AN AUTOMATIC OPTIMIZER FOR USE IN OPTIMAL PROCESS CONTROLLERS

by

KENNETH GEORGE WHALE

B.A.Sc., University of British Columbia, 1966

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in the Department of
Electrical Engineering

We accept this thesis as conforming to the
required standard

Members of the Department
of Electrical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA

September, 1968

Department of Elect. Eng.

The University of British Columbia
Vancouver 8, Canada

Date Sept. 3/68

ABSTRACT

The practical implementation of optimal control systems in large industrial process applications has been limited by the high costs of the required computing facilities. With the recent advances in component fabrication and the resultant decrease in hardware costs, special purpose computers, utilizing virtually no software at all, can be constructed as economical alternatives to presently available general purpose computers for use in optimal process controllers.

A design for one such special purpose machine, an automatic optimizer, is presented in this thesis. Tests conducted on a working optimizer constructed on the basis of the given design, demonstrate that it is suitably fast and powerful for use in process controllers. In addition, the optimizer is inexpensive enough to be used as part of an economical process controller.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure | Page

ACKNOWLEDGEMENT

Many thanks are due to Dr. E.V. Bohn for his general guidance of the thesis project, to Dr. E.L. Sigurdson for his numerous helpful suggestions, and to Mr. D. Holmes for many hours of valuable technical assistance in circuit construction and testing.

I would also like to thank those fellow graduate students who assisted by proof-reading the final draft of the thesis.

Grateful acknowledgement is given to the National Research Council for a bursary provided in 1966, for a research assistantship granted in 1967, and for general project support under Research Grant number 67-3134.

1. INTRODUCTION

1.1 The Optimal Controller Requirement

The rapid mathematical development of optimal control theory in recent decades has far outpaced its practical application to the realization of optimal control systems for industrial processes. This failure is due largely to the requirement, in realistic situations, of large-scale general purpose computing facilities for continuous numerical solution of the complex and multi-dimensional systems of equations which are encountered. The capabilities of today's most sophisticated computers, and indeed of those proposed for the near future, are severely taxed by such applications. More significantly, the high cost of such facilities, notwithstanding the drastic reduction in computer hardware costs brought about by micro-miniature integrated circuit technology, has restricted such installations to a relatively few large government or industrial establishments.

Recent investigations(1) have pointed the way toward the reduction of the obstacle posed by dimensionality, through the decomposition of the complete optimization problem into a number of much smaller and nearly independent sub-problems. A control hierarchy can then be constructed (Figure 1) in which locally independent, sub-optimal controllers are co-ordinated by one central computer at a supervisory level to achieve, as closely as possible, overall process optimization.

A hierarchical structure is economically attractive only if costly general purpose computing facilities can be confined to the supervisory levels, and relatively inexpensive sub-optimal controllers can be provided locally. Such controllers are not presently available. Optimal controllers have, however, been implemented (rather expensively) on general purpose hybrid analog and digital computers and the nature of the computations which they must perform are well established.

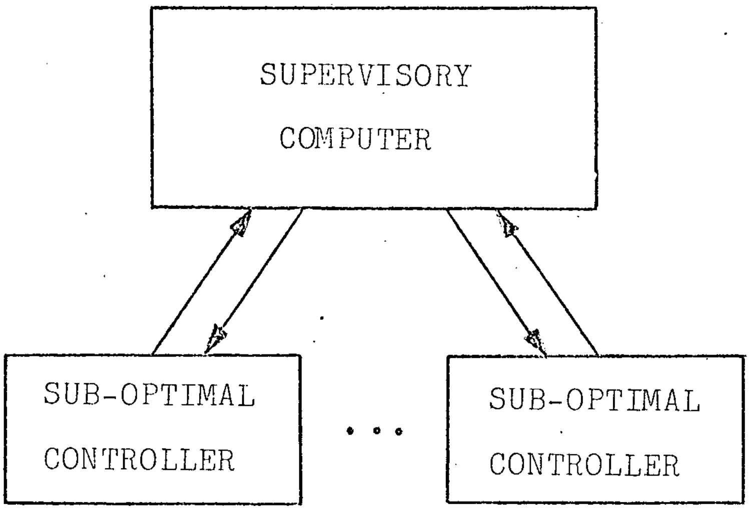


Figure 1. Hierarchical Control Structure

Special purpose machines are needed to replace these expensive computers at the local level, and it was with this requirement in mind that the work of this thesis was initiated. The remaining chapters are devoted to the specification, design, construction and evaluation of one of the constituent sub-assemblies, an automatic optimizer, required for the economical synthesis of local, sub-optimal process controllers.

1.2 The Optimization Problem

A representative sub-optimal controller is depicted in the block diagram of Figure 2. The process model consists of a set of differential equations describing, approximately, the dynamics of the actual process. On the basis of the information contained in the model and knowledge of the desired process output, the optimal control problem can be formulated--normally as a two-point boundary value problem subject to the minimization of some scalar performance function of the projected terminal state of the process.

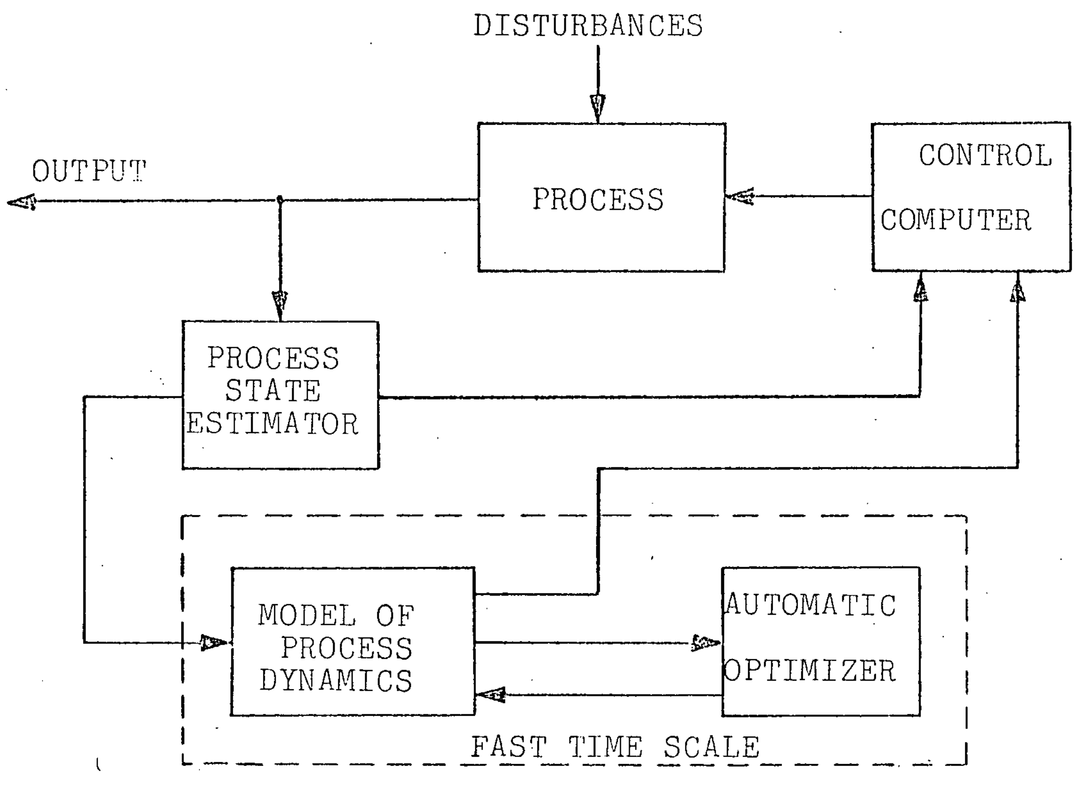


Figure 2. Sub-optimal Controller

If the process was modelled exactly and no external disturbances were present, then the boundary value problem could be solved, minimizing the performance function, and the optimal control could be calculated once, and generated thereafter by the control computer; no additional blocks would be required in the controller structure.

However, because the model is not exact and disturbances are usually present, the optimal control must be continually re-adjusted as the process proceeds. Each control up-dating implies the minimization of the performance function by repeated, fast-time solution of the process model equations. The time intervals between minimizations must obviously be much smaller than either the process duration or the time constants associated with changes in the process state (due to both control adjustments and external disturbances). Process time constants vary widely but can frequently be expressed in terms of minutes, so that intervals in the order of a few seconds between minimizations are realistic.

The performance function is computed only at the terminal points of fast-time solutions of differential equations. However, its discrete nature can be concealed if the solution repetition rate is sufficiently high and the function value is held constant between computations. The function may be assumed to be accessible for measurement, but not analytically available as a known function of the parameters (usually initial conditions of the model equations) which can be adjusted to achieve its minimization.

The task of an automatic optimizer between the control up-datings is basically one of static optimization, although over longer periods of time, under the influence of disturbances, the optimum may drift and the overall performance of the controller may thus be considered one of dynamic optimization. Due to the wide difference in time constants between the process and process model, however, the deviation from the optimum should never be large, so that it can be assumed that the optimizer need not distinguish between global or absolute minima and relative minima.

Although as many as fifty independent variables are not uncommon in practical industrial process control problems, consistent with the concept of decomposition, each sub-problem can reasonably be considered to contain not more than five or six variables. It can further be assumed that these variables are unconstrained, since such constraints as may exist can generally be incorporated into the performance function itself.

In summary, the automatic optimizer is faced with the local, unconstrained static optimization of a measurable, but analytically unknown, continuous scalar performance function of up to five or six independent variables.

Hill-climbing search strategies for static optimization problems are well developed and documented in the literature. It remains to choose from among them a strategy that, while being powerful enough to converge rapidly and effectively for five dimensional problems, is sufficiently simple that it can be economically implemented in a special purpose machine. Before presenting in the next section a necessarily brief survey of available techniques, the conditions required for a function to possess an optimum are given below along with some appropriate notations.

The performance function is designated as

$$P = P(\overline{x})$$

where

$$\overline{x} = (x_1, x_2, \ldots x_n)'$$

and, within a region R in the space defined by the independent variables, P is assumed to be continuous.

A necessary condition that P have an extreme at a point $\overline{x}^*$ in the region R is that the gradient $\overline{\nabla P}$, where

$$\overline{\nabla P} \triangleq \left(\frac{\partial P}{\partial x_1}, \frac{\partial P}{\partial x_2}, \ldots \frac{\partial P}{\partial x_n} \right), \qquad \ldots\ldots\ldots\ldots\ldots\ldots(1.1)$$

evaluated at $\overline{x} = \overline{x}^*$, must be zero.

Sufficient conditions for the point $\overline{x}^*$ to be a local minimum or maximum can be determined by approximating P with a Taylor series expansion about $\overline{x}^*$.

$$P(\overline{x}) \cong \sum_{i=0}^{\infty} \frac{1}{i!} \left(d^i \left[P(\overline{x}^*) \right] \right) \qquad \ldots\ldots\ldots\ldots\ldots\ldots(1.2)$$

In the above equation d^i is an operator defined as follows:

$$d^i \triangleq \left[\sum_{j=1}^{n} (x_j - x_j^*) \frac{\partial}{\partial x_j} \right]^i \triangleq \left[\sum_{j=1}^{n} a_j \frac{\partial}{\partial x_j} \right]^i$$

In the immediate neighbourhood of the optimum, only the first few terms of (1.2) are required, reducing the approximation to:

$$P(\overline{x}) \cong P(\overline{x}^*) + \sum_{j=1}^{n} a_j \frac{\partial P(\overline{x}^*)}{\partial x_j} + \frac{1}{2} \sum_{k=1}^{n} \sum_{j=1}^{n} a_j a_k \frac{\partial^2 P(\overline{x}^*)}{\partial x_j \partial x_k} + \ldots$$

In matrix notation

$$P(\overline{x}) \cong P(\overline{x}^*) + \overline{a}' \overline{\nabla P} + \frac{1}{2} \overline{a}' \overline{Q}\, \overline{a}$$

where
$$q_{ij} = \frac{\partial^2 P(\overline{x}^*)}{\partial x_i \partial x_j}$$

are the components of $\overline{Q}$.

Since the point $\overline{x} = \overline{x}^*$ was specified to be an extreme, the gradient term must be zero so the equation becomes

$$\overline{P}(\overline{x}) \approxeq P(\overline{x}^*) + \frac{1}{2}\, \overline{a}'\, \overline{Q}\, \overline{a} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.3)$$

The second term on the right hand side of Equation (1.3) is a standard quadratic form and it follows that if $\overline{x}^*$ is to be a minimum of $P(\overline{x})$ then

$$\overline{a}'\, \overline{Q}\, \overline{a} = P(\overline{x}) - P(\overline{x}^*) > 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.4)$$

so that $\overline{Q}$ must be a positive definite matrix. If $\overline{x}^*$ is to be a maximum

$$\overline{a}'\, \overline{Q}\, \overline{a} = P(\overline{x}) - P(\overline{x}^*) < 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.5)$$

and $\overline{Q}$ must be a negative definite matrix. If $\overline{Q}$ is semi-definite then higher order terms of the Taylor series expansion must be examined.

It can be seen from either (1.4) or (1.5) that a maximum can be transformed to a minimum simply by negating the performance function and so no generality is lost by hereafter assuming that all extremes sought are minima.

Consistent with the previous statement (page 5) that distinction between relative and absolute minima need not be made, it will be assumed also that, within the region of interest R in the independent variable space, the performance function is unimodal. That is, from any point in the region R there should exist only a strictly falling path on the response surface from that point to the minimum.

1.3 Survey of Available Search Techniques

In general, a hill-climbing or search technique in a static optimization problem consists of adjusting the values of the independent variables of the function under investigation according to some pre-established strategy and, on the basis of the results, locating the point or points at which the function value is a maximum or minimum. The nature of any particular function, the amount of information available about it, the manner in which the data accumulated by the search strategy is to be manipulated (by hand or by computer, for instance), and the number of dimensions or independent variables involved, are all vital considerations in the determination of the suitability of any particular search strategy to the solution of any given optimization problem.

The more general problem of choosing a search strategy that is suitable for a whole range of functions is naturally more difficult still. With the statement of the problem, as given in section (1.2), in mind, a survey of available methods was made, the results of which are summarized below. Consistent with the problem of section (1.2), techniques are not distinguished on the basis of their ability to differentiate between absolute and relative extremes, nor on their adaptability to problems in which constraints of various forms are imposed on the independent variables.

The search techniques to be considered can be conveniently classified into two general categories--direct methods and gradient methods. Although such a distinction appears somewhat artificial in a few cases, and, indeed, additional or alternative classifications are certainly possible, this one, with some flexibility, serves as a simple but useful guide.

Direct search methods are characterized by requiring nothing more of a performance function than its value at a number, frequently large, of points in the space of its independent variables. Included are a variety of "random search" techniques[2], the so-called "factorial method"[3] described by Box, Wilde's "lattice technique"[4], and the "pattern search"[5] developed by Hooke and Jeeves. There are, of course, many others and many variations of those mentioned, but most (with the notable exception of random searches) are subject to failure when applied to performance functions whose response surfaces contain ridges or valleys. Thus, although in general direct search methods do have the advantage of simplicity, they are also frequently extremely inefficient, even on simple functions.

Gradient search techniques, as the name suggests, make use of information about the slope of the performance function response surface to determine the direction in which to conduct experiments on it. This information is contained in the "gradient vector", $\overline{\nabla P(\overline{x})}$, of the performance function where

$$\overline{\nabla P(\overline{x})} \triangleq \left(\frac{\partial P}{\partial x_1} , \frac{\partial P}{\partial x_2} , \ldots \frac{\partial P}{\partial x_n} \right)$$

Use of the gradient implies that the component partial derivatives be calculated if P is analytically known, or approximated on the basis of local experimentation if this is not the case.

Although not, strictly speaking, a gradient method, the "univariate" search[6], which consists simply of searching along each of the directions of the independent variable axes in succession, is generally considered with gradient methods because of its close relation to "relaxation"(7) methods. Such methods also adjust only one independent variable at a time, but the order in which they are varied depends upon which component of the gradient vector is largest. Like the direct search methods, both the univariate and relaxation techniques are fundamentally simple but relatively inefficient and prone to failure in some situations.

Steepest descent techniques(8), first described by Cauchy in 1847, have been, in one form or another, the most widely used of all search strategies. They are all based on the approximation of the performance function by an n-dimensional hypersphere. Mathematically

$$P(\overline{x}) = p + \overline{c}'\overline{x} + \frac{1}{2}\,\overline{x}'\overline{K}\,\overline{x} \qquad \text{.......................(1.6)}$$

where

$$\overline{K} = k\,\overline{I} \qquad \text{.......................(1.7)}$$

and $\overline{I}$ is the identity matrix. Differentiating Equation (1.6) gives

$$\overline{\nabla P} = \overline{c} + \overline{K}\,\overline{x} \qquad \text{.......................(1.8)}$$

and solving Equation (1.8) for $\overline{x}$ yields

$$\overline{x} = \overline{K}^{-1}\,(\,\overline{\nabla P} - \overline{C}\,) \qquad \text{.......................(1.9)}$$

Substituting two specific points, $\overline{x}_i$ and $\overline{x}_{i+1}$, into Equation (1.9) and forming the difference, Equation (1.10) is obtained.

$$\overline{x}_{i+1} - \overline{x}_i = \overline{K}^{-1}\,(\,\overline{\nabla P}_{i+1} - \overline{\nabla P}_i\,) \qquad \text{................(1.10)}$$

Now, if it is assumed that the point $\overline{x}_{i+1}$ is the extreme $\overline{x}^*$ of the function as approximated by Equation (1.6), then $\overline{\nabla P_{i+1}}$ must vanish and Equation (1.10) becomes

$$\bar{x}^* - \bar{x}_i = -\bar{K}^{-1} \, \overline{\nabla P_i} \qquad \text{(1.11)}$$

Substituting Equation (1.7) into this result gives

$$\bar{x}^* - \bar{x}_i = -\frac{1}{k} \, \overline{\nabla P_i} \qquad \text{(1.12)}$$

Equation (1.12) says,essentially,that if the approximation of $P(\bar{x})$ by a hypersphere is accurate at the point $\bar{x}_i$, then the extreme can be reached by one search along the direction defined by the gradient $\overline{\nabla P_i}$, the so-called steepest descent direction. In most cases, of course, the approximation is crude and the extreme found is not the true optimum. A great many steepest descent variations[(9),(10)] have been developed which attempt both to improve the approximation by scaling the function or transforming the independent variables, and to compensate for its crudeness by iteratively up-dating the gradient (and hence the approximation). Such methods have been found to be considerably more effective than the direct search methods previously mentioned, but at the expense of the increased computational effort required to calculate or estimate the gradient vector. As suggested, the rate of convergence of steepest descent methods is highly dependent upon the nature of the response surface and tends to be very slow in the presence of ridges, and in the immediate vicinity of the optimum. In the former case a tendency to zig-zag along the ridge is exhibited, while near the extreme the gradient is very small.

A group of search techniques, closely related to gradient methods but based on a higher order approximation to the response surface, has been developed which exhibits much improved performance on ridges, particularly in the neighbourhood of the optimum. The approximation, similar in form to that expressed in Equation (1.6), is

$$P(\overline{x}) \cong P_o + \overline{c}'\overline{x} + \frac{1}{2}\overline{x}'\overline{Q}\,\overline{x} \qquad \text{..................(1.13)}$$

where $\overline{Q}$ is required to be a symmetric, positive definite, n by n matrix. Equation (1.13) is the generalized n-dimensional representation of a quadratic surface. Following the same steps used to derive the steepest descent direction, the equation

$$\overline{x}^* - \overline{x}_i = -\overline{Q}^{-1}\,\overline{\nabla P}_i \qquad \text{..................(1.14)}$$

can be obtained. This equation indicates that from any point, $\overline{x}_i$, on a quadratic surface, the optimum, $\overline{x}^*$, can be reached by a single search along the direction defined by Equation (1.14). For future reference such a direction will be called an "optimum direction".

By twice differentiating Equation (1.13), it can be seen that the matrix $\overline{Q}$ must be the so-called Hessian matrix whose components

$$q_{ij} = \frac{\partial^2 P}{\partial x_i \partial x_j}$$

are just the second partial derivatives of $P(\overline{x})$. Thus, if $P(\overline{x})$ is known analytically then $\overline{Q}^{-1}$ and $\overline{\nabla P}_i$ can be calculated and the required search direction determined exactly from any point. If $P(\overline{x})$ is unknown then $\overline{Q}^{-1}$ and $\overline{\nabla P}_i$ must be estimated.

Methods based on this principle, generally referred to as "Newton's"[11] methods, are said to possess "quadratic convergence" since they locate the optimum of a true quadratic function in a finite number of steps. Such methods in iterative form have been shown to be extremely powerful, converging efficiently from poor initial starting points even for non-quadratic response surfaces. Unlike steepest descent techniques, which can be seen to be a special case of Newton's methods, convergence is accelerated as the optimum is approached since a quadratic approximation becomes increasingly accurate. Ridges or valleys are also traversed efficiently.

In recent years a flurry of research, prompted by the suitability of digital computers to optimization problems, has produced several new and powerful variations of the search techniques discussed to this point. Two rather ingenious direct search methods, referred to respectively as the "sequential simplex"[12] method and "Rosenbrock's"[13] method, neither of which have any sound theoretical basis apart from being systematic procedures, have shown surprisingly good results. As well, four related methods, "conjugate gradients"[14], "deflected gradients"[15], "Partan"[16], and a technique which will be called "Powell's method"[17], are notable among many new strategies derived from the principle of Newton's method. All possess quadratic convergence and all have been demonstrated to be competent on difficult response surfaces up to many dimensions.

The deflected gradient method determines an optimum direction by actually estimating the matrix $\overline{Q}$ of Equation (1.14). The estimate is obtained progressively from a sequence of linear search steps whose directions are defined by the equation

$$\bar{d}_{i+1} = \bar{x}_{i+1} - \bar{x}_i = -\mu_i \bar{H}_i \overline{\nabla P}_i$$

where $\bar{H}_i$ is a matrix computed at the end of each step according to the following equations

$$\bar{H}_o = I$$
$$\bar{H}_i = \bar{H}_{i-1} + \bar{A}_i + \bar{B}_i \quad ; \quad i = 1,2,\ldots$$

If the matrices $\bar{A}_i$ and $\bar{B}_i$ are chosen at each step to be

$$\bar{A}_i = \frac{\bar{d}_i \bar{d}_i}{\bar{d}_i \overline{\nabla P}_i}$$

$$\bar{B}_i = - \frac{\bar{H}_{i-1} \overline{\nabla P}_i \bar{H}_i}{\overline{\nabla P}_i \bar{H}_{i-1} \overline{\nabla P}_i}$$

then it can be shown that for a quadratic function

$$\bar{H}_{n-1} = \bar{Q}^{-1}$$

so that

$$\bar{d}_n = \bar{x}_n - \bar{x}_{n-1} \equiv \bar{x}^* - \bar{x}_{n-1}$$

and the optimum is located after n steps.

The method of conjugate gradients generates an optimum direction for a quadratic function by a succession of steps based on the following iteration:

$$\bar{d}_i = - \overline{\nabla P}_i + \beta_i \bar{d}_{i-1} \qquad i = 1,2,\ldots n$$

The vectors $\bar{d}_i$, again, represent directions of linear one-dimensional searches, $\overline{\nabla P}_i$ is the gradient vector and β_i is an ordinary steepest descent direction with

$$\beta_i = 0$$

Thereafter the constant β_i is given by

$$\beta_i = \frac{\left|\overline{\nabla P_i}\right|}{\left|\overline{\nabla P_{i-1}}\right|}$$

For this value of β_i it can be shown(14) that the direction $\overline{d}_n$ is an optimum direction, as defined by Equation (1.14), from the point $\overline{x}_{n-1}$ located by minimizing $P(\overline{x})$ along the previous direction $\overline{d}_{n-1}$. The point $\overline{x}_n$ must then be the optimum of $P(\overline{x})$.

Partan (an acronym for parallel tangents) and Powell's method are two different n-dimensional generalizations of a method proposed for two-dimensional quadratics by Finkel(18). Based on the so-called theorem of parallel chords, his technique is illustrated in Figure 3 below.

