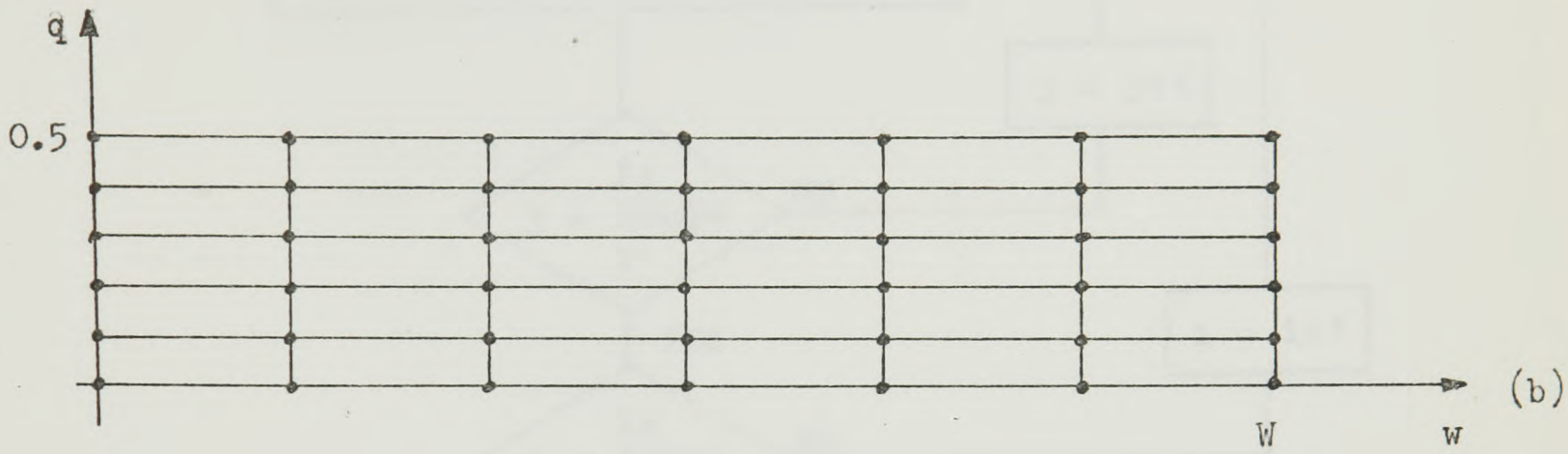
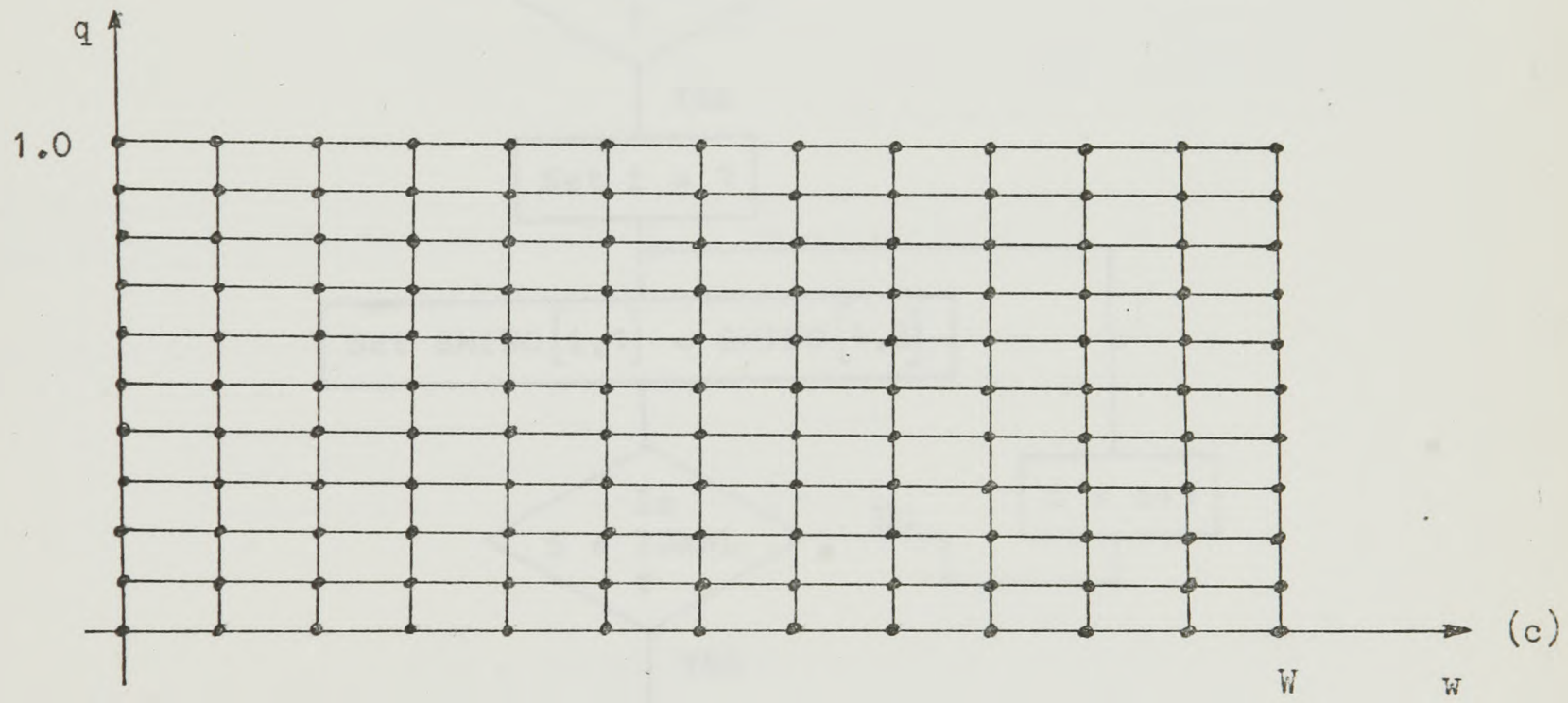

Start
Set i = 1
Set j = 1
Read in data to array element MINO[i,j]
Set j = j+1
Is j = IDQQM ?
NO
Set i = i+1
YES
Is i = IDWWL ?
NO
YES
End

Figure A3.2.1





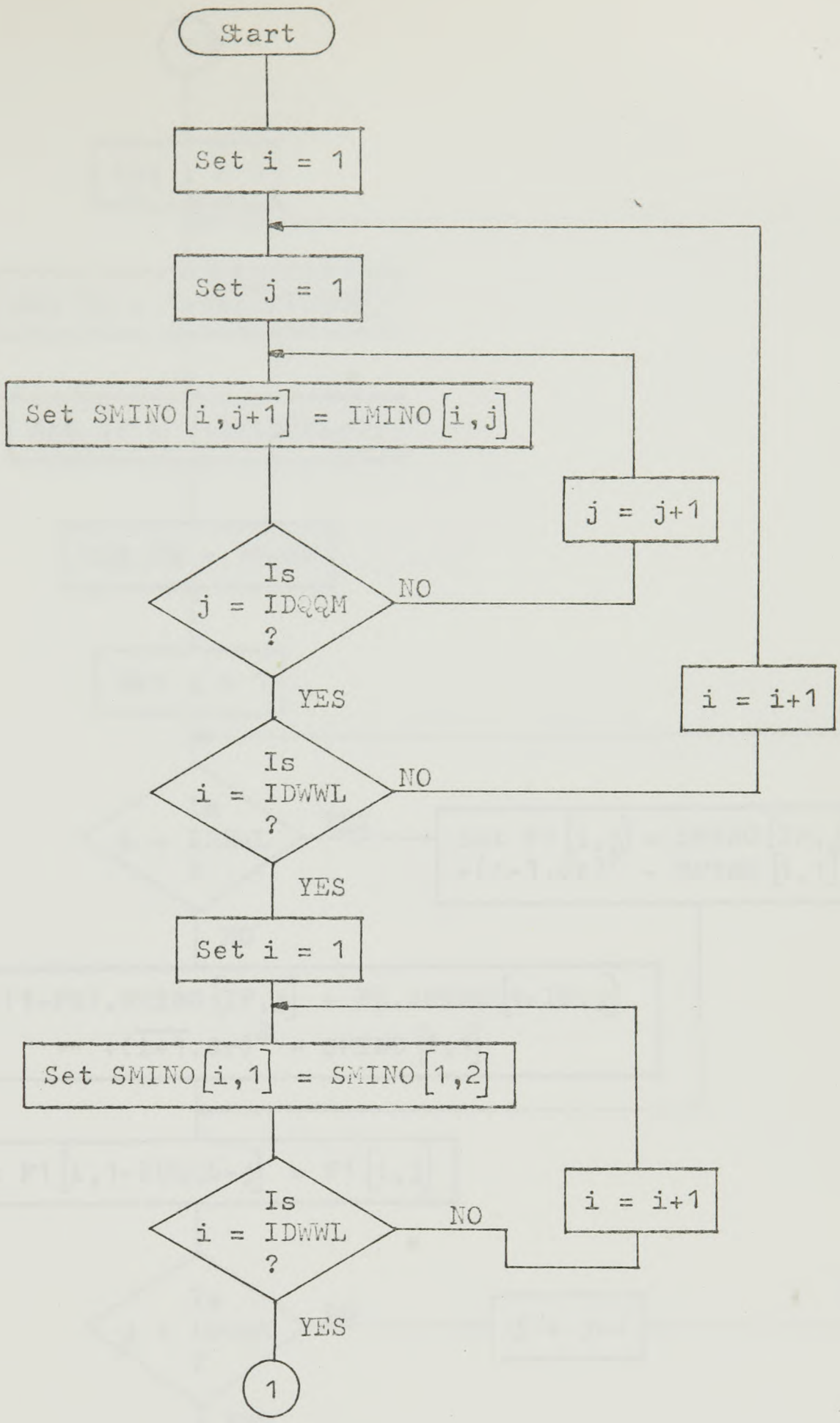


i	1	2	3	4	5	6	7	8	9	10	11	12	13
FL	1	$\frac{3}{2}$	2	$\frac{5}{2}$	3	$\frac{7}{2}$	4	$\frac{9}{2}$	5	$\frac{11}{2}$	6	$\frac{13}{2}$	7
IF	1	1	2	2	3	3	4	4	5	5	6	6	7
FR	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0

Example for the case IDRDW = 2, IDWWL = 7, dq = 0.1

. ≡ grid point

Figure A3.3.1


Start
Set i = 1
Set j = 1
Set SMINO[i,j+1] = IMINO[i,j]
j = j+1
Is j = IDQQM ?
NO
YES
i = i+1
Is i = IDWWL ?
NO
YES
Set i = 1
Set SMINO[i,1] = SMINO[1,2]
i = i+1
Is i = IDWWL ?
NO
YES
1

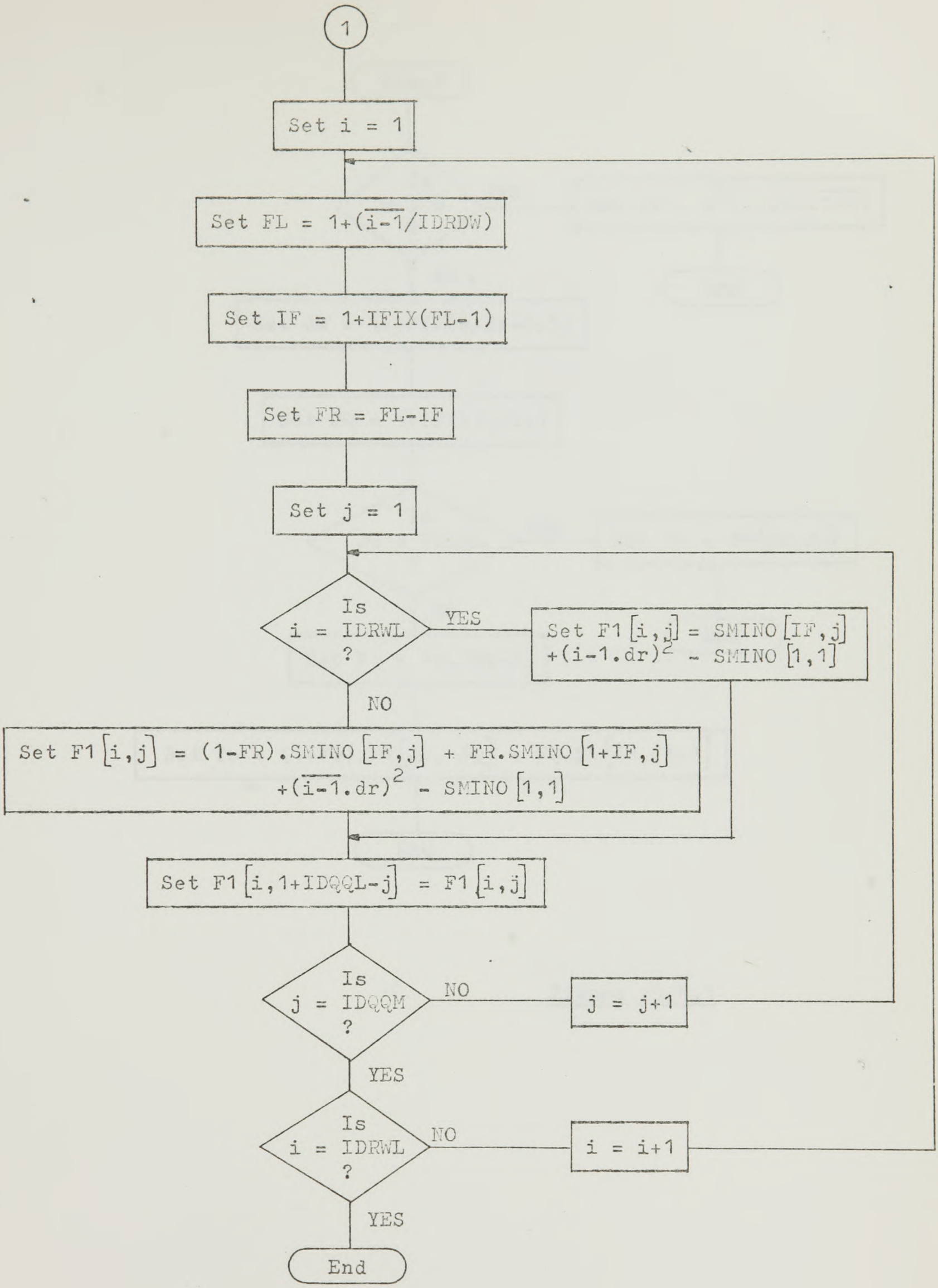

1
Set i = 1
Set FL = 1+(i-1/IDRDW)
Set IF = 1+IFIX(FL-1)
Set FR = FL-IF
Set j = 1
Is i = IDRWL ?
YES
Set F1 [i,j] = SMINO [IF,j] +(i-1.dr)^2 - SMINO [1,1]
NO
Set F1 [i,j] = (1-FR).SMINO [IF,j] + FR.SMINO [1+IF,j] +(i-1.dr)^2 - SMINO [1,1]
Set F1 [i,1+IDQQL-j] = F1 [i,j]
Is j = IDQQM ?
NO
j = j+1
YES
Is i = IDRWL ?
NO
i = i+1
YES
End

Figure A3.3.2

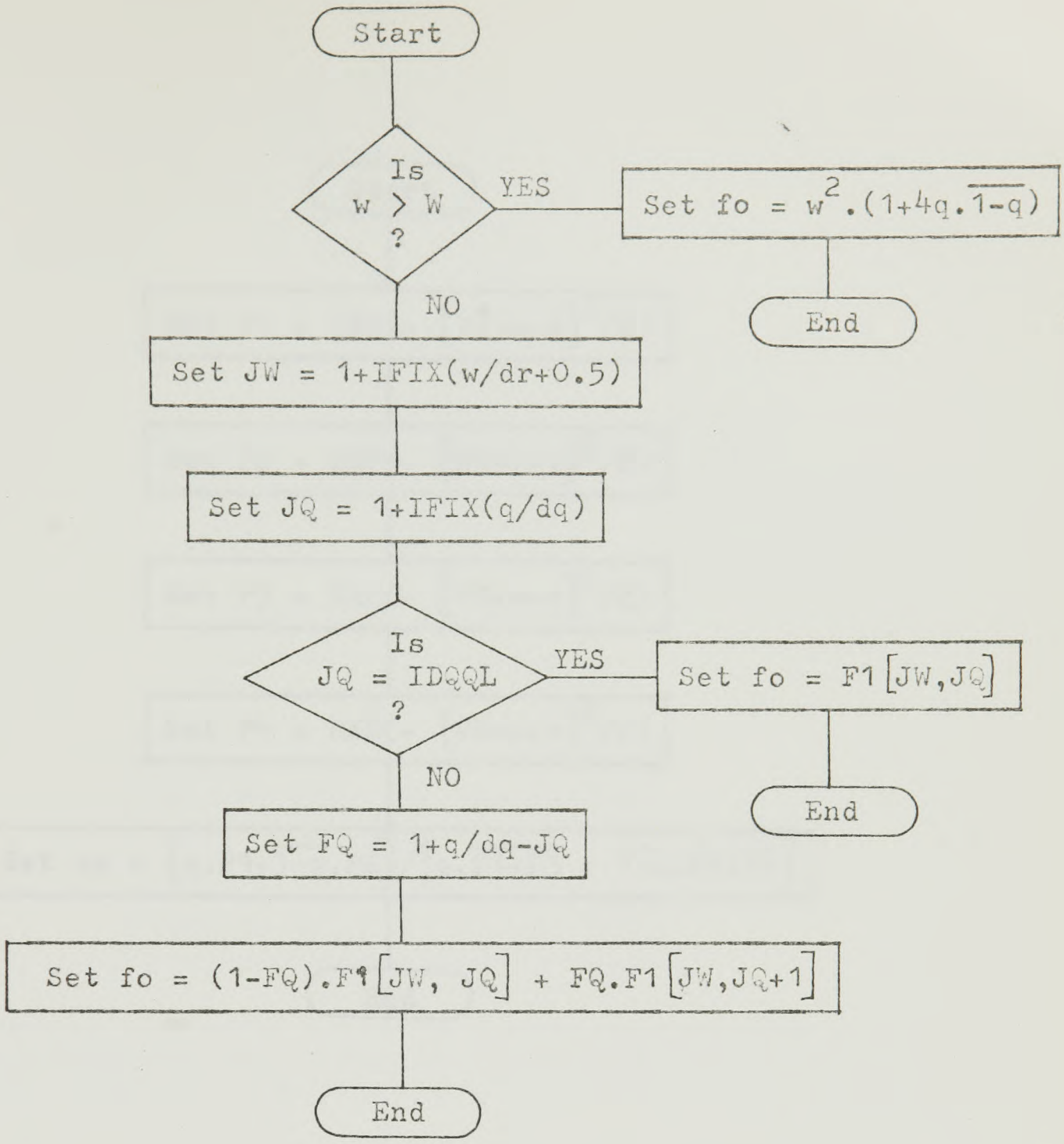

Start
Is w > W ?
YES
Set fo = w^2.(1+4q.√(1-q))
End
NO
Set JW = 1+IFIX(w/dr+0.5)
Set JQ = 1+IFIX(q/dq)
Is JQ = IDQQL ?
YES
Set fo = F1[JW,JQ]
End
NO
Set FQ = 1+q/dq-JQ
Set fo = (1-FQ).F1[JW, JQ] + FQ.F1[JW,JQ+1]
End

Figure A3.4.1

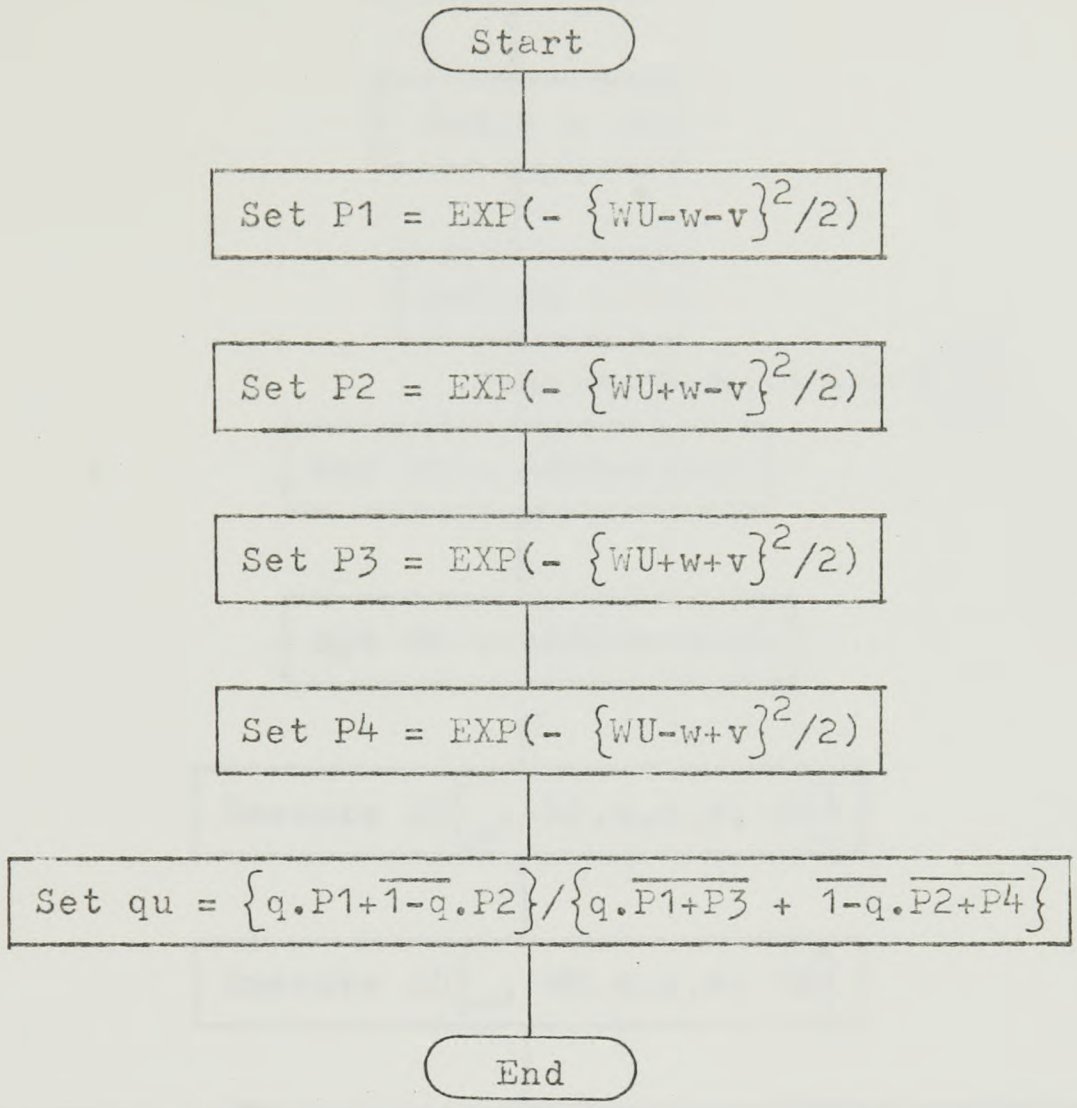
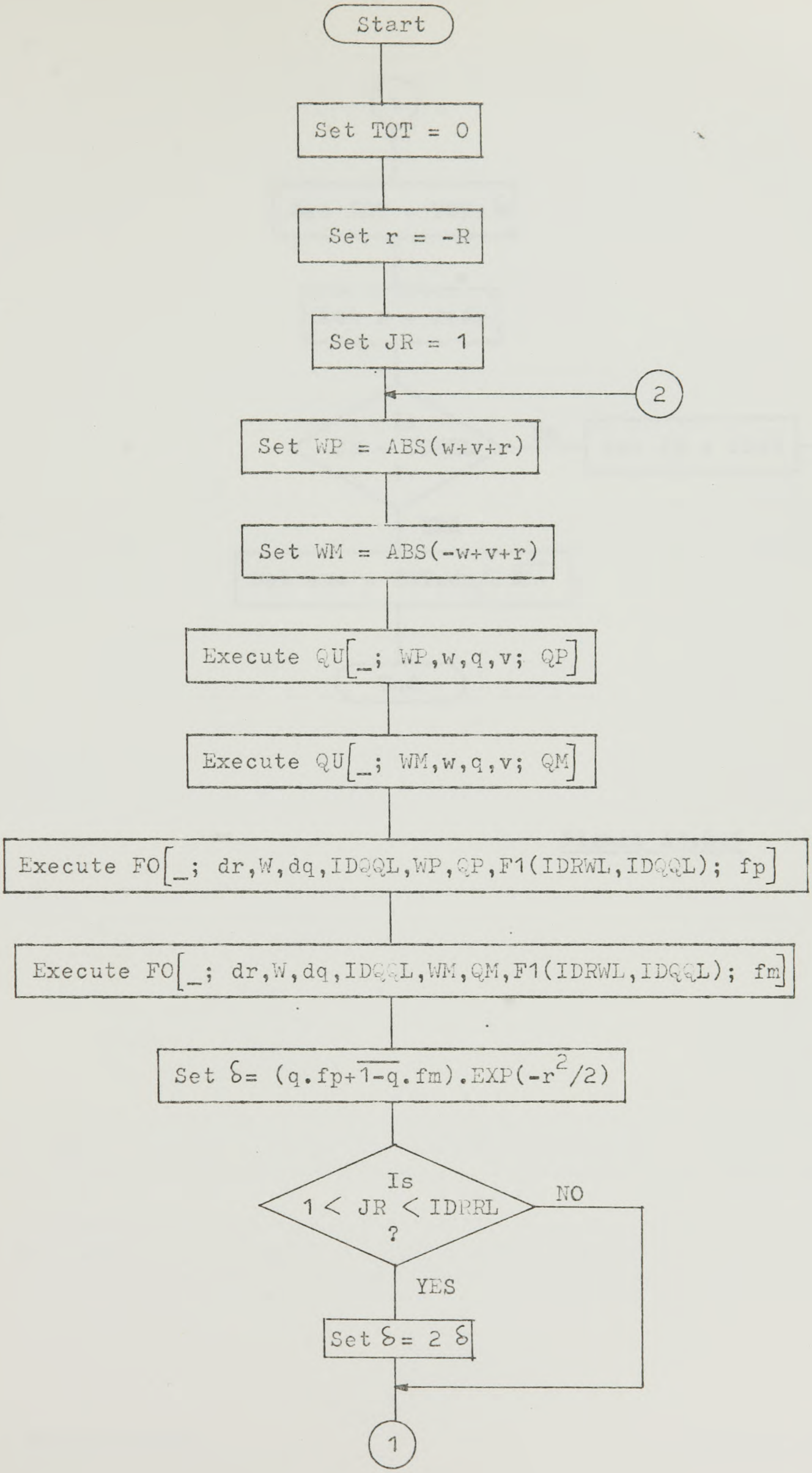

