APPLICATION OF A VECTOR COST FUNCTION TO ADAPTIVE CONTROL WITHOUT PLANT IDENTIFICATION

APPLICATION OF A VECTOR COST FUNCTION TO ADAPTIVE CONTROL WITHOUT PLANT IDENTIFICATION

by

Alain J-C Rey, B.Sc. (Electrical Engineering)

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Engineering

McMaster University
March 1969

MASTER OF ENGINEERING (Electrical Engineering)

McMASTER UNIVERSITY Hamilton, Ontario.

Title: Application of a Vector Cost Function to Adaptive Control Systems Without Plant Identification.

Author: Alain Jean-Claude Rey, B.Sc. (University of Manchester).

Supervisor: N. K. Sinha, B.Sc. Eng. (Banaras), Ph.D. (University of Manchester).

Number of Pages: v; 131.

Scope and Contents: The proposed Vector Cost Function technique for adaptive control of an unknown plant is investigated. The mathematical background of this algorithm is presented, and then the technique is applied to two second order plants, an over-damped one and an under-damped one, whose step responses are required. Also, the vector cost function parameters which affect the adapted responses of the plants are investigated, and finally the conclusions arrived at are presented.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr.Sinha, for his guidance and encouragement in the preparation of this thesis, and also my parents for their encouragement and understanding.

The scholarship provided by the National Research Council of Canada to carry out this work is gratefully acknowledged.

TABLE OF CONTENTS

Page No.

CHAPTER 1.

Introduction.

An adaptive control system is basically a feedback control system that achieves a desired response in the presence of extreme changes in the controlled system's parameters, and major external disturbances by adjusting the parameters of a controller. Adaptive control systems are usually characterized by devices which automatically measure the dynamics of the controlled system and other devices (usually called controllers) which are adjusted on the basis of a comparison between these measurements and some optimum figure of merit [1].

Conventional control systems do compensate for small variations in the parameters of the controlled system and other disturbances by using negative feedback. However, the reason why the control engineer resorts to an adaptive control system is that a conventional feedback control system is not capable of high performance in the presence of large changes in the controlled system's parameters and large external disturbances. For example, a large change in the feedback element of a conventional feedback control system will easily degrade the performance of the system;

on the other hand, an adaptive control system will be able to cope with such a change and hence keep the system's performance at an optimum level. Further, a conventional feedback control system designed to respond in an optimum manner to simple inputs may not respond to these signals if they are contaminated with noise or other disturbances. On the other hand, an adaptive control system can be designed to overcome these limitations of conventional feedback control systems.

In this thesis an adaptive control system will be taken to mean one which is a self-optimizing control system. This will often require that past data be stored and used for a learning and adapting scheme.

Two ideas form the basis of modern adaptive control techniques. One is to identify the system to be controlled and the changes that have taken place in it before calculating the new optimal values of the controller parameters. This method is usually called adaptive control "with identification". The other idea is based on the fact that complete knowledge of a plant is not necessary in order to control it. It is therefore referred to as adaptive control "without identification". For example, a human being does not have to know the laws of static and dynamic equilibrium before he learns to stand, walk or run. Here, the person learns the correct (optimum) way of distributing weight on his two legs and the angle at which his body should lie, depending on whether he is standing, walking or running.

Adaptive control without identification attempts very much to imitate man and his process of learning and adapting. Storing data from past experience is another way in which adaptive control systems can improve their performance faster - the analogy again to the human brain that can store a vast amount of information. Computers are usually used to store past data for control systems, and this data is later used in conjunction with adaptive control techniques in order to make the system respond in a more efficient manner and faster as more and more information is acquired about the system. Of course, the human brain is a master computer, and no adaptive control system will ever have such a learning and adapting ability. But computers can be programmed to do a specific job and, because of their speed and reliability, will do them better and faster than a man can. The major promise of the adaptive concept lies in the possibility of introducing a simple learning mechanism within the adaptive part of the system. Once learning is combined with adaptivity, the control system approaches the flexibility and capabilities of human controllers in more significant jobs.