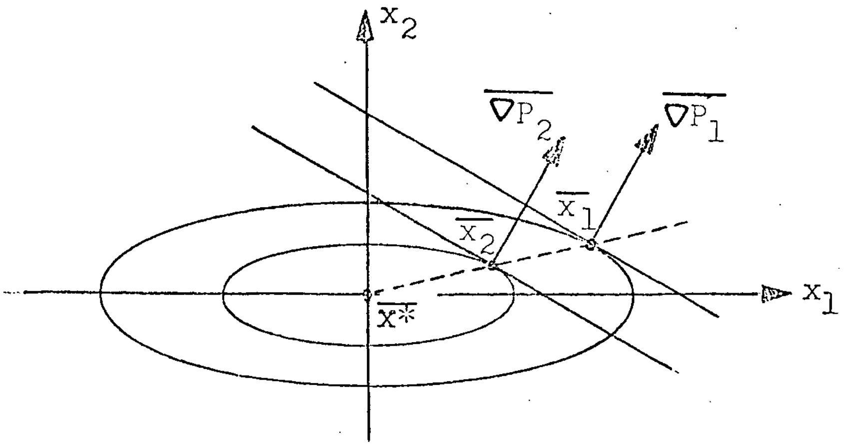


Figure 3. Finkel's Method

The generalization states essentially that if the gradients at $\bar{x}_1$ and $\bar{x}_2$ on a quadratic surface are parallel, then the optimum lies on the line joining the two points. This can be related to Newton's method by writing Equation (1.14) for the two points

$$\bar{x}_1 - \bar{x}^* = - \bar{Q}^{-1} \, \overline{\nabla P}_1$$

$$\bar{x}_2 - \bar{x}^* = - \bar{Q}^{-1} \, \overline{\nabla P}_2$$

and thus

$$\bar{x}_2 - \bar{x}_1 = - \bar{Q}^{-1} \left(\overline{\nabla P}_2 - \overline{\nabla P}_1 \right)$$

If, however, $\overline{\nabla P}_2$ and $\overline{\nabla P}_1$ are parallel then

$$\overline{\nabla P}_2 \equiv k \, \overline{\nabla P}$$

and hence

$$\bar{x}_2 - \bar{x}_1 = - \bar{Q}^{-1} \, \overline{\nabla P}_1 \, (k-1) \qquad \text{..................(1.15)}$$

so that the direction from $\bar{x}_1$ to $\bar{x}_2$ is indeed an optimum direction and must therefore pass through $\bar{x}^*$ as well.

Partan locates two such points at which the gradients are parallel by a sequence of linear searches in n-dimensional space whose directions are chosen parallel to certain tangent planes. The number of planes to which these directions are constrained to be parallel increases progressively until only one possible direction remains and that is shown to pass through the optimum.

Powell's method accomplishes the same thing by progressively reducing the dimensionality of the sub-spaces in which successive linear searches are performed.

These last four methods appear from the literature published to be the most generally powerful techniques presently available; hence, they are the most desirable methods for special purpose implementation. Unfortunately, they are also the most complex methods computationally, involving vector and matrix manipulations of various forms. Such computations imply both memory storage and arithmetical facilities in any automatic machine realization.

A more detailed analysis of Powell's method revealed, however, that it could be modified slightly so as to avoid virtually all of the computational requirements mentioned. This modified method which was adopted for an automatic optimizer is given, along with the original method of Powell, in the next section.

1.4 Powell's Method and a Modification

In his 1962 paper, M.J.D. Powell presented the following algorithm for the minimization of a function $P(\overline{x})$ of n independent variables.

(1) From any point $\overline{x}_1$ conduct a linear search in the n-dimensional gradient direction $\overline{\nabla P}_1$. The extreme is located at $\overline{x}_2$.

(2) Conduct a search in the n-1 dimensional hyperplane containing $\overline{x}_2$ and orthogonal to the direction $\overline{x}_2-\overline{x}_1$. The new extreme is at $\overline{x}_3$.

(3) Conduct a linear search along the direction $\overline{x}_3-\overline{x}_1$ locating the extreme at $\overline{x}_4$.

If $P(\overline{x})$ is quadratic then $\overline{x}_4$ is the minimum sought. If not, the algorithm must be iterated and some criterion satisfied to stop the search when it has converged sufficiently close to the minimum.

Examination of step two reveals that the method is recursive in that each n-dimensional search invokes steps one, two and three for its execution. Figures 4 and 5 illustrate the linear search sequences in two and three dimensions respectively.

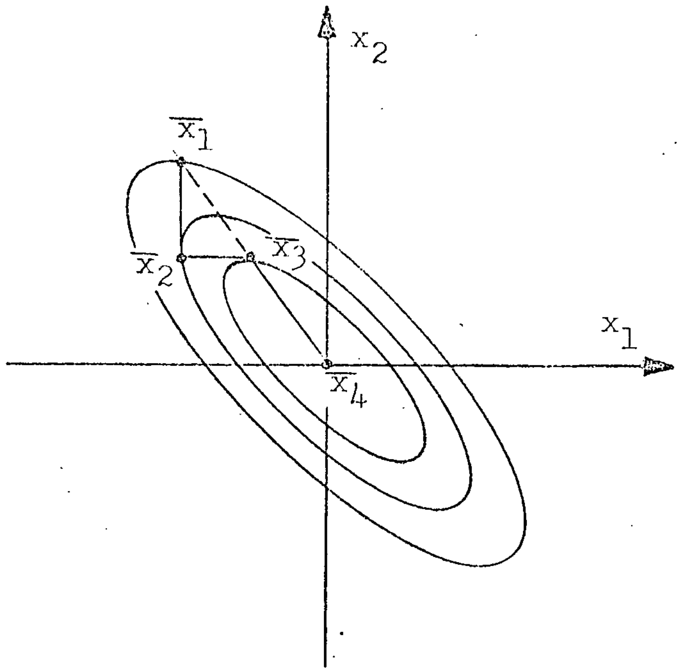


Figure 4. Powell's Method--Two Dimensions

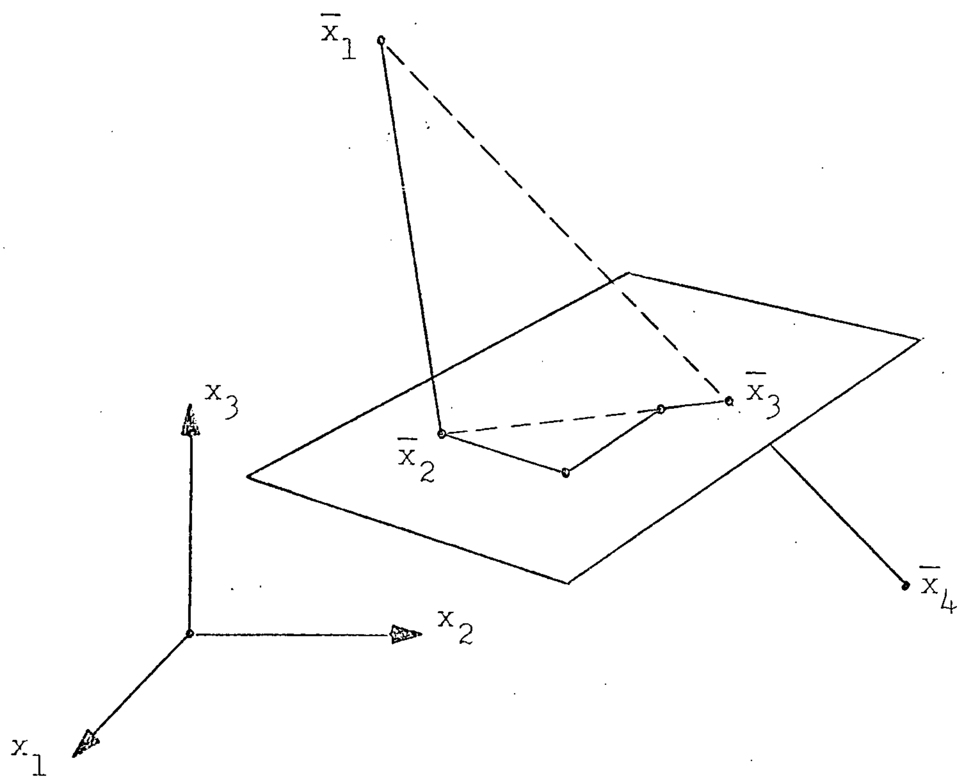


Figure 5. Powell's Method--Three Dimensions

The n-1 dimensional hyperplane containing $\overline{x}_2$ and $\overline{x}_3$ must be tangent to the n-dimensional contours of the response surface at point $\overline{x}_3$, since $\overline{x}_3$ is the minimum of $P(\overline{x})$ in the restricted, n-1 dimensional space. Hence, the n-dimensional gradient $\overline{\nabla P}_3$ at $\overline{x}_3$ is orthogonal to this tangent plane and must therefore be parallel to the gradient $\overline{\nabla P}_1$ at $\overline{x}_1$. Equation (1.15) of section (1.3) showed that the line joining two such points must pass through the optimum.

Except for the computations involved in determining the gradient direction for step one of the algorithm, Powell's method is well suited for machine implementation. A modification involving the addition of another step is suggested to eliminate these computations. The new algorithm, designated the P.M.A. algorithm (for Powell's Modified Algorithm) follows:

(1) Conduct an n-1 dimensional search from any point $\overline{x}_o$, locating the extreme at $\overline{x}_1$.

(2) Conduct a linear search from $\overline{x}_1$ along the direction orthogonal to the n-1 dimensional hyperplane in (1), locating the extreme at $\overline{x}_2$.

(3) Conduct a search in the n-1 dimensional hyperplane containing $\overline{x}_2$ and orthogonal to the direction $\overline{x}_2-\overline{x}_1$, locating the extreme at $\overline{x}_3$.

(4) Conduct a linear search along the direction $\overline{x}_3-\overline{x}_1$, locating the extreme at $\overline{x}_4$.

A new step one has been introduced to establish the initial (n-dimensional) gradient direction from $\overline{x}_1$ by locating $\overline{x}_1$ as the extreme in an n-1 dimensional hyperplane--the gradient $\overline{\nabla P}_1$ must be orthogonal to this hyperplane at $\overline{x}_1$ since, again, as in step two of the original algorithm, this hyperplane must be tangent to the n-dimensional contour surfaces of $P(\overline{x})$ at $\overline{x}_1$. The remaining steps of the new algorithm are identical to those of the original.

A further computational reduction can be obtained by arbitrarily choosing n-1 co-ordinate directions to define the n-1 dimensional hyperspace of step one. The gradient vector $\overline{\nabla P_1}$ is then constrained to be parallel to the remaining co-ordinate direction and, conveniently, the n-1 dimensional hyperplane searched by step three of P.M.A. must be parallel to the first one.

The linear search sequences for the modified algorithm are shown in Figures 6 and 7, again for two and three dimensions.

Since only the manner of obtaining the gradient direction at $\overline{x}_1$ has been altered, it is to be expected that the quadratic convergence property of Powell's original method should be retained in this modified version. This was indeed demonstrated to be so, as the results obtained by P.M.A. from digital computer tests compared favourably in all cases with those published by Powell in his original paper. The digital computer program and a summary of the results is contained in Appendix I.

A more revealing comparison between the two algorithms can be made on the basis of the number of linear searches required for one iteration, or equivalently, to optimize a quadratic function. This information is shown in Figure 8 for two, three, four and five dimensions. The entries in each column are the numbers of linear searches constituting one iteration of each algorithm.

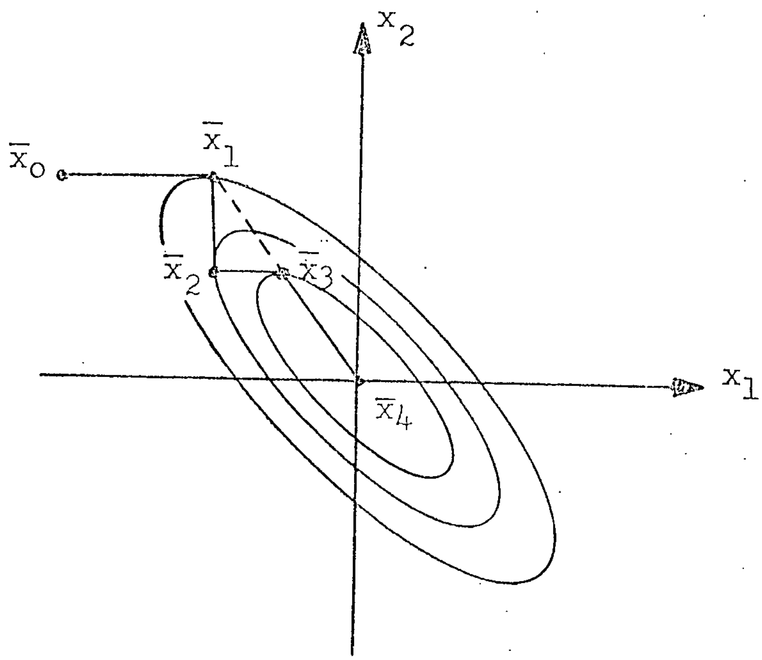


Figure 6. P.M.A.--Two Dimensions

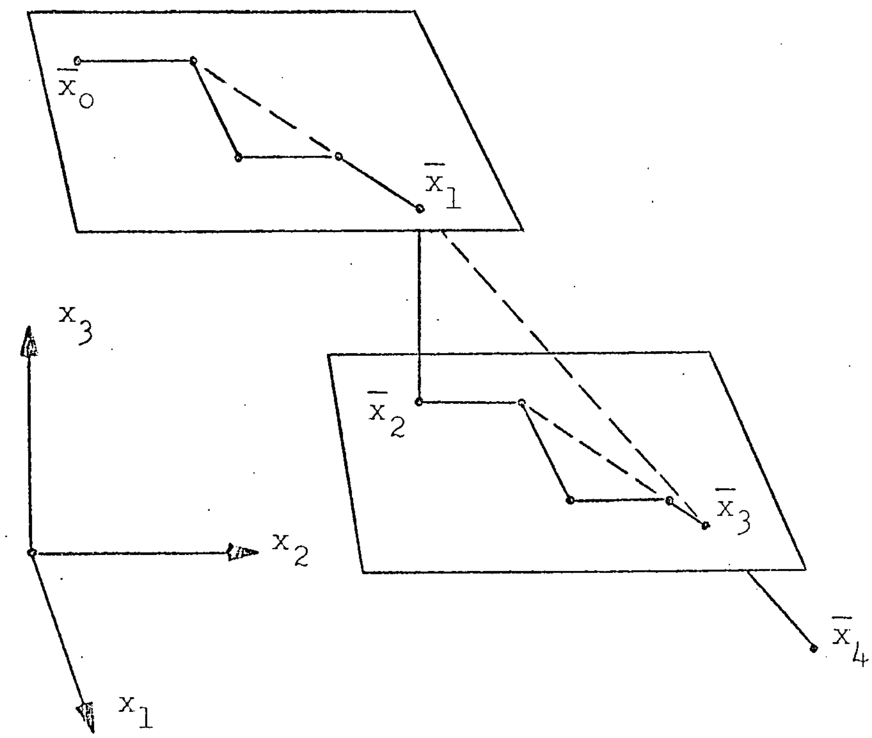


Figure 7. P.M.A.--Three Dimensions

n	2	3	4	5
Powell's method	3	5	7	9
Modified method	4	10	22	46

Figure 8. Comparison of Powell's Method and P.M.A.

As the table of Figure 8 clearly indicates, the modified algorithm eliminates computations at the expense of a rather large increase in the number of linear searches. Provided, however, that the average time consumed by a linear search is sufficiently small, this defect will not significantly detract from the performance of the method, at least up to the required five dimensions. It is also significant, of course, that the algorithm executes its search in such a fashion that the value of the performance function is ever diminishing until the optimum is attained.

The remaining chapters of this thesis are devoted to the design and evaluation of an economical automatic system to implement this modified version of Powell's search strategy.

2. AUTOMATIC OPTIMIZER--BASIC DESIGN

2.1 System Configuration

A special purpose automatic optimizer, designed to execute the modified form of Powell's search algorithm, is shown greatly simplified in the block diagram of Figure 9 below.

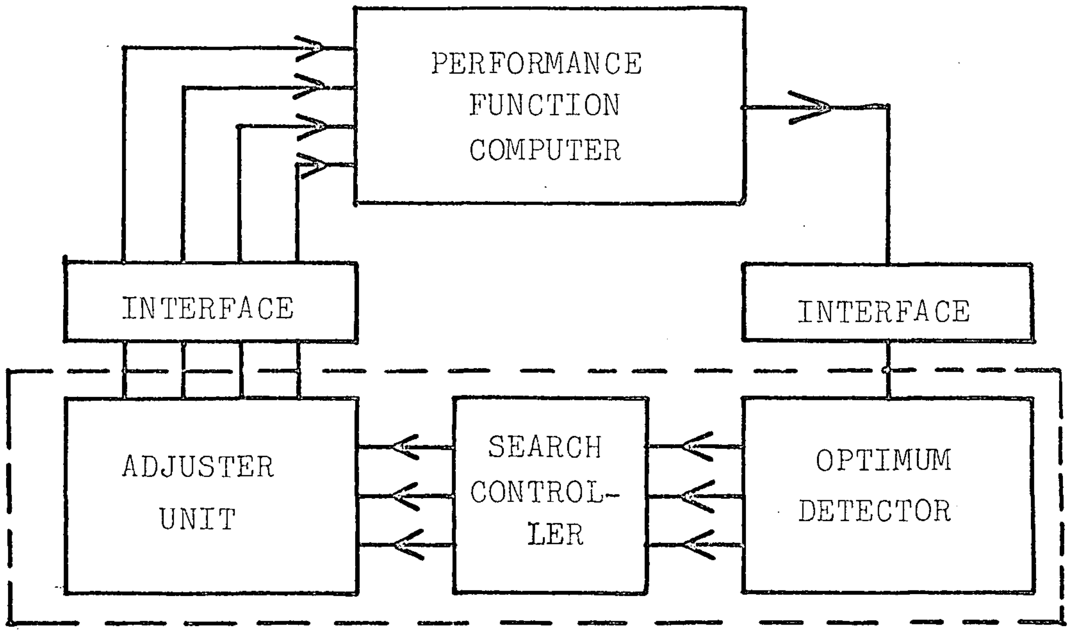


Figure 9. Automatic Optimizer

The blocks outside the dotted lines are external to the optimizer and represent, in some form, the performance function, $P(\overline{x})$, which is to be optimized. Its output is assumed to be an analog signal which is related in a continuous and unimodal manner to the values of its input signals, $\overline{x}(t)$. It is further assumed that the output response to an input change is so fast as to appear to be instantaneous. Any signal conditioning or scaling required to interconnect between the performance function block and the optimizer proper is performed by the "interface" blocks, shown here as distinct units; normally they can be considered as part of the performance function block. In any event, it will be assumed throughout the thesis that the analog input to the optimum detector is characterized by a signal to noise ratio of at least sixty decibels.

The "optimum detector" continuously monitors the performance function, P, and indicates the presence of local extremes, both minima and maxima. These indications are interpreted by the search controller which appropriately sequences control signals to the adjuster unit, so as to execute the desired search algorithm.

The iterative nature of the search strategy dictates a digital form for the search controller. The structures of the remaining two blocks are not so obvious, although functionally they can be thought of, simply, as types of analog to digital (A/D) and digital to analog (D/A) converters. With reference to the requirements of section (1.4), the next two sections of this chapter will describe the basic considerations involved in the design of the adjuster unit and the optimum detector.

2.2 The Adjuster Unit

As mentioned briefly in section (2.1), the purpose of the adjuster unit is to generate controllable signals, more specifically voltages, which are to be interpreted by the performance function block as instantaneous values of its independent variables, $x_i(t)$. There must thus, of course, be exactly n adjuster unit output signals for a performance function involving n independent variables. The manner in which these signals are to be controlled is revealed by close examination of the P.M.A. search algorithm given in section (1.4) along with a brief discussion. Two distinct types of searches are evident:

(1) Linear searches parallel to co-ordinate axes.

(2) Linear searches in arbitrary directions in r-dimensional space; where $r \leqq n$, the number of independent variables.

Both types of searches imply continuous adjustment of the independent variables, relying on the optimum detector to recognize that a minimum has been located along a given line. In addition, it is implied that the adjuster unit must be capable of holding constant any combination of these independent variables, possibly over extended time intervals.

Searches of type one, parallel to co-ordinate directions, can be realized merely by adjusting one output signal at a time, holding the remainder constant, The faster the rate of change of the adjusted variable, the faster will be the convergence to a minimum, assuming no limitations on either the performance function computer or the optimum detector.

Searches of the second type may be implemented by simultaneously varying r of the adjuster unit outputs at r independent rates, $\frac{dx_i}{dt}$. The resultant search direction in r dimensional space then depends only on the ratios of the $\frac{dx_i}{dt}$, and at no time is it necessary that the absolute values of any of the independent variables be known. This is significant in that the adjuster unit need not be capable of precisely setting its output voltage levels, but only of adjusting them.

Although several analog adjuster unit structures were considered (various forms of ramp generators and operational amplifier integrators), it was decided to use digital stepping motors for reasons of reliability, stability and ease of interconnection with the digital search controller.

Available in several types and sizes, both unidirectional and bidirectional, and complete with custom matched electronic driver circuits, all stepping motors are characterized by an incremental response exhibiting one-to-one correspondence between digital input pulses and rotor angular position, as shown in Figure 10.

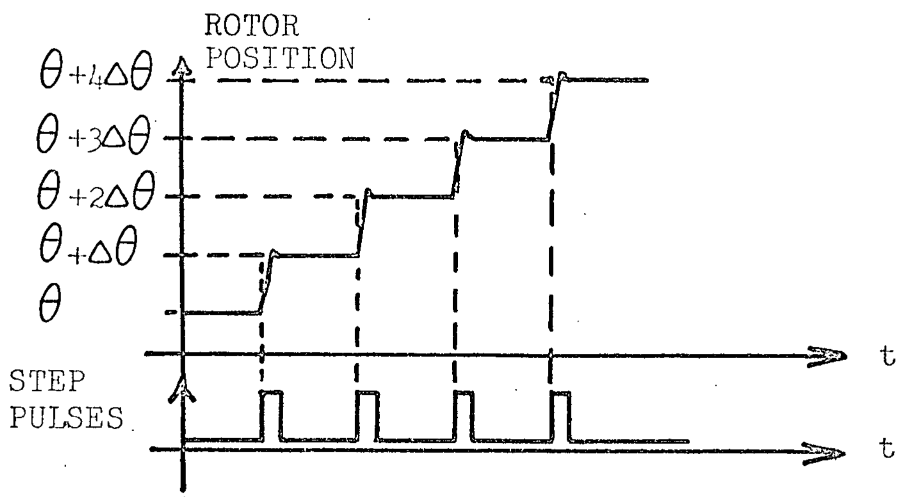


Figure 10. Stepping Motor Response

Although much faster motors can be obtained for specialized applications, stepping rates are normally limited to about 800 to 1000 steps per second. Specified angular step accuracies also vary considerably; however, such positional errors as do occur are non-cumulative, by virtue of the motor design, whereby stable rotor positions are fixed relative to the stator.

A system of such stepping motors, one for each independent variable, driving potentiometers through suitable gear ratios was adopted for the adjuster unit. Step pulses are obtained from variable frequency pulse generators, and, although adjustment of the outputs from the potentiometers is necessarily in discrete amounts because of the digital nature of the motors, the choice of gear ratios and frequency range are such that continuous linear searches can be closely approximated.

Search speed is less than that which could be achieved using completely electronic circuitry, but it is felt to be sufficient for the relatively slow processes to be encountered in most industrial optimization problems.

2.3 The Optimum Detector

The function of the optimum detector consists of monitoring the performance function as the adjuster unit conducts a linear search in the space of its independent variables, and indicating immediately that a minimum along that line is reached. The sensitivity and response time of this unit directly influences both the precision with which the steps of the search algorithm can be executed and the final resolution to which the minimum can be located.

There are three conceptually simple methods of detecting the extreme of a continuously varying analog signal, employing either direct differentiation, analog sampling, or analog to digital conversion.

The first method is clearly unsuitable for the slowly changing signals expected, due to the inherently poor low frequency response of differentiating circuits.

The second method involves the continual comparison of the analog signal value, P, with a time delayed sample of itself,

P_s. When the difference signal, $D = P - P_s$, changes sign, the presence of an extreme is indicated.

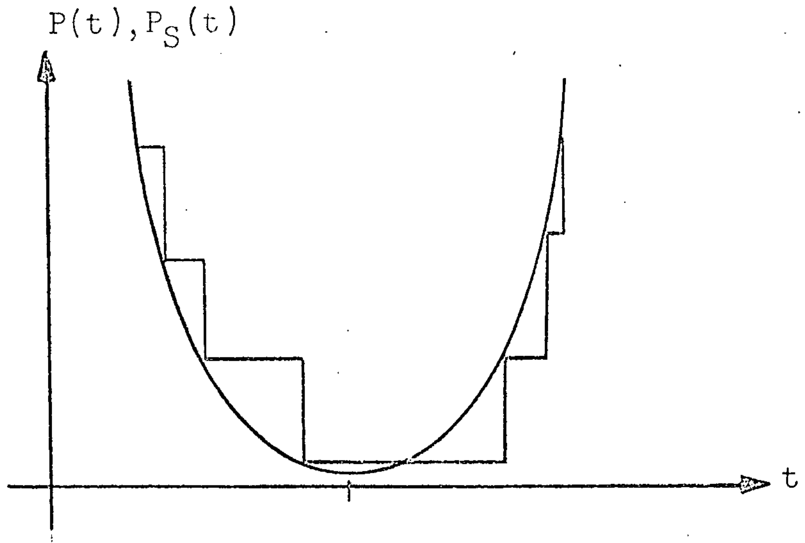


Figure 11. Analog Sampling Method

The precision to which the extreme is located can be made independent of the sampling "rate" by using a system configuration as shown in Figure 12 below.