Start
Set P1 = EXP(- {WU-w-v}^2/2)
Set P2 = EXP(- {WU+w-v}^2/2)
Set P3 = EXP(- {WU+w+v}^2/2)
Set P4 = EXP(- {WU-w+v}^2/2)
Set qu = {q.P1+1-q.P2}/{q.P1+P3 + 1-q.P2+P4}
End

Figure A3.5.1


Start
Set TOT = 0
Set r = -R
Set JR = 1
2
Set WP = ABS(w+v+r)
Set WM = ABS(-w+v+r)
Execute QU[_; WP,w,q,v; QP]
Execute QU[_; WM,w,q,v; QM]
Execute FO[_; dr,W,dq,IDQQL,WP,QP,F1(IDRWL,IDQQL); fp]
Execute FO[_; dr,W,dq,IDQQL,WM,QM,F1(IDRWL,IDQQL); fm]
Set δ= (q.fp+1-q.fm).EXP(-r²/2)
Is 1 < JR < IDRRL ?
NO
YES
Set δ= 2 δ
1

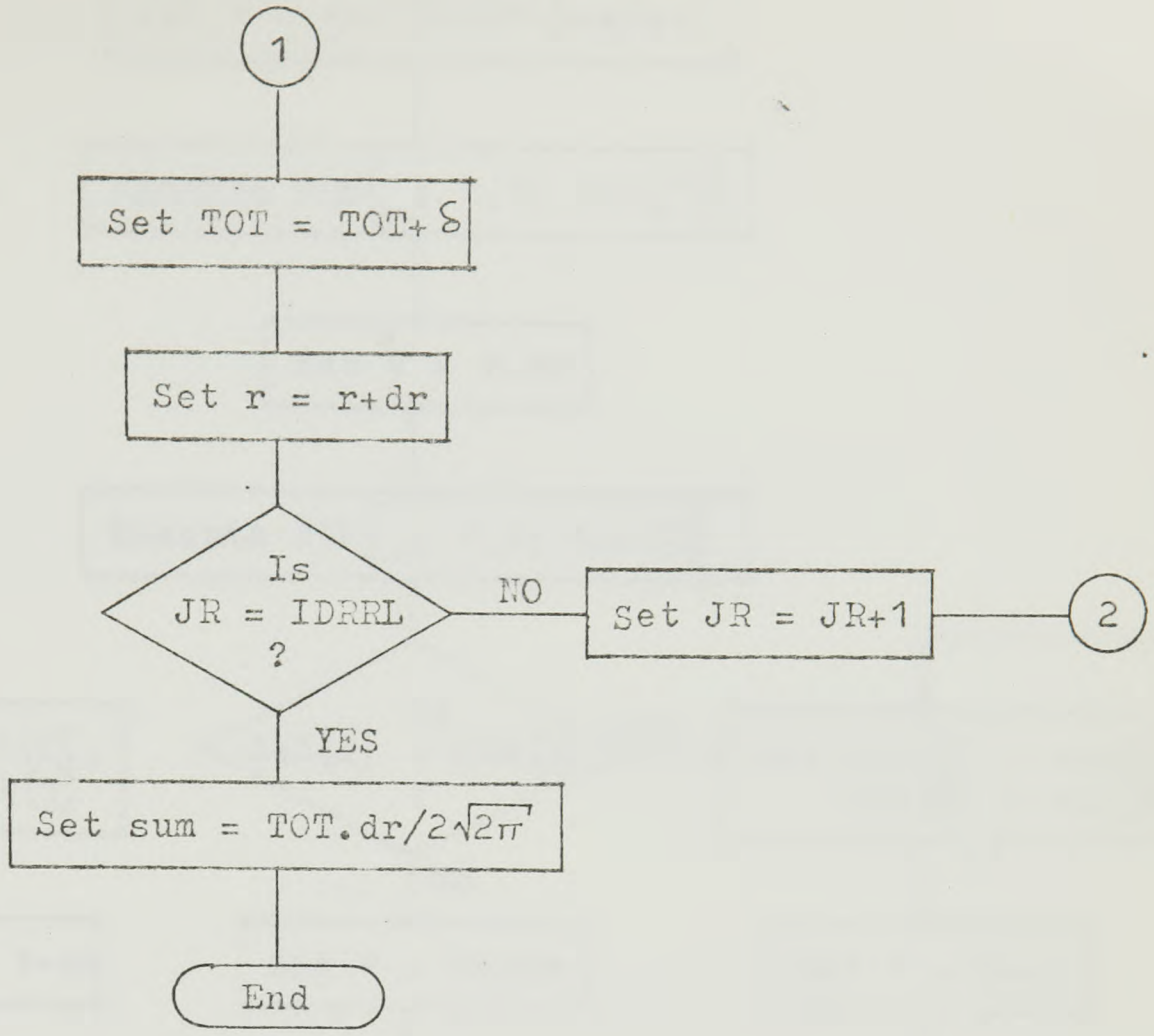
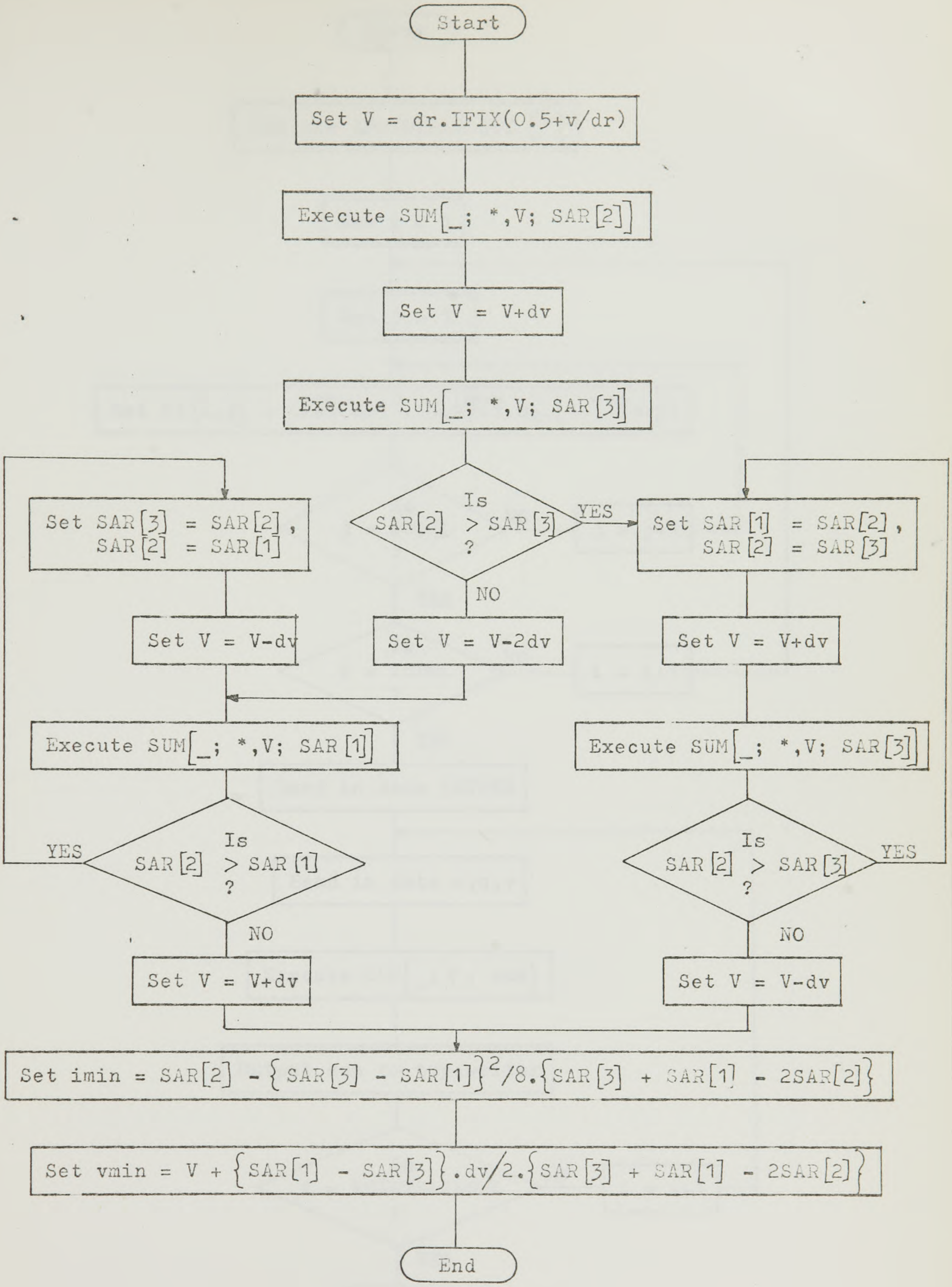
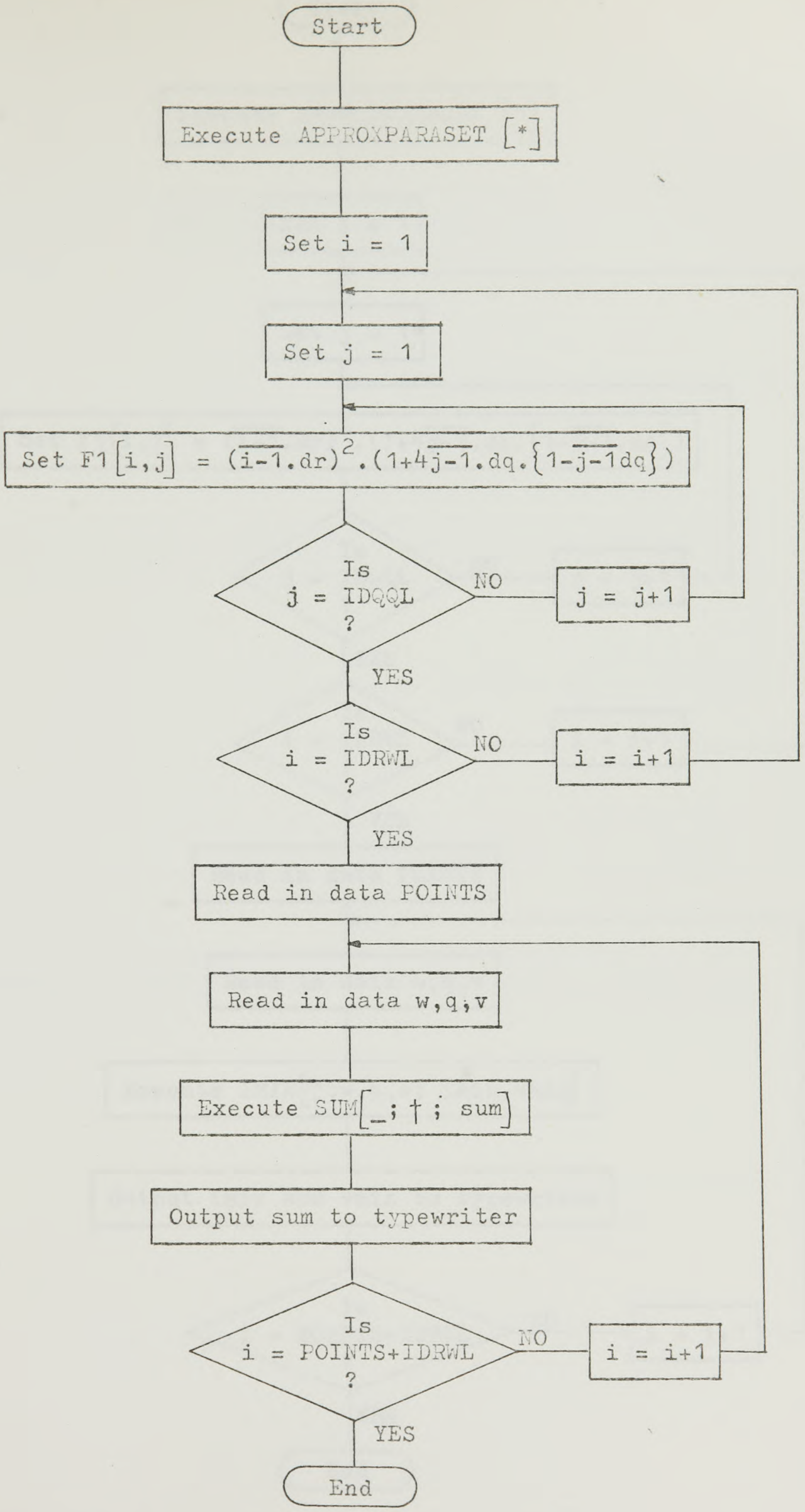

1
Set TOT = TOT+ δ
Set r = r+dr
Is JR = IDRRL ?
NO
Set JR = JR+1
2
YES
Set sum = TOT.dr/2√2π
End

Figure A3.6.1



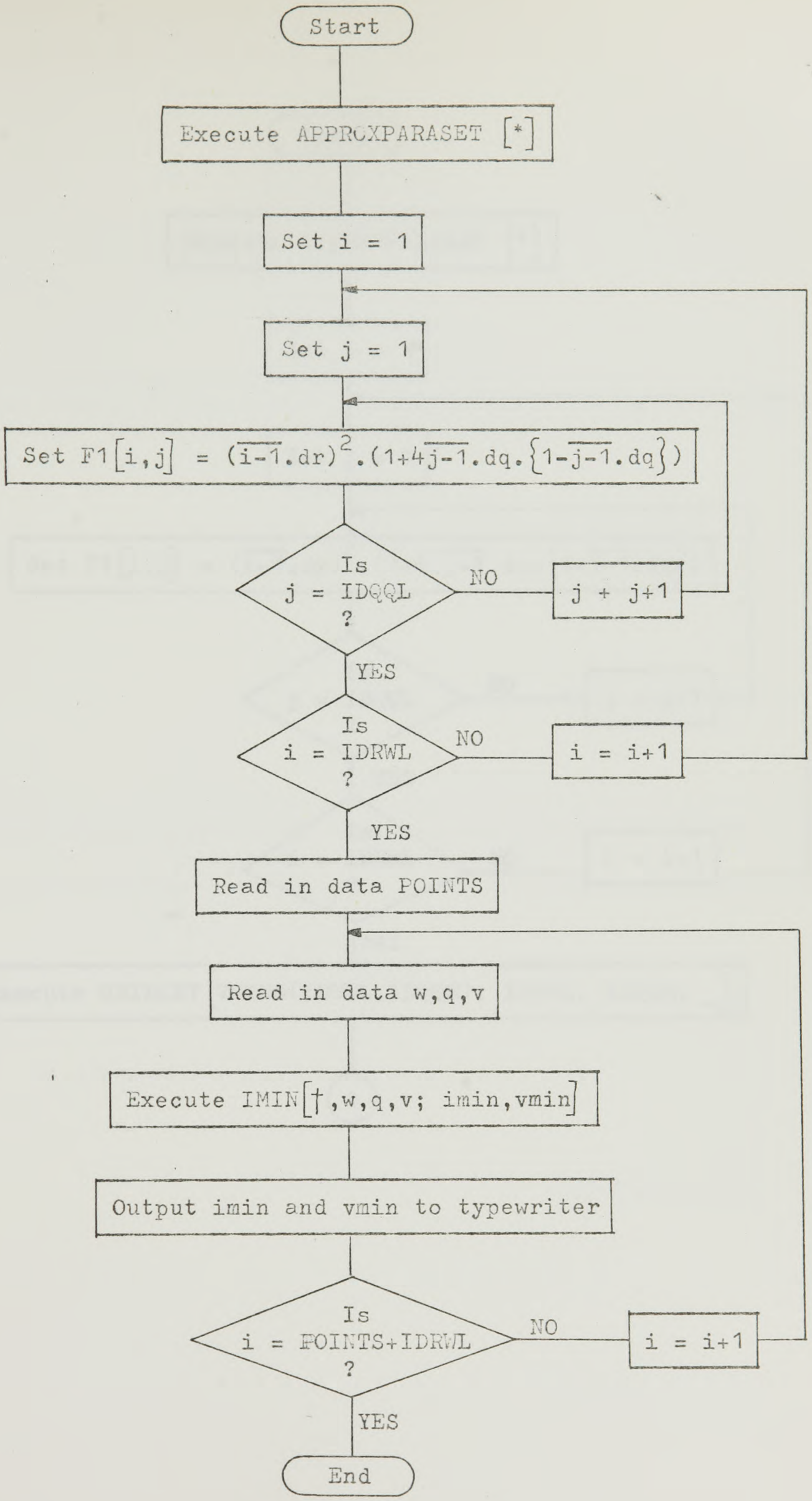
* See comments.

Figure A3.7.1



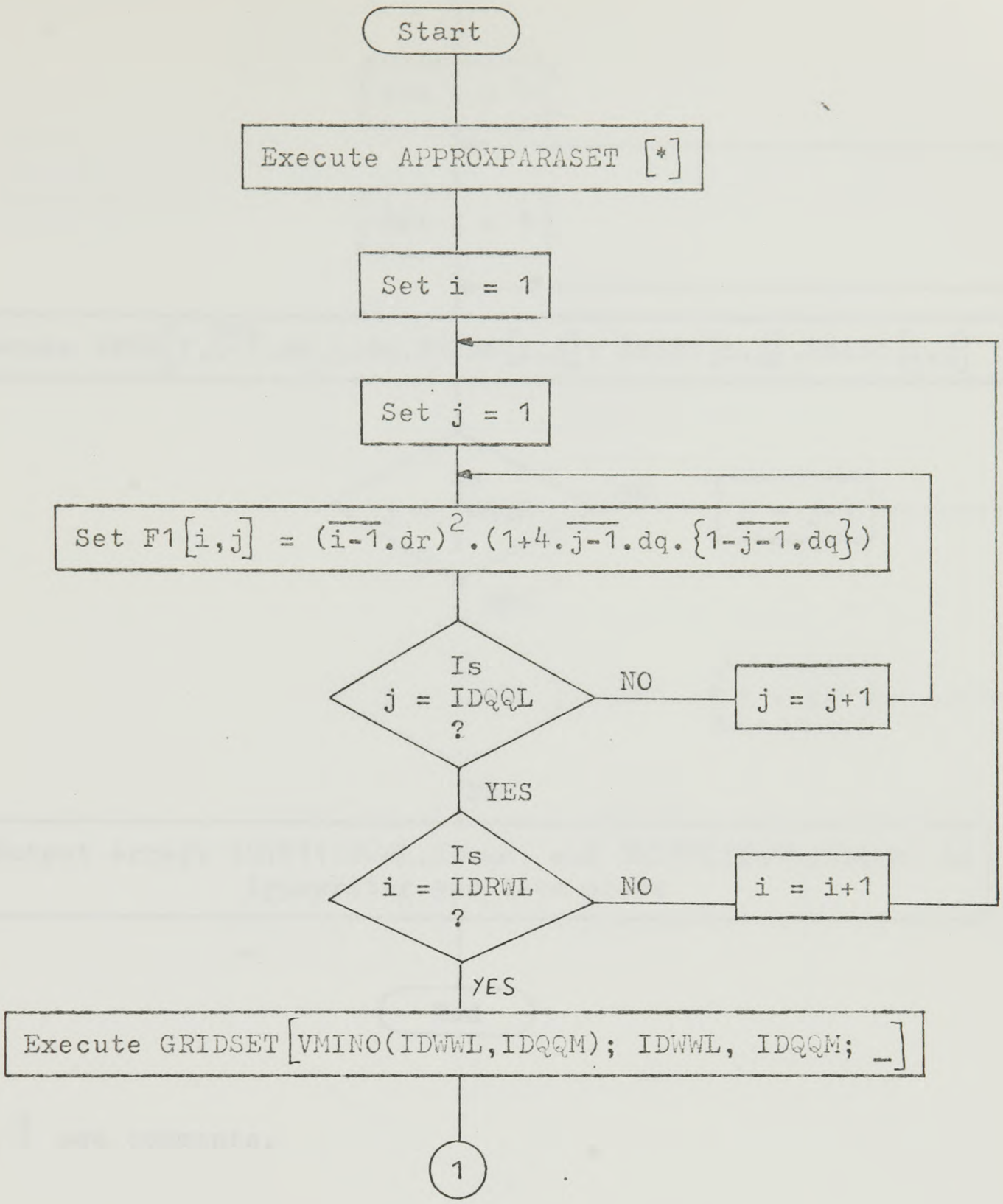
*, † see comments.