With the advent of the electronic digital computer, engineering approach to control problems has changed considerably. Tasks which took days or weeks to perform can now be done in a few seconds on a computer. Consequently, adaptive control strategies, which previously could not have been justified because of the time factor involved, are now within reach of any control engineer. Very few modern adaptive control techniques can be used efficiently without taking advantage of computer techniques. Reduction in size and weight of present day computers together with increase in speed of operation resulting from the use of integrated circuits and micro-electronics, have extended their application to forming an integral part of aircrafts, space vehicles and other similar systems.

In the present thesis, a digital-computer-controlled adaptive control technique is considered, namely the Vector Cost Function algorithm. This technique has been recently proposed [7] and here its merits and drawbacks are investigated. Chapter Two is a review of some of the modern adaptive control methods (without identification) using digital computers. In Chapter Three the theoretical background of the Vector Cost Function is presented together with some illustrative examples. Chapter Four deals with the implementation of the actual systems used, and Chapter Five gives the results obtained together with partial conclusions. Chapter Six presents the particular and general conclusions drawn and is followed by the appendices.

CHAPTER 2

Adaptive Control Techniques Based on Hill-Climbing

2.1 Introduction:

Economic considerations often motivate the engineer to design and operate a system in the best possible fashion i.e. he tries to optimize the performance and to minimize the overall cost. It is in connection with the performance optimization that adaptive control plays a very important role. Usually, the success of the overall system, rather than assuring that each part functions itself in an optimum fashion, is the primary objective.

The thought that it may be advantageous to the system to alter the control characteristics during the operation to adapt to various changes in different parts of the overall system has resulted in the increasingly extensive use of adaptive control techniques. The control parameters are changed in accordance with other changes in the system to improve the performance.

There are two basic philosophies in adaptive control. One attempts to identify the system and the changes which have taken place in it in order that new values of the control parameters be calculated, and the parameters changed accordingly, so as to cancel out these changes. The other makes no attempt to identify the system (also referred to later as "plant"), but causes it to measure its own performance against a figure of merit and uses this information to reach an optimum. The latter method will be the one dealt with in the remainder of the thesis.

Adaptive control without identification has the advantage that an optimum can be reached though the changes in the system's parameters or the external disturbances are not known, and that sufficient data about the system is not available. Indeed, the use of feedback is often motivated by this very ignorance or lack of data. Also, one of the major challenges in adaptive control is to design systems which perform satisfactorily using the information available, however inadequate it may be [2]. The index of performance, or cost function as it is also known, is determined either by measurement or computation, and the control variables are then operated in such a way or with such settings as to yield an extremum value of this cost function. It must be noted here that the cost function is chosen by the designer, and consequently the final response of the system can be no better than the criterion chosen. Depending on the criterion, different emphasis is placed on various parts of the response, and the most suitable one to choose is only the designer's choice.

Modern control systems are built around computers, and all the adaptive control techniques mentioned in this chapter make use of computers in some way or other. Today's engineering projects tend to be of such magnitude and scale that the use of computers is not only justifiable, it is essential and economical to accomplish the job in the time allowed and with the performance desired. Computers, especially digital computers, have made possible the solution of problems which were previously almost impossible using analytical techniques; linear, non-linear and time varying equations are handled with nearly equal facility since they are treated in much the same way. The use of computing for on-line process control is to provide timely signals to a process to enhance the value of its outputs. The computer is in fact an intimate part of the process; it receives inputs from the process and in turn its outputs may serve as inputs to the process. Indeed, without a computer, changes in adaptive control systems could only be made very infrequently, and probably the use of adaptive control would not be justified then. In fact, one of the major requirements is that the adaptive controller's ability to alter its parameters be relatively fast compared to the time rate of change of the cost function surface.

There are several techniques of adaptive control without identification which extremalize an assigned index of performance, perhaps with constraints, using automatic iterative procedures. A few of the most recently developed ones are given in the sections following.

2.2 Steepest Descent Methods.

What is sought in this method is an extremum value on a hyper-surface in the presence or absence of constraints*. Steepest-descent methods provide a set of algorithms or control laws which usually lead to the desired solution eventually [2]. As in the case of other optimization methods, there is always the difficulty of determining a global, rather than a local, extremum point.