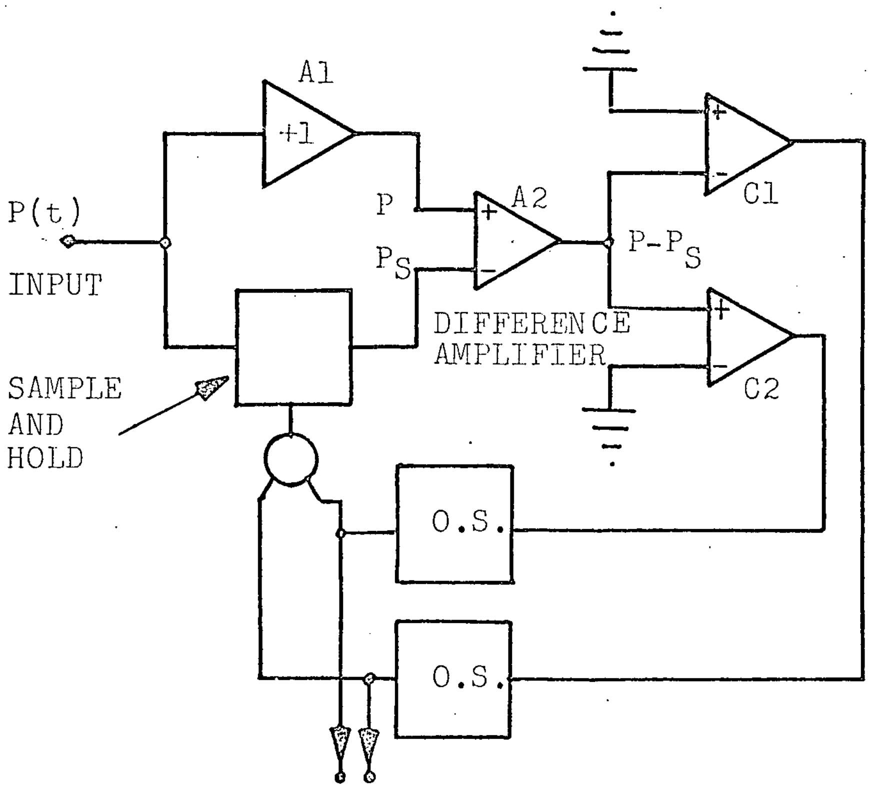


Figure 12. Analog Sampling Circuit

Here, samples are taken only after the analog input has changed by some small fixed amount. The two comparators C1 and C2 are connected so that increments of either polarity in P may be detected.

The third method consists of converting the analog input to a digital equivalent and performing some sort of manipulations on the digital signal to determing at what point an extreme is reached. The number of binary "bits" used to digitally approximate the analog signal is the primary limitation on the achievable detector precision.

Although such techniques are generally comparatively costly, the A/D conversion method was chosen for implementation in favour of analog sampling because of the higher inherent stability, reliability, and noise immunity of digital circuitry. A complete discussion of the design used is given in the next chapter.

3. AUTOMATIC OPTIMIZER--COMPLETE DESIGN

3.1 The Optimum Detector

A digital optimum detector was built around the principle of A/D conversion, using the tracking circuit shown below.

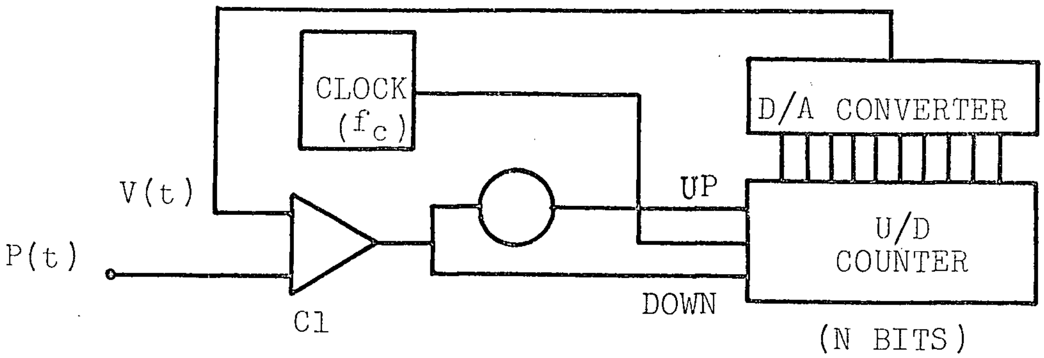


Figure 13. A/D Tracking Circuit

An up/down reversible binary counter supplied by uninterrupted pulses from a fixed frequency (f_c) clock is directed by the comparator, C1, to count up or down according to the difference between the analog inputs, P(t) and V(t), to the comparator. P(t) represents the value of the performance function $P(\overline{x}(t))$, and V(t) is the continuous analog conversion of the binary number, C_N, in the counter. The comparator is connected in such a way that V(t) provides negative feedback, forcing the counter always to count in the direction that tends to reduce the difference $P(t) - V(t)$. $V(t)$ and the equivalent binary number, C_N, thus continuously track the input, $P(t)$.

The number, N, of bits in the up/down counter, the clock frequency, f_c, the sensitivity and speed of the comparator, and the D/A converter transfer relation, (Figure 14), are all important factors governing the permissable range of the input $P(t)$, both in terms of frequency and voltage magnitude. Of these factors, only the comparator characteristics are fixed (by a choice of comparator), and the remainder can be adjusted to give any desired, practical performance.

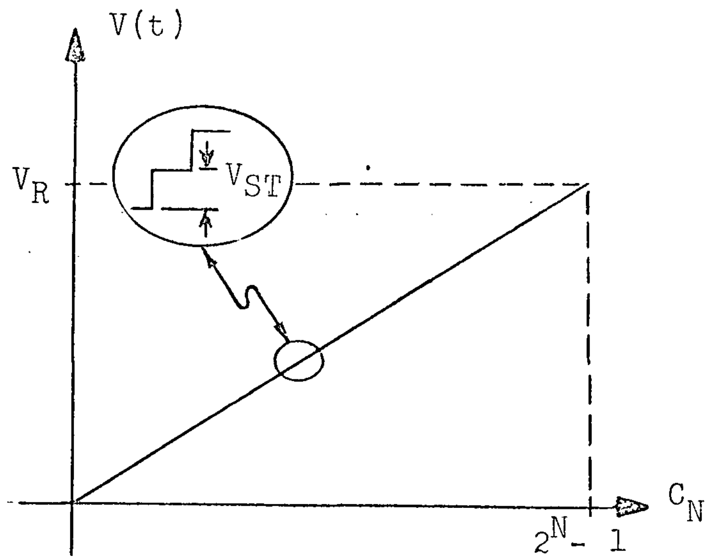


Figure 14. D/A Transfer Relation

By choosing the D/A converter output step size, V_{ST}, to be

$$V_{ST} > 2V_D ,$$

where V_D is the comparator window, the tracking signal, $V(t)$ for a constant input $P(t) = C$, can be made to resemble the waveform of Figure 15.

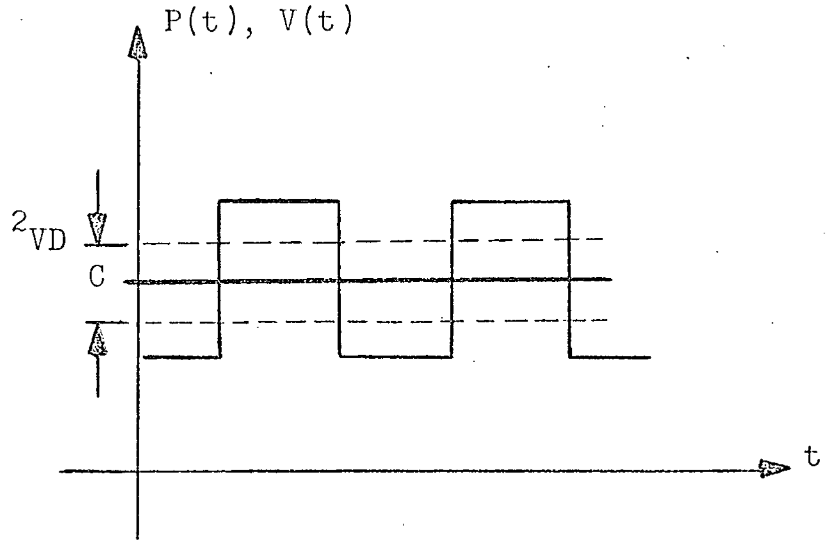


Figure 15. Tracking Waveform

V_{ST} can be related to the length of the counter by the following equations:

$$V(t) = V_{ST}\ C_N$$

and

$$V_{ST} = \frac{V_R}{2^N - 1} \qquad \text{......................(3.1)}$$

where V_R is the maximum D/A output voltage (and hence also the maximum input signal voltage). The precision with which P(t) can be tracked is thus directly proportional to the length, N, of the up/down counter.

This counter length can also be related to the maximum rate of change of P(t), (the so-called "slewing rate"), which can be followed by V(t). If ΔT is the total time required to count from $C_N = 0$ to $C_N = 2^N - 1$, then obviously

$$\Delta T = \frac{2^N - 1}{f_c} \quad \text{seconds}$$

for a clock frequency, f_c. The maximum slewing rate of V(t) must then be

$$S_V = \frac{V_R\ f_c}{2^N - 1} \ \frac{\text{volts}}{\text{second}} \qquad \text{...................(3.2)}$$

and S_V must also correspond to the maximum input slewing rate that can be handled by the optimum detector.

The tracking system of Figure 13 does not in itself constitute an A/D converter. Additional circuitry is generally included to make C_N accessible and to relate it in sign and magnitude to the input signal P(t). Since the search algorithm of section (1.4) does not require explicitly the value of P(t), this circuitry was omitted, and logic was developed, instead, for locating the extremes of P(t).

A diagram of the system, showing the extreme detection circuitry, is given in Figure 16. Another up/down counter (the detector counter), driven in parallel with the original (tracking) counter and identical to it except in length, has been added.

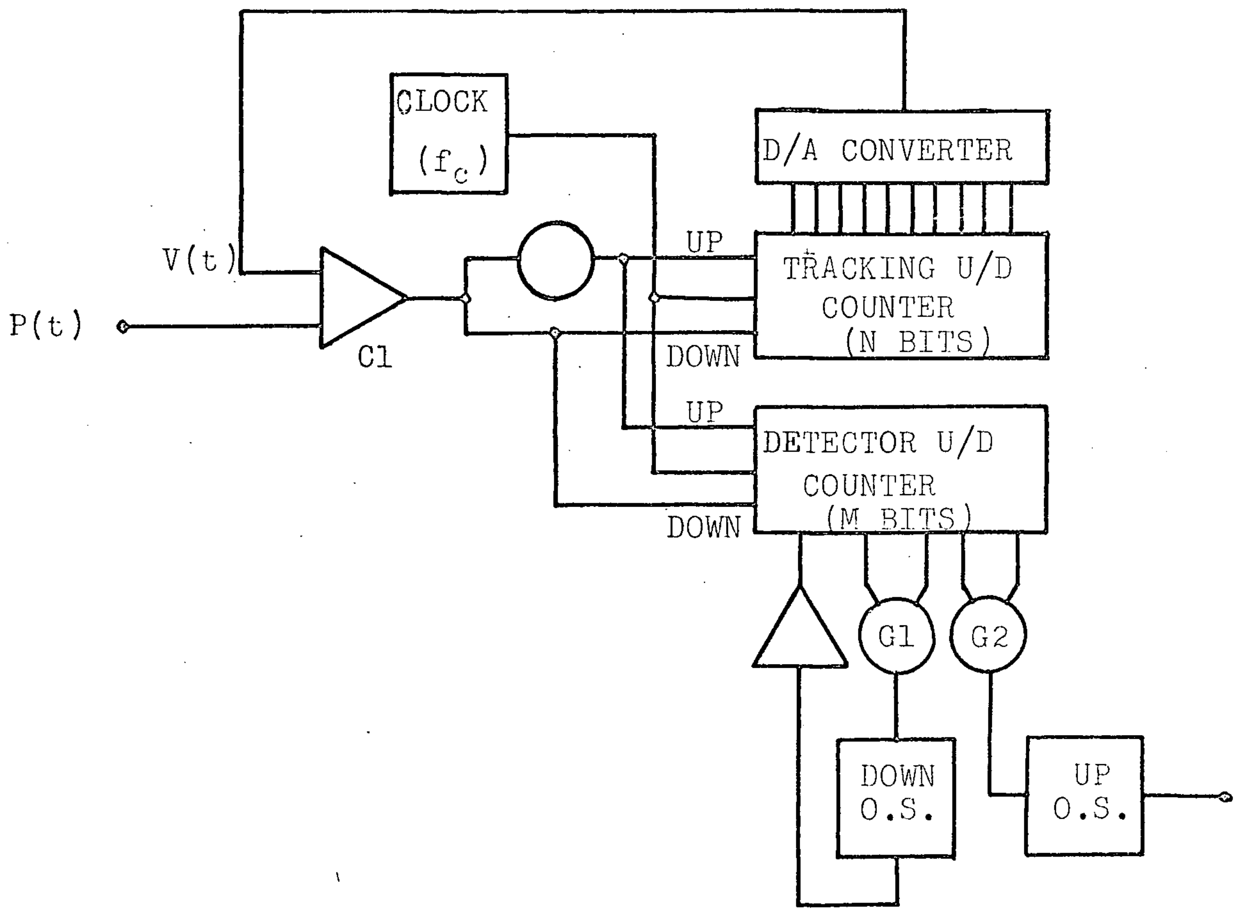


Figure 16. Tracking Circuit with Detection Logic

The logical sequence table (Figure 17) for this new counter illustrates the principle used, assuming its length to be four bits ($M = 4$). The most significant bits are shown on the right.

```
                      0 1 1 0
                      1 0 1 0
        Count         0 0 1 0  ---- Detect 10 for increasing P
         up           1 1 0 0
                      0 1 0 0
                      1 0 0 0
"0"-state ----        0 0 0 0
                      1 1 1 1
        Count         0 1 1 1
        down          1 0 1 1
                      0 0 1 1
                      1 1 0 1  ---- Detect 01 for decreasing P.
                      0 1 0 1
```

Figure 17. Detector Counter Sequence

Clearly, if the detector counter is initially reset to zero, then monitoring the two right-most bit combinations, 10 and 01, is sufficient to determine whether $P(t)$ is locally increasing or decreasing. (Actually, in order for detection symmetry with increasing or decreasing $P(t)$, the combination $1\lambda 10$ must be detected on the up count and $\lambda\lambda 01$ on the down count.) In Figure 16 this is done by gates G1 and G2 in conjunction, respectively, with the up and down one-shots.

The precision with which changes in $P(t)$ can be detected by this method is primarily dependent on the length of the detector counter. A sort of quality figure for the optimum detector can accordingly be defined as $E(M,N)$, where

$$E(M,N) = \frac{\Delta C_M}{2^N - 1} \times 100\% \qquad \ldots\ldots\ldots\ldots\ldots\ldots (3.3)$$

and ΔC_M is the number of counts in the detector counter (and hence also in the tracking counter) required to generate a detection pulse. E(M,N) thus indicates the achievable detection precision as a percentage of the allowable range of the input P(t). A theoretical lower limit of three counts for ΔC_M, corresponding to M = 3, is imposed by the natural tracking oscillation of V(t), as illustrated in Figure 15.

A complete optimum detector designed and constructed on the basis of the foregoing considerations is shown in Figure 19. Some additional control circuitry (U/D control flip-flop and strobe pulse) is included to avoid possible logical hazards both in counting and in generation of detector pulses. A standard up/down counter circuit, given in Figure 18, was used.

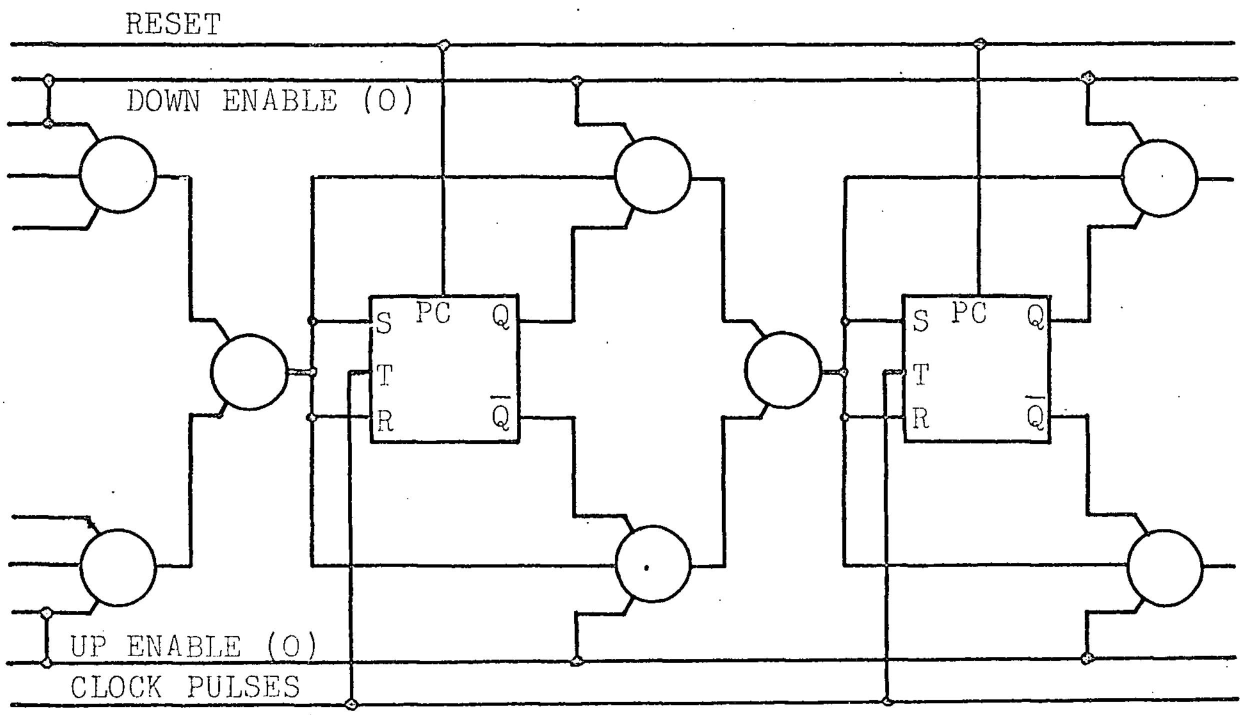


Figure 18. Up/Down Counter

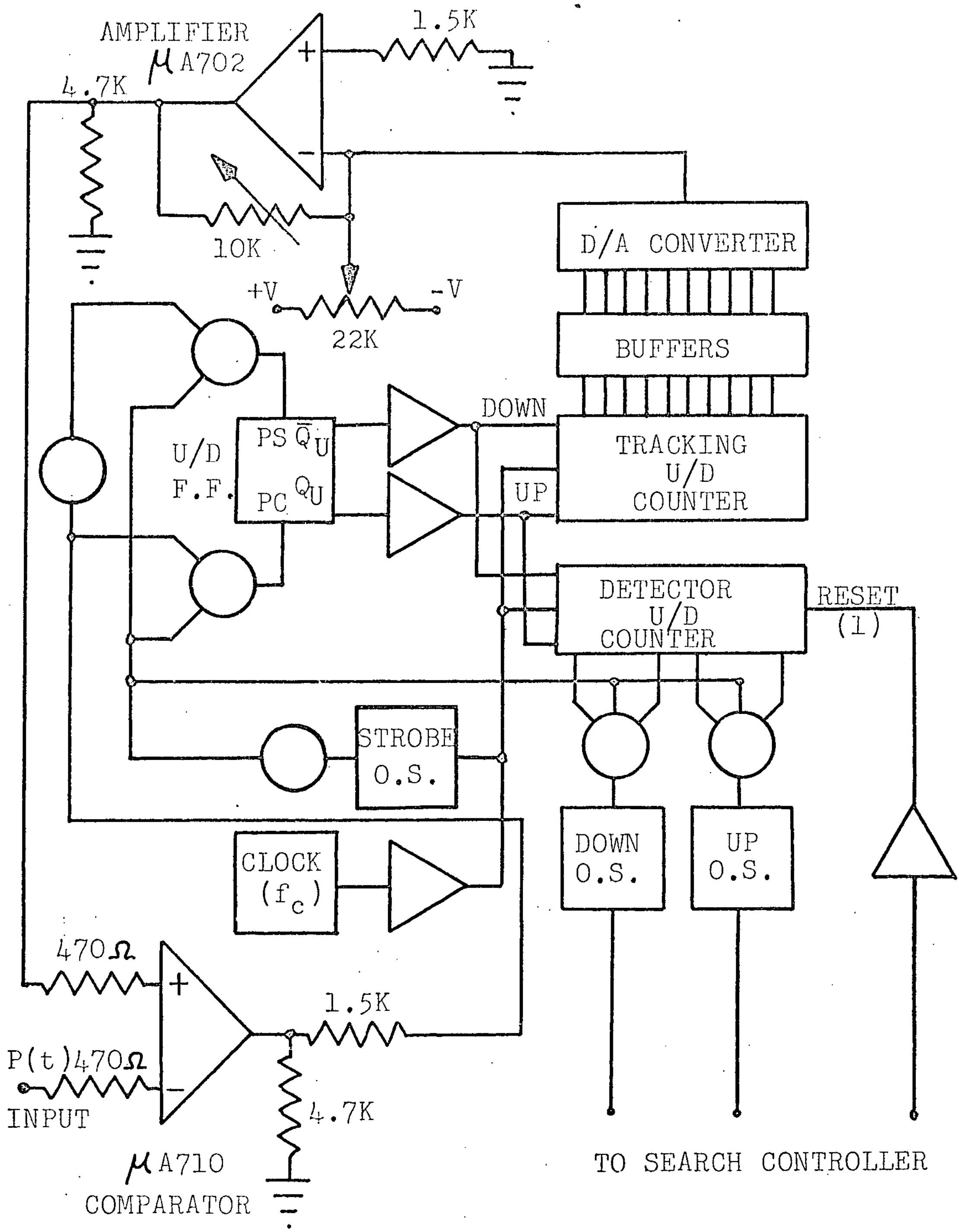


Figure 19. Complete Optimum Detector

3.2 The Adjuster Unit

The complete adjuster unit consists of n stepping motor-potentiometer combinations, as discussed in section (2.2). Each potentiometer, motor, and the associated driving circuitry is designated as an adjuster unit "channel", and is represented by the simplified block diagram below.

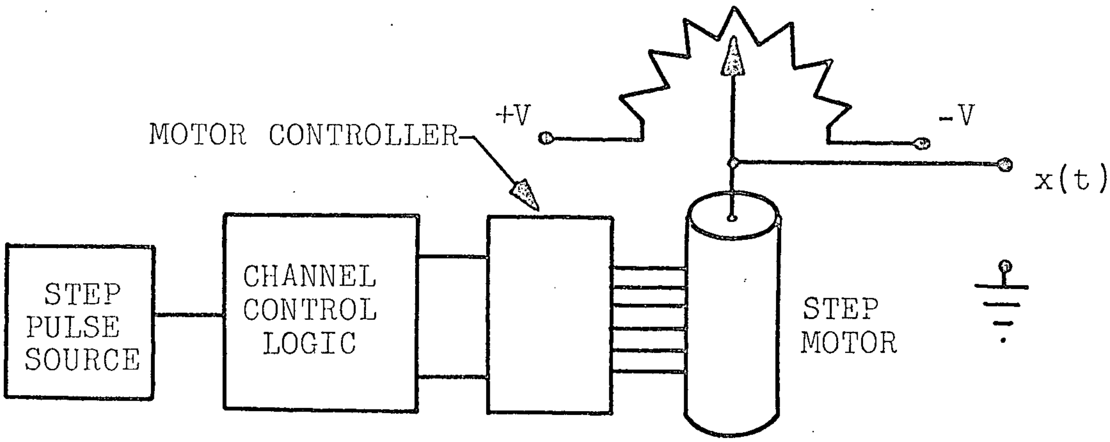


Figure 20. Adjuster Unit Channel

A voltage to frequency (V/F) converter is used to generate the step pulses required to actuate the motors. Its output pulse repetition rate is directly proportional to the analog voltage applied to its input terminal

$$f_o = K_p V_i$$

The circuit used[19], a variation of the standard unijunction transistor relaxation oscillator, is shown in Figure 21. A voltage versus frequency transfer characteristic, given in Figure 22, was obtained for the circuit, revealing that it is essentially linear over the frequency range from zero to about one thousand hertz, which covers the entire working range of the stepping motors. The output pulses are suitable for direct input to the micro-logic gates of the channel control logic section.