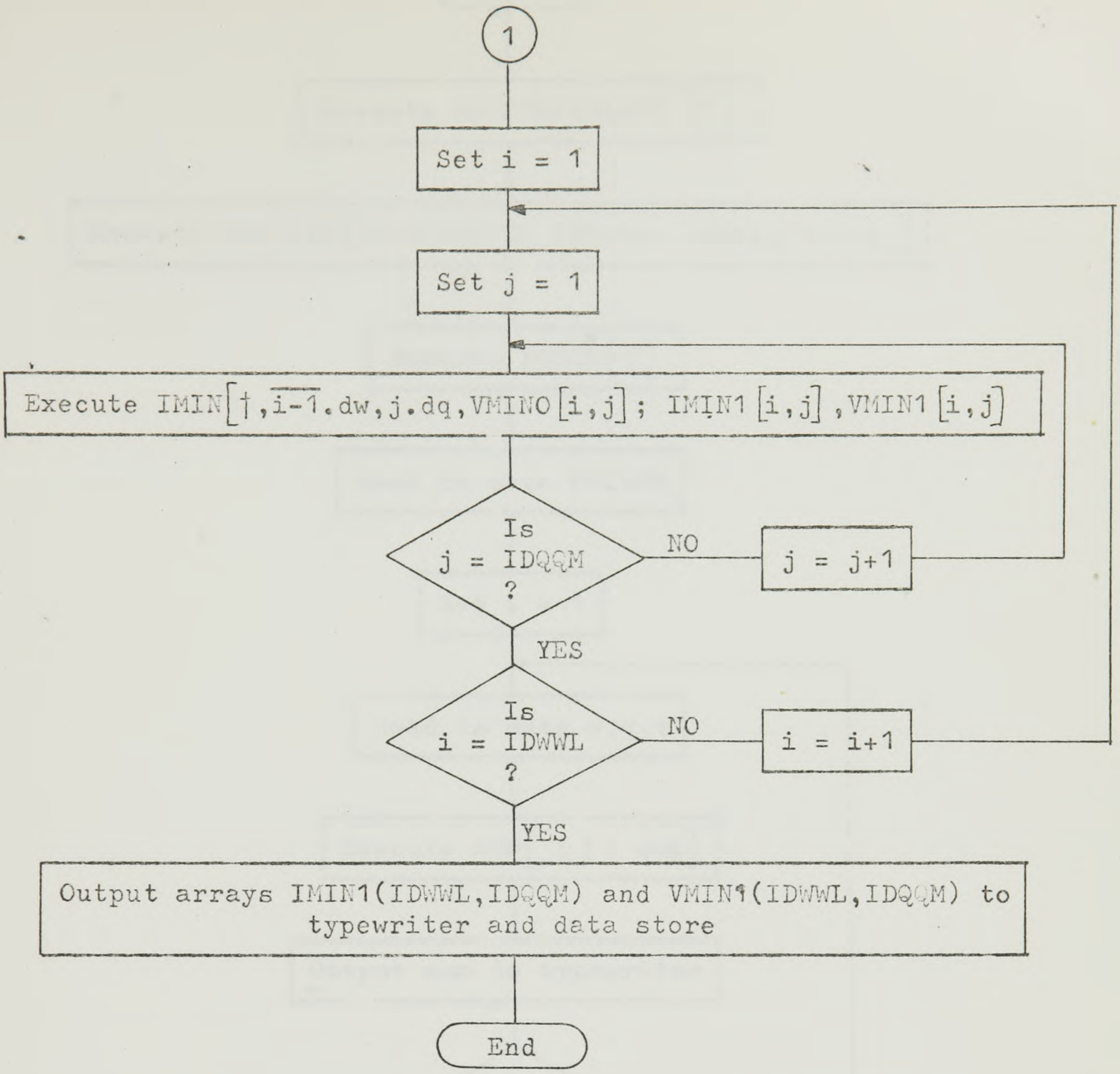
Figure A3.8.1



*, † see comments

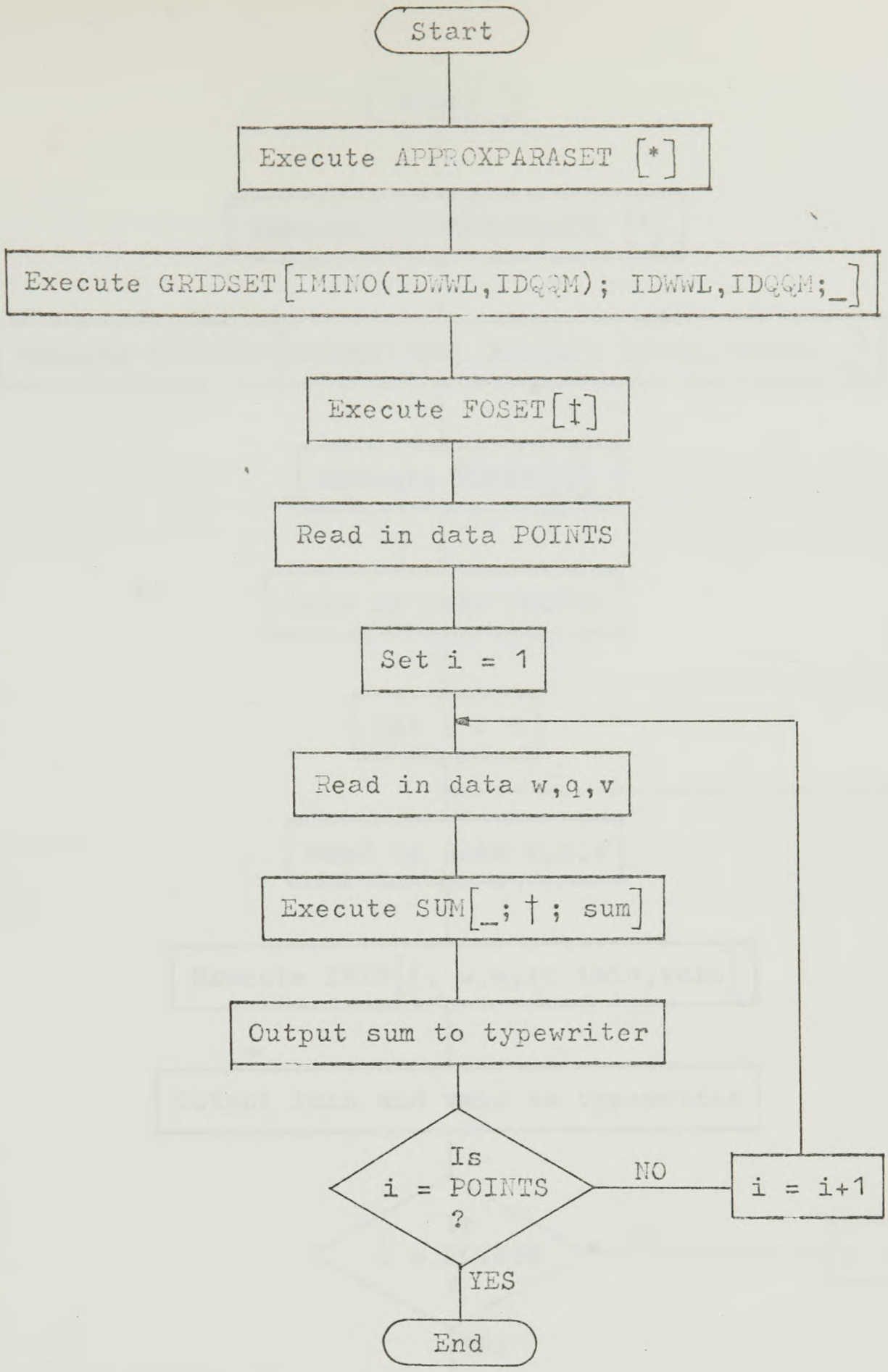
Figure A3.9.1


Start
Execute APPROXPARASET [*]
Set i = 1
Set j = 1
Set F1[i,j] = (i-1.dr)^2.(1+4.j-1.dq.{1-j-1.dq})
Is j = IDQQL ?
NO
j = j+1
YES
Is i = IDRWL ?
NO
i = i+1
YES
Execute GRIDSET[VMINO(IDWWL,IDQQM); IDWWL, IDQQM; _]
1



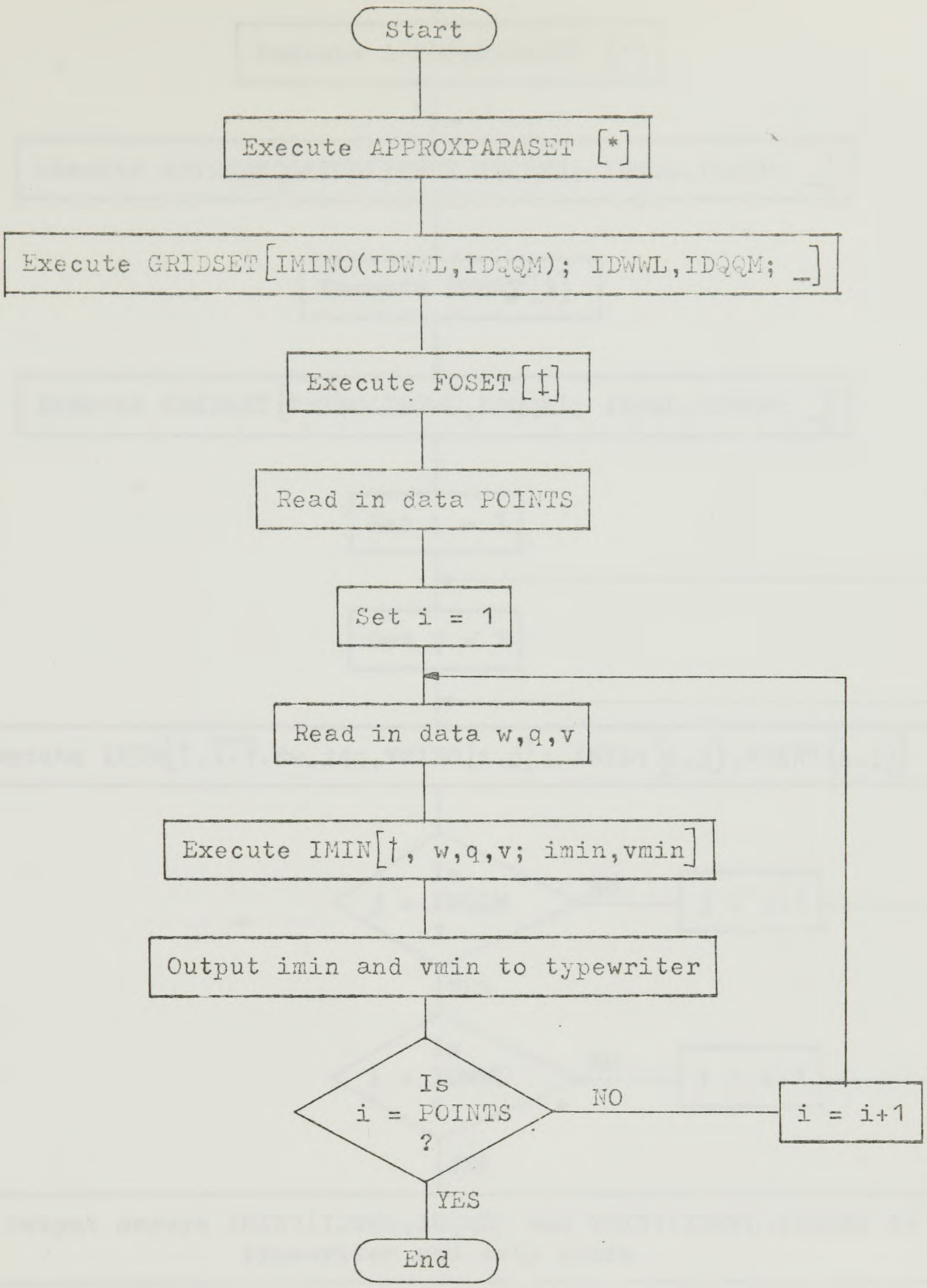
*, † see comments.

Figure A3.10.1



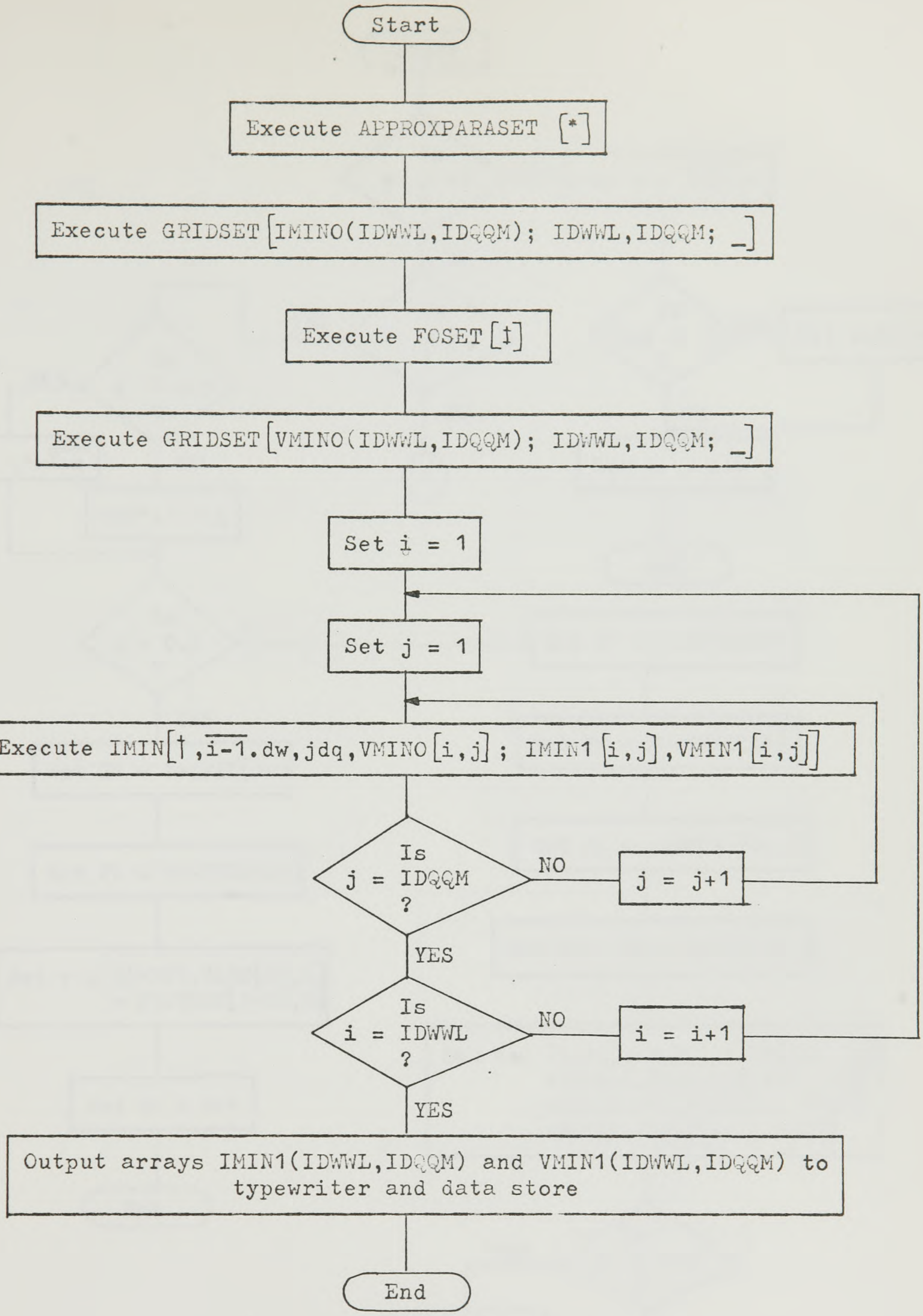
*, ‡, † see comments.

Figure A3.11.1



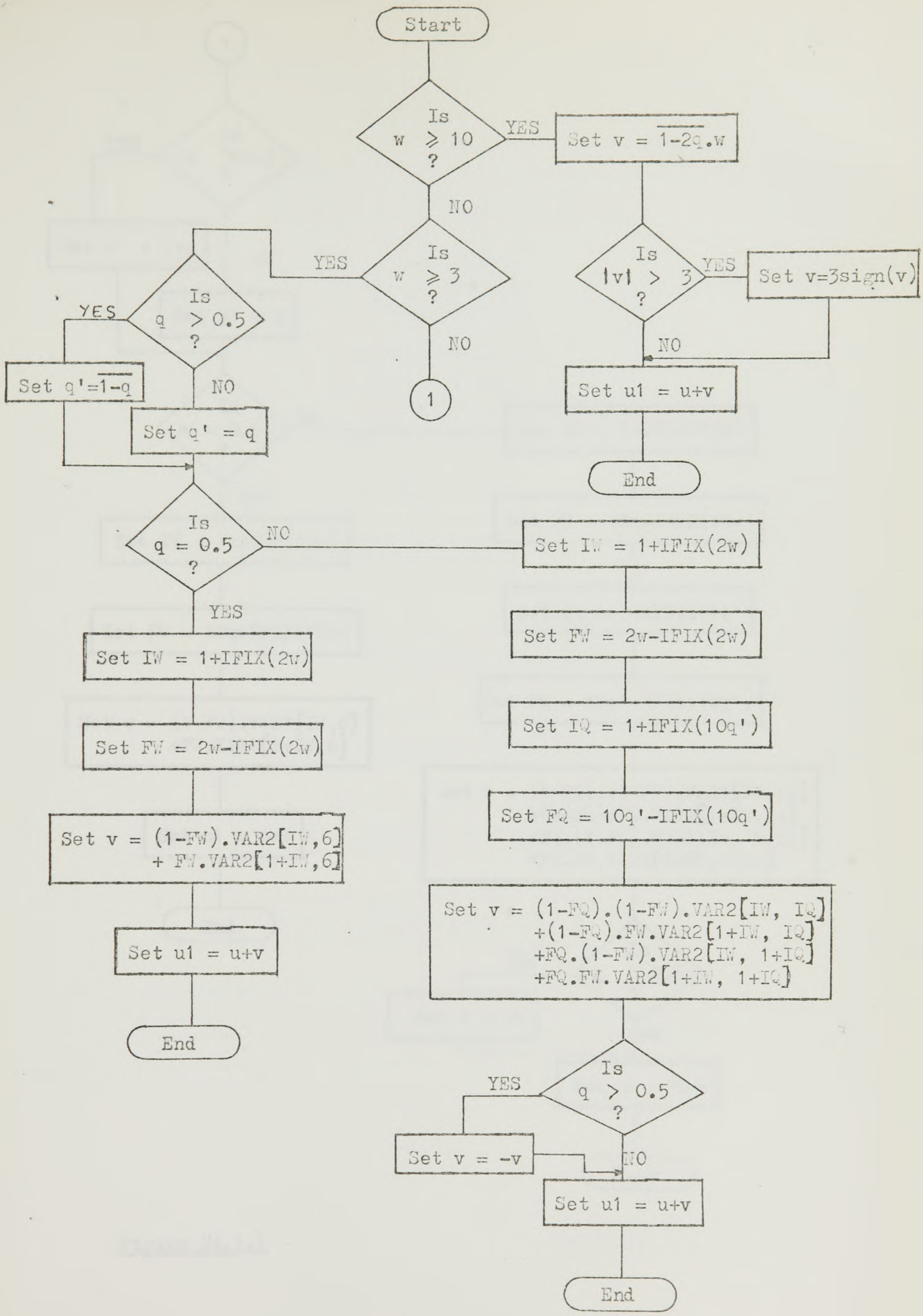
*, ‡, † see comments

Figure A3.12.1



*, ‡, † see comments.

Figure A3.13.1


Start
Is w ⩾ 10 ?
YES
Set v = 1-2q.w
NO
Is w ⩾ 3 ?
YES
NO
1
Is |v| > 3 ?
YES
Set v=3sign(v)
NO
Set u1 = u+v
End
Is q > 0.5 ?
YES
Set q'=1-q
NO
Set q' = q
Is q = 0.5 ?
NO
YES
Set IW = 1+IFIX(2w)
Set FW = 2w-IFIX(2w)
Set v = (1-FW).VAR2[IW,6] + FW.VAR2[1+IW,6]
Set u1 = u+v
End
Set IW = 1+IFIX(2w)
Set FW = 2w-IFIX(2w)
Set IQ = 1+IFIX(10q')
Set FQ = 10q'-IFIX(10q')
Set v = (1-FQ).(1-FW).VAR2[IW, IQ] +(1-FQ).FW.VAR2[1+IW, IQ] +FQ.(1-FW).VAR2[IW, 1+IQ] +FQ.FW.VAR2[1+IW, 1+IQ]
Is q > 0.5 ?
YES
Set v = -v
NO
Set u1 = u+v
End

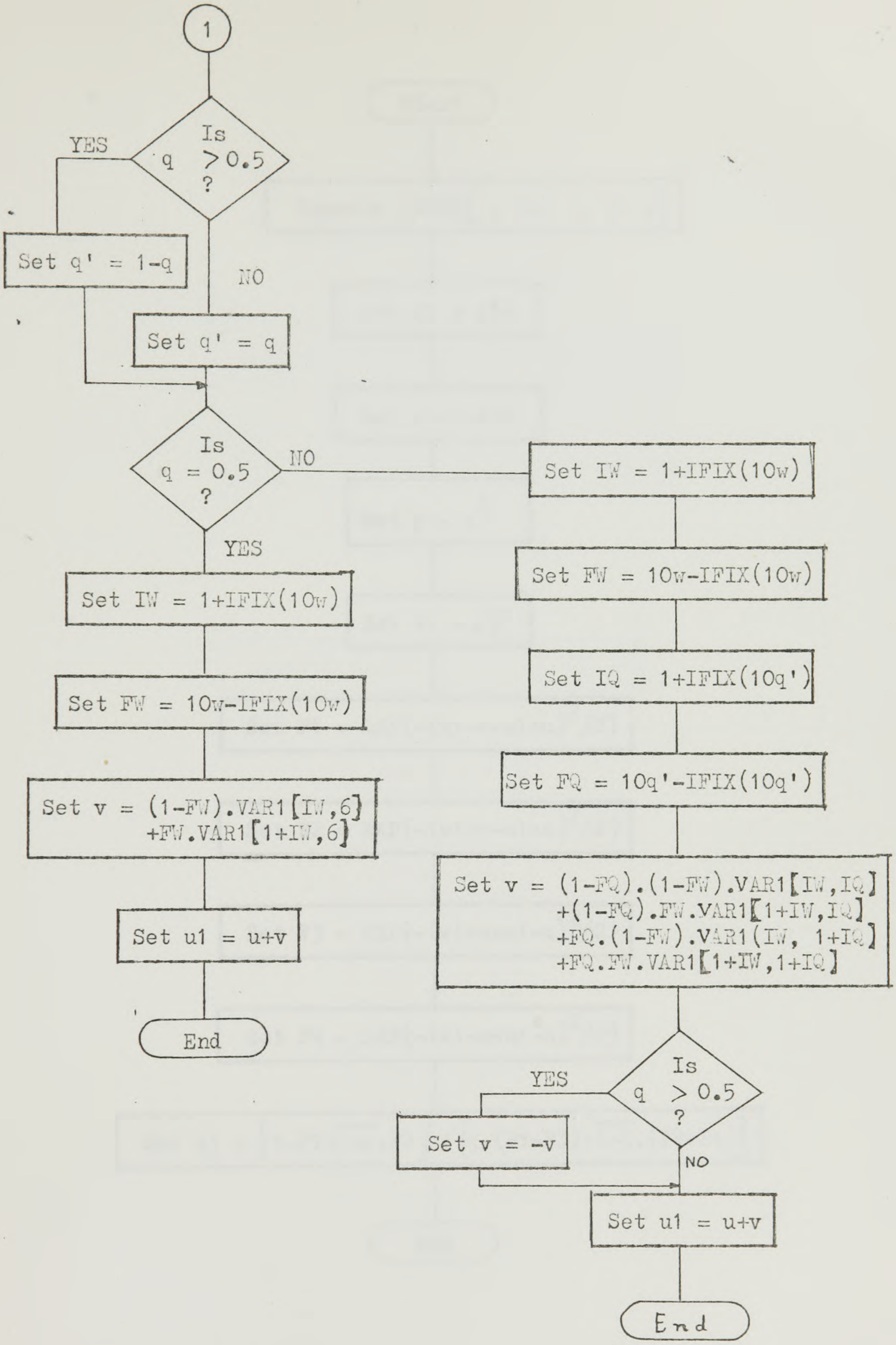

1
Is q > 0.5 ?
YES
Set q' = 1-q
NO
Set q' = q
Is q = 0.5 ?
NO
YES
Set IW = 1+IFIX(10w)
Set FW = 10w-IFIX(10w)
Set v = (1-FW).VAR1[IW,6] +FW.VAR1[1+IW,6]
Set u1 = u+v
End
Set IW = 1+IFIX(10w)
Set FW = 10w-IFIX(10w)
Set IQ = 1+IFIX(10q')
Set FQ = 10q'-IFIX(10q')
Set v = (1-FQ).(1-FW).VAR1[IW,IQ] +(1-FQ).FW.VAR1[1+IW,IQ] +FQ.(1-FW).VAR1(IW, 1+IQ] +FQ.FW.VAR1[1+IW,1+IQ]
Is q > 0.5 ?
YES
Set v = -v
NO
Set u1 = u+v
End