In discrete optimization methods using steepest descent (or ascent depending on the goal to be reached), the slopes in different directions of the hypersurface are assessed and steps of carefully chosen lengths taken in the steepest direction of the hypersurface. Even though the problem space may not be discrete, these methods are of value because of the speed of digital computers. Gradient methods may be used to optimize non-linear systems of equations and non-linear cost functions either alone or in conjunction with statistical methods when knowledge

* A constraint is a limit which exists on one quantity that prevents a better value of another from being obtained.

acquired previously from the system is used to calculate the future steps to be taken. The problem may be formulated as follows: given the cost function and the constraint functions, it is required to find the values of the variables - system or controller variables - which extremalize the cost function within the constraints.

The cost function may be represented in three-dimensional space by a hill having arbitrary contours. The constraints form curved boundaries on the hill and define the region within which the parameters (variables) may be adjusted. Figure (2.2.a) shows an example of a two-dimensional space with constraint boundaries. The parameters x_1 and x_2 can only take on those values which lie inside the unshaded region.

Gradient methods are well suited for the solution of non-linear problems because they can be easily programmed for digital computations. For example, if maximization of the cost function is required, then equation (2.2.1) must be satisfied.

$$Z(\underline{x})_{p+1} > Z(\underline{x})_p \quad \ldots\ldots (2.2.1)$$

where $Z(\underline{x})_p$ is the response for values of the variables after the pth step. In order to achieve this, all methods use the recurrence formula (2.2.2).

$$\underline{x}_{p+1} = \underline{x}_p + \lambda_p \, \underline{d}_p \quad \ldots\ldots (2.2.2)$$