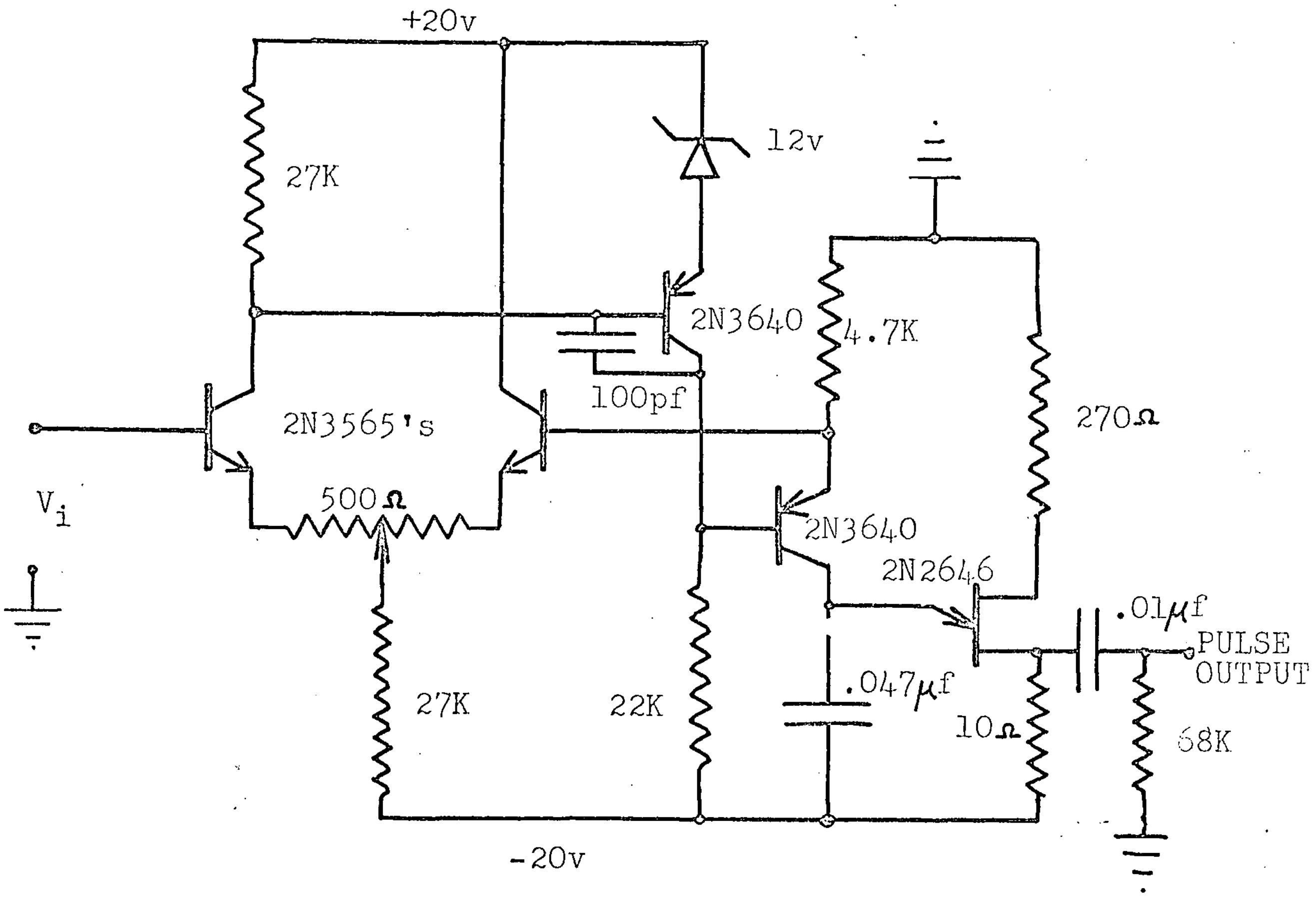


Figure 21. Voltage to Frequency Converter

Analog input to the V/F converter is obtained from the search controller from a D/A converter connected to the buffered outputs of a digital rate register whose contents, C_R, can be altered by the search controller. The step pulse frequency can hence be expressed as

$$f_o = K_p K_d C_R \qquad \text{..................(3.4)}$$

where K_d is a gain constant associated with the D/A converter.

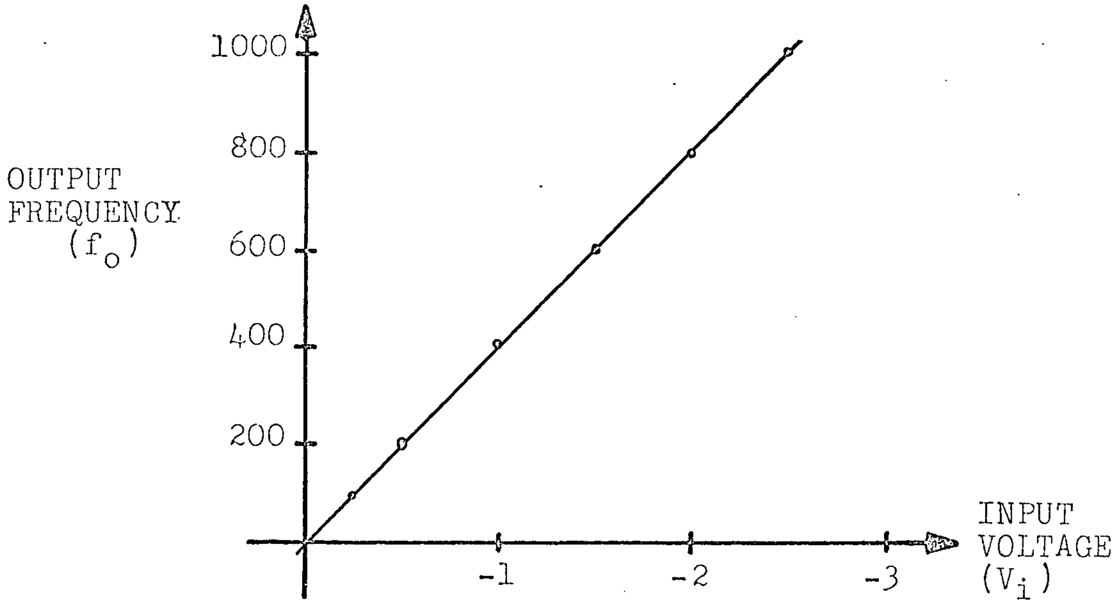


Figure 22. Voltage-Frequency Transfer Characteristic

A complete diagram of one adjuster unit channel is given in Figure 24. Channel control logic is enclosed by dotted lines. Depending on the state of the sign flip-flop, step pulses from the V/F converter are routed, through level conversion (L/C) circuits, (Figure 23), to either the clockwise or counterclockwise input terminals of the motor controllers. Enable and inhibit lines are also available, permitting the step pulses to be passed to the motors, or blocked, in a variety of ways.

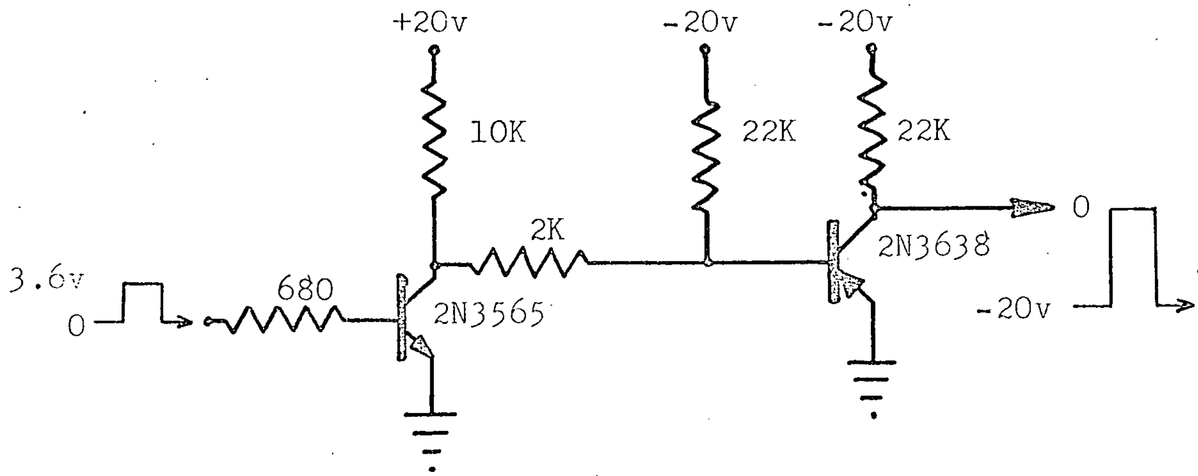


Figure 23. Step Pulse Level Converter

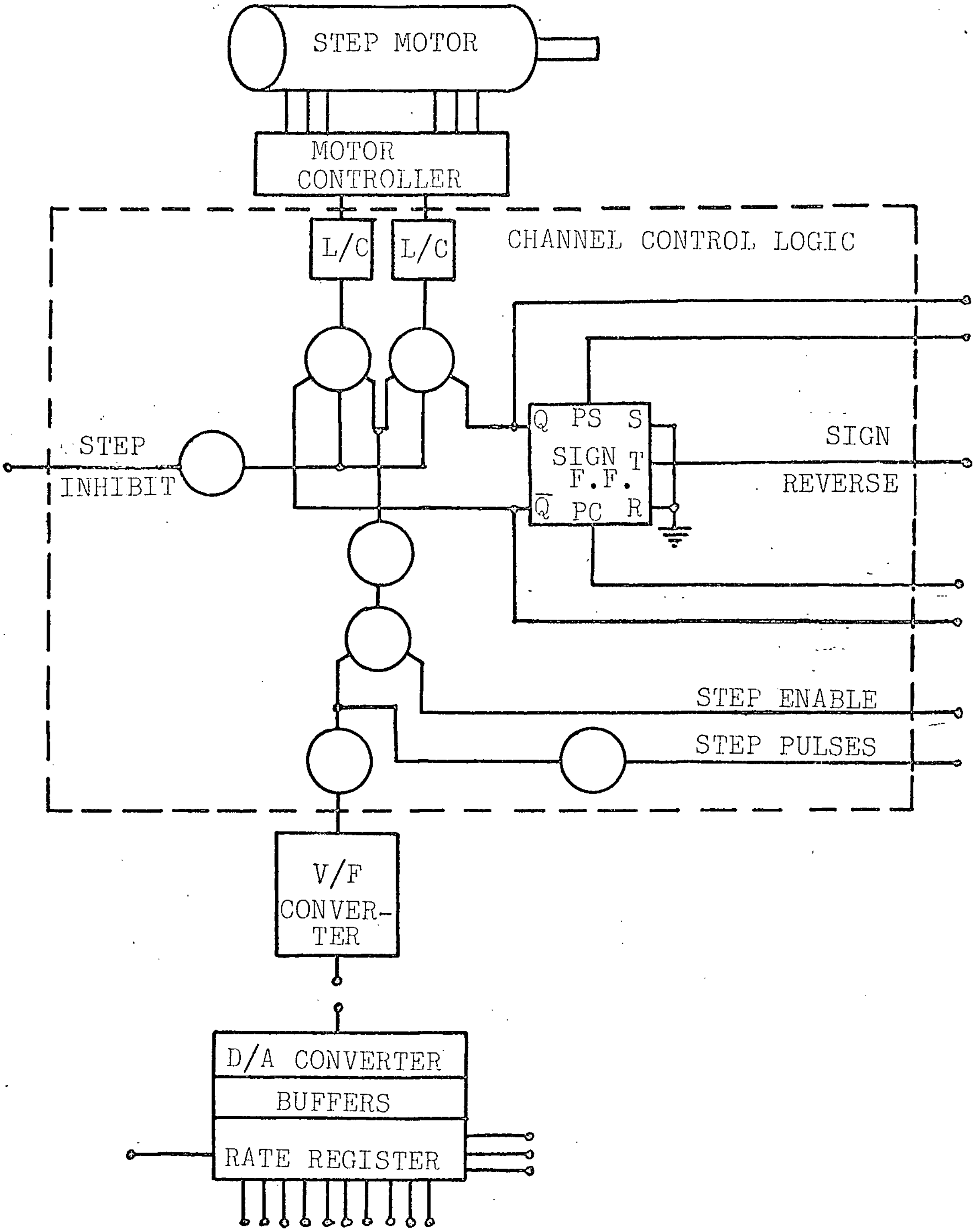


Figure 24. Adjuster Channel Logic

3.3. The Search Controller

The search controller must, in response to the pulses received from the optimum detector, co-ordinate stepping of the adjuster unit motors, so as to perform linear searches in the sequence prescribed by the P.M.A. search strategy of section (1.4). A block diagram of the search controller is given in Figure 25 below.

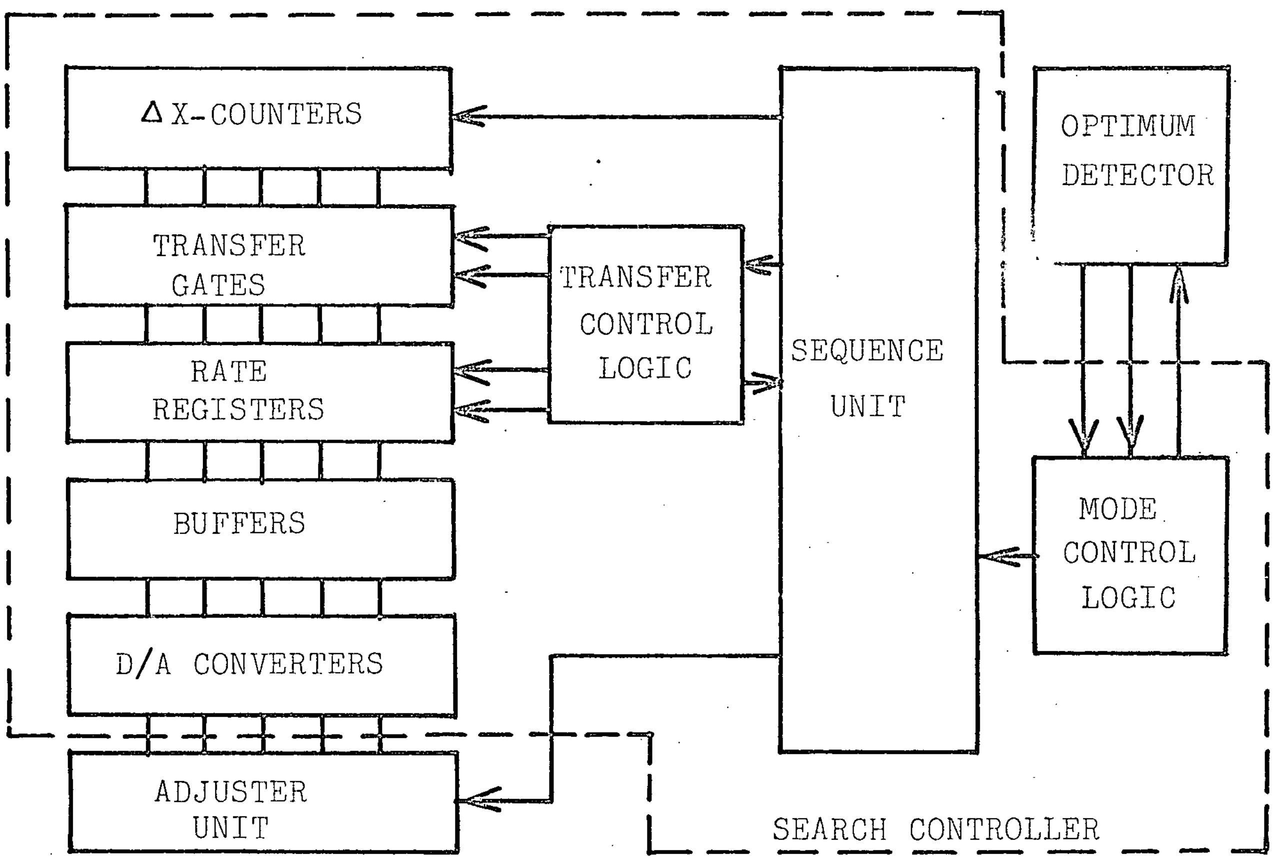


Figure 25. Search Controller Structure

Before discussing in detail the structure of the various units shown in Figure 25, a few points regarding the execution of these linear searches must be elaborated.

3.3.1. Linear Searches

Any specified linear search may begin, from a given point in space, in either of two opposing directions. Unless the point is in fact the local extreme being sought, proceeding in one of these directions will improve the performance function, while proceeding in the other direction will cause it to deteriorate. Since no a priori information is available to distinguish between these directions, the one that is favorable can only be determined by an experimental search in either one or the other of the two directions. For this purpose a preliminary experimental search can be accommodated as the first mode (Sign Test mode) of a standardized, two mode form of linear search. The second mode (Step mode), the search proper, then continues in the same direction, or reverses, depending upon the success or failure of the preliminary search.

In addition to logically supervising these two modes, the search controller must also specify for the adjuster unit the orientation in r-dimensional space of the line along which the search is to be executed. The orientation is, of course, obvious for type one linear searches, for which only one motor is involved. For searches of type two, as was stated in section (2.2) and as is illustrated below for r equal to two, the orientation is defined by the incremental changes accrued during previous type one searches in each of the r independent variables.

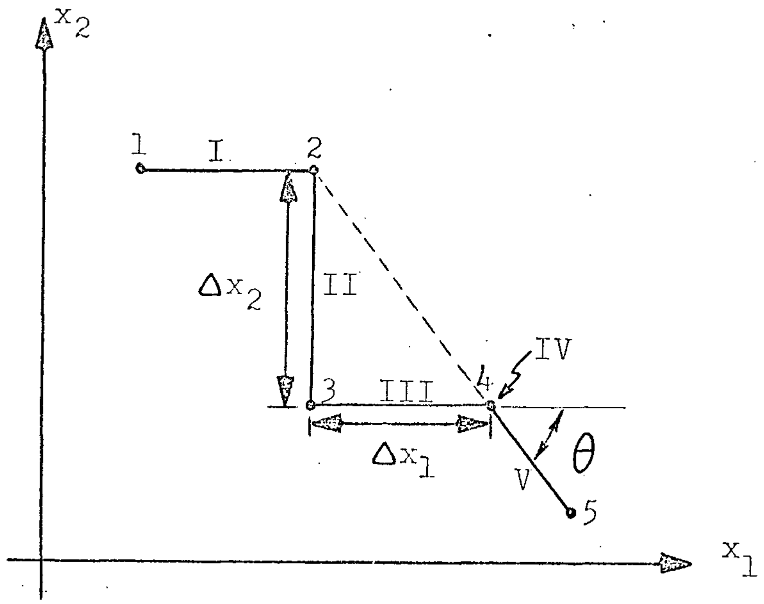


Figure 26. Two Dimensional Search Sequence

For instance, the line from point $\bar{x}_2$ to point $\bar{x}_4$ in Figure 26 is established by the previous increments $\Delta x1$ and $\Delta x2$ (stages III and II, respectively) according to the relation

$$\tan \theta = \frac{\Delta x2}{\Delta x1}$$

These increments are not known directly, but their ratio can be determined readily by counting the numbers, n_{s1} and n_{s2}, of pulses supplied to the respective stepping motors, since

$$\frac{\Delta x2}{\Delta x1} = \frac{n_{s2}}{n_{s1}}$$

By loading these pulse counts (Δx-counts) into the appropriate channel rate registers at stage IV of the search in Figure 26, it follows from Equation (3.4), section (3.2), that, for the ensuing stage V search

$$\frac{f_{o2}}{f_{o1}} = \frac{n_{s2}}{n_{s1}} \qquad \text{.......................(3.5)}$$

constraining it to proceed along the line joining $\overline{x}_1$ and $\overline{x}_3$, from $\overline{x}_3$ to the minimum at $\overline{x}_4$.

Clearly, the angular resolution between adjacent search directions is directly related to the precision with which the ratio expressed by Equation (3.5) can be defined; and this, in turn, depends upon the number of bits, R, chosen for the rate register. The minimum angle, θ_{min}, must be

$$\theta_{min} = \tan^{-1} \frac{1}{2^R - 1} \qquad \text{.................(3.6)}$$

Values of R in the range of eight to ten bits are felt to provide, in general, adequate angular resolution.

With reference once more to Figure 26, it is essential for rapid convergence of the search toward the optimum, that the stepping motors should run at or near maximum rate for all stages of the search. This requirement is readily met for search stages I, II, and III, where only one motor is running at a time, by setting the appropriate rate registers for maximum V/F converter output rates. Although not as easily done for searches of type two, since the motor speeds are set by the various pulse counts loaded into the rate registers at stage IV, search speed can still be maintained near maximum by simultaneously shifting the contents of all registers toward their most significant ends. Such shifting results in a simple binary scaling of the transferred pulse counts, leaving their ratios unchanged, yet forcing at least one motor channel to operate close to top speed.

3.3.2 ΔX-Counters and Data Transfer

To illustrate clearly the requirements of the Δx-counters involved in establishing the directions of type two linear searches, the three dimensional search sequence of Figure 7 is repeated below, with all points and all pertinent Δx increments labelled.

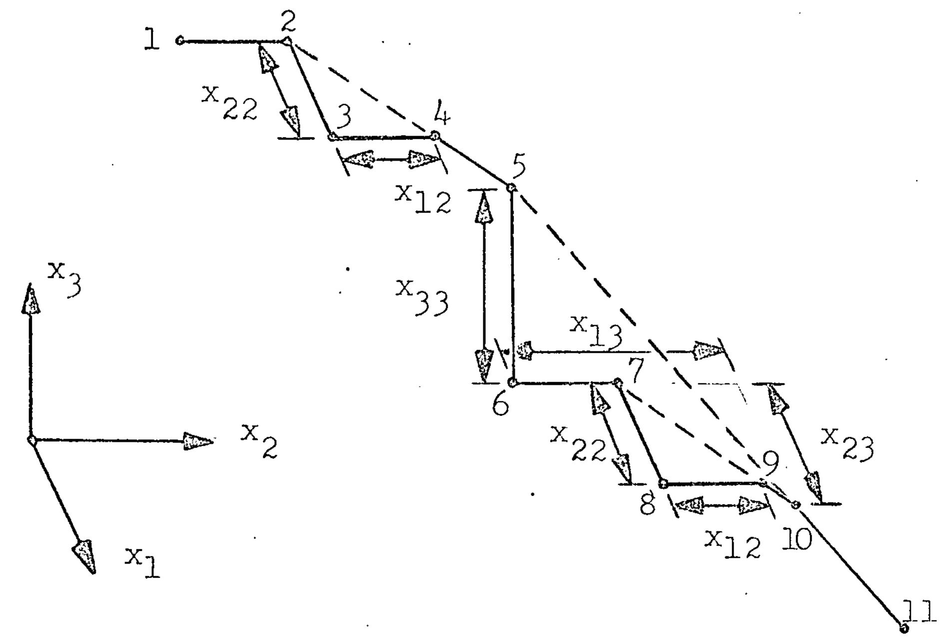


Figure 27. Three Dimensional Search Sequence

Three type two linear searches are shown. Two of these, from points four to five and nine to ten, involve only two dimensions (or variables), while the third, from points ten to eleven, involves all three. A double subscript notation is used for the related Δx increments, the first subscript referring to the variable involved, and the second to the dimensionality of the type two search for which the increment applies. For example, Δx_{12} designates an increment in the variable x_1, while Δx_{13} denotes an increment in the same variable but related to a type two search in three dimensions.

From Figure 27, it is clear that there is some overlapping of the increments required for the two and three dimensional type two searches. This overlapping necessitates the use of two Δx-counters to define the two dimensional type two search directions, and an additional three Δx-counters to define the three dimensional search direction. For higher dimensions the extension is expressed by the recursion equations

$$K_N = N + K_{N-1}$$

and

$$K_2 = 2$$

where K_N is the total number of Δx-counters required for a search in N dimensions.

An array of fourteen counters as required for five dimensions is shown in Figure 28.

x12	x22			
x13	x23	x33		
x14	x24	x34	x44	
x15	x25	x35	x45	x55
Transfer Gates	Transfer Gates	Transfer Gates	Transfer Gates	Transfer Gates
Rate Register	Rate Register	Rate Register	Rate Register	Rate Register

Figure 28. Five Dimensional ΔX-Counter Array

The notation follows that introduced for the three dimensions of Figure 27. The counters of each column are associated with the same variable. Those of each row correspond to a type two search of a given dimension, and only when Δx-counts have been obtained for all counters in that row is the type two search defined. Then, in order for it to be executed, the entire row of Δx-counts must be transferred into the appropriate rate registers which are shown below each column.

As can be seen for Δx_{12}, Δx_{22} and Δx_{33} in Figure 27, the shaded counters of Figure 28 are distinguished from the rest in that their pulse counts are always attained over only one linear search, while the others involve several sequential searches. The shaded counters are designated (arbitrarily) as primary counters, and the others as secondary counters.