Figure A4.1.1

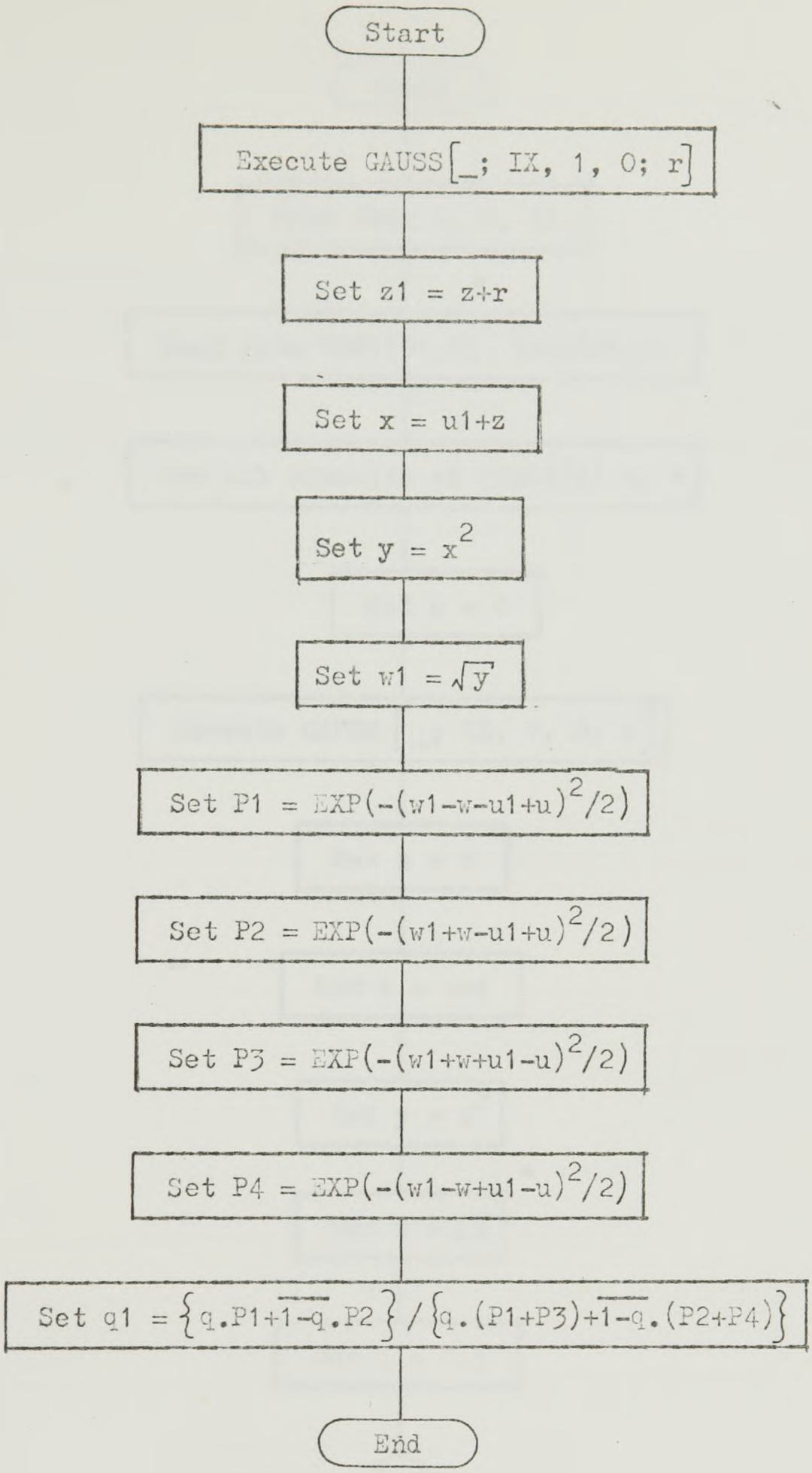
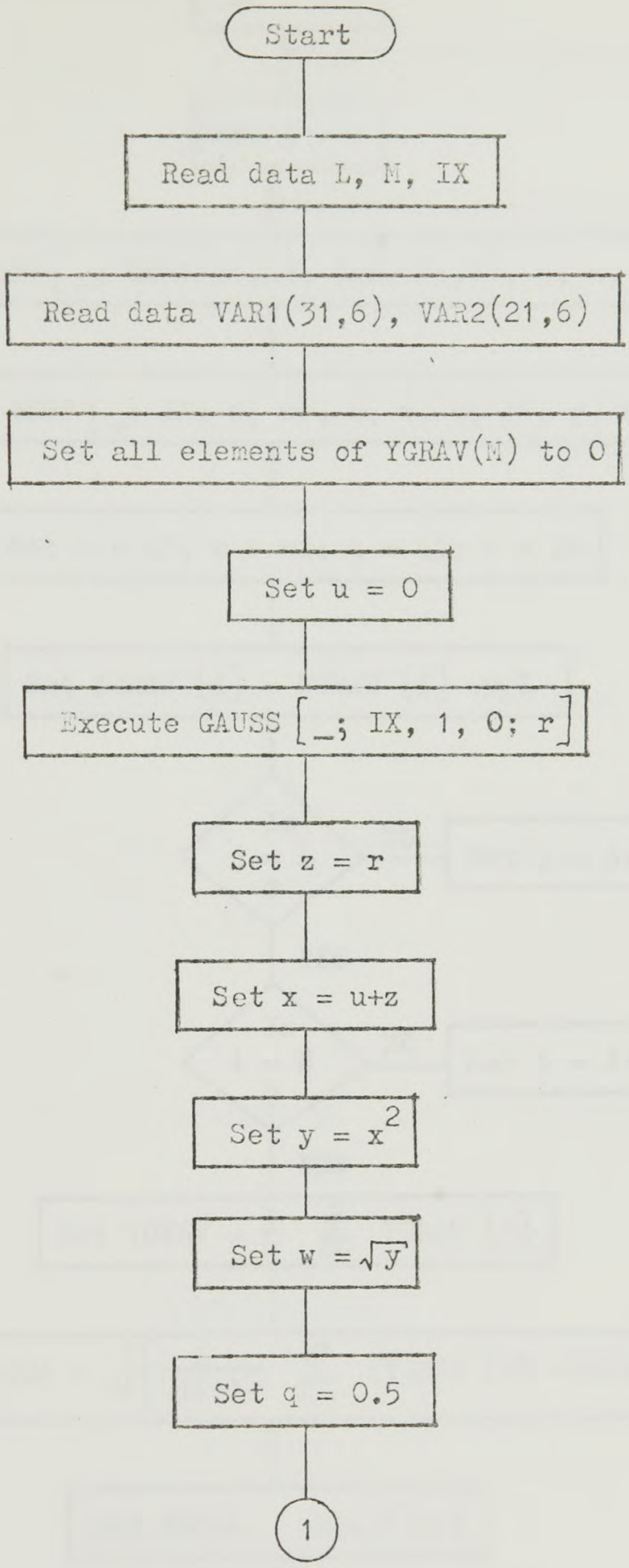

Start
Execute GAUSS[_; IX, 1, 0; r]
Set z1 = z+r
Set x = u1+z
Set y = x^2
Set w1 = √y
Set P1 = EXP(-(w1-w-u1+u)^2/2)
Set P2 = EXP(-(w1+w-u1+u)^2/2)
Set P3 = EXP(-(w1+w+u1-u)^2/2)
Set P4 = EXP(-(w1-w+u1-u)^2/2)
Set q1 = {q.P1+1-q.P2} / {q.(P1+P3)+1-q.(P2+P4)}
End

Figure A4.2.1


Start
Read data L, M, IX
Read data VAR1(31,6), VAR2(21,6)
Set all elements of YGRAV(M) to 0
Set u = 0
Execute GAUSS [_; IX, 1, 0; r]
Set z = r
Set x = u+z
Set y = x^2
Set w = √y
Set q = 0.5
1

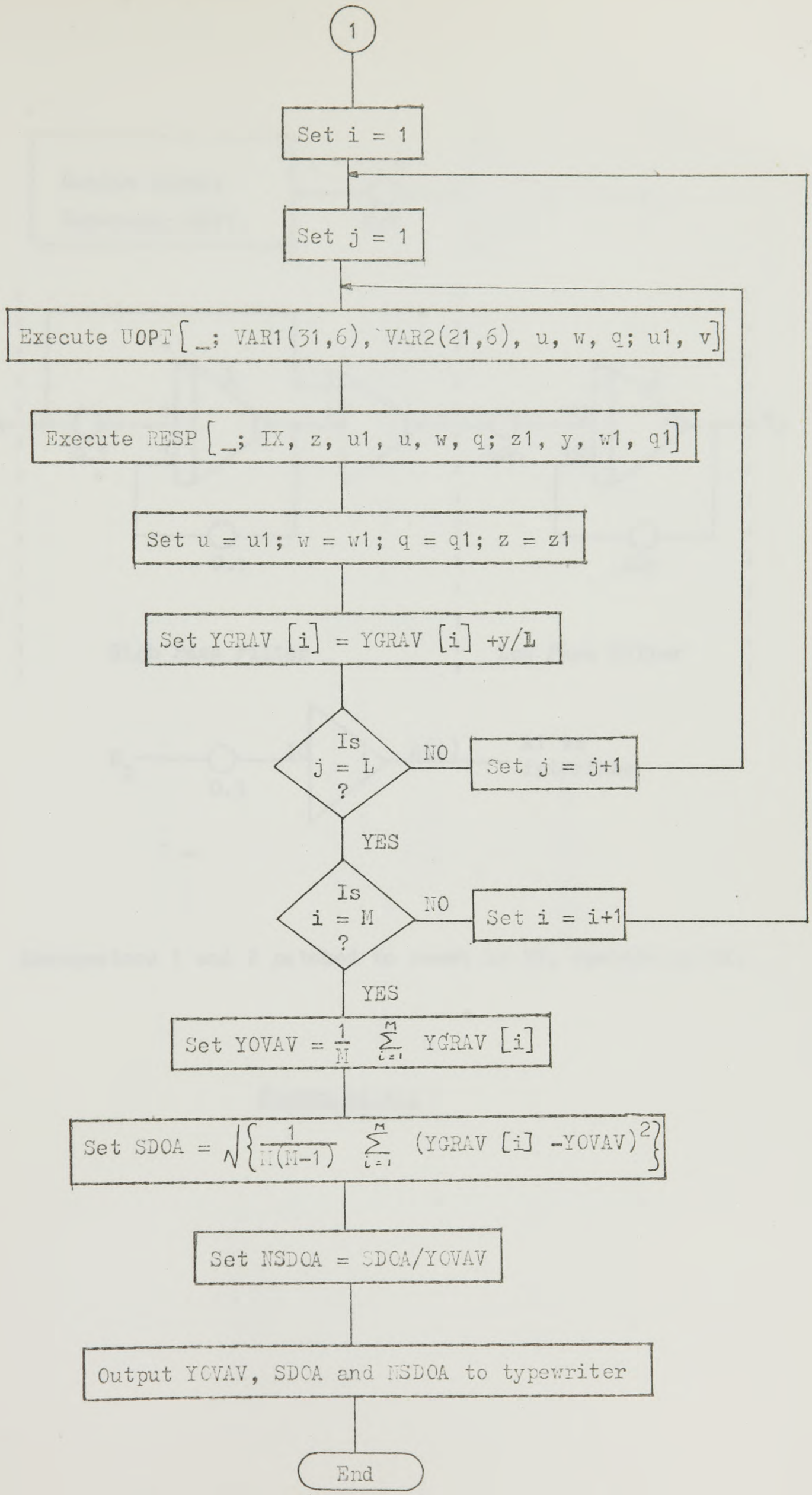
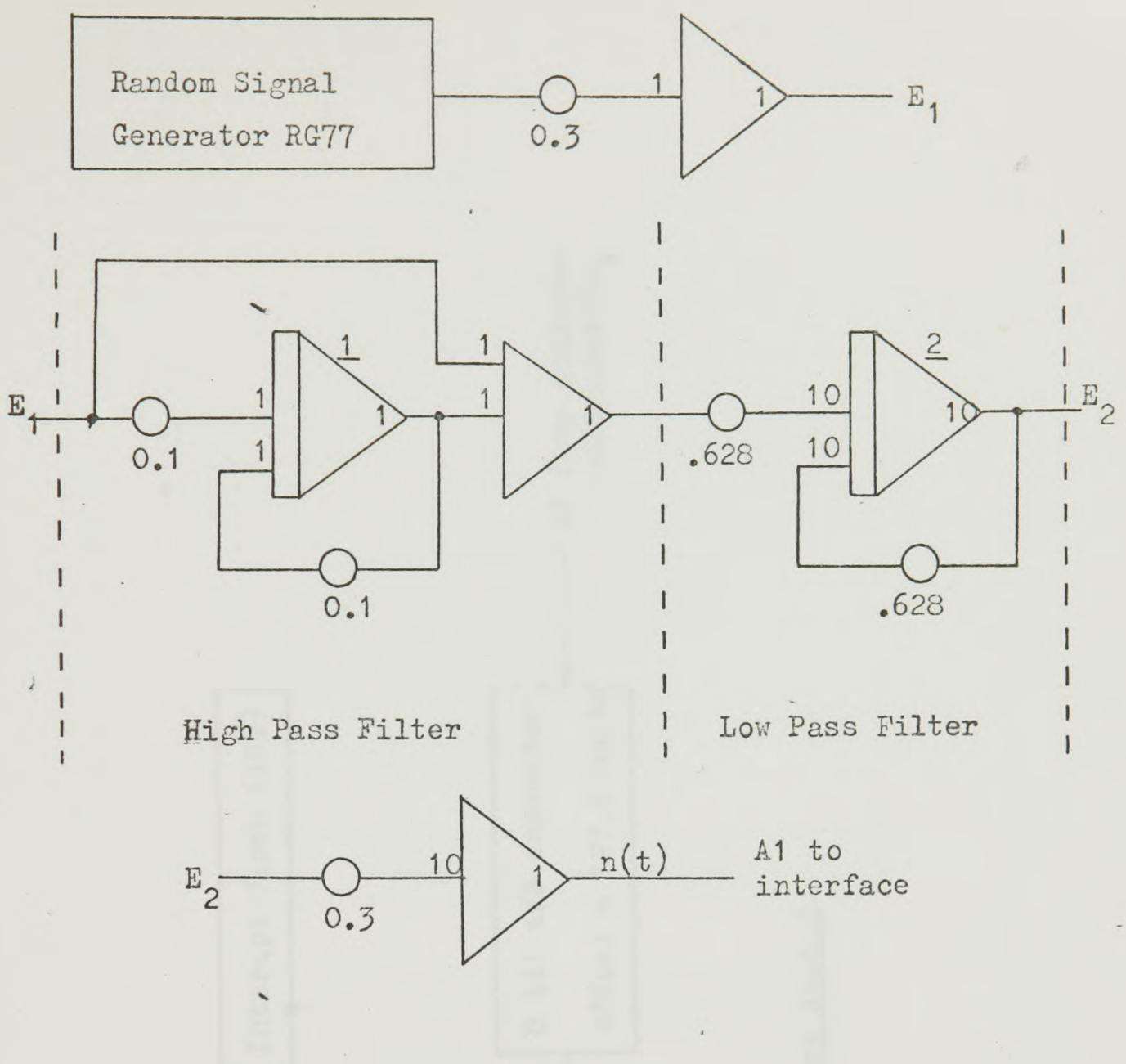

1
Set i = 1
Set j = 1
Execute UOPT [_; VAR1(31,6), VAR2(21,6), u, w, q; u1, v]
Execute RESP [_; IX, z, u1, u, w, q; z1, y, w1, q1]
Set u = u1; w = w1; q = q1; z = z1
Set YGRAV [i] = YGRAV [i] +y/L
Is j = L ?
NO
Set j = j+1
YES
Is i = M ?
NO
Set i = i+1
YES
Set YOVAV = 1/M Σ YGRAV [i]
Set SDOA = √{1/M(M-1) Σ (YGRAV [i] -YOVAV)²}
Set NSDOA = SDOA/YOVAV
Output YOVAV, SDOA and NSDOA to typewriter
End

Figure A4.3.1



Integrators 1 and 2 patched to reset in RS, operate in OP.

Figure A4.4.1

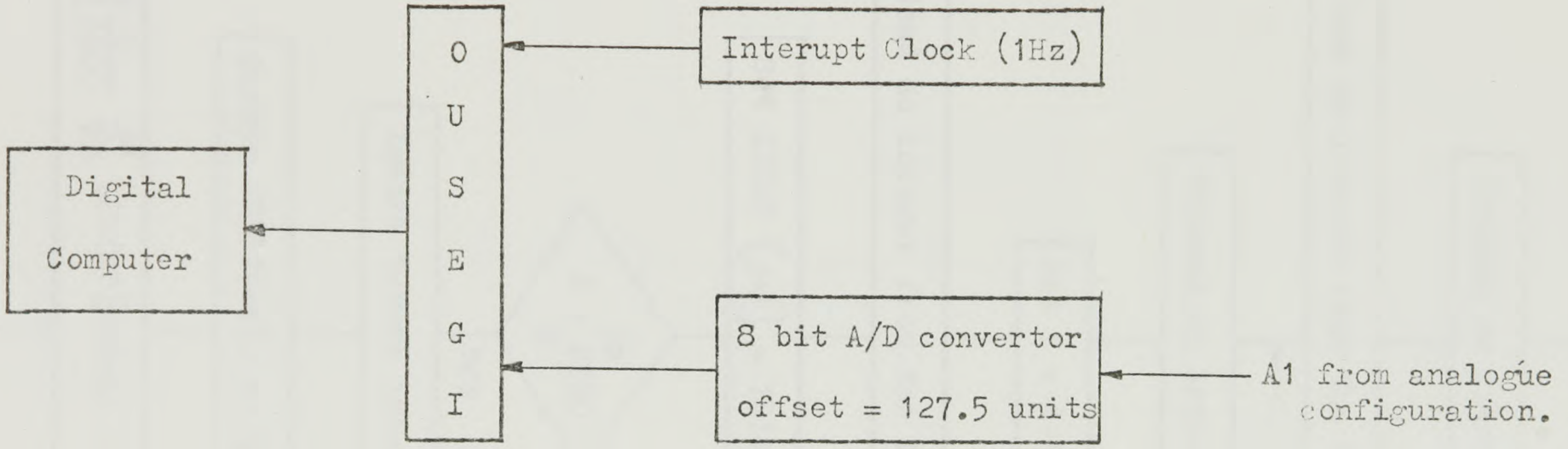

O
U
S
E
G
I
Interupt Clock (1Hz)
Digital
Computer
8 bit A/D convertor
offset = 127.5 units
A1 from analogue
configuration.

Figure A4.4.2.

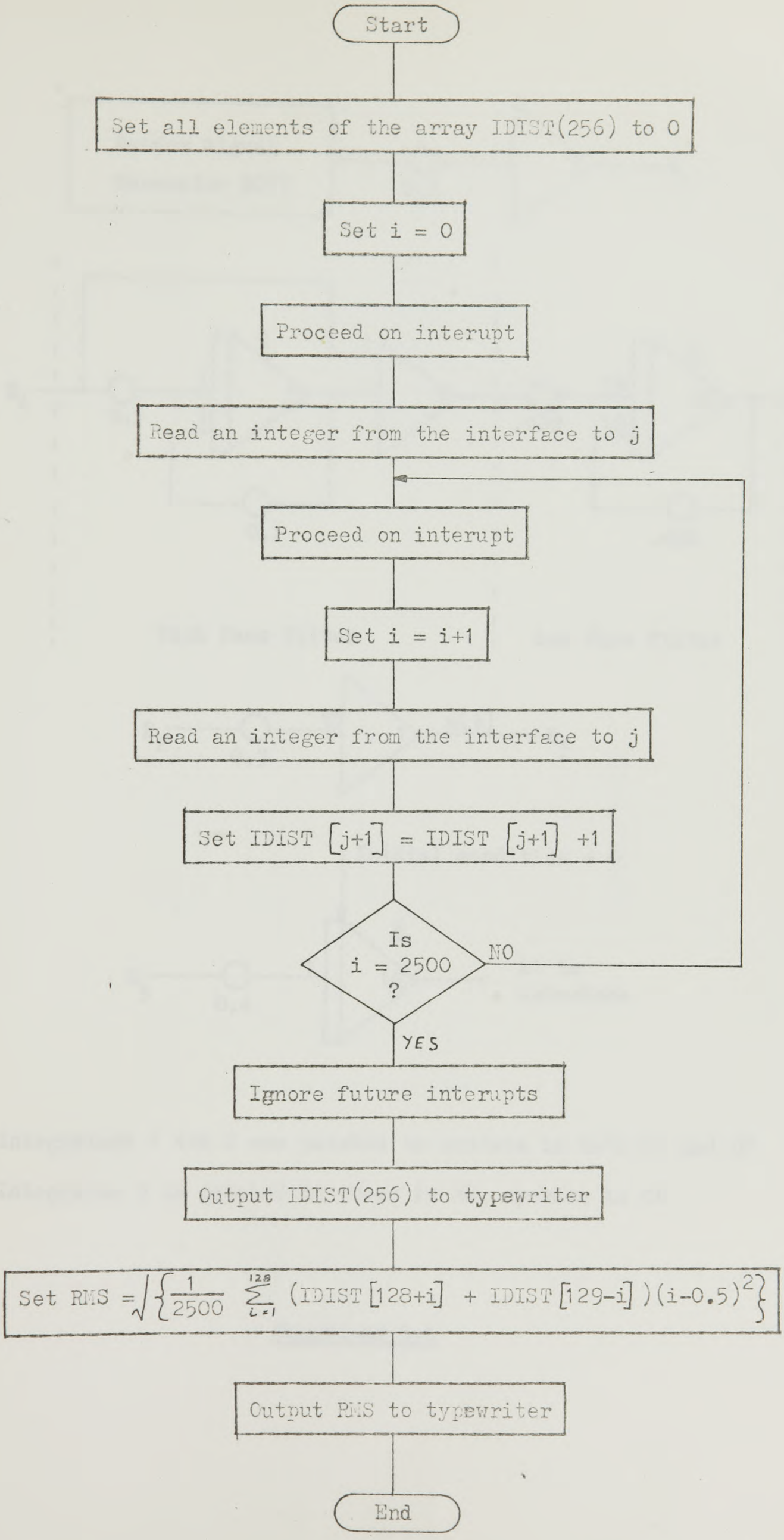
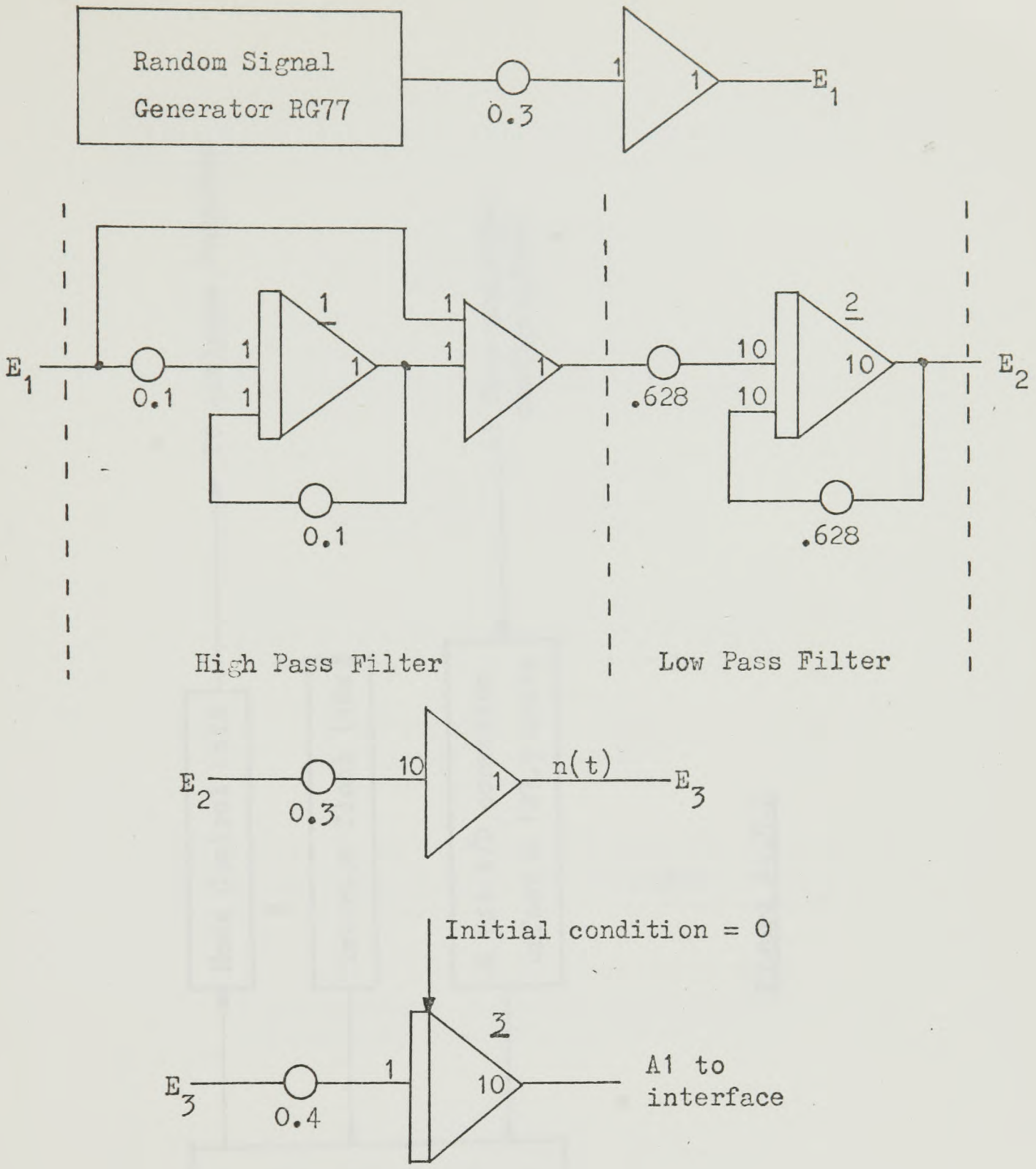

Start
Set all elements of the array IDIST(256) to 0
Set i = 0
Proceed on interupt
Read an integer from the interface to j
Proceed on interupt
Set i = i+1
Read an integer from the interface to j
Set IDIST [j+1] = IDIST [j+1] +1
Is i = 2500 ?
NO
YES
Ignore future interupts
Output IDIST(256) to typewriter
Set RMS = $\sqrt{\left\{\frac{1}{2500}\sum_{i=1}^{128}(IDIST[128+i] + IDIST[129-i])(i-0.5)^2\right\}}$
Output RMS to typewriter
End