Only the magnitude of any primary Δx-count is required, since the state of the corresponding adjuster channel sign flip-flop must, at the end of the type one search which produced the Δx-increment concerned, already represent the correct initial polarity for that channel for the ensuing type two search. The secondary counters, however, must count through a number of linear searches, and hence a succession of Sign Test and Step modes, and must therefore record not only the correct Δx-count, but its net polarity as well.

Both primary and secondary counters use the same basic up/down counter complete with U/D control and strobe pulse circuitry that was described for the optimum detector counters.

The primary counters require up/down capability in order to correctly determine the Δx-count in situations for which the preliminary Sign Test mode results in a motor reversal. Such a situation is depicted in Figure 29.

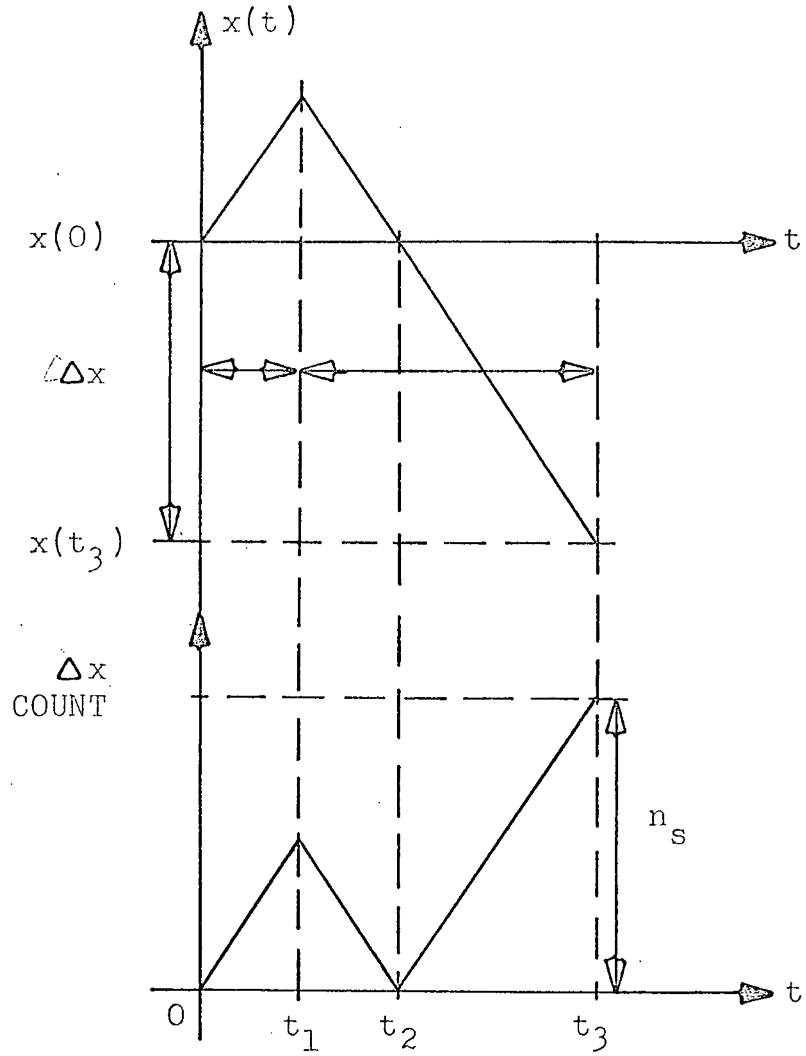


Figure 29. Primary ΔX-Counter Operation

Secondary counters require up/down capability to account for a possible succession of such motor reversals.

The complete Δx-counter circuitry is given in Figure 30. Circuit connections shown boldface are used by the primary counters, those shown dotted by the secondary counters,

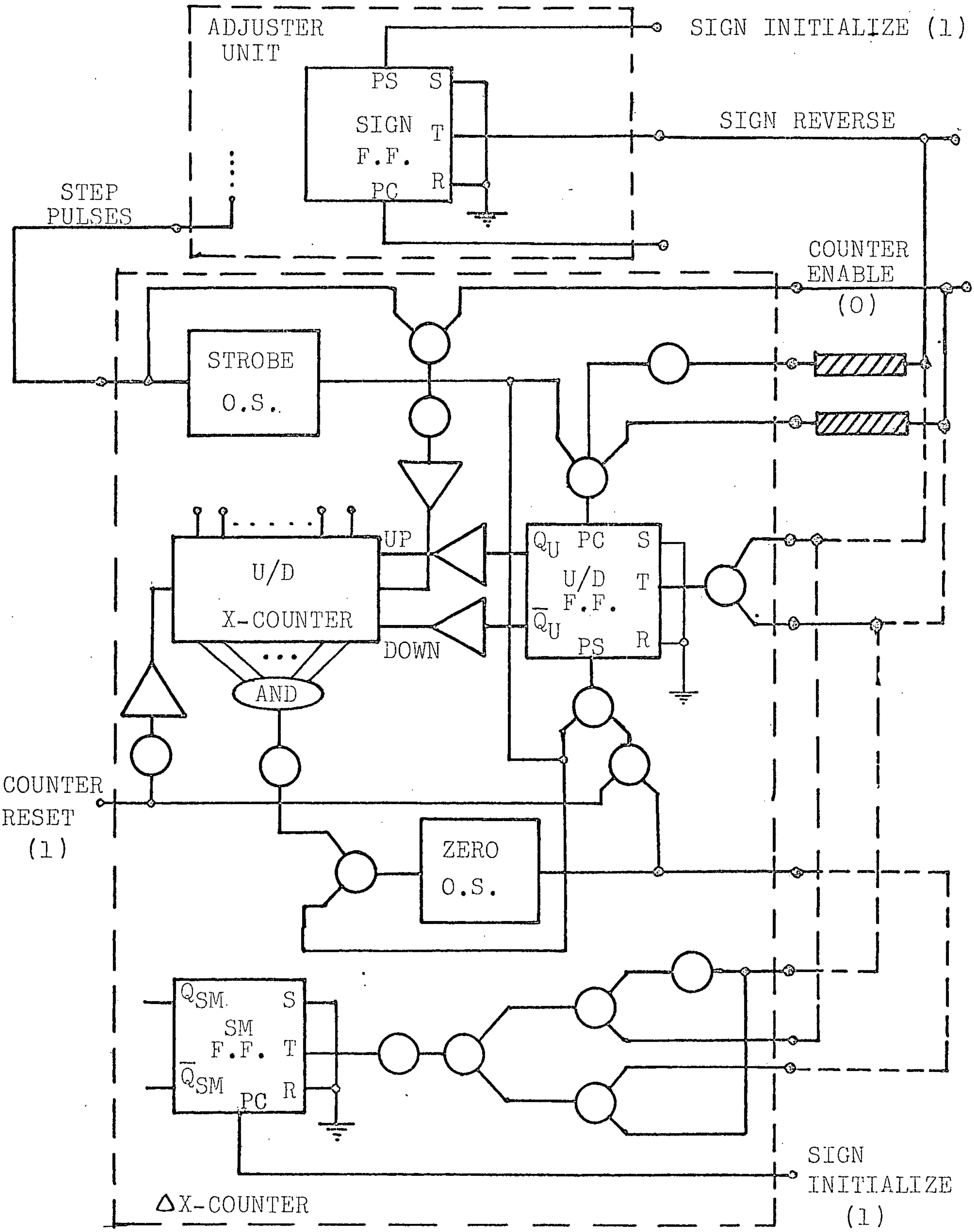


Figure 30. Complete ΔX-Counter Circuit

and the rest are common to both. A full counter length AND gate is provided to detect, in conjunction with the zero one-shot, the transition of the counter into the reset (zero) state, and to accordingly reverse the counting direction (refer to Figure 29).

A sign memory (SM) flip-flop is provided for each secondary counter to record the net polarity of any Δx-increment. It must first be "initialized", along with the corresponding sign flip-flop. It then receives the same sign reversal pulses applied to the sign flip-flop until the Δx-count begins, after which it is reversed only by pulses from the zero one-shot. The SM flip-flop state at the conclusion of the Δx-counts must thus represent the net sign associated with the Δx-increment.

Immediately following the accumulation of all the Δx-counts for any one level of the counter array of Figure 28, and before the associated type two linear search can be executed, this data must be transferred into the rate registers which drive, through D/A converters, the V/F converters of the corresponding adjuster channels.

To avoid the much increased circuit complexity for the Δx-counters that would be required to enable them to shift in addition to counting up and down, it was decided to perform the transfer in parallel rather than serially. One rate register, the associated parallel transfer gates and one of the up to four possible Δx-counters from which the rate register may be loaded, are depicted in Figure 31.

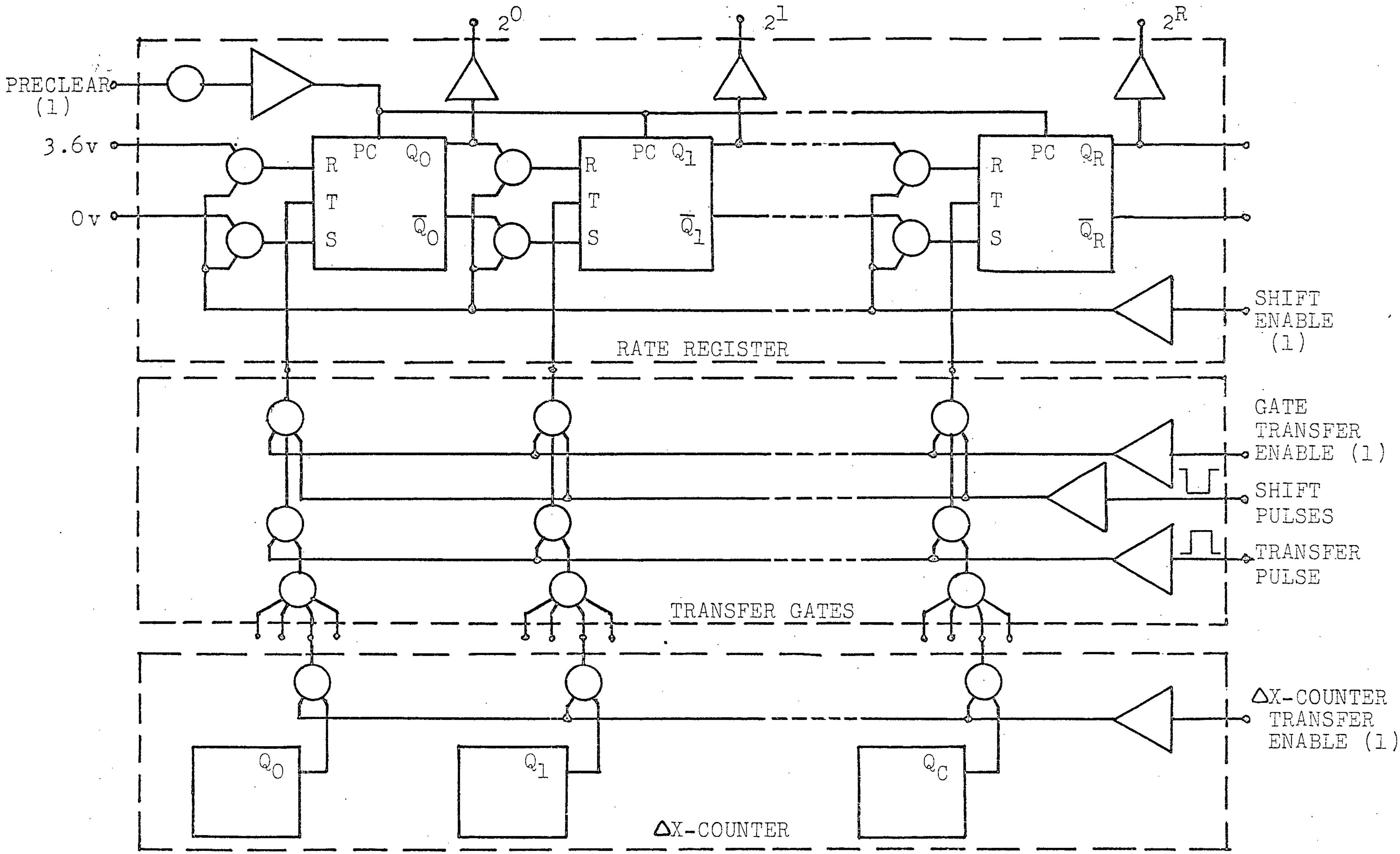


Figure 31. Rate Register, Transfer Gates and ΔX-Counter

A complete transfer is executed in two stages--a parallel transfer from the Δx-counters into the rate registers followed by a serial shift in the rate registers. The shift is toward the most significant ends of all rate registers, and ensures that at least one adjuster channel will operate close to maximum rate in the subsequent type two linear search.

In order for a parallel transfer to occur, it is necessary that all rate registers involved have previously been reset, and that logical one levels are present on both the transfer enable and shift enable lines. This being the case, a positive pulse applied at the transfer pulse input will be transmitted through the gates to the toggle inputs of those rate register flip-flops for which the corresponding Δx-counter flip-flops contain logical zeros. The Δx-counts are thus transferred in complement form to the rate registers. Inverting buffers on the rate register flip-flops, however, provide the transferred Δx-counts in uncomplemented form to the D/A converters.

Transfer of SM data to the sign flip-flops is complicated by the fact that other levels of Δx-counters may be monitoring the state of the sign flip-flops for which the SM data are destined. This precludes the required initial pre-clearing of the sign flip-flops, since the counters, which monitor the flip-flop toggle inputs (Figure 30), would be disrupted. To circumvent this problem, the SM data parallel transfer is executed in two steps, from the SM flip-flop to an extra scratch pad (SP) flip-flop, and from there to the sign flip-flop. The required circuitry is shown in Figure 32.

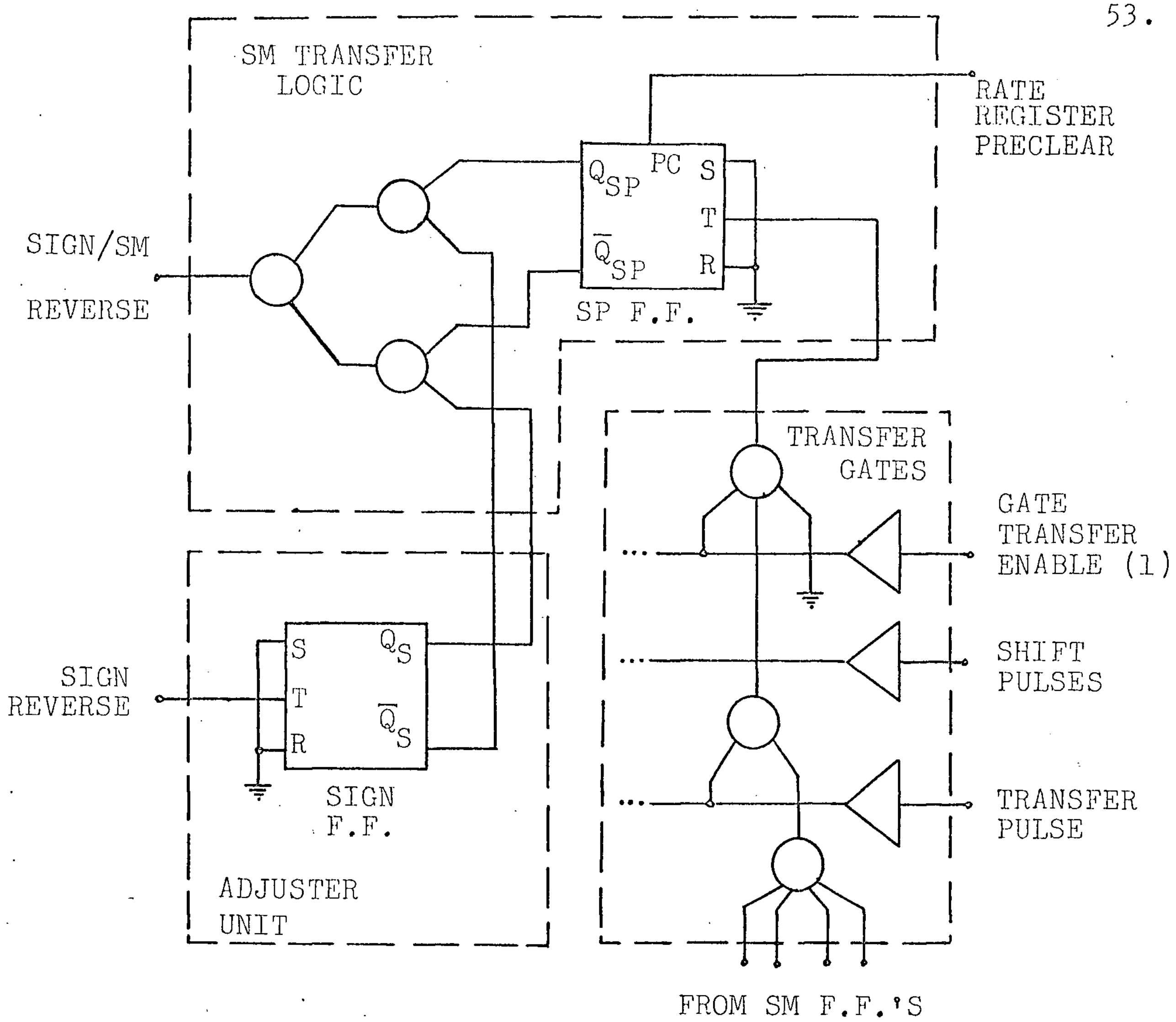
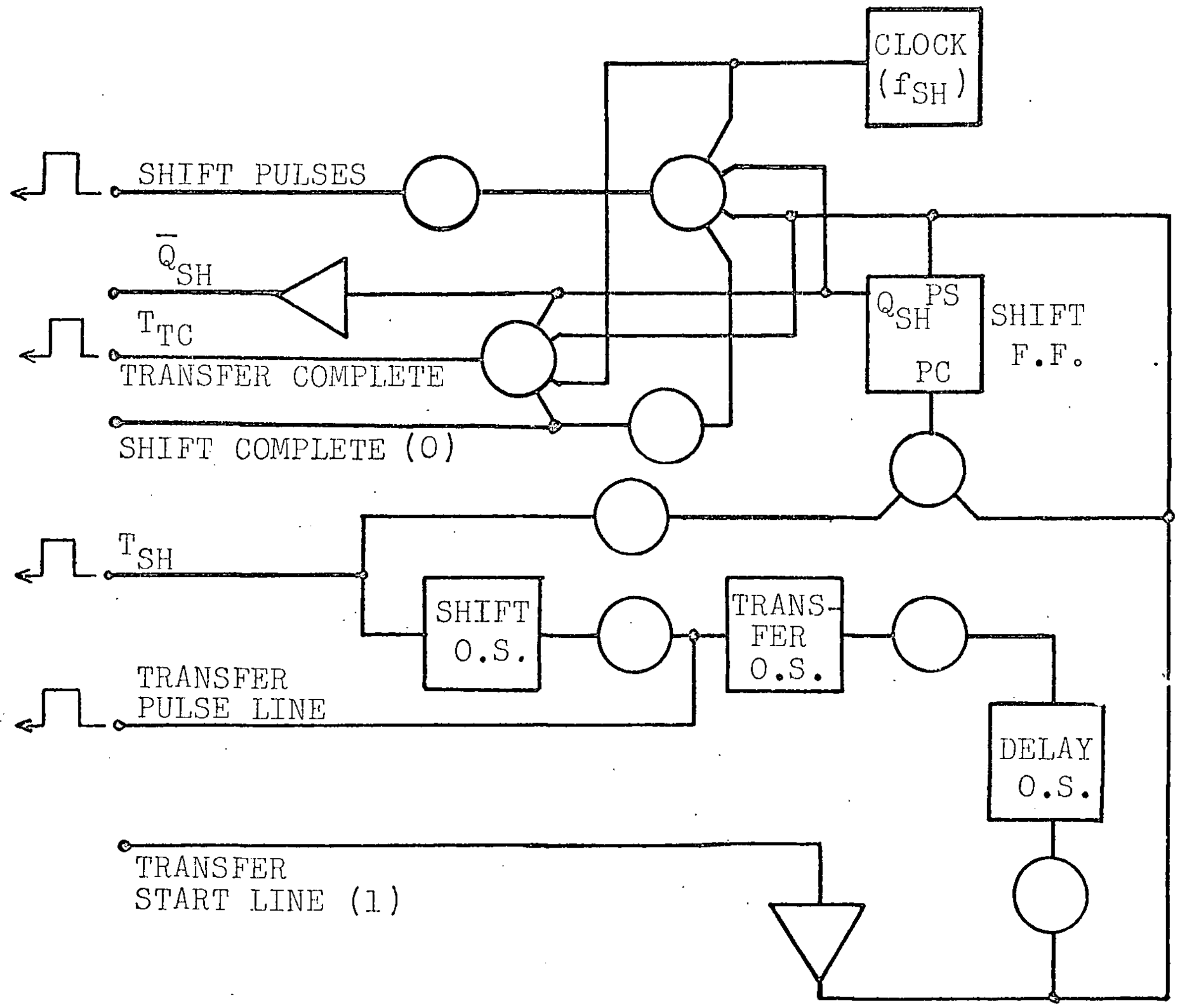


Figure 32. Sign Memory Transfer Logic

The SP flip-flop can be pre-cleared so that the first step can be executed using similar transfer gates and the same transfer pulse as used for transfer of the Δx-counts. The second step uses an additional pulse to toggle the sign flip-flop whenever its state and that of the SP flip-flop differ.

Following the parallel transfer stage, a logical zero appears on the shift enable line, and the rate register shift begins. It continues until the most significant bits of at least one of the transferred Δx-counts occupies the most significant bit position of its rate register.

Transfer control logic which supervises all phases of the above parallel and shift stages is shown in Figure 33. A transfer is initiated by a logical zero to one level shift on the transfer start line. Several delays are inserted to ensure quiescent conditions immediately before both the parallel transfer and serial shift stages.



<u>Figure 33.</u> Transfer Control Logic

Following completion of the shift stage, a transfer complete signal is generated to indicate that the appropriate type two search may commence.

A timing diagram is provided in Figure 34, indicating the waveforms at pertinent points of both the transfer control logic and the Δx-counter, transfer gate, rate register configuration of Figure 31.

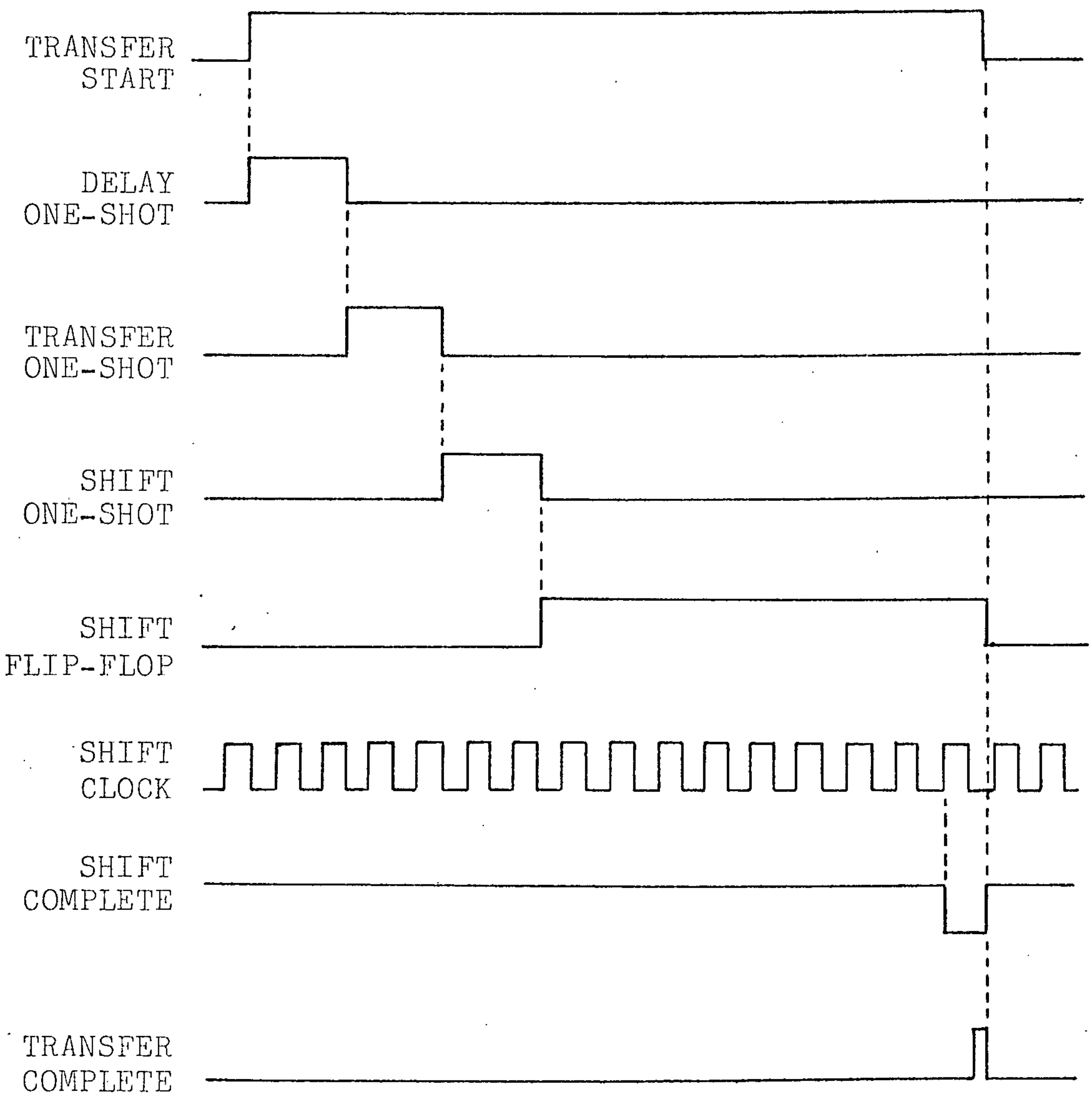


Figure 34. Transfer Timing Diagram

3.3.3 Sequence Control

The bulk of the circuitry required for the automatic optimizer has now been presented. It remains to correctly sequence the series of linear searches, Δx-counts and data transfers so that the steps of the P.M.A. search are executed. For this purpose, a sequence unit was designed as the heart of the search controller. It provides co-ordination not only among the remaining blocks of the search controller itself, but also between it and both the adjuster unit and the optimum detector.

The sequence unit consists of a hierarchy of six-bit ring counters, each with its own input logic network, as depicted in Figure 35.

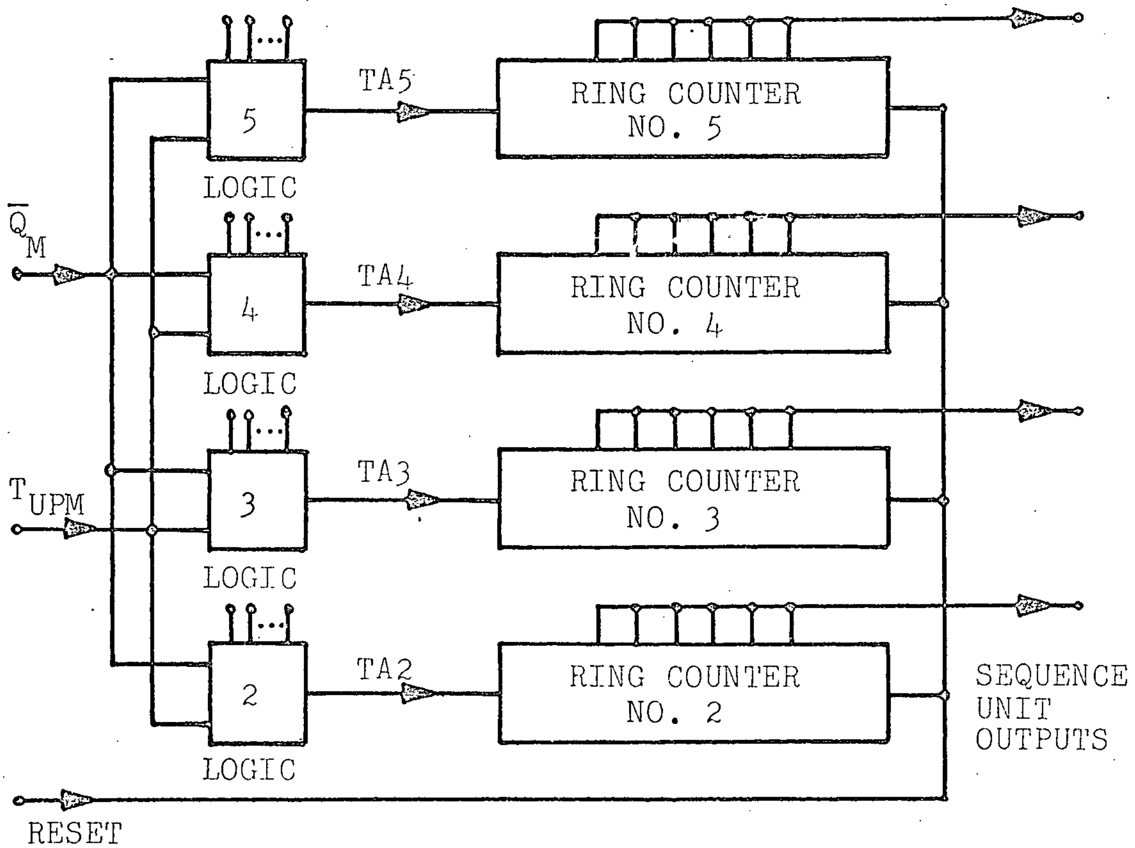


Figure 35. Sequence Unit Structure

Each individual ring counter consists of the circuit shown in Figure 36. Both external and automatic internal resets are available, and access is provided to all six flip-flop outputs, only one of which at any time may be a logical one. Each counter thus has six possible "states", each of which may be entered in sequence by triggering the advance one-shot through the input logic network.

According to the recursive nature of the P.M.A. search strategy, for which a two dimensional search consists of a set of linear searches, a three dimensional search a set of two dimensional searches, and so on, the ring counter levels shown in Figure 35 correspond in number to the dimension of the search over which each has direct control. No level is indicated for linear or one dimensional searches. Instead, they are supervised by the mode control logic given in Figure 37.

Pulses from the optimum detector are routed either to the adjuster channel sign flip-flops for motor reversal, or to the sequence unit input logic networks for ring counter advances, depending upon the state (Sign Test or Step) of the mode flip-flop. The detector reset flip-flop, in addition to resetting the detector counter of the optimum detector, also inhibits all stepping motors for a short time following each detector up pulse, thereby avoiding any possible logical hazards associated with motor reversal or counter advances.

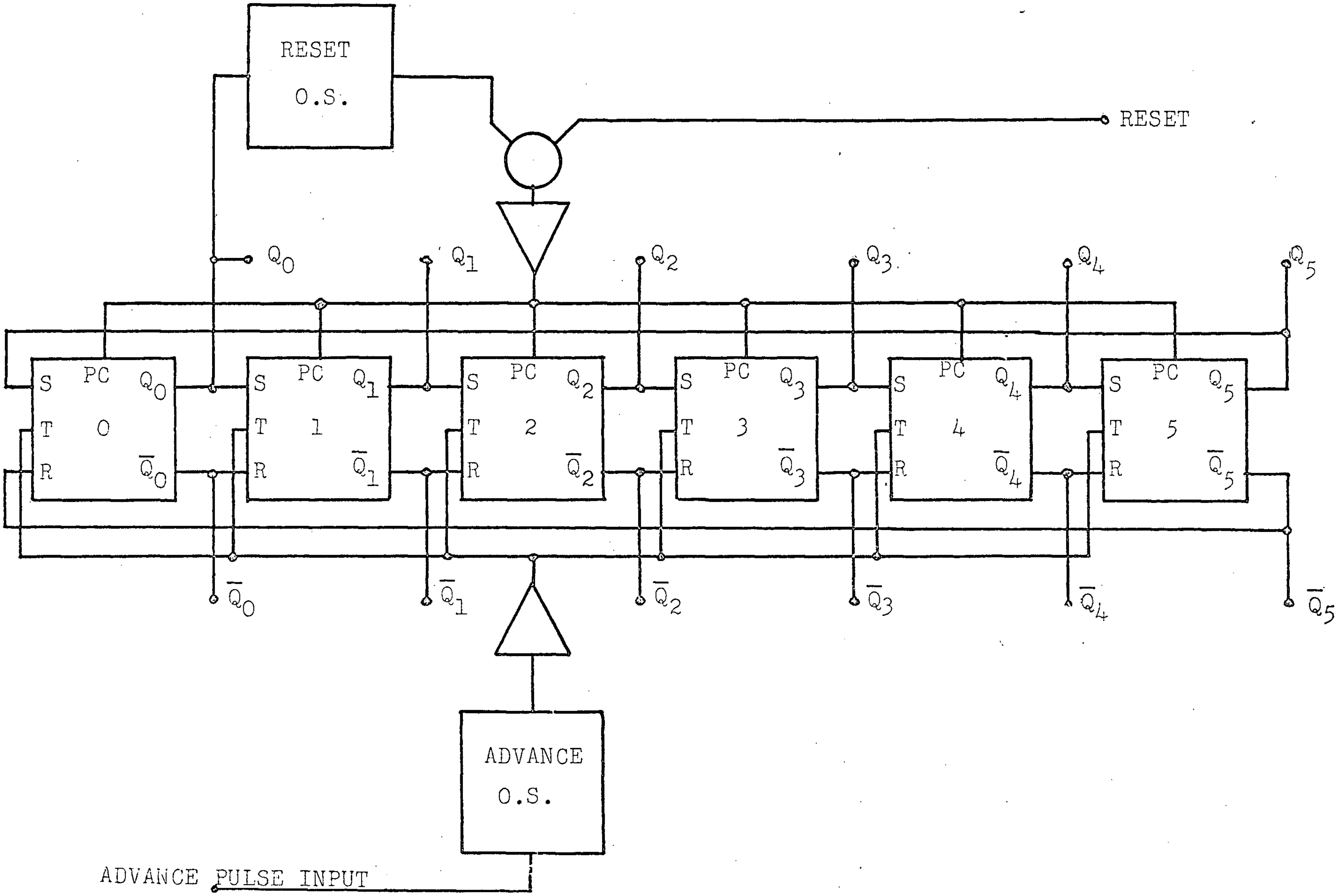


Figure 36. Ring Counter Circuit

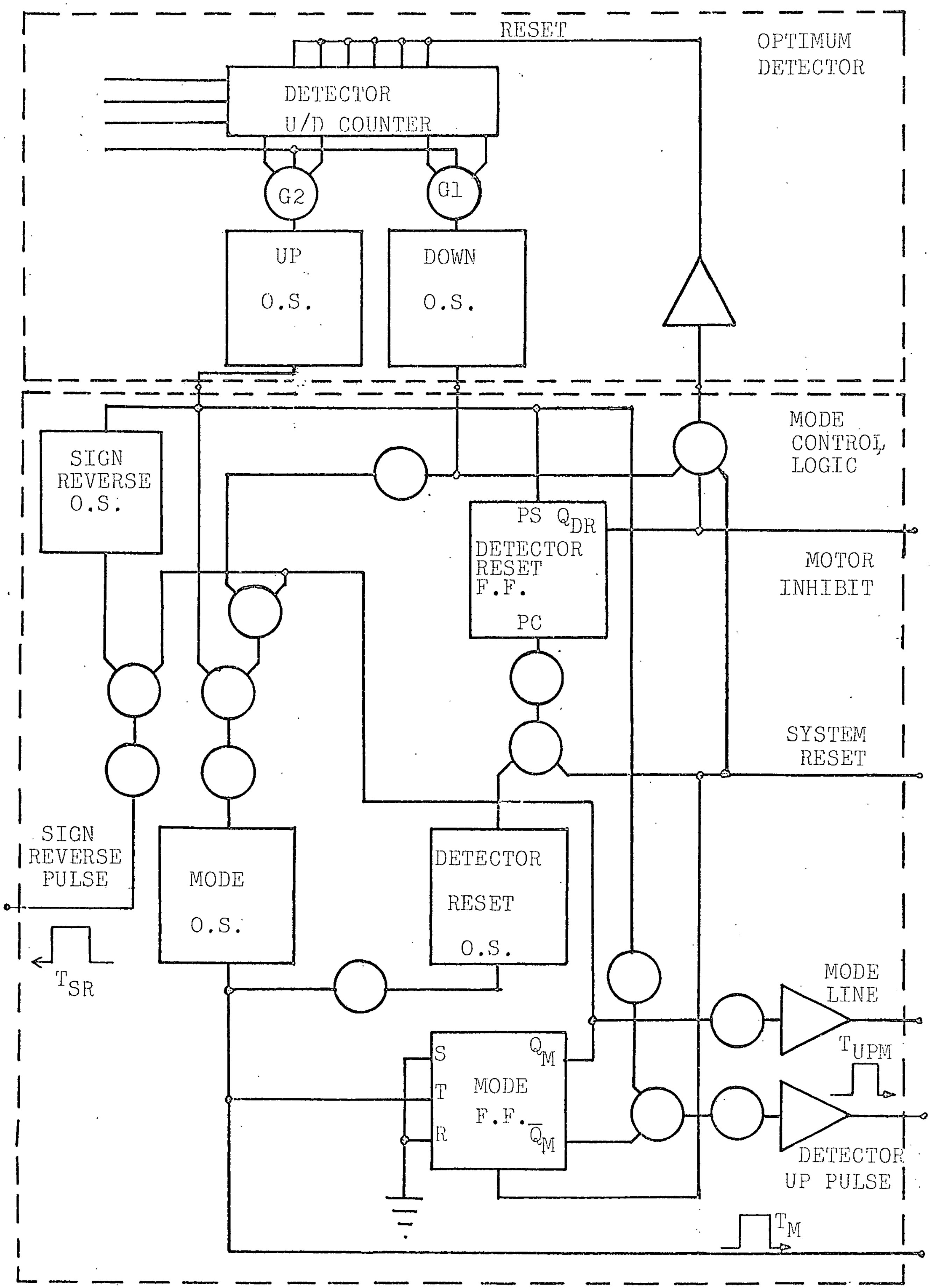


Figure 37. Mode Control Logic

The complete sequence of optimizer events required for a five dimensional P.M.A. search is presented in the chart of Figure 38. Each step of the search algorithm implementation corresponds to one of the sequence unit "states" (combination of ring counter states) listed in the left-most column of the chart. The states must progress in the exact sequence shown in order to execute the desired search. The second column lists those step motors involved in each of the search steps. The transfers required at various stages of the sequence are indicated in the third column, according to the number of dimensions or variables involved, and hence, according to the various levels shown in the Δx-counter array of Figure 28. The fourth column of the chart indicates the points in the search sequence at which the secondary counters must be initialized, as mentioned in connection with the sign and sign memory flip-flops. The various counters, both primary and secondary, which must count during each search step, are listed in the final column.

Although not included in the chart, it must be remembered that each of the search steps begins with a Sign Test mode and concludes with a Step mode. On the other hand, each of the transfers can, for convenience, be considered to occur entirely during a Sign Test mode.

It may be helpful, for complete understanding of the chart, to study its first fourteen steps along with the illustration of the three dimensional search sequence shown previously in Figure 27.

From this chart, the interconnections required between the sequence unit and the remainder of the optimizer can readily be derived.

For example, if Q_{ij} refers to the Q output of flip-flop i in ring counter number j, then it can be determined by inspection of the M2 column of the chart that the step enable input (which will be abbreviated SE2) to adjuster unit channel two must be:

$$SE2 = Q_{22} + Q_{52} + Q_{53} + Q_{54} + Q_{55}$$

Similarly, the input logic network required to produce the correct trigger pulses (TA2) for the advance one-shot of ring counter number two is defined by the RC2 column of the chart to be:

$$TA2 = T_{UPM} \quad Q_{12} + Q_{22} + Q_{32} + Q_{52} \quad + Q_{02} \quad Q_{23} + Q_{24} + Q_{25}$$

$$+ \overline{Q}_M \; T_{TC} \; Q_{42}$$

where $\overline{Q}_M$ refers to the mode flip-flop output and T_{UPM} to the detector up pulses, both of which are produced by the mode control logic of Figure 37. T_{TC} is a transfer complete pulse from the transfer control logic. The remaining interconnection functions required are tabulated in Appendix III.

STATE				MOTORS					TRANSFERS				ΔX-COUNTERS			
RC5	RC4	RC3	RC2	M1	M2	M3	M4	M5	2D	3D	4D	5D	INITIATE 3D	4D	5D	COUNT
0	0	0	0													
1	1	1	1	X												
			2		X											ΔX22
			3	X												ΔX12
			4						X							
			5	X	X											
1	1	2	0			X							X			ΔX33
1	1	3	1	X												ΔX13
			2		X											ΔX22, ΔX23
			3	X												ΔX12, ΔX13
			4						X							
			5	X	X											ΔX13, ΔX23
1	1	4	0							X						
1	1	5	0	X	X	X										
1	2	0	0				X							X		ΔX44
1	3	1	1	X												ΔX14
			2		X											ΔX22, ΔX24
			3	X												ΔX12, ΔX14
			4						X							
			5	X	X											ΔX14, ΔX24
1	3	2	0			X										ΔX33, ΔX34
1	3	3	1	X												ΔX13, ΔX14
			2		X											ΔX22, ΔX23, ΔX24
			3	X												ΔX12, ΔX13, ΔX14
			4						X							
			5	X	X											ΔX13, ΔX23, ΔX14, ΔX24
1	3	4	0							X						
1	3	5	0	X	X	X										ΔX14, ΔX24, ΔX34
1	4	0	0								X					
1	5	0	0	X	X	X	X									
2	0	0	0					X							X	ΔX55

Figure 38. Five Dimensional Sequence Chart

STATE				MOTORS					TRANSFERS				ΔX-COUNTERS			
RC5	RC4	RC3	RC2	M1	M2	M3	M4	M5	2D	3D	4D	5D	INITIATE 3D	4D	5D	COUNT
3	1	1	1	X												ΔX15
			2		X											ΔX22, ΔX25
			3	X												ΔX12, ΔX15
			4						X							
			5	X	X											ΔX15, ΔX25
3	1	2	0			X										ΔX33, ΔX35
3	1	3	1	X												ΔX13, ΔX15
3	1	3	2		X											ΔX22, ΔX23, ΔX25
			3	X												ΔX12, ΔX13, ΔX15
			4						X							
			5	X	X											ΔX13, ΔX23, ΔX15, ΔX25
3	1	4	0							X						
3	1	5	0	X	X	X										ΔX15, ΔX25, ΔX35
3	2	0	0				X							X		ΔX44, ΔX45
3	3	1	1	X												ΔX14, ΔX15
			2		X											ΔX22, ΔX24, ΔX25
			3	X												ΔX12, ΔX14, ΔX15
			4						X							
			5	X	X											ΔX14, ΔX24, ΔX15, ΔX25
3	3	2	0			X							X			ΔX33, ΔX34, ΔX35
3	3	3	1	X												ΔX13, ΔX14, ΔX15
			2		X											ΔX22, ΔX23, ΔX24, ΔX25
			3	X												ΔX22, ΔX13, ΔX14, ΔX15
			4						X							
			5	X	X											ΔX13, ΔX23, ΔX14, ΔX24, ΔX15, ΔX25
3	3	4	0							X						
3	3	5	0	X	X	X										ΔX14, ΔX24, ΔX34, ΔX15, ΔX25, ΔX35
3	4	0	0								X					
3	5	0	0	X	X	X	X									ΔX15, ΔX25, ΔX35, ΔX45
4	0	0	0									X				
5	0	0	0	X	X	X	X	X								

Figure 38. (continued)

4. SYSTEM EVALUATION AND CONCLUSIONS

An automatic optimizer based on the design presented in the previous chapter was built and tested for two of the proposed five dimensions. Before describing the tests performed and the results obtained, a few points regarding the actual system construction will be elaborated.

4.1 Optimizer Construction

Motorola MRTL digital integrated circuits were used for all logic functions, minimizing both system bulk and component cost.

The optimum detector was constructed with a ten bit (N) tracking counter and a four bit (M) detector counter. A precision ten bit Pastoriza RSN2698 D/A converter module was used in conjunction with a Fairchild μA702 integrated operational amplifier for the required D/A conversion of the tracking counter contents. The comparator was a Fairchild μA710 integrated circuit. The D/A amplifier gain was adjusted for a total input signal capability of ten volts, fixing the tracking step size, V_{ST}, at

$$V_{ST} = \frac{10}{2^{10} - 1} = 10 \text{ millivolts,}$$

and providing optimum detector sensitivity to fifty millivolt changes in the performance function, P(t). The quality factor, E(M,N), defined in section (3.1), is

$$E(4,10) = \frac{5 \text{ X } 100}{2^{10} - 1} = 0.5\%$$

A clock frequency of sixty kilohertz was selected, permitting an input slewing rate, S_V, of

$$S_V = \frac{10 \times 60 \times 10^3}{(2^N - 1)} = 600 \ \frac{\text{volts}}{\text{second}}$$

This was found to be adequate for all performance functions tested.

Turning now to the adjuster unit, IMC stepping motors size 008-008 were used. A gear ratio was chosen for each channel to give a step motor to potentiometer relation of 1805 steps per potentiometer revolution. This imposed a limit of about 0.05 per cent on the resolution, parallel to each co-ordinate, with which the performance function space could be searched for the optimum. An additional result of this choice was to require that all Δx-counters contain at least eleven bits in order to permit the maximum possible count of 1805 steps to be recorded. In fact, twelve bit counters were used, giving a maximum count of 4095 steps and allowing for some future variation in the search resolution.

The length of the rate registers was then necessarily also twelve bits, so that all possible Δx-counts could be transferred and accepted without discarding any bits. This is, of course, essential, since low order bits, associated with small Δx-counts, may well become significant after shifting in the rate register.

Although the desired achievable search resolution along any given line in the space of the independent variables fixed the rate register length at twelve bits, it was pointed out in section (3.3.1) that only seven or eight rate register bits were needed to provide adequate angular resolution between adjacent search lines. For instance, assuming eight bits and substituting into Equation (3.6) gives

$$\theta_{min} = \tan^{-1} \frac{1}{255} \cong 0.23 \text{ degrees.}$$

For seven bits,

$$\theta_{min} = \tan^{-1} \frac{1}{127} \cong 0.45 \text{ degrees.}$$

Since further precision would have been wasted, only the eight most significant rate register bits were used by the D/A converters to obtain the analog input voltages for each V/F converter. Discrete component D/A converters, as shown in Figure 39, with precision[20] to plus or minus one bit in eight bits, were used.

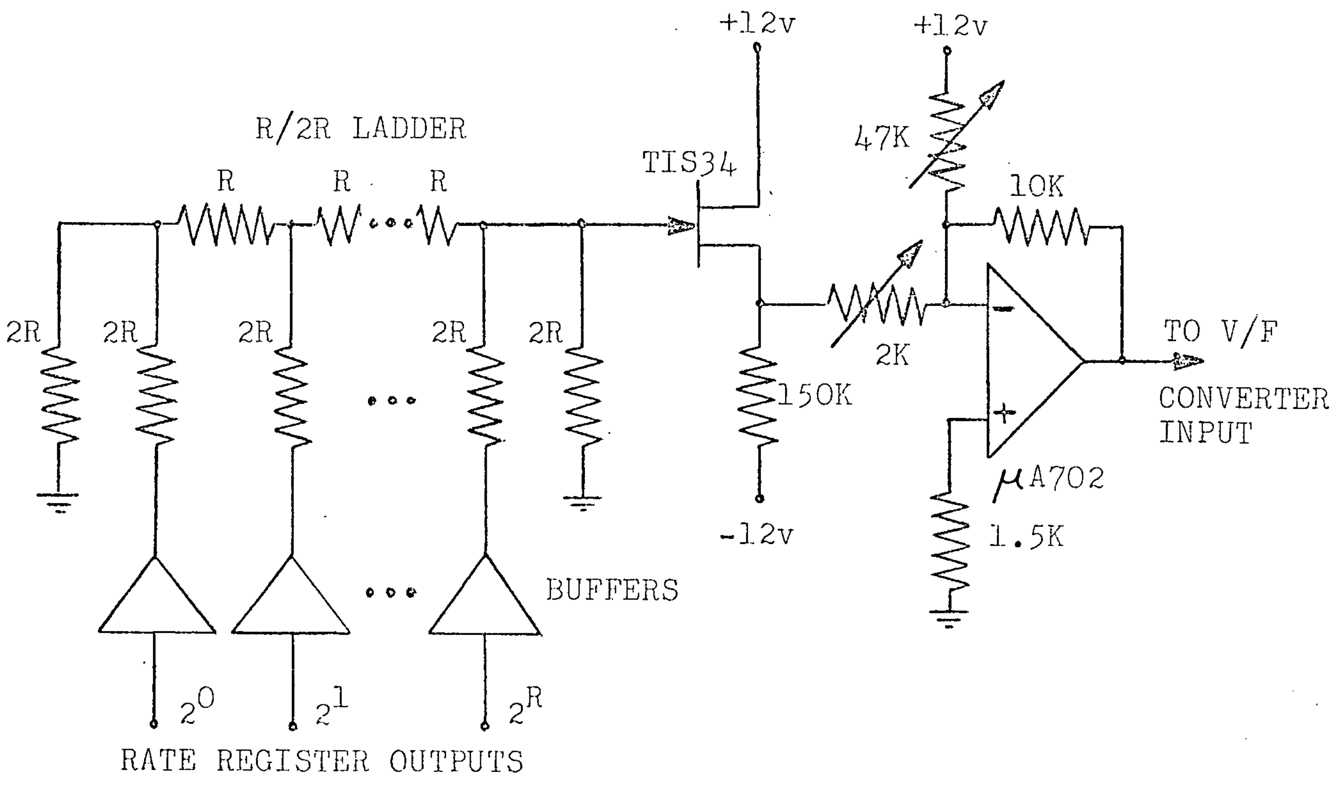


Figure 39. Rate Register D/A Converter

Based on the cost of the two dimensional system constructed, and using the most recent component price schedules (unit quantity), a total cost of just under $1000 was estimated for the complete five dimensional automatic optimizer. This is roughly one tenth of the cost of any small, presently available, general purpose computer capable of performing the equivalent optimization function. The cost of the five stepping motors ($750) was not included in the above estimate, since they essentially perform D/A conversion between the optimizer and the performance function computer, and such conversion would not normally be included in the cost of a general purpose computer.