Figure A4.4.3



Integrators 1 and 2 are patched to operate in both RS and OP

Integrator 3 is patched to reset in RS, operate in OP

Figure A4.5.1

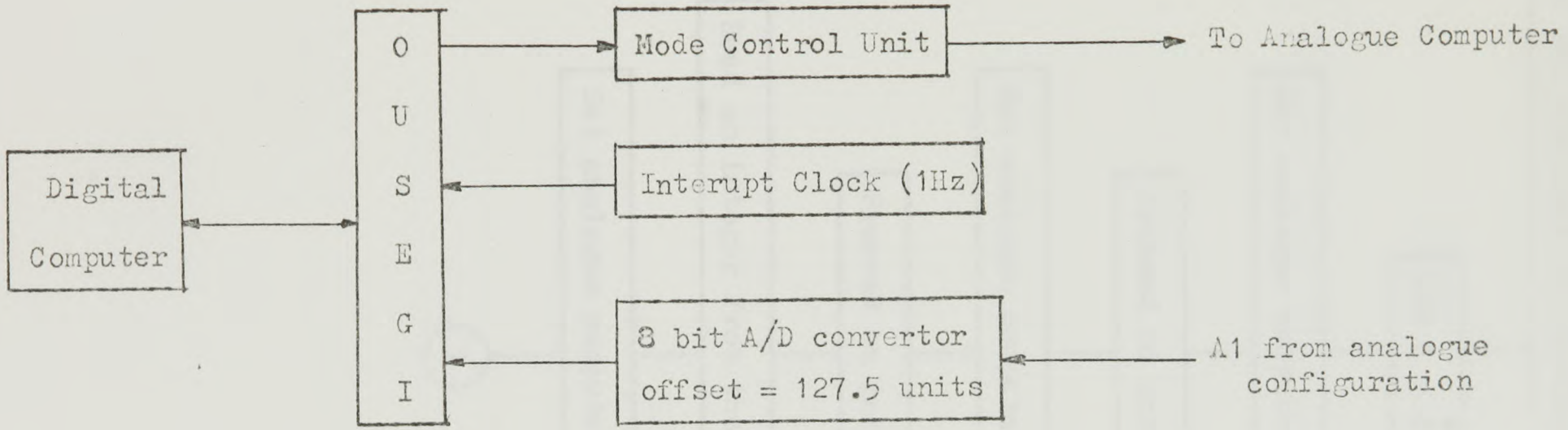
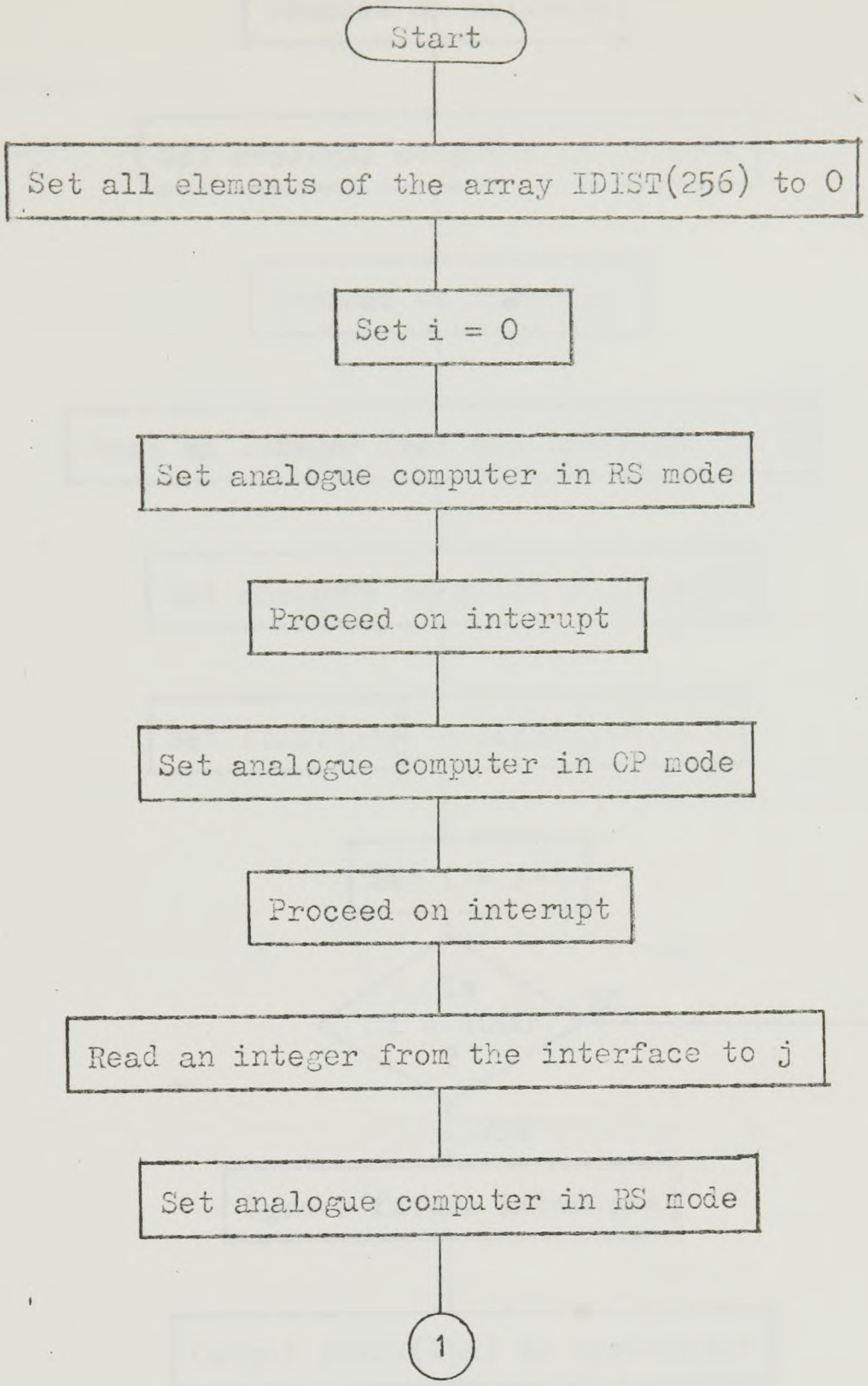

Digital Computer
O U S E G I
Mode Control Unit
To Analogue Computer
Interupt Clock (1Hz)
8 bit A/D convertor
offset = 127.5 units
A1 from analogue configuration

Figure A4.5.2


Start
Set all elements of the array IDIST(256) to 0
Set i = 0
Set analogue computer in RS mode
Proceed on interupt
Set analogue computer in CP mode
Proceed on interupt
Read an integer from the interface to j
Set analogue computer in RS mode
1

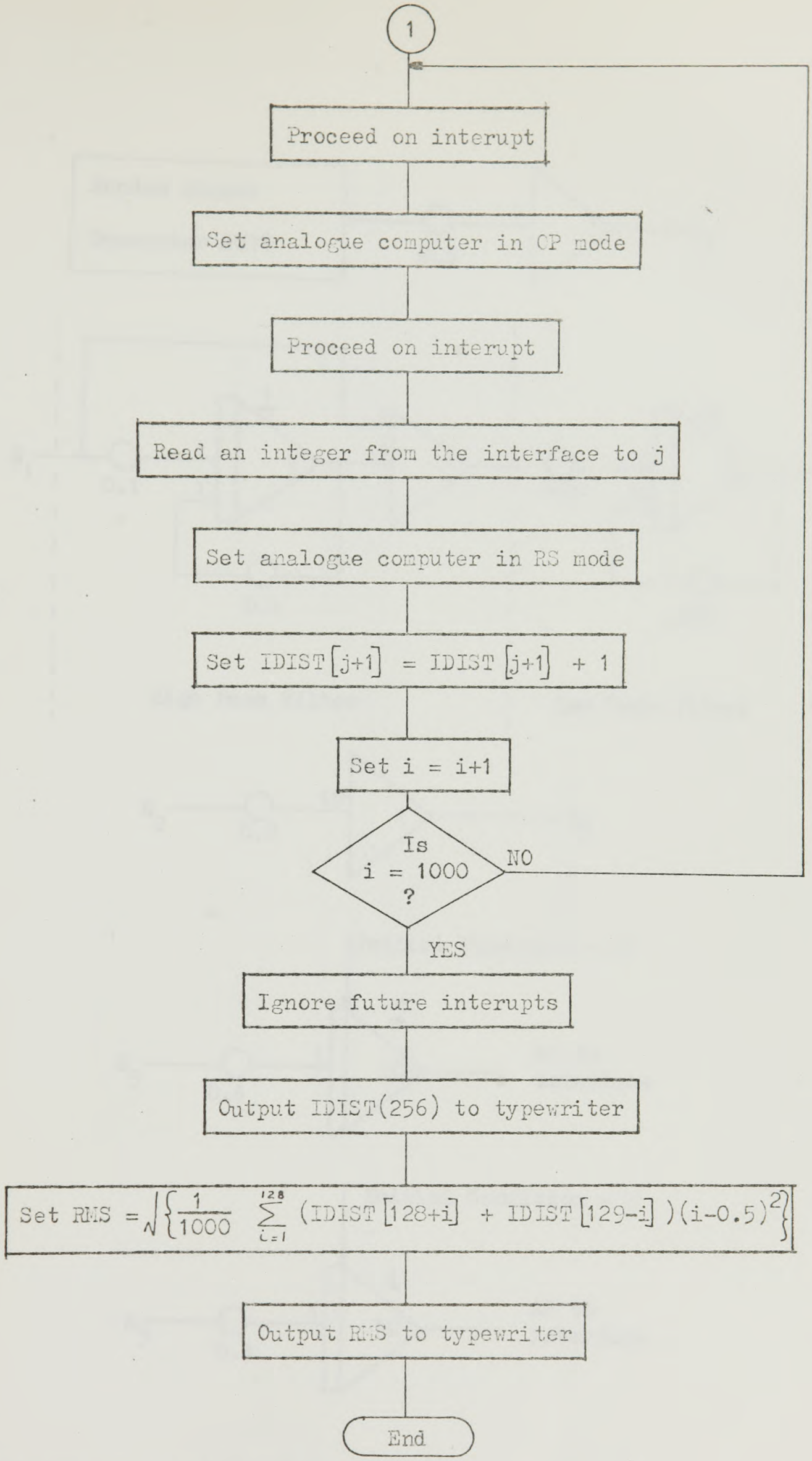
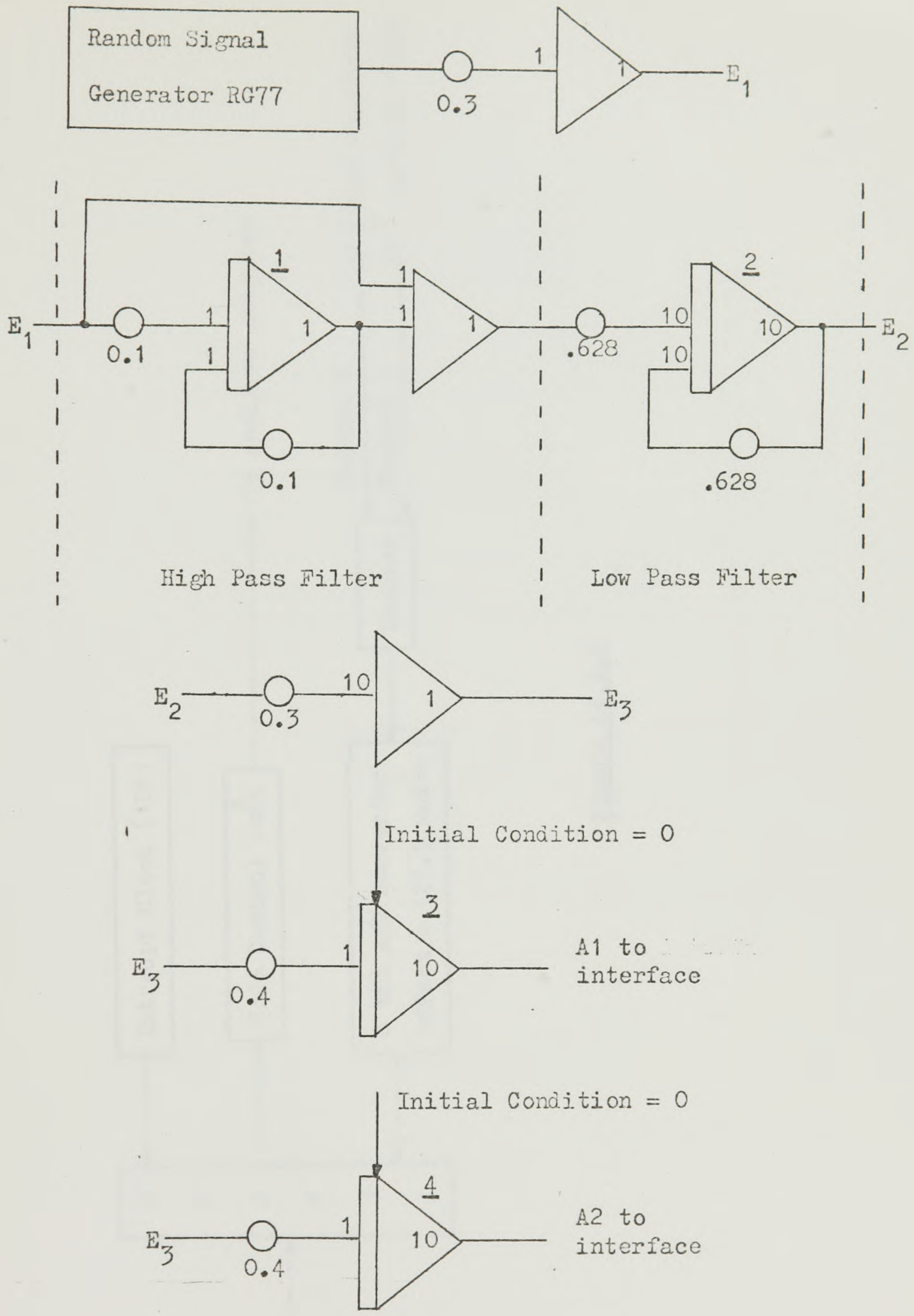

1
Proceed on interupt
Set analogue computer in CP mode
Proceed on interupt
Read an integer from the interface to j
Set analogue computer in RS mode
Set IDIST[j+1] = IDIST[j+1] + 1
Set i = i+1
Is i = 1000 ?
NO
YES
Ignore future interupts
Output IDIST(256) to typewriter
Set RMS = $\sqrt{\left\{\frac{1}{1000}\sum_{i=1}^{128}(IDIST[128+i] + IDIST[129-i])(i-0.5)^2\right\}}$
Output RMS to typewriter
End

Figure A4.5.3



Integrators 1 and 2 set to operate in both RS and OP

Integrator 3 set to operate in OP, reset in RS

Integrator 4 set to operate in RS, reset in OP

Figure A4.6.1

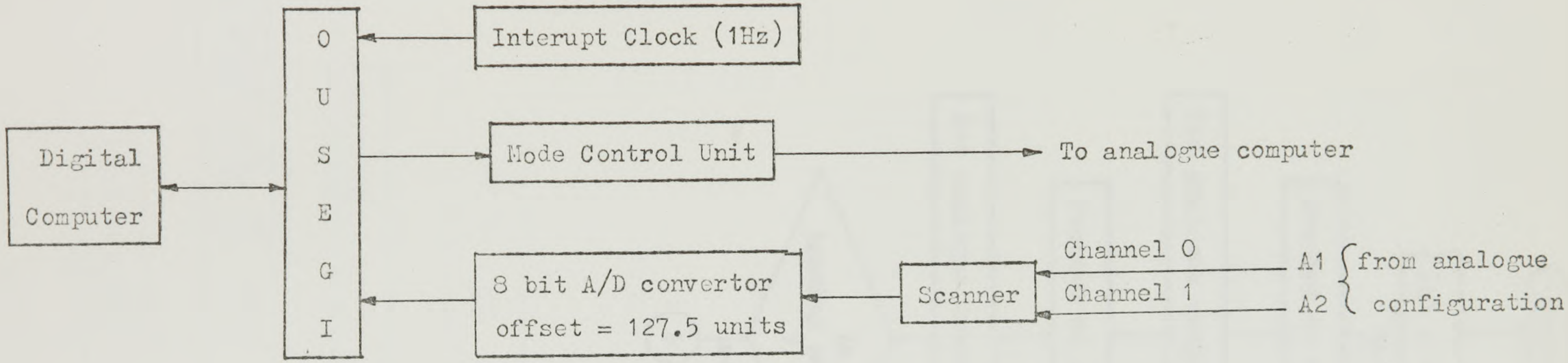
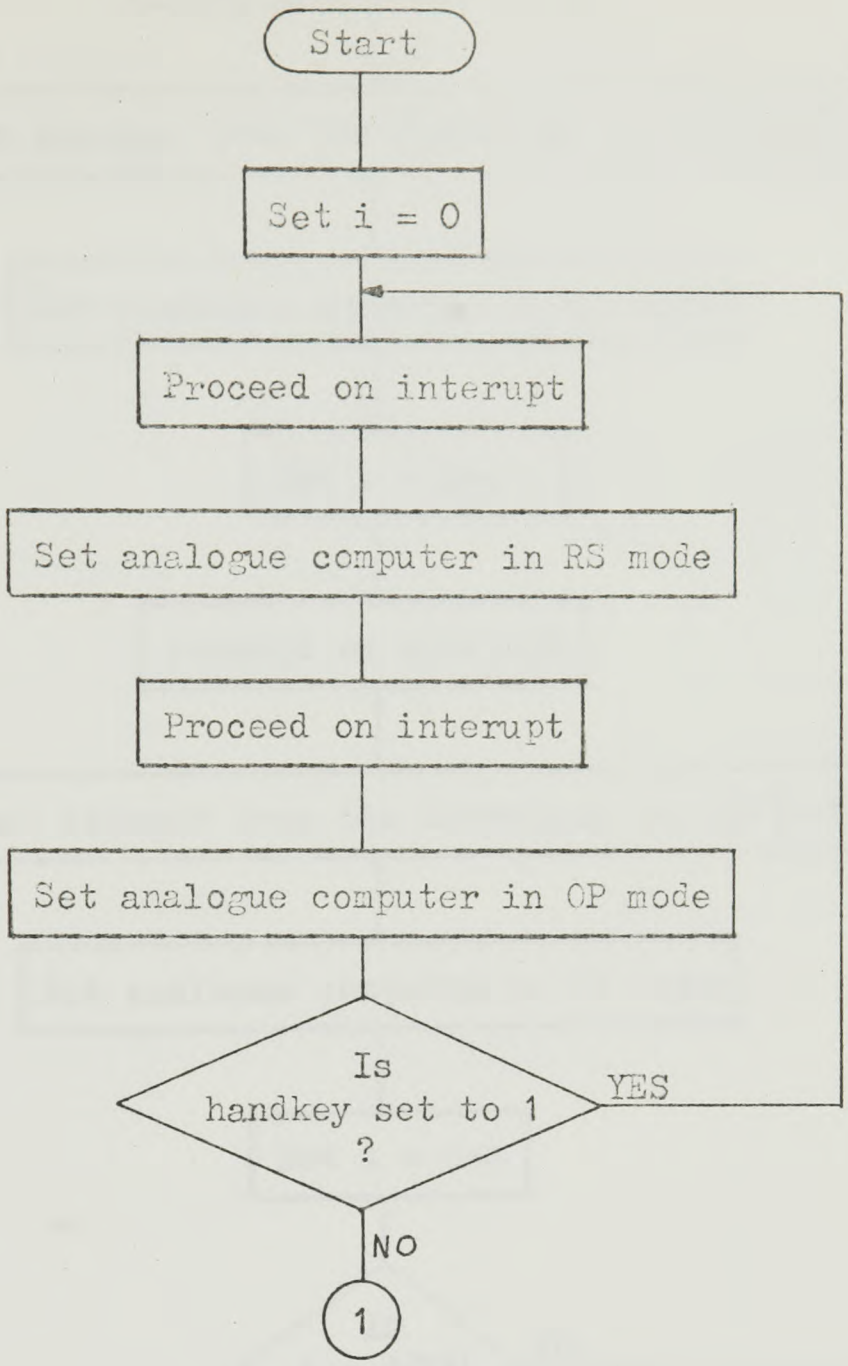

Digital Computer
O U S E G I
Interupt Clock (1Hz)
Mode Control Unit
To analogue computer
8 bit A/D convertor
offset = 127.5 units
Scanner
Channel 0
Channel 1
A1
A2
from analogue configuration

Figure A4.6.2


Start
Set i = 0
Proceed on interupt
Set analogue computer in RS mode
Proceed on interupt
Set analogue computer in OP mode
Is handkey set to 1 ?
YES
NO
1

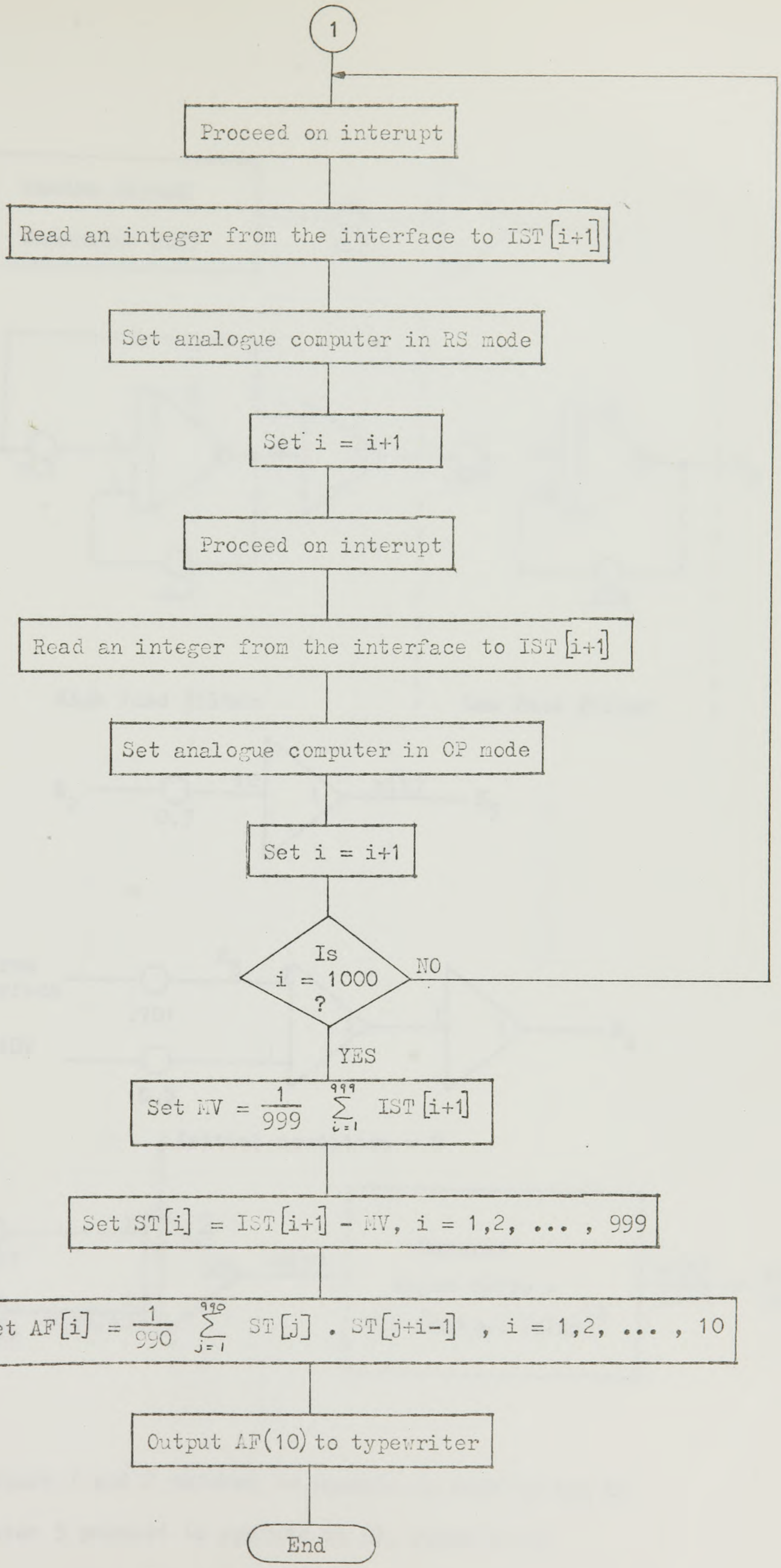
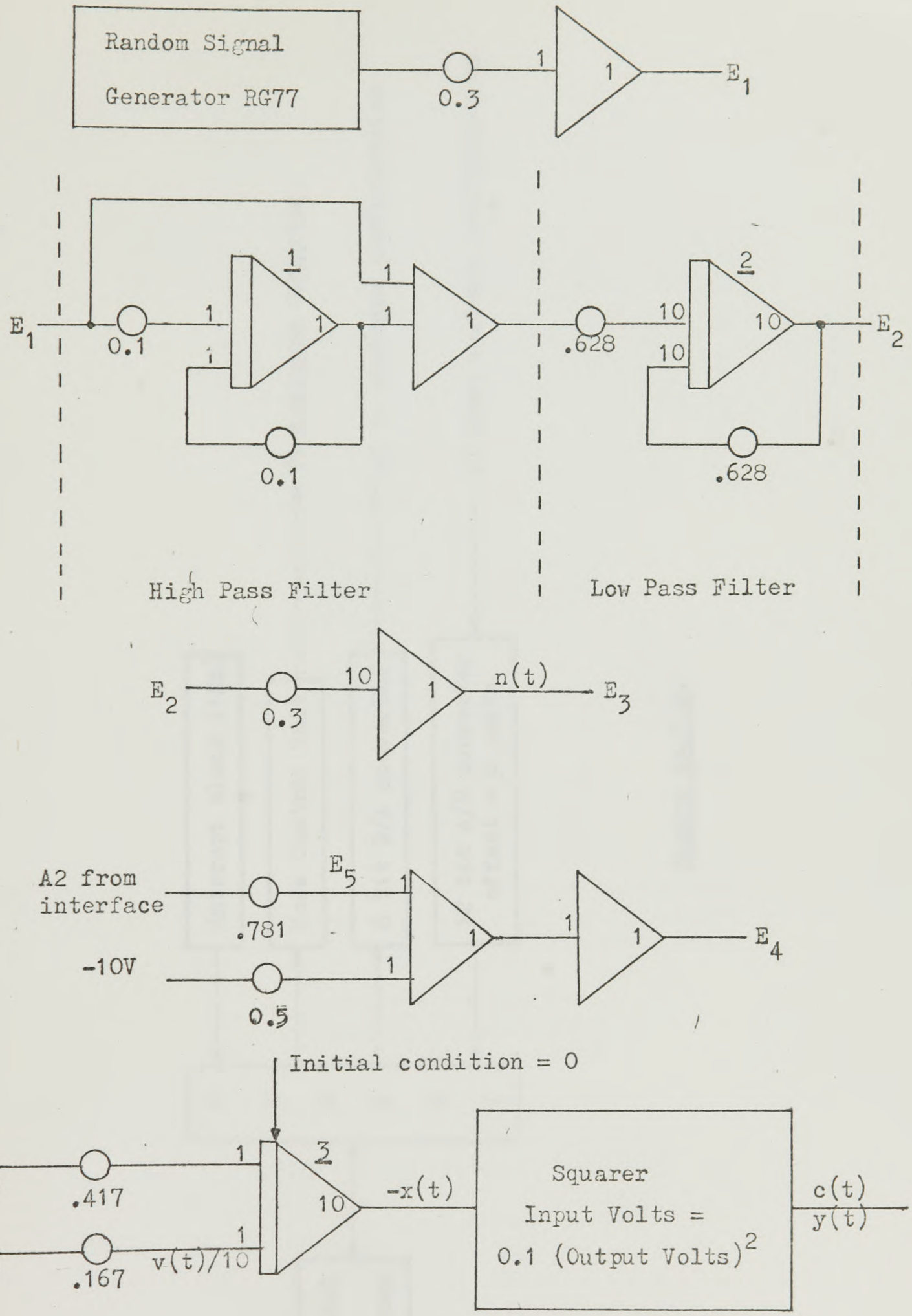

1
Proceed on interupt
Read an integer from the interface to IST [i+1]
Set analogue computer in RS mode
Set i = i+1
Proceed on interupt
Read an integer from the interface to IST [i+1]
Set analogue computer in OP mode
Set i = i+1
Is i = 1000 ?
NO
YES
Set MV = 1/999 Σ (i=1 to 999) IST [i+1]
Set ST[i] = IST[i+1] - MV, i = 1,2, ... , 999
Set AF[i] = 1/990 Σ (j=1 to 990) ST[j] . ST[j+i-1] , i = 1,2, ... , 10
Output AF(10) to typewriter
End

Figure A4.6.3



Integrators 1 and 2 patched to operate in both RS and OP

Integrator 3 patched to operate in OP, reset in RS

Figure A4.7.1

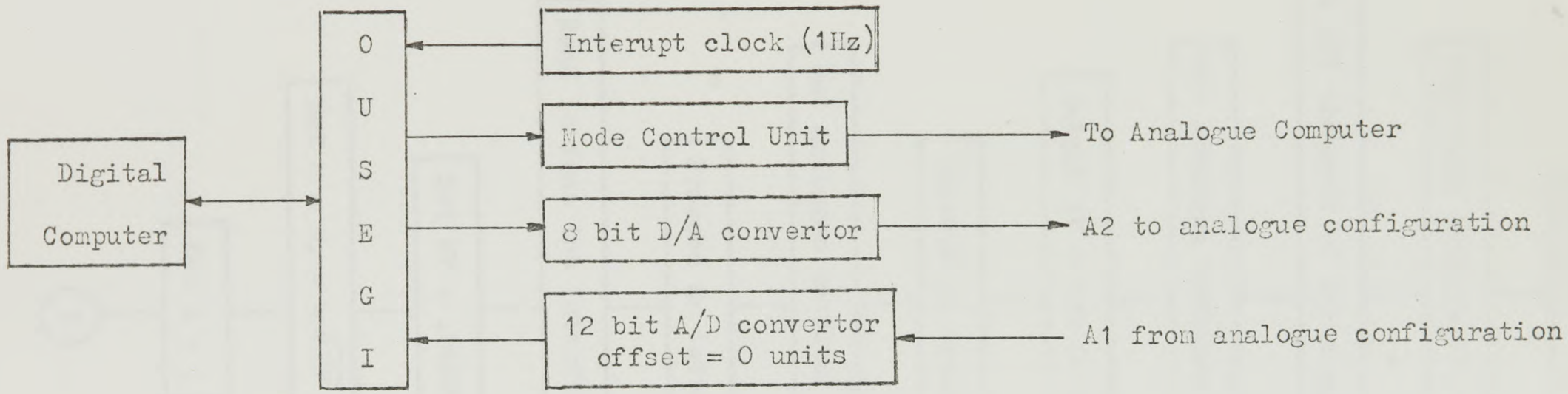
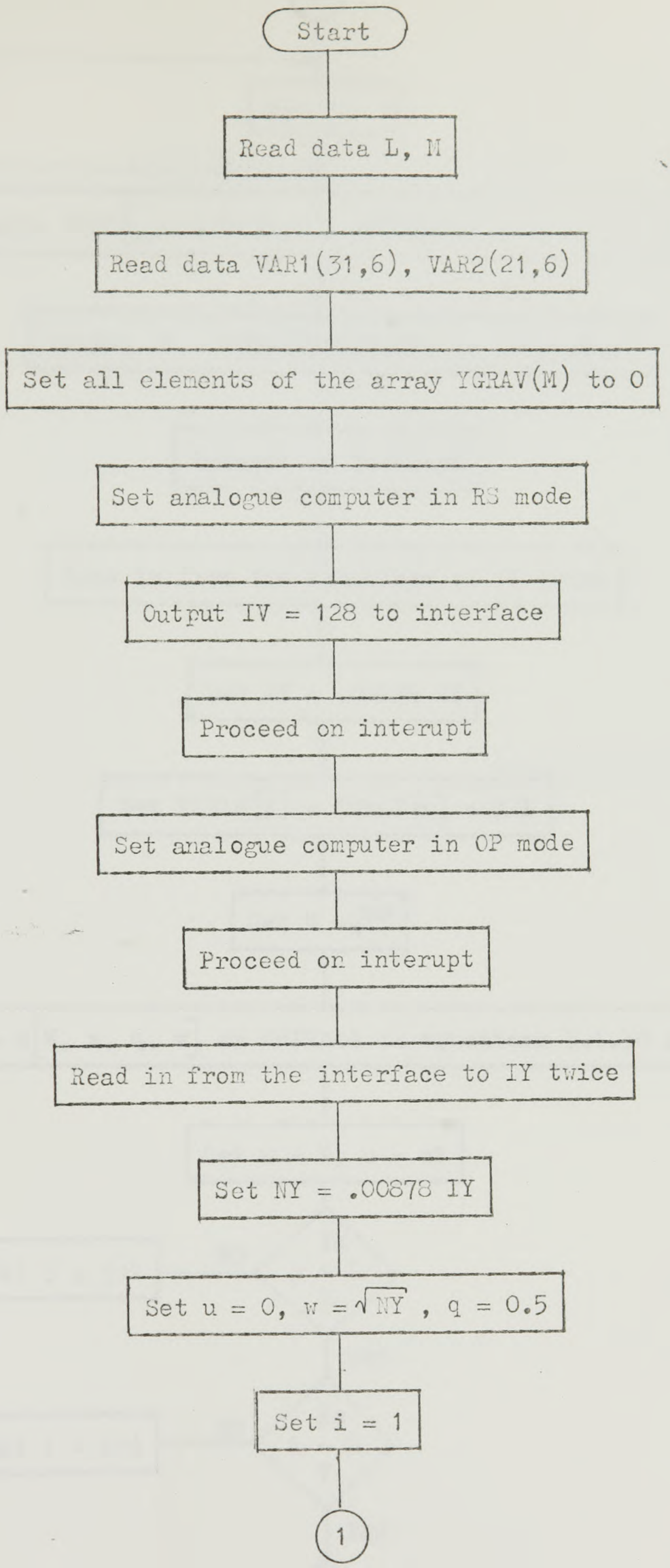
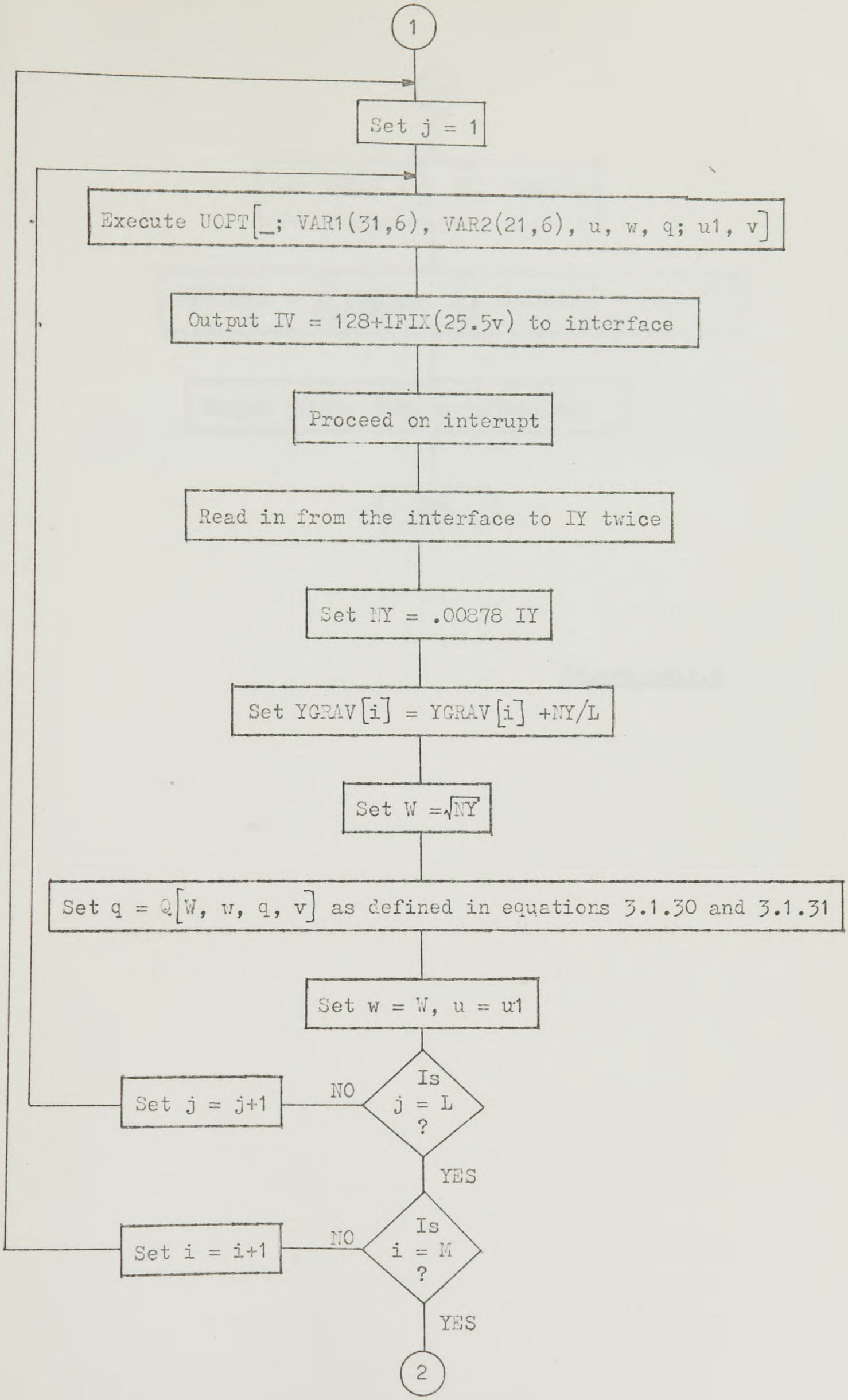

Digital
Computer
O
U
S
E
G
I
Interupt clock (1Hz)
Mode Control Unit
To Analogue Computer
8 bit D/A convertor
A2 to analogue configuration
12 bit A/D convertor
offset = 0 units
A1 from analogue configuration

Figure A4.7.2.


Start
Read data L, M
Read data VAR1(31,6), VAR2(21,6)
Set all elements of the array YGRAV(M) to 0
Set analogue computer in RS mode
Output IV = 128 to interface
Proceed on interupt
Set analogue computer in OP mode
Proceed on interupt
Read in from the interface to IY twice
Set NY = .00878 IY
Set u = 0, w = √NY , q = 0.5
Set i = 1
1


1
Set j = 1
Execute UOPT[_; VAR1(31,6), VAR2(21,6), u, w, q; u1, v]
Output IV = 128+IFIX(25.5v) to interface
Proceed on interupt
Read in from the interface to IY twice
Set NY = .00878 IY
Set YGRAV[i] = YGRAV[i] +NY/L
Set W =√NY
Set q = Q[W, w, q, v] as defined in equations 3.1.30 and 3.1.31
Set w = W, u = u1
Is j = L ?
NO
Set j = j+1
YES
Is i = M ?
NO
Set i = i+1
YES
2

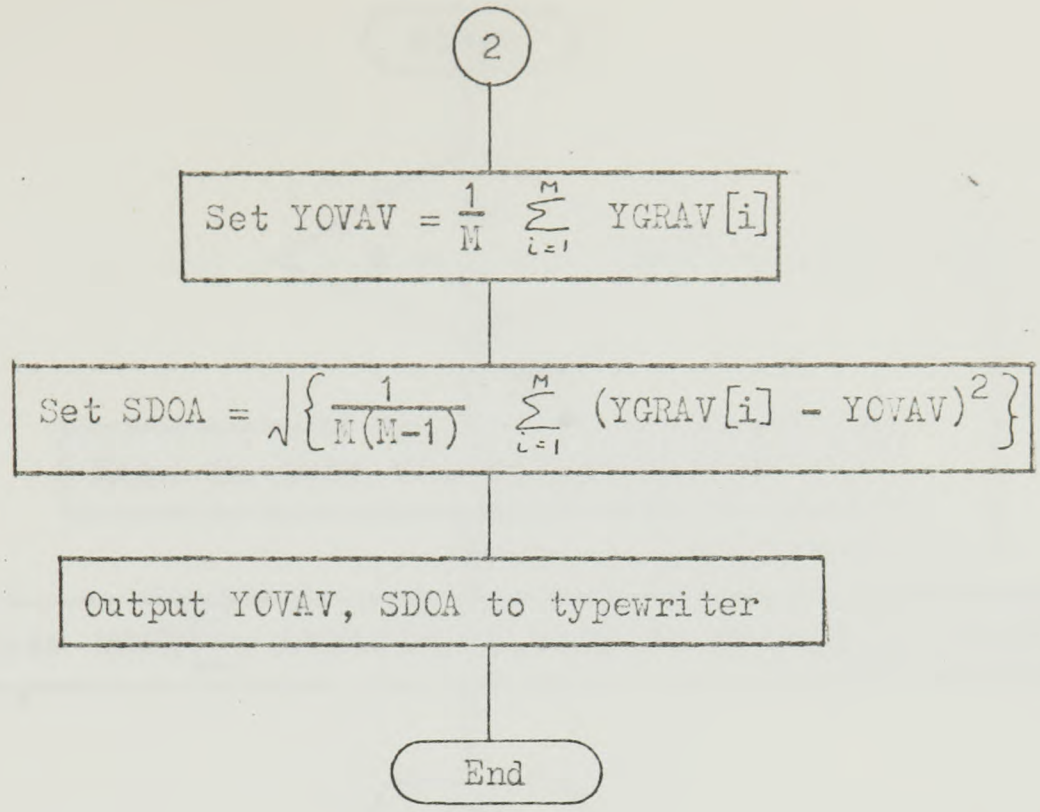

2
Set YOVAV = 1/M Σ YGRAV[i]
Set SDOA = √{1/M(M-1) Σ (YGRAV[i] - YOVAV)²}
Output YOVAV, SDOA to typewriter
End

Figure A4.7.3

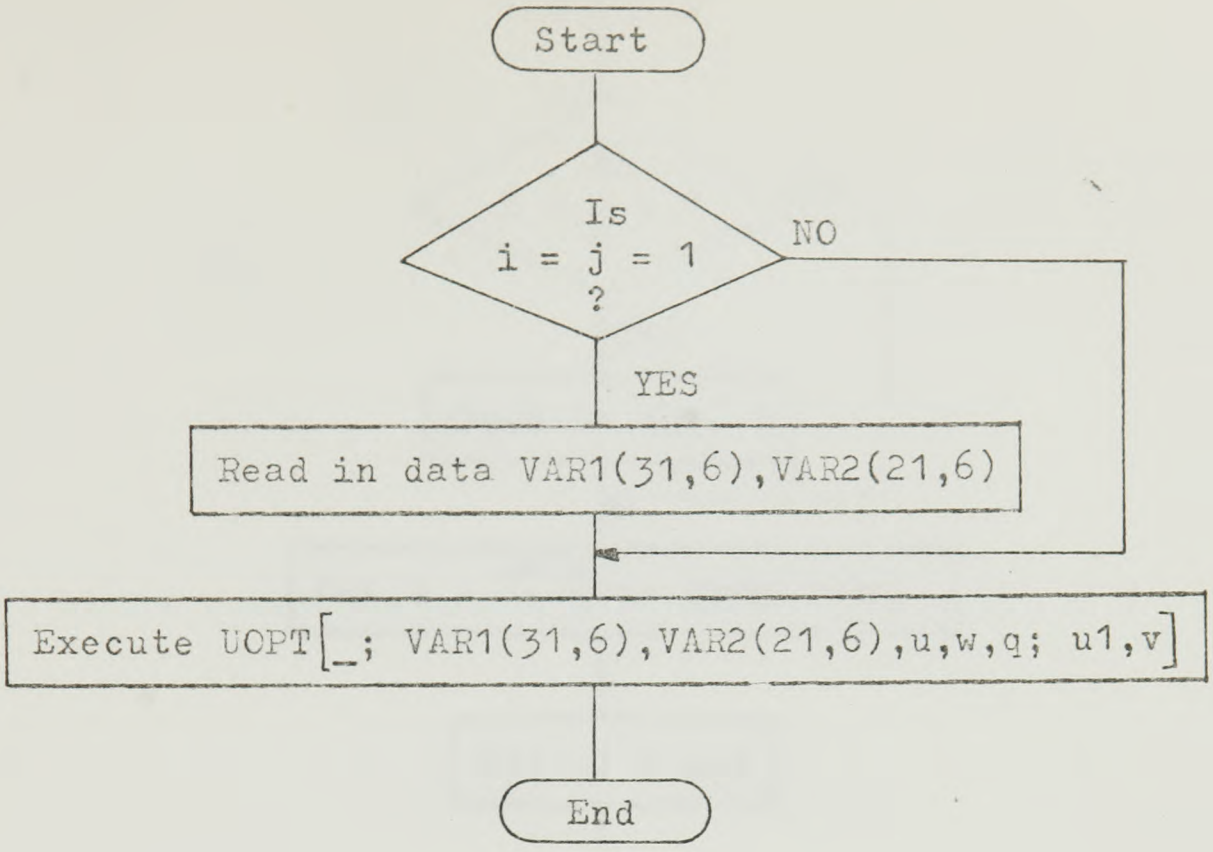

Start
Is
i = j = 1
?
NO
YES
Read in data VAR1(31,6),VAR2(21,6)
Execute UOPT[_; VAR1(31,6),VAR2(21,6),u,w,q; u1,v]
End

Figure A4.8.1

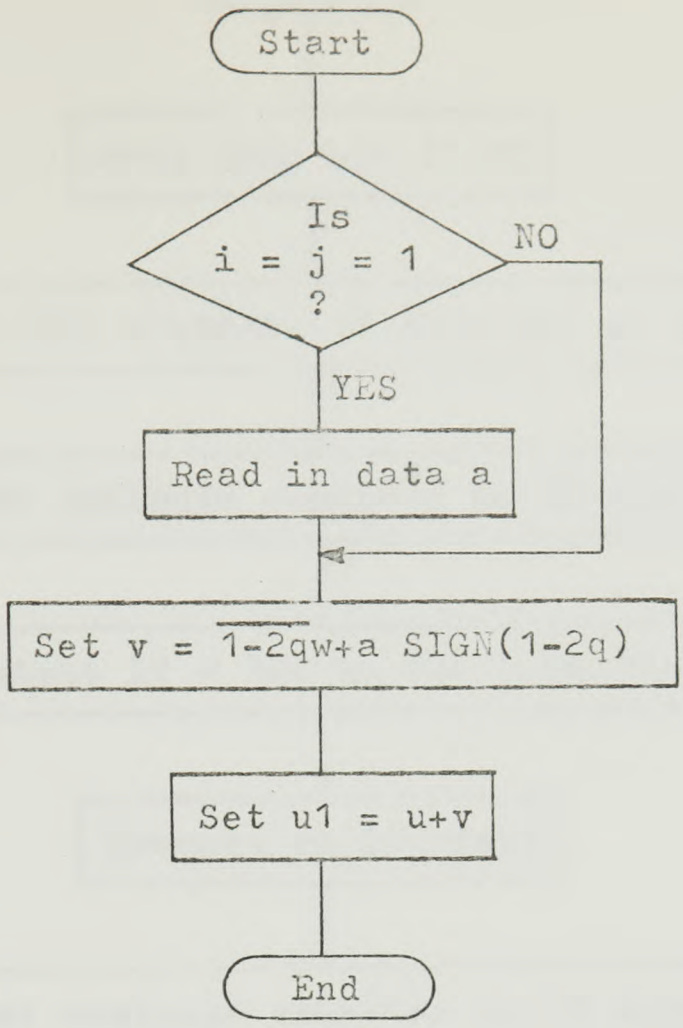
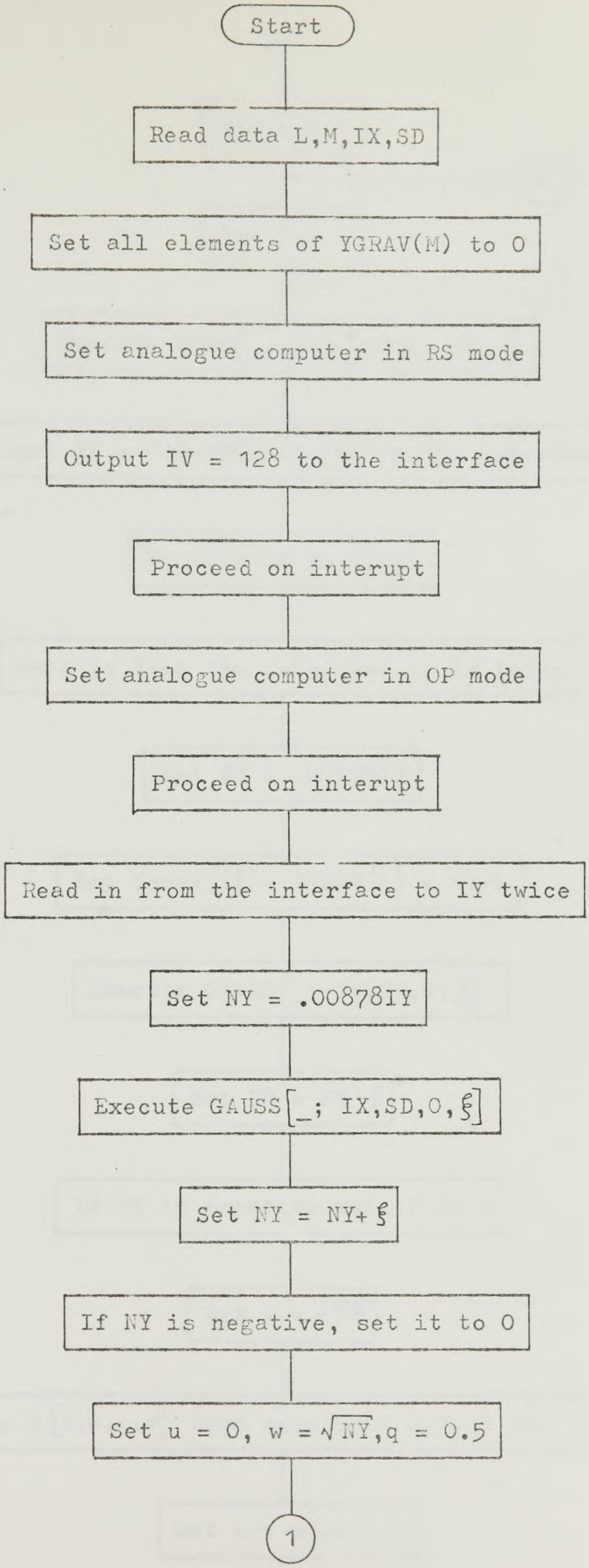
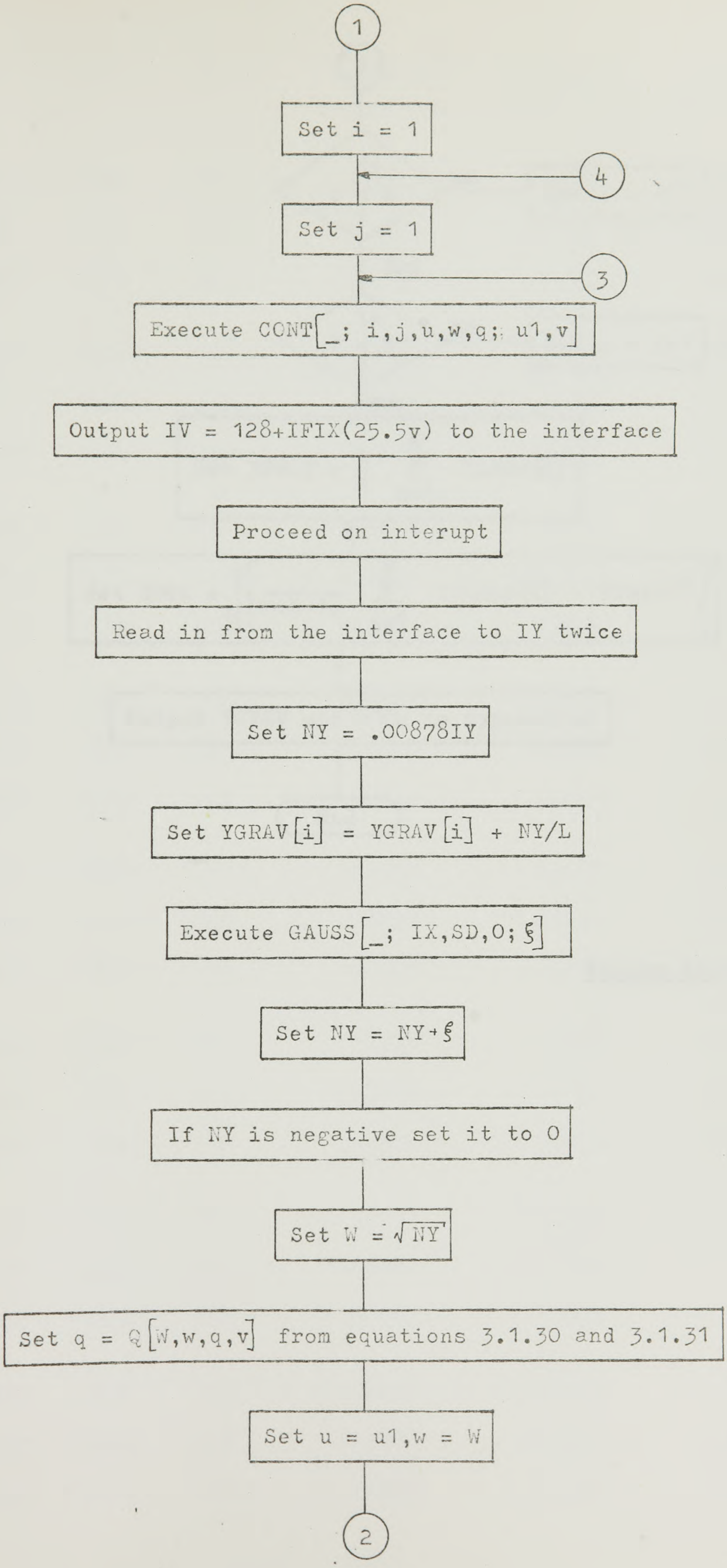