A further substantial reduction in the estimated cost may be possible in the near future if the full potential of large-scale integrated circuit technology (LSI), presently under intensive investigation, can be realized. The large number of identical circuit configurations used in the optimizer design are particularly suitable for adaptation to LSI. Comparable reductions in the cost of general purpose machines cannot be expected, unless the present high cost of their software systems can also be dramatically reduced.

4.2 Tests and Results

The two dimensional optimizer was tested in the configuration of Figure 9 with a PACE 231-R analog computer serving as the performance function computer. In all tests, the optimizer was operated in a "repetitive" mode.

The first test function used, $P = x_1^2 + 2x_2^2$, is a simple quadratic function with a minimum value of zero at the origin, and having symmetrical elliptical contours, as shown in Figure 40 below.

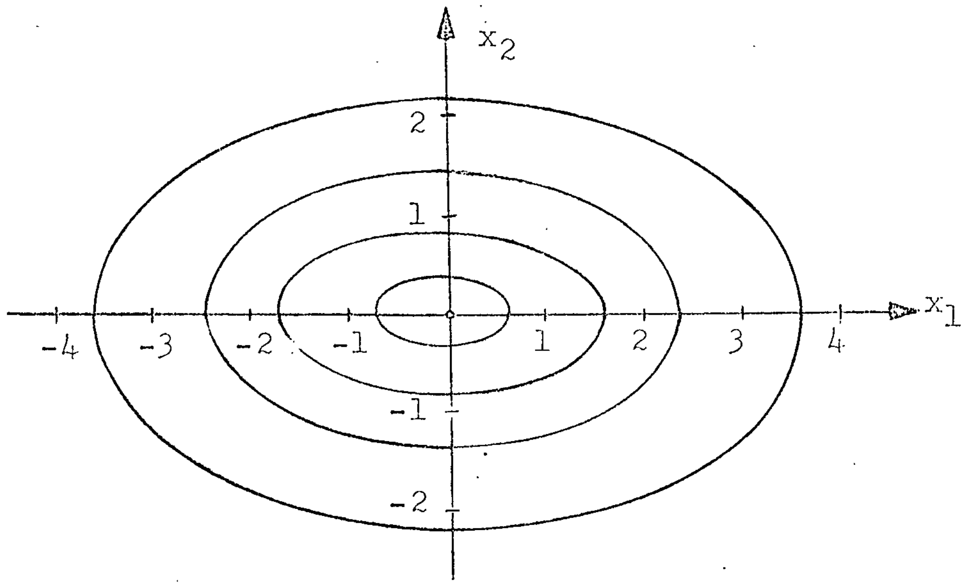
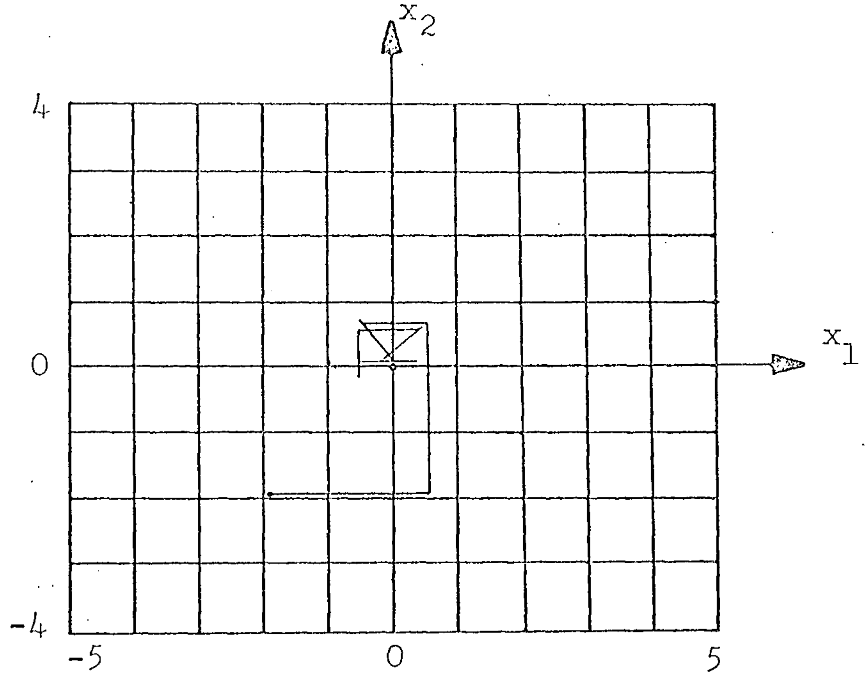


Figure 40. Contours of First Test Function

Searches were started at a variety of points distributed in all four quadrants, and, in each case, comparable results were obtained. The two photographs shown in Figure 41, representative of a typical test run, clearly indicate that quadratic convergence is in fact approximated by the optimizer, since the performance function is minimized in one iteration. Any deviation from the path which the P.M.A. steps should theoretically follow can be attributed to the limited sensitivity of the optimum detector to changes in the performance function. In all tests made with this function, as these particular photographs reveal, those iterations following the initial one were confined to a small region about the optimum within which the function deviations did not exceed about two per cent of the starting values.



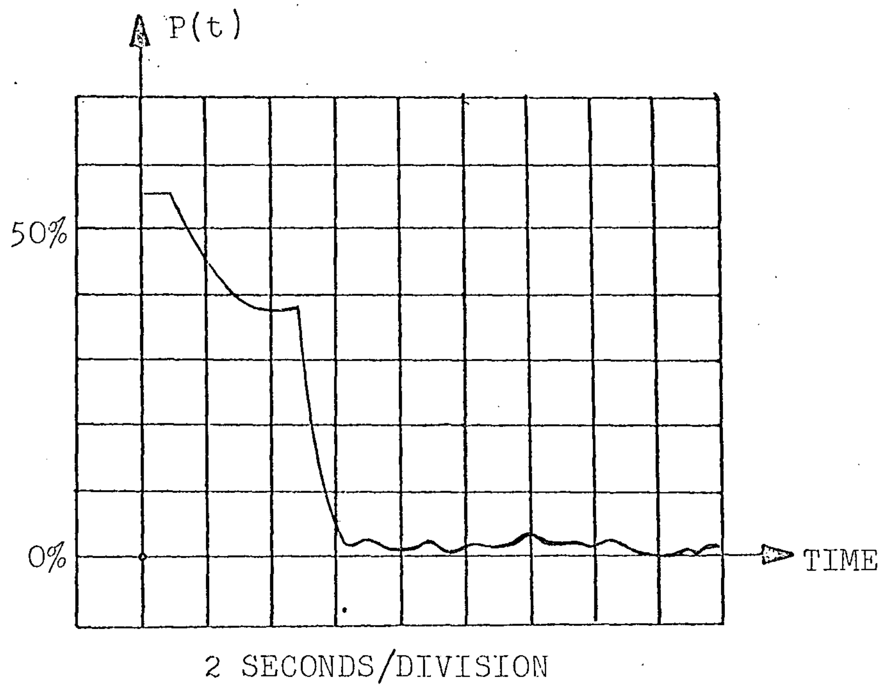


Figure 41. Results of First Test

A second test function was obtained from a simple optimal control problem whose solution can be obtained by the minimization of the function

$$P = 1 + \frac{x_1^2 + x_1 x_2}{2} + \frac{x_2^2}{6}$$

subject to the constraint that

$$g = 9x_1 + 4x_2 + 14 = 0 .$$

The constrained minimum is at the point $(x_1, x_2) = (-2, 1)$ (as can readily be verified using a Lagrange multiplier) at which point the minimum value of P is

$$P_{min} = \frac{13}{6} = 2.17$$

The problem in this form was not suitable for direct solution by the automatic optimizer, which cannot handle constraints, and so an equivalent unconstrained problem was obtained by applying the constraint g as a penalty function to P. The augmented performance function

$$P_A = P + kg^2$$

which also has a minimum of 2.17 at $(x_1, x_2) = (-2, 1)$, was then presented to the automatic optimizer for solution.

With a value of k = 1, the function P_A is

$$P_A = 1 + \frac{x_1^2 + x_1 x_2}{2} + \frac{x_2^2}{6} + \left(9x_1 + 4x_2 + 14\right)^2$$

This is also a quadratic function, since only second powers and linear terms in the independent variables are present; however, the contours of P_A, sketched in Figure 42, represent a long, narrow, sloping valley, the type of response surface found particularly troublesome by many of the direct search and steepest descent techniques mentioned in section (1.3).

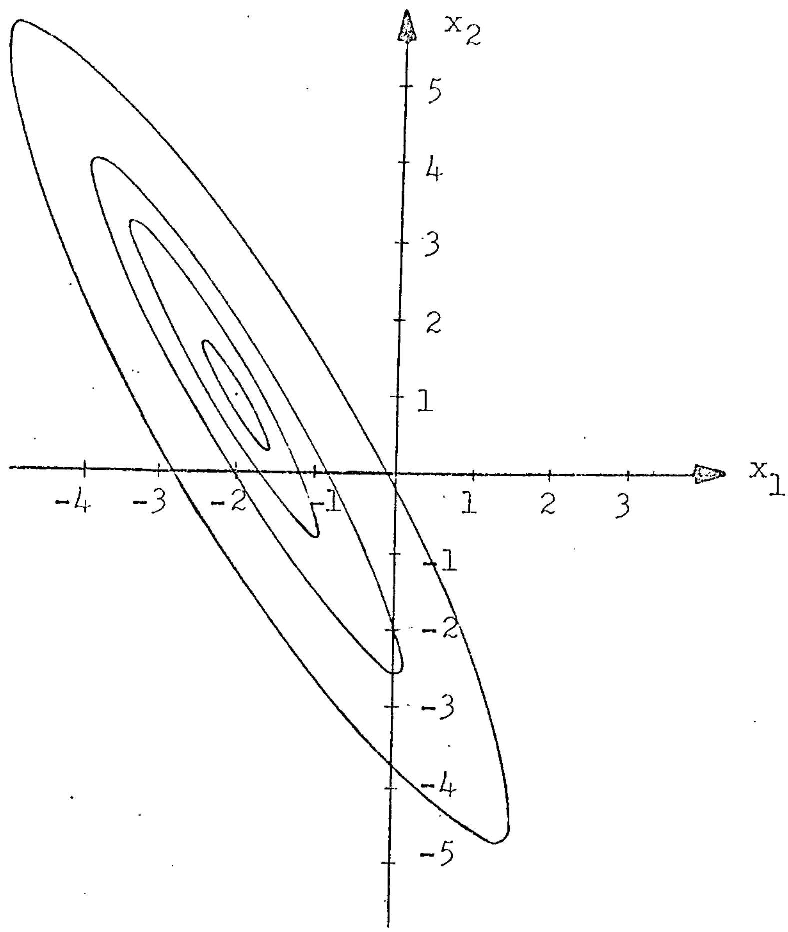


Figure 42. Contours of Second Test Function

Photographs of the optimizer's performance with this function are given in Figure 43 for starting points in the first quadrant and the fourth quadrant. In both cases, the bottom of the valley was reached after only one iteration and thereafter the search iterated about the minimum. No difficulty following the valley was encountered. The convergence time was roughly five seconds.

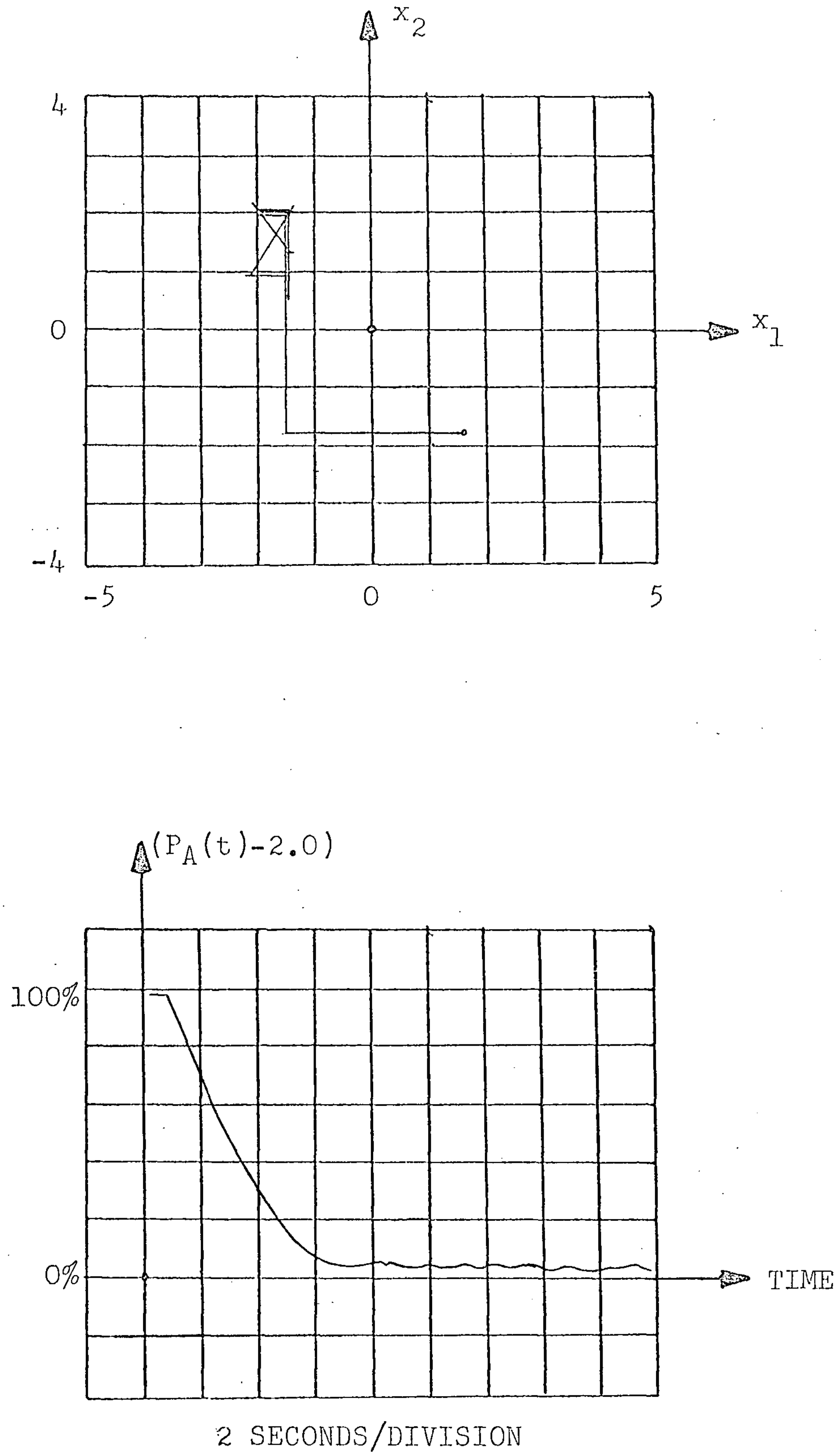


Figure 43. Results of Second Test

The final performance function chosen to test the optimizer was the function

$$P = 100(x_2 - x_1{}^2)^2 + (1 - x_1)^2$$

proposed by Rosenbrock[13] in 1960 and used by Powell in 1962 to test his algorithm. It represents a highly asymmetric curved valley, as shown in Figure 44, and is generally regarded as a "difficult" function to optimize. Its minimum value is zero at the point (1, 1).

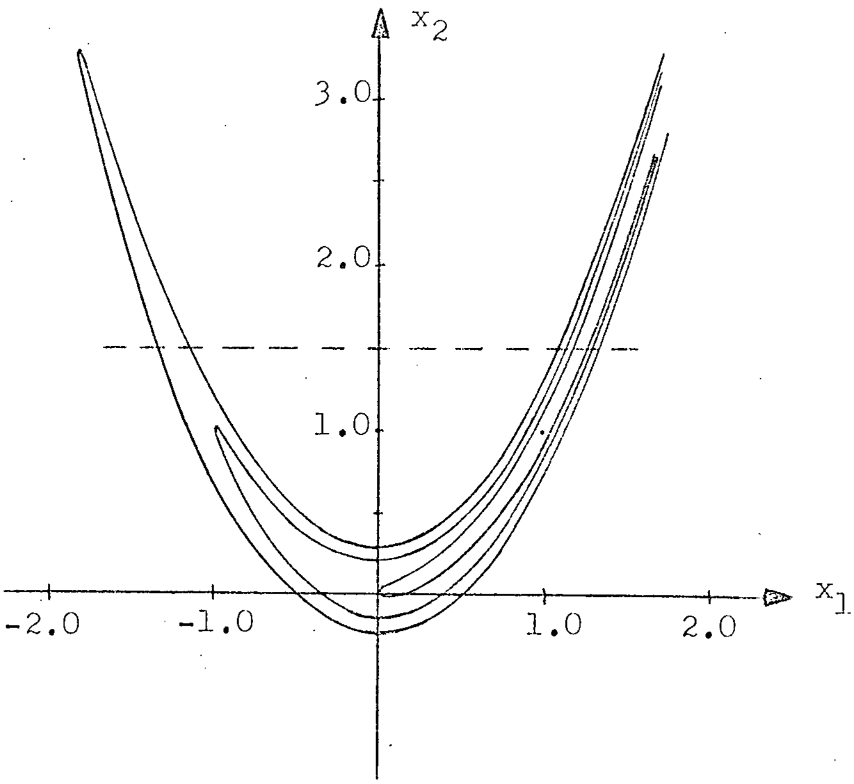


Figure 44. Contours of Third Test Function

The first time the optimizer was tested with this function, the search followed the bottom of the valley right up to the upper right corner of the first quadrant, since the optimum detector could not detect the slight increase in P. To alleviate this difficulty, a simple penalty function of the form

$$J = x_2{}^2 \ ; \text{ if } x_2 \geqq k$$

$$J = 0 \quad ; \text{ if } x_2 < k$$

was instrumented on the analog computer. The modified function optimized was then

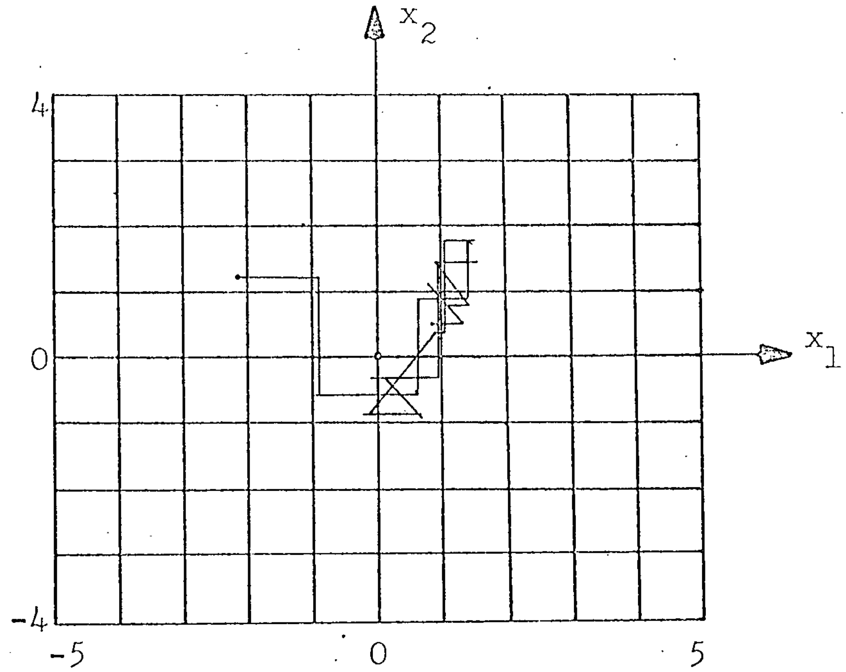
$$P_A = P + J$$

The effect of this penalty was to constrain the search to the area below the horizontal line $x = k$; the value of k was chosen to be 1.5, as shown in Figure 44. With this constraint, the optimizer produced the results shown by the photographs of Figure 45, successfully locating the optimum after about four or five iterations. The time required to minimize P was not much more than that taken with the previous two functions.

The excellent results obtained for the special purpose optimizer from these tests in two dimensions demonstrate that it is suitable for the type of optimization problems likely to be encountered in optimal controller applications.

No difficulties are anticipated in the expansion of the machine from two to the proposed five dimensions, since the algorithm itself has been tested up to five dimensions and found satisfactory, and the additional circuitry required for the five dimensional implementation simply duplicates that used for the two dimensional machine.

Since, as mentioned in section (1.2), a wide range of process time constants may be encountered, the optimization speed achieved in the test results, (roughly one second per linear search), may or may not be adequate for any particular controller application. However, if necessary, considerable speed improvement could be made by several methods without significantly altering the basic structure of the optimizer.



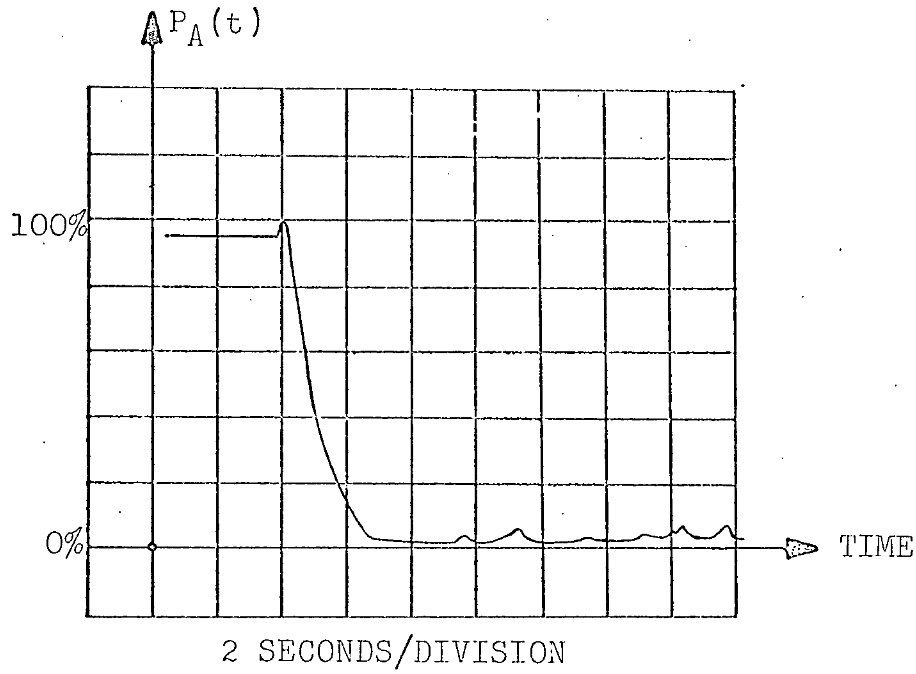


Figure 45. Results of Third Test

For instance, faster stepping motors could be employed, or, at some sacrifice to search resolution, smaller gear ratios could be used between the motors and potentiometers. Alternatively, the motors and potentiometers could be replaced entirely by some form of purely electronic D/A conversion. However, it must be remembered that in a practical optimal controller such as is illustrated in Figure 2, the maximum search speed is ultimately limited by the solution speed achievable for the process model equations.

4.3 Summary and Conclusions

A special purpose automatic optimizer has been designed for use in practical optimal process controllers. The design was based on the implementation of a modified version of a quadratically convergent search algorithm developed in 1962 by M.J.D. Powell, and is suitable for application to the local, unconstrained minimization of analytically unknown but otherwise accessible performance functions of up to five independent variables.

The optimizer was constructed for two of the five dimensions and was tested on several known functions of varying difficulty. The results obtained clearly demonstrated that the special purpose machine was both powerful and fast. In addition, it was found to be sufficiently inexpensive to justify its use in hierarchical control structures at a local, sub-optimal level.

Full benefit of the economic advantages of the automatic optimizer over a general purpose digital computer can only be realized if it is used in conjunction with other inexpensive special purpose machines for the synthesis of complete sub-optimal controllers, and it is both hoped and expected that such machines will soon be forthcoming.

REFERENCES

1. Masak, M., "Decomposition and Optimal Control Theory", Ph. D. Thesis, Department of Electrical Engineering, Faculty of Applied Science, University of British Columbia, 1968.

2. Brooks, S.H., "A Discussion of Random Methods for Seeking Maxima", Operations Research, Vol. 6, pp. 244-251, 1958.

3. Box, G.E.P., and J.S. Hunter, "Multi-Factor Experimental Designs", Annals of Mathematical Statistics, Vol. 28, pp. 195-241, 1957.