Start
Is
i = j = 1
?
NO
YES
Read in data a
Set v = $\overline{1-2q}w+a$ SIGN(1-2q)
Set u1 = u+v
End

Figure A4.9.1


Start
Read data L,M,IX,SD
Set all elements of YGRAV(M) to 0
Set analogue computer in RS mode
Output IV = 128 to the interface
Proceed on interupt
Set analogue computer in OP mode
Proceed on interupt
Read in from the interface to IY twice
Set NY = .00878IY
Execute GAUSS[_; IX,SD,0,ξ]
Set NY = NY+ξ
If NY is negative, set it to 0
Set u = 0, w = √NY, q = 0.5
1


1
Set i = 1
4
Set j = 1
3
Execute CONT[_; i,j,u,w,q; u1,v]
Output IV = 128+IFIX(25.5v) to the interface
Proceed on interupt
Read in from the interface to IY twice
Set NY = .00878IY
Set YGRAV[i] = YGRAV[i] + NY/L
Execute GAUSS[_; IX,SD,0; §]
Set NY = NY+§
If NY is negative set it to 0
Set W = √NY
Set q = Q[W,w,q,v] from equations 3.1.30 and 3.1.31
Set u = u1,w = W
2

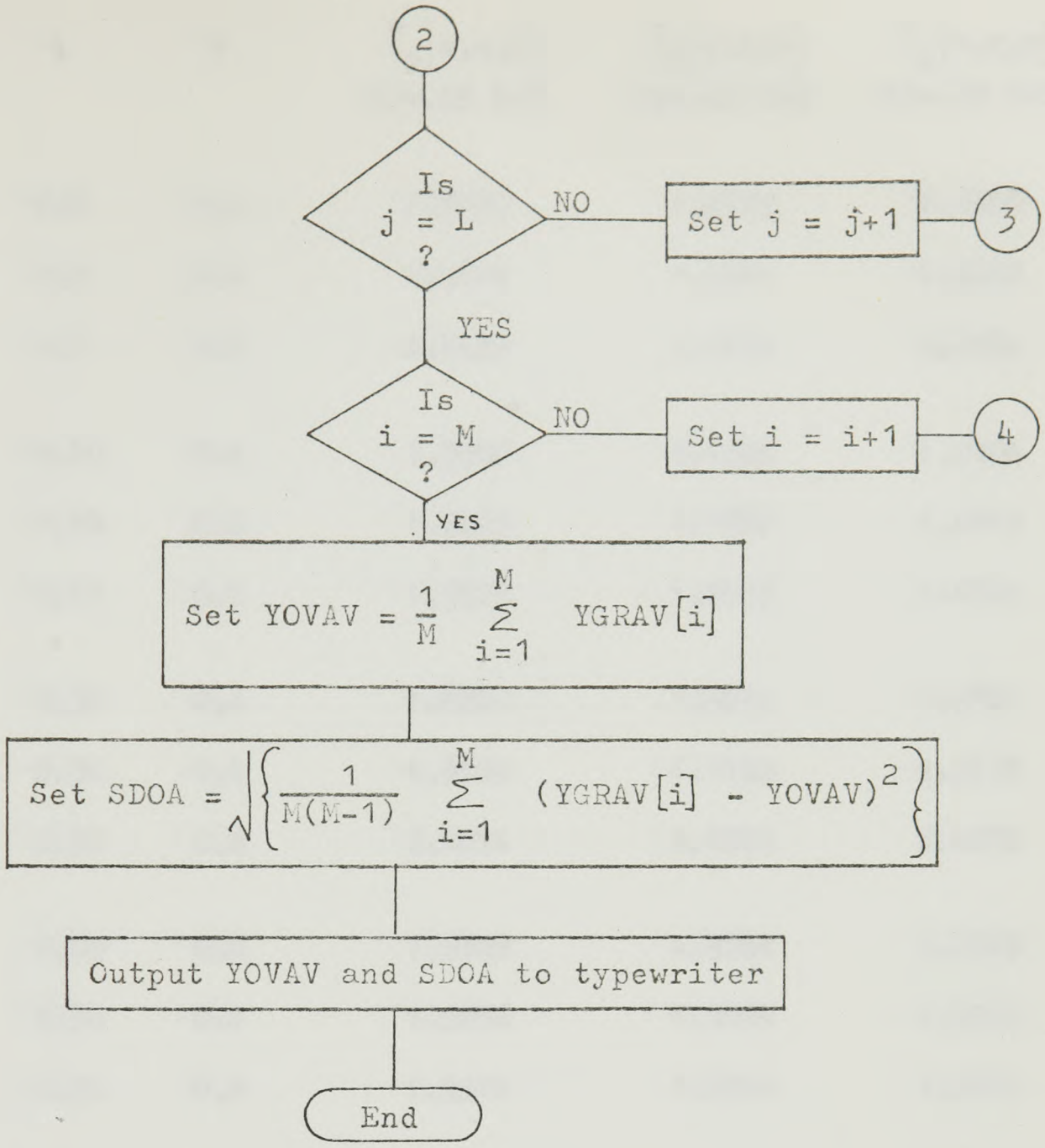


Figure A4.10.1

w	q	v	$\bar{I}_2[w,q,v]$ dr=.05 R=5	$\bar{I}_2[w,q,v]$ dr=.02 R=5	$\bar{I}_2[w,q,v]$ dr=.05 R=10
0.0	0.0	0.2	1.9590	1.9589	1.9589
0.0	0.0	0.4	1.9062	1.9061	1.9062
0.0	0.0	0.6	1.9134	1.9133	1.9134
0.10	0.10	0.4	1.9306	1.9305	1.9306
0.10	0.10	0.6	1.9063	1.9062	1.9063
0.10	0.10	0.8	1.9725	1.9723	1.9724
0.10	0.30	0.4	1.9300	1.9299	1.9300
0.10	0.30	0.6	1.9199	1.9198	1.9199
0.10	0.30	0.8	2.0076	2.0074	2.0075
0.10	0.50	0.2	1.9789	1.9788	1.9788
0.10	0.50	0.4	1.9256	1.9255	1.9256
0.10	0.50	0.6	1.9318	1.9316	1.9317
0.20	0.10	0.4	1.9740	1.9739	1.9740
0.20	0.10	0.6	1.9268	1.9267	1.9268
0.20	0.10	0.8	1.9508	1.9507	1.9507
0.20	0.30	0.4	1.9876	1.9875	1.9876
0.20	0.30	0.6	1.9607	1.9606	1.9607
0.20	0.30	0.8	2.0231	2.0229	2.0230
0.20	0.50	0.2	2.0385	2.0384	2.0385
0.20	0.50	0.4	1.9837	1.9835	1.9836
0.20	0.50	0.6	1.9867	1.9865	1.9866
0.50	0.10	0.8	2.0535	2.0533	2.0534
0.50	0.10	1.0	2.0533	2.0532	2.0533
0.50	0.10	1.2	2.1563	2.1561	2.1562

w	q	v	$\bar{I}_2[w,q,v]$ dr=.05 R=5	$\bar{I}_2[w,q,v]$ dr=.02 R=5	$\bar{I}_2[w,q,v]$ dr=.05 R=10
0.50	0.30	0.6	2.2770	2.2769	2.2770
0.50	0.30	0.8	2.2640	2.2638	2.2639
0.50	0.30	1.0	2.3515	2.3513	2.3514
0.50	0.50	0.4	2.3858	2.3857	2.3858
0.50	0.50	0.6	2.3672	2.3670	2.3671
0.50	0.50	0.8	2.4446	2.4444	2.4445
1.20	0.10	1.2	2.6518	2.6517	2.6517
1.20	0.10	1.4	2.5977	2.5976	2.5977
1.20	0.10	1.6	2.6134	2.6132	2.6133
1.20	0.30	1.0	3.7032	3.7030	3.7031
1.20	0.30	1.2	3.6903	3.6901	3.6902
1.20	0.30	1.4	—	3.7936	3.7938
1.20	0.50	0.6	4.4514	4.4512	4.4514
1.20	0.50	0.8	4.3846	4.3843	4.3845
1.20	0.50	1.0	4.4409	4.4406	4.4408
3.00	0.10	2.0	5.0616	5.0616	5.0617
3.00	0.10	2.2	4.8906	4.8907	4.8907
3.00	0.10	2.4	4.8919	4.8920	4.8920
3.00	0.30	1.8	10.0249	10.0248	10.0250
3.00	0.30	2.0	10.0058	10.0057	10.0059
3.00	0.30	2.2	10.2104	10.2103	10.2105
3.00	0.50	1.4	14.2213	14.2210	14.2214
3.00	0.50	1.6	14.1316	14.1316	14.1317
3.00	0.50	1.8	14.3219	14.3218	14.3220

Table 3.3.2.1

$$I_2[w,q] - I_2[0,0]$$

w	q	A	B	C	D
0.0	—	0.0000	0.0000	0.0000	0.0000
0.1	0.1	0.0064	0.0065	0.0066	0.0064
0.1	0.3	0.0160	0.0160	0.0156	0.0161
0.1	0.5	0.0190	0.0190	0.0190	0.0190
0.2	0.1	0.0260	0.0260	0.0254	0.0253
0.2	0.3	0.0613	0.0615	0.0614	0.0613
0.2	0.5	0.0758	0.0760	0.0758	0.0759
0.5	0.1	0.1439	0.1443	0.1439	0.1439
0.5	0.3	0.3604	0.3613	0.3600	0.3604
0.5	0.5	0.4661	0.4671	0.4656	0.4661
1.2	0.1	0.6951	0.6956	0.6951	0.6951
1.2	0.3	1.7834	1.7859	1.7832	1.7834
1.2	0.5	2.4857	2.4889	2.4857	2.4856
3.0	0.1	2.9726	2.9744	2.9725	2.9723
3.0	0.3	8.0861	8.0881	8.0861	8.0852
3.0	0.5	12.2278	12.2297	12.2276	12.2265
I_2 0,0		1,8986	1.9043	1.8987	1.9008

NOTE

A Initial values of approximation parameters used

B dq set to 0.05

C dv set to 0.05

D $f_1[w,q]$ interpolated

Table 3.3.2.2

		$v_2[w,q]$			
w	q	A	B	C	D
0.0	—	0.4836	0.4809	0.4870	0.4840
0.1	0.1	0.5659	0.5630	0.5681	0.5662
0.1	0.3	0.5259	0.5225	0.5294	0.5263
0.1	0.5	0.4859	0.4832	0.4895	0.4863
0.2	0.1	0.6499	0.6460	0.6499	0.6503
0.2	0.3	0.5725	0.5697	0.5756	0.5728
0.2	0.5	0.4930	0.4903	0.4965	0.4933
0.5	0.1	0.9003	0.8973	0.9034	0.9006
0.5	0.3	0.7311	0.7274	0.7333	0.7314
0.5	0.5	0.5487	0.5449	0.5488	0.5491
1.2	0.1	1.4625	1.4607	1.4634	1.4628
1.2	0.3	1.1229	1.1203	1.1248	1.1231
1.2	0.5	0.8135	0.8105	0.8147	0.8137
3.0	0.1	2.2983	2.3012	2.2971	2.2981
3.0	0.3	1.9159	1.9163	1.9151	1.9159
3.0	0.5	1.5614	1.5621	1.5612	1.5614

NOTE A Initial values of approximation parameters used

B dq set to 0.05

C dv set to 0.05

D $f_1[w,q]$ interpolated.

Table 3.3.2.3

w	q	$I_2[w,q]$	$I_3[w,q]$	$I_4[w,q]$	$I_5[w,q]$	$I_6[w,q]$	$I_7[w,q]$
0.00	0.10	1.899	2,109	2.162	2.168	2.168	2.168
0.00	0.20	1.899	2.109	2.162	2.168	2.168	2.168
0.00	0.30	1.899	2.109	2.162	2.168	2.168	2.168
0.00	0.40	1.899	2.109	2.162	2.168	2.168	2.168
0.00	0.50	1.899	2.109	2.162	2.168	2.168	2.168
0.10	0.10	1.906	2.116	2.169	2.175	2.175	2.175
0.10	0.20	1.911	2.122	2.174	2.181	2.181	2.181
0.10	0.30	1.914	2.126	2.178	2.185	2.185	2.185
0.10	0.40	1.917	2.128	2.181	2.188	2.188	2.187
0.10	0.50	1.918	2.129	2.182	2.188	2.188	2.188
0.20	0.10	1.924	2.136	2.189	2.196	2.196	2.196
0.20	0.20	1.944	2.158	2.211	2.218	2.218	2.218
0.20	0.30	1.960	2.174	2.228	2.234	2.234	2.234
0.20	0.40	1.970	2.185	2.239	2.245	2.245	2.245
0.20	0.50	1.975	2.189	2.243	2.250	2.250	2.250
0.30	0.10	1.954	2.168	2.221	2.228	2.228	2.228
0.30	0.20	1.999	2.216	2.270	2.277	2.277	2.277
0.30	0.30	2.034	2.252	2.308	2.314	2.314	2.314
0.30	0.40	2.057	2.277	2.333	2.340	2.340	2.339
0.30	0.50	2.068	2.288	2.345	2.351	2.351	2.351
0.40	0.10	1.994	2.210	2.264	2.271	2.271	2.271
0.40	0.20	2.073	2.294	2.350	2.357	2.356	2.356
0.40	0.30	2.134	2.359	2.416	2.423	2.423	2.422
0.40	0.40	2.177	2.403	2.462	2.469	2.468	2.468
0.40	0.50	2.199	2.426	2.485	2.491	2.491	2.491

w	q	$I_2[w,q]$	$I_3[w,q]$	$I_4[w,q]$	$I_5[w,q]$	$I_6[w,q]$	$I_7[w,q]$
0.50	0.10	2.043	2.262	2.317	2.324	2.324	2.324
0.50	0.20	2.163	2.390	2.448	2.455	2.455	2.454
0.50	0.30	2.259	2.491	2.551	2.557	2.557	2.557
0.50	0.40	2.327	2.562	2.623	2.630	2.630	2.630
0.50	0.50	2.365	2.600	2.662	2.669	2.669	2.669
0.60	0.10	2.100	2.323	2.379	2.386	2.386	2.386
0.60	0.20	2.269	2.503	2.563	2.569	2.569	2.569
0.60	0.30	2.406	2.647	2.709	2.716	2.716	2.716
0.60	0.40	2.506	2.751	2.815	2.822	2.822	2.821
0.60	0.50	2.564	2.810	2.875	2.882	2.882	2.881
0.70	0.10	2.164	2.392	2.450	2.456	2.456	2.456
0.70	0.20	2.390	2.632	2.693	2.700	2.700	2.700
0.70	0.30	2.574	2.826	2.890	2.897	2.897	2.897
0.70	0.40	2.712	2.969	3.036	3.042	3.042	3.042
0.70	0.50	2.795	3.053	3.122	3.128	3.128	3.128
0.80	0.10	2.237	2.470	2.528	2.535	2.535	2.534
0.80	0.20	2.525	2.777	2.839	2.846	2.845	2.845
0.80	0.30	2.762	3.026	3.092	3.099	3.099	3.098
0.80	0.40	2.943	3.214	3.283	3.289	3.289	3.288
0.80	0.50	3.058	3.329	3.401	3.407	3.407	3.406
0.90	0.10	2.316	2.555	2.614	2.621	2.620	2.620
0.90	0.20	2.672	2.935	2.999	3.005	3.005	3.005
0.90	0.30	2.968	3.247	3.314	3.320	3.320	3.320
0.90	0.40	3.197	3.484	3.554	3.560	3.560	3.560
0.90	0.50	3.350	3.636	3.709	3.715	3.715	3.714

w	q	$I_2[w,q]$	$I_3[w,q]$	$I_4[w,q]$	$I_5[w,q]$	$I_6[w,q]$	$I_7[w,q]$
1.00	0.10	2.402	2.648	2.707	2.714	2.714	2.714
1.00	0.20	2.831	3.108	3,172	3.178	3.178	3.178
1.00	0.30	3.190	3.487	3.554	3.560	3.559	3.559
1.00	0.40	3.474	3.779	3.849	3.855	3.854	3.854
1.00	0.50	3.669	3.972	4.045	4.051	4.051	4.050
1.10	0.10	2.495	2.748	2.808	2.815	2.815	2,815
1.10	0.20	3.002	3.294	3.358	3.364	3.364	3.363
1.10	0.30	3.429	3.745	3.810	3.816	3.816	3.816
1.10	0.40	3.772	4.097	4.165	4.170	4.170	4.170
1.10	0.50	4.014	4.336	4.408	4.413	4.413	4.413
1.20	0.10	2.594	2.856	2.916	2.923	2.923	2.923
1.20	0.20	3.183	3.493	3.556	3.562	3.562	3.562
1.20	0.30	3.682	4.021	4.084	4.090	4.089	4.089
1.20	0.40	4.089	4.437	4.502	4.507	4.506	4.506
1.20	0.50	4.384	4.726	4.796	4.801	4.800	4.800
1.30	0.10	2.699	2.971	3.032	3.038	3.038	3.038
1.30	0.20	3.373	3.706	3.767	3.773	3.773	3.772
1.30	0.30	3.948	4.315	4.374	4.380	4.379	4.379
1.30	0.40	4.422	4.797	4.858	4.863	4.862	4.862
1.30	0.50	4.777	5.139	5.204	5.209	5.208	5.208
1.40	0.10	2.811	3.094	3.155	3.161	3.161	3.161
1.40	0.20	3.573	3.932	3.991	3.997	3.996	3.996
1.40	0.30	4.226	4.626	4.681	4.687	4.686	4.686
1.40	0.40	4.772	5.167	5.223	5.228	5.227	5.227
1.40	0.50	5.190	5.568	5.629	5.633	5.632	5.632

w	q	$I_2[w,q]$	$I_3[w,q]$	$I_4[w,q]$	$I_5[w,q]$	$I_6[w,q]$	$I_7[w,q]$
1.50	0.10	2,927	3.224	3.285	3.291	3.291	3.291
1.50	0.20	3.779	4.171	4.227	4.233	4.232	4.232
1.50	0.30	4.514	4.941	4.993	4.998	4.997	4.997
1.50	0.40	5.139	5.540	5.593	5.597	5.597	5.596
1.50	0.50	5.625	6.011	6.067	6.070	6.069	6.069
1.60	0.10	3.046	3.361	3.422	3.428	3.428	3.428
1.60	0.20	3.988	4.422	4.476	4.482	4.481	4.481
1.60	0.30	4.814	5.237	5.289	5.293	5.292	5.291
1.60	0.40	5.518	5.919	5.968	5.972	5.971	5.970
1.60	0.50	6.078	6.465	6.517	6.520	6.519	6.519
1.70	0.10	3.171	3.506	3.567	3.573	3.573	3.573
1.70	0.20	4.206	4.649	4.704	4.708	4.707	4.707
1.70	0.30	5.120	5.527	5.578	5.581	5.580	5.580
1.70	0.40	5.911	6.299	6.346	6.349	6.348	6.347
1.70	0.50	6.548	6.929	6.977	6.979	6.978	6.978
1.80	0.10	3.291	3.659	3.719	3.725	3.725	3.725
1.80	0.20	4.425	4.837	4.891	4.894	4.894	4.893
1.80	0.30	5.435	5.819	5.868	5.871	5.870	5.870
1.80	0.40	6.316	6.686	6.733	6.734	6.734	6.733
1.80	0.50	7.036	7.406	7.452	7.452	7.451	7.451
1.90	0.10	3.408	3.814	3.873	3.878	3.877	3.877
1.90	0.20	4.646	5.026	5.079	5.082	5.082	5.081
1.90	0.30	5.758	6.115	6.163	6.165	6.164	6.164
1.90	0.40	6.732	7.081	7.126	7.127	7.126	7.126
1.90	0.50	7.538	7.894	7.936	7.936	7.935	7.935

w	q	$I_2[w,q]$	$I_3[w,q]$	$I_4[w,q]$	$I_5[w,q]$	$I_6[w,q]$	$I_7[w,q]$
2.00	0.10	3.525	3.889	3.948	3.953	3.952	3.952
2.00	0.20	4.873	5.217	5.269	5.272	5.272	5.271
2.00	0.30	6.087	6.415	6.462	6.464	6.463	6.463
2.00	0.40	7.160	7.485	7.527	7.528	7.527	7.527
2.00	0.50	8.057	8.395	8.434	8.434	8.432	8.432
2.10	0.10	3.642	3.968	4.027	4.032	4.032	4.032
2.10	0.20	5.103	5.414	5.466	5.470	5.469	5.469
2.10	0.30	6.426	6.726	6.771	6.773	6.772	6.772
2.10	0.40	7.600	7.902	7.942	7.942	7.942	7.941
2.10	0.50	8.591	8.911	8.947	8.946	8.945	8.945
2.20	0.10	3.758	4.052	4.108	4.113	4.112	4.112
2.20	0.20	5.338	5.619	5.667	5.671	5.671	5.670
2.20	0.30	6.772	7.045	7.087	7.089	7.088	7.088
2.20	0.40	8.051	8.330	8.366	8.367	8.366	8.366
2.20	0.50	9.139	9.440	9.473	9.472	9.471	9.471
2.30	0.10	3.877	4.139	4.193	4.198	4.198	4.198
2.30	0.20	5.578	5.830	5.876	5.880	5.879	5.879
2.30	0.30	7.128	7.375	7.413	7.416	7.415	7.415
2.30	0.40	8.516	8.771	8.804	8.805	8.804	8.804
2.30	0.50	9.704	9.986	10.015	10.014	10.013	10.013
2.40	0.10	3.999	4.232	4.284	4.290	4.289	4.289
2.40	0.20	5.827	6.051	6.094	6.099	6.098	6.098
2.40	0.30	7.497	7.718	7.754	7.756	7.756	7.756
2.40	0.40	8.995	9.229	9.258	9.258	9.258	9.258
2.40	0.50	10.286	10.548	10.574	10.573	10.572	10.572

w	q	$I_2[w,q]$	$I_3[w,q]$	$I_4[w,q]$	$I_5[w,q]$	$I_6[w,q]$	$I_7[w,q]$
2.50	0.10	4.127	4.334	4.384	4.390	4.390	4.389
2.50	0.20	6.086	6.282	6.322	6.326	6.326	6.326
2.50	0.30	7.877	8.073	8.105	8.108	8.107	8.107
2.50	0.40	9.488	9.699	9.725	9.726	9.725	9.725
2.50	0.50	10.883	11.127	11.149	11.148	11.148	11.148
2.60	0.10	4.261	4.447	4.493	4.499	4.499	4.498
2.60	0.20	6.354	6.519	6.554	6.558	6.558	6.558
2.60	0.30	8.269	8.436	8.464	8.467	8.467	8.467
2.60	0.40	9.995	10.181	10.204	10.205	10.205	10.205
2.60	0.50	11.497	11.720	11.739	11.739	11.738	11.738
2.70	0.10	4.402	4.571	4.614	4.619	4.619	4.619
2.70	0.20	6.632	6.768	6.798	6.802	6.802	6.802
2.70	0.30	8.674	8.811	8.834	8.837	8.837	8.836
2.70	0.40	10.517	10.677	10.696	10.697	10.697	10.697
2.70	0.50	12.127	12.327	12.344	12.344	12.343	12.343
2.80	0.10	4.550	4.707	4.747	4.753	4.753	4.752
2.80	0.20	6.923	7.036	7.063	7.067	7.067	7.067
2.80	0.30	9.095	9.198	9.217	9.219	9.219	9.219
2.80	0.40	11.057	11.184	11.200	11.200	11.200	11.200
2.80	0.50	12.775	12.948	12.963	12.962	12.961	12.961
2.90	0.10	4.706	4.853	4.891	4.897	4.897	4.897
2.90	0.20	7.227	7.325	7.349	7.353	7.353	7.353
2.90	0.30	9.532	9.608	9.624	9.626	9.626	9.626
2.90	0.40	11.614	11.703	11.715	11.715	11.715	11.715
2.90	0.50	13.441	13.581	13.592	13.591	13.591	13.591

w	q	$I_2[w,q]$	$I_3[w,q]$	$I_4[w,q]$	$I_5[w,q]$	$I_6[w,q]$	$I_7[w,q]$
3.00	0.10	4.871	5.010	5.047	5.052	5.052	5.052
3.00	0.20	7.545	7.632	7.654	7.658	7.658	7.658
3.00	0.30	9.985	10.044	10.057	10.059	10.059	10.059
3.00	0.40	12.189	12.246	12.254	12.255	12.255	12.255
3.00	0.50	14.126	14.223	14.230	14.229	14.229	14.229