4. Wilde, D.J., Optimum Seeking Methods, Prentice-Hall, 1964.

5. Hooke, R., and T.A. Jeeves, "Direct Search Solution of Numerical and Statistical Problems", Journal of the Association of Computing Machinery, pp. 212-229, 1961.

6. Friedman, M., and L.J. Savage, "Planning Experiments Seeking Maxima", Techniques of Statistical Analysis, McGraw-Hill, New York, 1947.

7. Booth, A.D., "An Application of the Method of Steepest Descent to the Solution of Systems of Non-Linear Simultaneous Equations", Quarterly Journal of Mechanics and Applied Mathematics, Vol. 2, pp. 460-468, 1949.

8. Curry, H.B., "The Method of Steepest Descent for Non-Linear Minimization Problems", Quarterly Journal of Applied Mathematics, Vol. 2, pp. 258-260, 1944.

9. Box, G.E.P., and K.B. Wilson, "The Experimental Attainment of Optimum Conditions", Journal of the Royal Statistics Society, Series B, Vol. 13, pp. 1-37, 1951.

10. Leitman, G.(ed.), Optimization Techniques, Academic Press, New York, 1962.

11. Crocket, J.B., and H. Chernoff, "Gradient Methods of Maximization, Pacific Journal of Mathematics, Vol. 5, pp. 33-50, 1955.

12. Nelder, J.A., and R. Mead, "A Simplex Method for Function Minimization", Computer Journal, Vol. 7, p. 308, 1965.

13. Rosenbrock, H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer Journal, Vol. 3, pp. 175-184, 1960.

14. Fletcher, R., and C.M. Reeves, "Function Minimization by Conjugate Gradients, Computer Journal, Vol. 7, pp. 149-154, 1964.

15. Fletcher, R., and M.J.D. Powell, "A Rapidly Convergent Descent Method for Minimization", Computer Journal, Vol. 6, pp. 163-168, 1963.

16. Shaw, B.V., R.J.Buehler, and O. Kempthorne, "The Method of Parallel Tangents (Partan) for Finding an Optimum", Office of Naval Research Report, NR-042-207, (No. 2), 1961.

17. Powell, M.J.D., "An Iterative Method for Finding Stationary Values of Functions of Several Variables", Computer Journal, Vol. 5, pp. 147-151, 1962.

18. Finkel, R.W., "The Method of Resultant Descents for the Minimization of an Arbitrary Function", Paper 71, Preprints of Papers Presented at the 14th National Meeting of the Association of Computing Machinery, 1959.

19. Voelker, W.H., "Transistor Circuit Converts Voltage to Regulated Frequency", Electronics, p. 73, November 16, 1964.

20. Donaldson, R.W., and D. Chan, "Calculation of Steady State Errors in Digital-to-Analog Ladder Converters", Electronics Letters, Vol. 3, No. 5, May, 1967.

P.M.A. Search (Fortran) Program

The main subroutine of the program follows.

```
$IBFTC PMA
      DIMENSION FMU(5,5),DELTA(5,5),X(5),DIRTN(5),K(5)
      N=5
      L=n
      FK=α
C     STATEMENT(S) TO INITIALIZE X(I)
2     CALL SRCH (N,X,DIRTN,P,PNEW,FK,FMU,DELTA,L,K)
      WRITE(6,5)P
5     FORMAT(F15.8)
      IF(P.GT. β )GO TO 2
      RETURN
      END
```

In the above subroutine, α , β , and n must be set according to the problem. n is the number of dimensions involved, and α and β are parameters defining the step size and the stopping criterion, respectively.

The performance function evaluation subroutine is given below.

```
$IBFTC POLY
      SUBROUTINE POLY(P,X,N)
      DIMENSION X(N)
C     P=DESIRED FUNCTION OF X(I)---MUST BE INSERTED
      RETURN
      END
```

The above subroutine is called before and after each linear search step by the SRCH1 subroutine which follows.

```
$IBFTC SRCH1
      SUBROUTINE SRCH1(N,X,DIRTN,P,PNEW,FK,FMU,DELTA,L,K)
      DIMENSION FMU(N,N),DELTA(N,N),X(N),DIRTN(N),K(N)
      CALL POLY (P,X,N)
      DO 1 I=1,N
      X(I)=X(I)+FK*DIRTN(I)
      CALL POLY(PNEW,X,N)
      IF (PNEW.GT.P)FK=-FK
      DO 2 J=1,5000
      P=PNEW
      DO 3 I=1,N
3     X(I)=X(I)+FK*DIRTN(I)
      CALL POLY(PNEW,X,N)
      IF(PNEW.GT.P)GO TO 20
2     CONTINUE
20    RETURN
      END
```

The rest of the program consists of identical subroutines SRCH2, SRCH3, SRCH4 and SRCH5 which control the sequencing of linear searches to execute P.M.A. A representative subroutine is given below where the parameter γ must be 2, 3, 4, or 5.

```
$IBFTC SRCHγ
      SUBROUTINE SRCHγ (N,X,DIRTN,P,PNEW,FK,FMU,DELTA,L,K))
      DIMENSION FMU(N,N),DELTA(N,N),DIRTN(N),K(N)
      K(L)=1
3     DO 2 I=1,N
2     DIRTN(I)=0.0
      DIRTN(1)=1.0
      LM1=L-1
      CALL SRCH(γ-1)(N,X,DIRTN,P,PNEW,FK,FMU,DELTA,LM1,K)
7     IF(K(1).EQ.1)GO TO 10
      IF(K(1).EQ.2)GO TO 20
      IF(K(1).EQ.3)GO TO 30
      IF(K(1).EQ.4)GO TO 40
10    DO 1 I=1,N
      FMU(L,I)=X(I)
1     DIRTN(I)=0.0
      DIRTN(L)=1.0
      K(L)=K(L)+1
      CALL SRCH1(N,X,DIRTN,P,PNEW,FK,FMU,DELTA,1,K)
      GO TO 7
20    K(L)=K(L)+1
      GO TO 3
30    DO 4 I=1,N
      DELTA(L,I)=X(I)
4     DIRTN(I)=DELTA(L,I)-FMU(L,I)
      K(L)=K(L)+1
      GO TO 6
40    RETURN
      END
```

The P.M.A. program was executed on an IBM 7040 digital computer for the functions tested originally by Powell with his 1962 algorithm. The first function used was

$$P = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

which has a minimum of zero at (1, 1) A starting point of (-1.2, 1.0) was chosen as was used by Powell. The second function chosen was

$$P = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + (x_1 - x_4)^4$$

which has a minimum of zero at the origin. The starting point was (3, -1, 0, 1).

A comparison of the computer results with those obtained by Powell is presented in the following chart. The P.M.A. method clearly is equally as powerful as the original method.

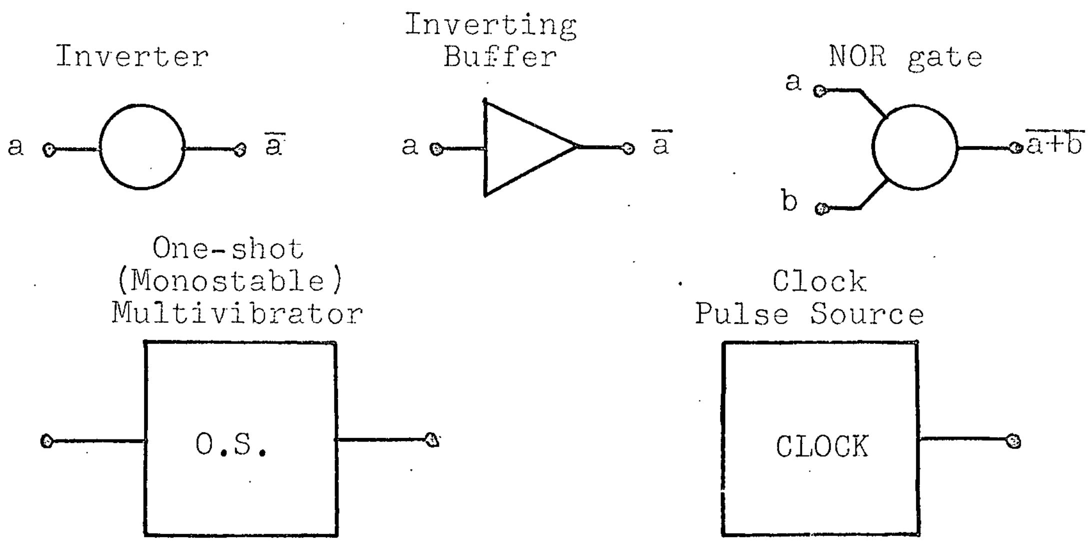
	Function One		Function Two	
Iteration	Powell	P.M.A.	Powell	P.M.A.
0	24.200	24.200	215.00	215.00
1	3.643	1.645	0.009	0.056
2	2.898	0.988	9×10^{-5}	1.3×10^{-4}
3	2.195	0.532	2×10^{-6}	9.5×10^{-5}
4	1.412	0.402		
5	0.831	0.354		
6	0.432	0.343		
7	0.182	0.336		

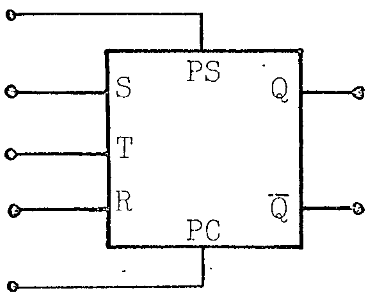
APPENDIX II

Circuit Components Used

Motorola series MC700P digital integrated circuits were used for all logic circuits. Logic levels of +3.6 volts (logical one) and zero volts (logical zero) apply; loading rules, pin arrangements, circuit details, and operating specifications are supplied for all units in the MC700P series technical data sheet.

The logic symbols used in the thesis are defined below.



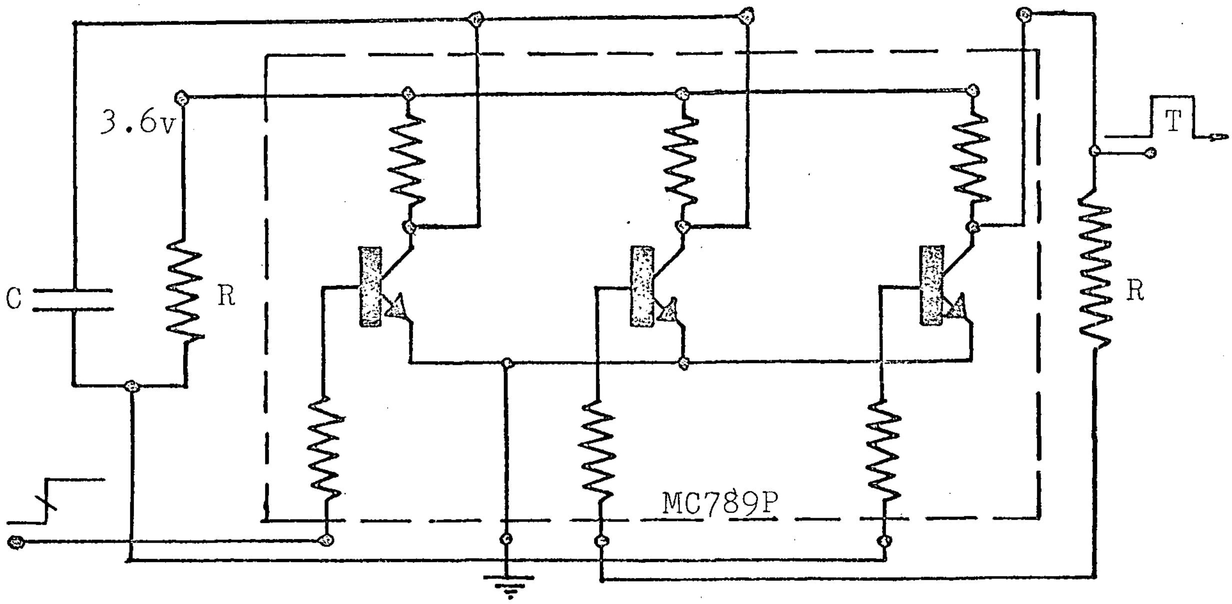


J-K flip-flop

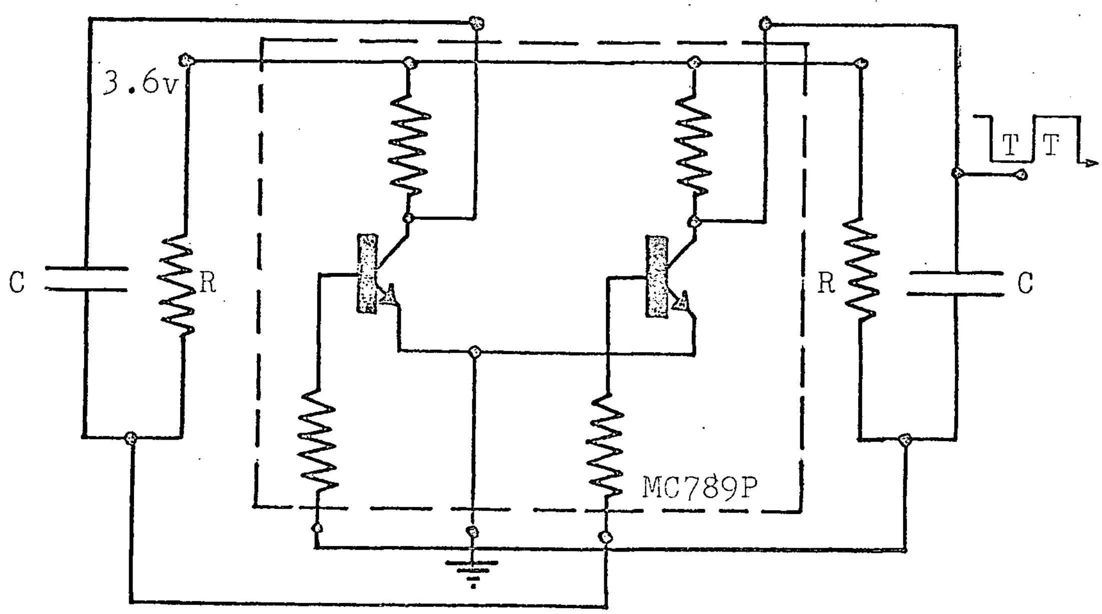
t_n		t_{n+1}	
S	R	Q_{n+1}	$\overline{Q}_{n+1}$
0	0	Q_n	$\overline{Q}_n$
0	1	1	0
1	0	1	0
1	1	$\overline{Q}_n$	Q_n

The chart shows the flip-flop output states in response to a clock pulse on the toggle (T) input at time, t_n. Response is to the falling edge of the clock pulse. Direct preset (PS) and preclear (PC) inputs over-ride the toggle input.

Of the elements shown, all but the clock and one-shot are directly available in the MC700P series, and these two were constructed using the MC789P hex inverter circuit and a few external discrete components as shown below.



$T \cong 0.69\ RC$



The one-shot durations and clock frequencies used in the system constructed are listed below.

1. Optimum Detector

   Clock

   $$f_c = 60 \text{ kilohertz}$$

   One-shots

   $$T_B = 5 \text{ microseconds}$$
   $$T_{UP} = 2 \text{ microseconds}$$
   $$T_{DN} = 2 \text{ microseconds}$$

2. Search Controller

   Mode Control Logic

   $$T_M = 500 \text{ microseconds}$$
   $$T_{DR} = 10 \text{ microseconds}$$

   Transfer Control Logic

   Clock

   $$f_{SH} = 10 \text{ kilohertz}$$

   One-shots

   $$T_{DY} = T_T = T_{SH} = 100 \text{ microseconds}$$

   X-Counters

   $$T_Z = 200 \text{ microseconds}$$
   $$T_B = 100 \text{ microseconds}$$

   Sequence Unit

   $$T_A = 100 \text{ microseconds}$$
   $$T_R = 100 \text{ microseconds}$$

APPENDIX III

System Interconnections

Below are listed all of the interconnections required between the optimizer circuit blocks described in the text of the thesis. The ring counter states are designated by the Q_{ij} notation defined on page 61; other symbols used correspond to those used in the referenced figures.

1. Adjuster Unit (Refer to Figure 24)

Step Enable Inputs

$$SE1 = Q_{12} + Q_{32} + Q_{52} + Q_{53} + Q_{54} + Q_{55}$$
$$SE2 = Q_{22} + Q_{52} + Q_{53} + Q_{54} + Q_{55}$$
$$SE3 = Q_{23} + Q_{53} + Q_{54} + Q_{55}$$
$$SE4 = Q_{24} + Q_{54} + Q_{55}$$
$$SE5 = Q_{25} + Q_{55}$$

Step Inhibit Inputs

$$SI1 = SI2 = SI3 = SI4 = SI5 = Q_{DR}$$

Sign Reverse Inputs (Refer to Figures 24, 32, and 37)

$$SRi = T_{SR}\ (SEi) + (TEGi)\ (T_{SH})\ (S/SMi) \qquad \text{for } i = 1, 2, 3, 4$$

$$SR5 = T_{SR}\ (SE5)$$

where S/SM is the Sign/SM signal shown in Figure 27 and TEGi are the enable signals to the transfer gates listed on page 87.

2. Search Controller (Refer to Figure 30)

Δ X-Counters

Counter Enable Inputs (Refer to Figure 30)

$$CE12 = Q_{32},\ CE22 = Q_{22}$$
$$CE13 = Q_{33}\ (Q_{12} + Q_{32} + Q_{52}),\ CE23 = Q_{33}\ (Q_{22} + Q_{52})$$

$CE33 = Q_{33}$, $CE14 = Q_{34}(Q_{22} + Q_{32} + Q_{52} + Q_{53})$

$CE24 = Q_{34}\ (Q_{22} + Q_{52} + Q_{53})$, $CE34 = Q_{34}\ (Q_{23} + Q_{53})$

$CE44 = Q_{24}$, $CE15 = Q_{35}\ (Q_{12} + Q_{32} + Q_{52} + Q_{53} + Q_{54})$

$CE25 = Q_{35}\ (Q_{22} + Q_{52} + Q_{53} + Q_{54})$

$CE35 = Q_{35}\ (Q_{23} + Q_{53} + Q_{54})$, $CE45 = Q_{35}\ (Q_{24} + Q_{54})$

$CE55 = Q_{25}$

Counter Initialize Inputs (Refer to Figure 30)

$CI13 = CI23 = Q_{23}$

$CI14 = CI24 = CI34 = Q_{24}$

$CI15 = CI25 = CI35 = CI45 = Q_{25}$

Counter Reset Inputs

$CR12 = CR22 = Q_{52} + Q_{RESET}$

$CR13 = CR23 = CR33 = Q_{53} + Q_{RESET}$

$CR14 = CR24 = CR34 = CR44 = Q_{54} + Q_{RESET}$

$CR15 = CR25 = CR35 = CR45 = CR55 = Q_{55} + Q_{RESET}$

where

$$Q_{RESET} = Q_{02}\ Q_{03}\ Q_{04}\ Q_{05}$$

Data Transfers

Transfer Start Signal (Refer to Figure 33)

$TST = Q_{42} + Q_{43} + Q_{44} + Q_{45}$

Transfer Enable Signals (Refer to Figure 31)

Δ X-Counters

$TEC12 = TEC22 = Q_{42}$

$TEC13 = TEC23 = TEC33 = Q_{43}$

$TEC14 = TEC24 = TEC34 = TEC44 = Q_{44}$

$TEC15 = TEC25 = TEC35 = TEC45 = TEC55 = Q_{45}$

Transfer Gates

$$TEG1 = TEG2 = TST$$

$$TEG3 = Q_{43} + Q_{44} + Q_{45}$$

$$TEG4 = Q_{44} + Q_{45}$$

$$TEG5 = Q_{45}$$

Shift Enable Signals (Refer to Figures 31 and 33)

$$SHE\ i = (\overline{Q}_{SH})\ (TEG\ i);\ \text{for } i = 1, 2, 3, 4, 5$$

Shift Complete Signals (Refer to Figures 31 and 33)

$$SC = \overline{Q}_{SH}\ (A + B + C + D)$$

where

$$A = \overline{Q}_{R5}\ (TEG5),\ B = \overline{Q}_{R4}\ (TEG4)$$

$$C = \overline{Q}_{R3}\ (TEG3),\ D = (\overline{Q}_{R2} + \overline{Q}_{R1})\ (TST)$$

Rate Registers

Preclear Inputs (Refer to Figures 31 and 37)

$$PC1 = Q_{12} + Q_{32} + S$$

$$PC2 = Q_{22} + S$$

$$PC3 = Q_{23} + S$$

$$PC4 = Q_{24} + S$$

$$PC5 = Q_{25} + S$$

where

$$S = T_M\ Q_M\ (Q_{32} + Q_{52} + Q_{53} + Q_{54})$$

The logic function S preclears all the rate registers just prior to each transfer, as stipulated on page 52.

Sequence Unit

Ring Counter Advance Pulses (Refer to Figures 33 and 36)

$$TA2 = T_{UPM}\ (Q_{12} + Q_{22} + Q_{32} + Q_{52} + Q_{23} + Q_{24} + Q_{25}) + T_{TC}\ \overline{Q}_M\ Q_{42} + START$$

$$TA3 = T_{UPM}\,(Q_{52} + Q_{23} + Q_{53} + Q_{24} + Q_{25}) + T_{TC}\,\overline{Q}_M\,Q_{43} + START$$

$$TA4 = T_{UPM}\,(Q_{53} + Q_{24} + Q_{54} + Q_{25}) + T_{TC}\,\overline{Q}_M\,Q_{44} + START$$

$$TA5 = T_{UPM}\,(Q_{54} + Q_{25} + Q_{55}) + T_{TC}\,\overline{Q}_M\,Q_{45} + START$$

where

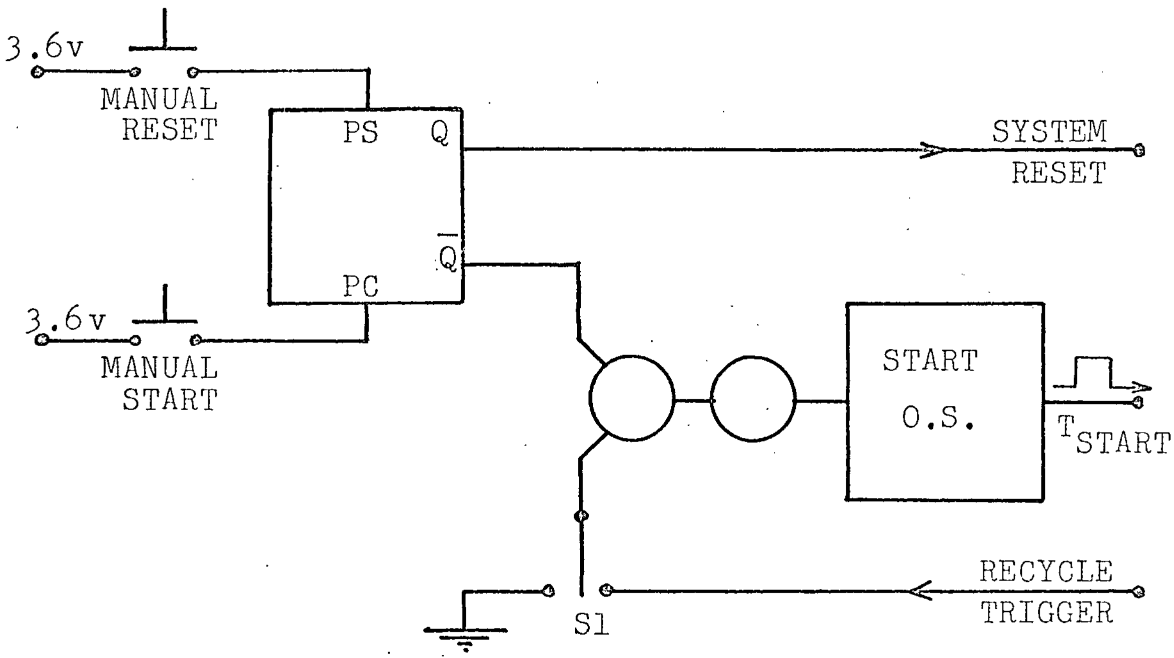
$$START = Q_{RESET}\,\overline{Q}_M\,T_{START}$$

and T_{START} is a one-shot pulse originating in the circuit shown below.

Ring Counter Reset Inputs (Refer to Figure 36)

$$RCR2 = RCR3 = RCR4 = RCR5 = SYSTEM\ RESET$$

and SYSTEM RESET also originates in the circuit below.



The above logic is used to manually reset and start the optimizer (in that sequence). If switch S1 is grounded, the optimizer stops after one iteration and must be reset and started again manually. If switch S1 is connected to the recycle trigger, the system operates repetitively. The system reset signal must be connected to the points shown in Figure 37 as well as to the ring counter reset terminals. The recycle trigger signal can be conveniently obtained as

$$RCT = T_{UPM}\,(Q_{55})$$

This being the case, T_{START} should be at least twice as long as T_M (Figure 37).