Table 3.3.2.4

w	q	$v_2[w,q]$	$v_3[w,q]$	$v_4[w,q]$	$v_5[w,q]$	$v_6[w,q]$	$v_7[w,q]$
0.00	0.10	0.484	0.615	0.644	0.643	0.643	0.643
0.00	0.20	0.484	0.615	0.644	0.643	0.643	0.643
0.00	0.30	0.484	0.615	0.644	0.643	0.643	0.643
0.00	0.40	0.484	0.615	0.644	0.643	0.643	0.643
0.00	0.50	0.484	0.615	0.644	0.643	0.643	0.643
0.10	0.10	0.564	0.696	0.723	0.724	0.724	0.724
0.10	0.20	0.545	0.677	0.703	0.704	0.704	0.704
0.10	0.30	0.526	0.657	0.685	0.685	0.685	0.685
0.10	0.40	0.505	0.638	0.664	0.665	0.665	0.665
0.10	0.50	0.486	0.618	0.646	0.645	0.645	0.645
0.20	0.10	0.647	0.779	0.805	0.806	0.806	0.806
0.20	0.20	0.609	0.741	0.768	0.769	0.769	0.768
0.20	0.30	0.573	0.703	0.730	0.731	0.731	0.731
0.20	0.40	0.533	0.664	0.692	0.692	0.691	0.691
0.20	0.50	0.496	0.625	0.651	0.652	0.652	0.652
0.30	0.10	0.732	0.862	0.890	0.890	0.889	0.889
0.30	0.20	0.677	0.808	0.835	0.835	0.835	0.835
0.30	0.30	0.622	0.753	0.779	0.780	0.779	0.779
0.30	0.40	0.566	0.696	0.722	0.723	0.723	0.722
0.30	0.50	0.508	0.637	0.663	0.664	0.663	0.663
0.40	0.10	0.815	0.947	0.974	0.975	0.974	0.974
0.40	0.20	0.746	0.877	0.903	0.903	0.903	0.903
0.40	0.30	0.675	0.805	0.831	0.831	0.831	0.831
0.40	0.40	0.603	0.731	0.756	0.756	0.756	0.756
0.40	0.50	0.523	0.654	0.679	0.679	0.679	0.679

w	q	$v_2[w,q]$	$v_3[w,q]$	$v_4[w,q]$	$v_5[w,q]$	$v_6[w,q]$	$v_7[w,q]$
0.50	0.10	0.900	1.032	1.058	1.059	1.059	1.059
0.50	0.20	0.816	0.948	0.973	0.974	0.973	0.973
0.50	0.30	0.731	0.860	0.885	0.885	0.885	0.885
0.50	0.40	0.643	0.770	0.794	0.794	0.793	0.793
0.50	0.50	0.549	0.676	0.698	0.698	0.698	0.698
0.60	0.10	0.987	1.118	1.149	1.144	1.144	1.144
0.60	0.20	0.887	1.019	1.044	1.043	1.043	1.043
0.60	0.30	0.788	0.918	0.941	0.941	0.940	0.940
0.60	0.40	0.685	0.813	0.834	0.834	0.834	0.833
0.60	0.50	0.574	0.702	0.722	0.722	0.721	0.721
0.70	0.10	1.070	1.204	1.229	1.229	1.229	1.229
0.70	0.20	0.957	1.092	1.114	1.114	1.114	1.114
0.70	0.30	0.846	0.977	0.997	0.997	0.997	0.997
0.70	0.40	0.731	0.858	0.877	0.877	0.876	0.876
0.70	0.50	0.608	0.732	0.749	0.749	0.748	0.748
0.80	0.10	1.152	1.290	1.314	1.314	1.314	1.314
0.80	0.20	1.027	1.164	1.184	1.184	1.184	1.184
0.80	0.30	0.904	1.037	1.054	1.054	1.053	1.053
0.80	0.40	0.779	0.906	0.921	0.921	0.920	0.920
0.80	0.50	0.645	0.766	0.780	0.780	0.779	0.779
0.90	0.10	1.234	1.375	1.398	1.398	1.399	1.398
0.90	0.20	1.093	1.237	1.253	1.253	1.253	1.253
0.90	0.30	0.961	1.098	1.110	1.110	1.110	1.110
0.90	0.40	0.829	0.955	0.966	0.966	0.965	0.965
0.90	0.50	0.683	0.803	0.813	0.813	0.813	0.813

w	q	$v_2[w,q]$	$v_3[w,q]$	$v_4[w,q]$	$v_5[w,q]$	$v_6[w,q]$	$v_7[w,q]$
1.00	0.10	1.312	1.461	1.482	1.482	1.482	1.481
1.00	0.20	1.157	1.309	1.321	1.321	1.321	1.321
1.00	0.30	1.017	1.158	1.166	1.166	1.165	1.165
1.00	0.40	0.878	1.005	1.012	1.012	1.011	1.011
1.00	0.50	0.725	0.842	0.849	0.849	0.849	0.849
1.10	0.10	1.389	1.546	1.565	1.565	1.564	1.564
1.10	0.20	1.217	1.380	1.388	1.388	1.387	1.387
1.10	0.30	1.071	1.217	1.221	1.221	1.220	1.220
1.10	0.40	0.927	1.056	1.059	1.059	1.058	1.058
1.10	0.50	0.769	0.884	0.888	0.888	0.888	0.888
1.20	0.10	1.461	1.631	1.647	1.647	1.647	1.646
1.20	0.20	1.274	1.450	1.454	1.454	1.453	1.453
1.20	0.30	1.123	1.276	1.276	1.276	1.276	1.276
1.20	0.40	0.975	1.106	1.107	1.107	1.107	1.107
1.20	0.50	0.814	0.927	0.929	0.929	0.929	0.929
1.30	0.10	1.531	1.716	1.729	1.729	1.728	1.728
1.30	0.20	1.329	1.520	1.519	1.519	1.519	1.519
1.30	0.30	1.169	1.334	1.333	1.333	1.333	1.333
1.30	0.40	1.021	1.144	1.145	1.144	1.144	1.144
1.30	0.50	0.858	0.963	0.966	0.965	0.965	0.965
1.40	0.10	1.594	1.800	1.811	1.810	1.810	1.810
1.40	0.20	1.374	1.588	1.584	1.585	1.584	1.584
1.40	0.30	1.215	1.380	1.378	1.379	1.378	1.378
1.40	0.40	1.069	1.164	1.166	1.165	1.164	1.164
1.40	0.50	0.904	0.988	1.001	1.001	1.000	1.000

w	q	$v_2[w,q]$	$v_3[w,q]$	$v_4[w,q]$	$v_5[w,q]$	$v_6[w,q]$	$v_7[w,q]$
1.50	0.10	1.646	1.885	1.892	1.892	1.891	1.891
1.50	0.20	1.406	1.653	1.649	1.650	1.649	1.649
1.50	0.30	1.263	1.340	1.344	1.341	1.341	1.341
1.50	0.40	1.117	1.189	1.193	1.192	1.191	1.191
1.50	0.50	0.951	1.035	1.039	1.038	1.038	1.038
1.60	0.10	1.683	1.969	1.973	1.973	1.973	1.973
1.60	0.20	1.453	1.720	1.715	1.715	1.715	1.715
1.60	0.30	1.305	1.340	1.344	1.341	1.341	1.341
1.60	0.40	1.158	1.215	1.221	1.219	1.219	1.219
1.60	0.50	0.995	1.072	1.076	1.075	1.074	1.074
1.70	0.10	1.697	2.054	2.055	2.055	2.055	2.054
1.70	0.20	1.488	1.439	1.451	1.448	1.448	1.448
1.70	0.30	1.345	1.360	1.363	1.361	1.361	1.361
1.70	0.40	1.205	1.250	1.254	1.253	1.253	1.253
1.70	0.50	1.041	1.112	1.115	1.114	1.114	1.114
1.80	0.10	1.715	2.137	2.137	2.137	2.137	2.137
1.80	0.20	1.526	1.474	1.477	1.477	1.477	1.477
1.80	0.30	1.390	1.397	1.400	1.398	1.398	1.398
1.80	0.40	1.249	1.292	1.294	1.292	1.292	1.292
1.80	0.50	1.085	1.153	1.157	1.156	1.155	1.155
1.90	0.10	1.732	1.541	1.541	1.540	1.541	1.541
1.90	0.20	1.566	1.515	1.518	1.517	1.517	1.517
1.90	0.30	1.430	1.437	1.437	1.436	1.436	1.436
1.90	0.40	1.290	1.330	1.331	1.330	1.330	1.330
1.90	0.50	1.129	1.194	1.196	1.195	1.195	1.195

w	q	$v_2[w,q]$	$v_3[w,q]$	$v_4[w,q]$	$v_5[w,q]$	$v_6[w,q]$	$v_7[w,q]$
2.00	0.10	1.744	1.599	1.598	1.598	1.598	1.598
2.00	0.20	1.600	1.563	1.563	1.562	1.562	1.562
2.00	0.30	1.475	1.480	1.480	1.479	1.479	1.479
2.00	0.40	1.336	1.373	1.373	1.372	1.372	1.372
2.00	0.50	1.174	1.236	1.238	1.237	1.237	1.237
2.10	0.10	1.782	1.661	1.656	1.656	1.656	1.656
2.10	0.20	1.652	1.614	1.611	1.611	1.611	1.611
2.10	0.30	1.522	1.527	1.524	1.524	1.524	1.524
2.10	0.40	1.379	1.415	1.415	1.414	1.414	1.414
2.10	0.50	1.217	1.277	1.279	1.278	1.277	1.277
2.20	0.10	1.829	1.720	1.712	1.712	1.712	1.713
2.20	0.20	1.694	1.662	1.657	1.657	1.657	1.657
2.20	0.30	1.562	1.570	1.566	1.565	1.565	1.565
2.20	0.40	1.423	1.456	1.455	1.454	1.454	1.454
2.20	0.50	1.260	1.317	1.318	1.317	1.317	1.317
2.30	0.10	1.877	1.782	1.772	1.771	1.772	1.772
2.30	0.20	1.740	1.714	1.707	1.706	1.706	1.706
2.30	0.30	1.609	1.616	1.611	1.610	1.610	1.610
2.30	0.40	1.467	1.498	1.497	1.496	1.496	1.496
2.30	0.50	1.303	1.357	1.358	1.357	1.357	1.357
2.40	0.10	1.927	1.844	1.833	1.833	1.833	1.833
2.40	0.20	1.792	1.767	1.758	1.757	1.757	1.757
2.40	0.30	1.656	1.661	1.657	1.656	1.656	1.656
2.40	0.40	1.510	1.540	1.538	1.537	1.537	1.537
2.40	0.50	1.344	1.396	1.396	1.395	1.395	1.395

w	q	$v_2[w,q]$	$v_3[w,q]$	$v_4[w,q]$	$v_5[w,q]$	$v_6[w,q]$	$v_7[w,q]$
2.50	0.10	1.988	1.908	1.893	1.892	1.892	1.892
2.50	0.20	1.844	1.812	1.803	1.802	1.802	1.802
2.50	0.30	1.699	1.702	1.696	1.695	1.695	1.695
2.50	0.40	1.549	1.577	1.575	1.574	1.574	1.574
2.50	0.50	1.383	1.431	1.432	1.431	1.430	1.430
2.60	0.10	2.049	1.971	1.955	1.954	1.954	1.954
2.60	0.20	1.890	1.856	1.846	1.845	1.845	1.845
2.60	0.30	1.743	1.741	1.736	1.735	1.735	1.735
2.60	0.40	1.590	1.613	1.611	1.610	1.610	1.610
2.60	0.50	1.422	1.466	1.466	1.465	1.465	1.465
2.70	0.10	2.104	2.036	2.019	2.018	2.018	2.018
2.70	0.20	1.942	1.906	1.896	1.895	1.895	1.895
2.70	0.30	1.787	1.781	1.775	1.774	1.773	1.773
2.70	0.40	1.630	1.648	1.646	1.645	1.645	1.645
2.70	0.50	1.459	1.498	1.499	1.498	1.498	1.498
2.80	0.10	2.167	2.102	2.087	2.086	2.086	2.086
2.80	0.20	1.993	1.958	1.949	1.948	1.948	1.948
2.80	0.30	1.831	1.818	1.813	1.812	1.811	1.812
2.80	0.40	1.669	1.680	1.677	1.677	1.676	1.676
2.80	0.50	1.495	1.528	1.529	1.528	1.528	1.528
2.90	0.10	2.232	2.171	2.155	2.154	2.154	2.154
2.90	0.20	2.045	2.011	2.002	2.001	2.001	2.001
2.90	0.30	1.874	1.859	1.853	1.852	1.852	1.852
2.90	0.40	1.706	1.709	1.707	1.706	1.706	1.706
2.90	0.50	1.529	1.556	1.556	1.555	1.555	1.555

w	q	$v_2[w,q]$	$v_3[w,q]$	$v_4[w,q]$	$v_5[w,q]$	$v_6[w,q]$	$v_7[w,q]$
3.00	0.10	2.298	2.239	2.224	2.223	2.223	2.223
3.00	0.20	2.097	2.064	2.055	2.054	2.054	2.054
3.00	0.30	1.916	1.899	1.894	1.893	1.893	1.893
3.00	0.40	1.742	1.740	1.738	1.737	1.737	1.737
3.00	0.50	1.562	1.580	1.580	1.579	1.579	1.579

Table 3.3.2.5

w	q	$f_1[w,q]$	$f_2[w,q]$
0.0	0.1	0.00	0.00
0.0	0.3	0.00	0.00
0.0	0.5	0.00	0.00
0.5	0.1	0.34	0.39
0.5	0.3	0.46	0.61
0.5	0.5	0.50	0.72
1.2	0.1	1.96	2.14
1.2	0.3	2.65	3.22
1.2	0.5	2.88	3.93
3.0	0.1	12.24	11.97
3.0	0.3	16.56	17.09
3.0	0.5	18.00	21.23

Table 3.3.2.6

w	q	I[w,q]	v[w,q]
0.00	0.10	2.205	0.653
0.00	0.20	2.205	0.653
0.00	0.30	2.205	0.653
0.00	0.40	2.205	0.653
0.00	0.50	2.205	0.653
0.50	0.10	2.361	1.067
0.50	0.20	2.490	0.979
0.50	0.30	2.593	0.887
0.50	0.40	2.665	0.794
0.50	0.50	2.703	0.697
1.00	0.10	2.750	1.478
1.00	0.20	3.209	1.312
1.00	0.30	3.587	1.156
1.00	0.40	3.877	1.000
1.00	0.50	4.068	0.834
1.50	0.10	3.322	1.882
1.50	0.20	4.259	1.644
1.50	0.30	5.009	1.309
1.50	0.40	5.599	1.173
1.50	0.50	6.062	1.020
2.00	0.10	4.082	2.302
2.00	0.20	5.268	1.543
2.00	0.30	6.447	1.460
2.00	0.40	7.499	1.353
2.00	0.50	8.390	1.214

w	q	I[w,q]	v[w,q]
2.50	0.10	4.426	1.898
2.50	0.20	6.331	1.793
2.50	0.30	8.084	1.678
2.50	0.40	9.674	1.552
2.50	0.50	11.066	1.403
3.00	0.10	5.079	2.214
3.00	0.20	7.678	2.050
3.00	0.30	10.072	1.891
3.00	0.40	12.253	1.733
3.00	0.50	14.188	1.568
3.50	0.10	6.006	2.566
3.50	0.20	9.432	2.307
3.50	0.30	12.532	2.082
3.50	0.40	15.326	1.885
3.50	0.50	17.805	1.698
4.00	0.10	7.201	2.942
4.00	0.20	11.598	2.558
4.00	0.30	15.463	2.252
4.00	0.40	18.895	2.014
4.00	0.50	21.930	1.808
4.50	0.10	8.635	3.344
4.50	0.20	14.162	2.809
4.50	0.30	18.850	2.402
4.50	0.40	22.939	2.125
4.50	0.50	26.547	1.904

w	q	I[w,q]	v[w,q]
5.00	0.10	10.281	3.762
5.00	0.20	17.105	3.064
5.00	0.30	22.681	2.539
5.00	0.40	27.448	2.222
5.00	0.50	31,644	1.988
5.50	0.10	12.122	4.188
5.50	0.20	20.411	3.331
5.50	0.30	26.949	2.669
5.50	0.40	32.420	2.311
5.50	0.50	37.217	2.065
6.00	0.10	14.152	4.616
6.00	0.20	24.063	3.612
6.00	0.30	31.654	2.796
6.00	0.40	37.855	2.392
6.00	0.50	43.267	2.133
6.50	0.10	16.368	5.041
6.50	0.20	28.050	3.903
6.50	0.30	36.796	2.924
6.50	0.40	43.756	2.466
6.50	0.50	49.795	2.195
7.00	0.10	18.768	5.465
7.00	0.20	32.364	4.200
7.00	0.30	42.377	3.054
7.00	0.40	50.128	2.536
7.00	0.50	56.805	2.251

w	q	I[w,q]	v[w,q]
7.50	0.10	21,353	5,886
7.50	0.20	37.002	4.499
7.50	0.30	48.396	3.191
7.50	0.40	56.971	2.603
7.50	0.50	64.299	2.303
8.00	0.10	24.123	6.305
8.00	0.20	41.961	4.799
8.00	0.30	54.853	3.335
8.00	0.40	64.290	2.665
8.00	0.50	72.277	2.350
8.50	0.10	27.075	6.722
8.50	0.20	47.240	5.100
8.50	0.30	61.746	3.490
8.50	0.40	72.086	2.724
8.50	0.50	80.743	2.394
9.00	0.10	30.211	7.136
9.00	0.20	52.840	5.400
9.00	0.30	69.072	3.656
9.00	0.40	80.361	2.782
9.00	0.50	89.697	2.435
9.50	0.10	33.530	7.549
9.50	0.20	58.760	5.700
9.50	0.30	76.827	3.832
9.50	0.40	89.117	2.837
9.50	0.50	99.140	2.474

w	q	I[w,q]	v[w,q]
10.00	0.10	37.030	7.959
10.00	0.20	65.000	6.000
10.00	0.30	85.008	4.017
10.00	0.40	98.354	2.891
10.00	0.50	109.074	2.510
10.50	0.10	40.713	8.368
10.50	0.20	71.560	6.300
10.50	0.30	93.614	4.209
10.50	0.40	108.074	2.944
10.50	0.50	119.501	2.544
11.00	0.10	44.577	8.775
11.00	0.20	78.440	6.600
11.00	0.30	102.642	4.404
11.00	0.40	118.279	2.996
11.00	0.50	130.424	2.576
11.50	0.10	48.623	9.181
11.50	0.20	85.640	6.900
11.50	0.30	112.091	4.602
11.50	0.40	128.972	3.050
11.50	0.50	141.855	2.611
12.00	0.10	52.850	9.585
12.00	0.20	93.160	7.200
12.00	0.30	121.960	4.801
12.00	0.40	140.145	3.101
12.00	0.50	153.768	2.641

Table 3.3.2.7

Random start	Mean stage cost
769	2.21 ± 0.09
3497	2.29 ± 0.10
8153	2.27 ± 0.10
12843	2.23 ± 0.09
18375	2.29 ± 0.10
27927	2.27 ± 0.10

N.B. The random start is the parameter IX referred to in section A4.2.

Table 4.1.1

'THE APPLICATION OF OPTIMAL CONTROL THEORY TO A CLASS OF EXTREMUM CONTROL SYSTEMS'

Abstract of thesis

by S.M. LANGDON

Abstract

This thesis constitutes a contribution to the theory of extremal control systems which come within the automatic control field of learning in engineering science.

An extremal control system comprises an extremal process (which is to be controlled) and an extremal controller. The class of extremal control systems referred to in the title of the thesis refers to those systems in which the extremal process concerned can be represented by a particular form of mathematical model.

Investigation of the application of optimal control theory to this class of extremal control systems is motivated by two basic requirements. The first of these originates in the control of complex industrial processes, particularly in the direct digital control situation. Here there are a number of possible applications for extremal control, and since costs involved are generally high, the requirement is generally for extremal controllers maintaining as low a system operating cost as possible, rather than for controllers with a simple structure. Hence there is a need for optimal extremal controllers.

The second requirement for establishing optimal extremal controllers lies in the design of simpler extremal controllers. The performance of the optimal extremal controller is required as a basis on which to judge the simpler controllers and decide whether any more significant improvement can be made.

The major part of this investigation of the application of optimal control theory to extremal control systems is concerned with the simplest possible case, where no account is taken of input or output lags or noise, or of multiple inputs. Since no

previous optimal extremal controllers are known this is the natural starting point.

A rigorous analysis of state variables, probability distributions and sufficient statistics for this simplified case is presented, and this leads on to a new approach to the application of dynamic programming which in turn results in an original functional recurrence equation. Analysis of this functional recurrence equation leads to a numerical solution procedure including many checks. This eventually establishes the optimal extremal controller.

Simulation techniques are used to confirm the performance of the optimal extremal controller, and hybrid computing facilities used to show that it can be implemented on a small on-line computer and used in a direct digital control situation without suffering from interfacing effects. Thus for the first time there is now available an optimal extremal controller, and moreover it can be confidently expected to perform optimally in a practicall situation.

Comparison of the performance of the optimal extremal controller with the performance of simpler controllers shows that, in the simplified case, there is still room for considerable improvement on the simple controllers, and a quantitative measure of just how much improvement might be possible is now available.

An initial investigation of the application of optimal control theory to more complex processes, involving noise, lags and multiple inputs, is presented and this shows that theoretical difficulties are likely to prevent further optimal extremal controllers from being established. In these cases there is therefore a requirement to establish closely optimal controllers. One approach would be just to use the optimal controller from the

simplified case in these more complex situations. Simulations are presented establishing the performance of the optimal extremal controller, and a suboptimal extremal controller, controlling not the process for which they are designed but a similar process involving output noise. These simulations show that these controllers are certainly not optimal in the more complex situation. A different approach must therefore be taken to establishing closely optimal controllers in the more complex situation. The thesis finishes with a discussion of how this might be achieved.