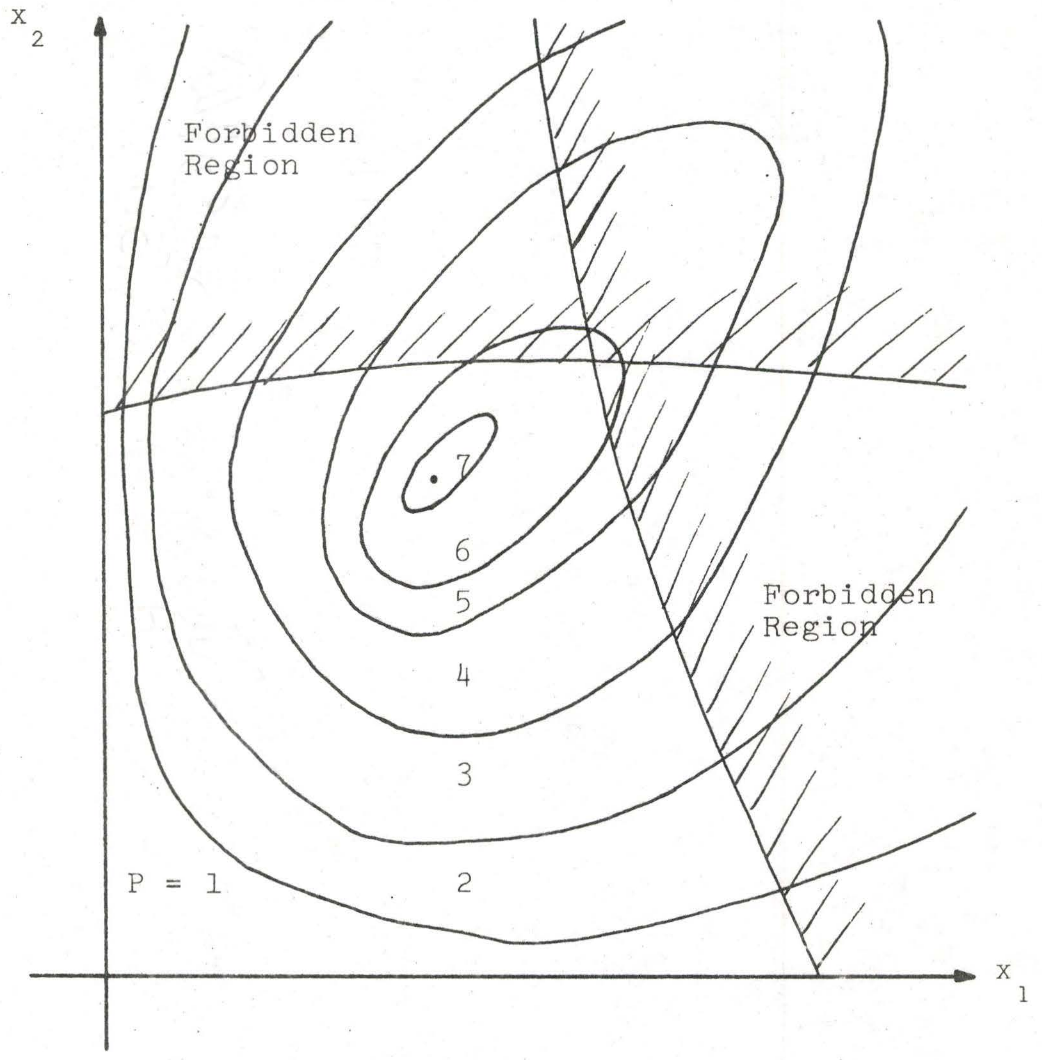


FIGURE (2.2.a)

Two-dimensional Space with Constraint Boundaries.

where λ is the magnitude of the step size, and $\underline{d}$ is the direction vector which establishes the extent in terms of the various x's to which the step size, λ, is apportioned.

Figure (2.2.b) shows a hill climbing procedure for extremalizing the cost function $Z(\underline{x})$.

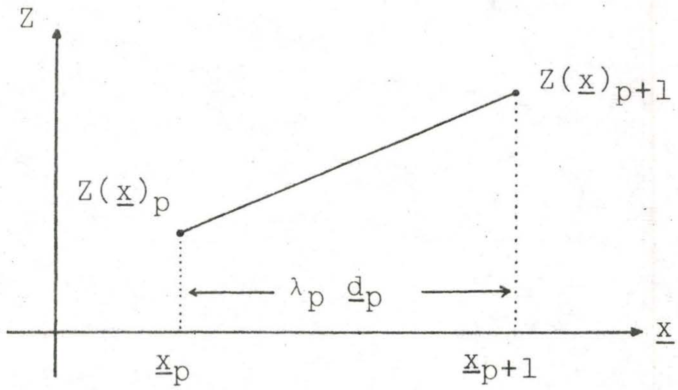


FIGURE (2.2.b)

The difference between the various optimization methods, i.e. gradient methods, depends upon the choice of the step size and the direction vector $\underline{d}$. The latter is given by equation (2.2.3).

$$\underline{d} = \nabla Z = \Sigma \frac{dZ}{dx_i} \left(\Sigma \left(\frac{dZ}{dx_i}\right)^2 \right)^{-1/2} \qquad \text{..........(2.2.3)}$$

$$(i = 1 \text{ ....} n)$$

All these methods require that the starting point be selected in a manner that satisfies all the constraints which may be imposed on the problem of extremalizing the function $Z(\underline{x})$. This implies that the control engineer must be in a position to assign initial values to all the parameters $\underline{x}$ in the problem. This requirement is in fact one of the limitations connected with the use of gradient methods. Depending upon the complexity of the system, it is not always possible to ascertain whether or not the starting point chosen will lead to an extremum value of the cost function. Whenever this is not possible, preliminary investigation must be made to ensure that such a starting point is found.

The method of Steepest Descent determines successive steps of $\underline{x}_{p+1}$ using equation (2.2.4).

$$\underline{x}_{p+1} = \underline{x}_p - \lambda_p \nabla Z_p \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.2.4)$$

The choice of whether the extremum is a minimum (descent) or a maximum (ascent) is merely a question of choice of sign. The step size can be made arbitrary, but equation (2.2.1) will not always be satisfied. Usually λ_p is made variable and after a new point is computed, i.e. $\underline{x}_{p+1}$, the validity of equation (2.2.1) is checked. If this equation is not satisfied, it is necessary to reduce λ_p and recompute $\underline{x}_{p+1}$ until the equation is satisfied. Figure (2.2.c) shows a flow chart of a possible steepest gradient algorithm to ensure that an extremum of the cost function is found. Each step calls for the

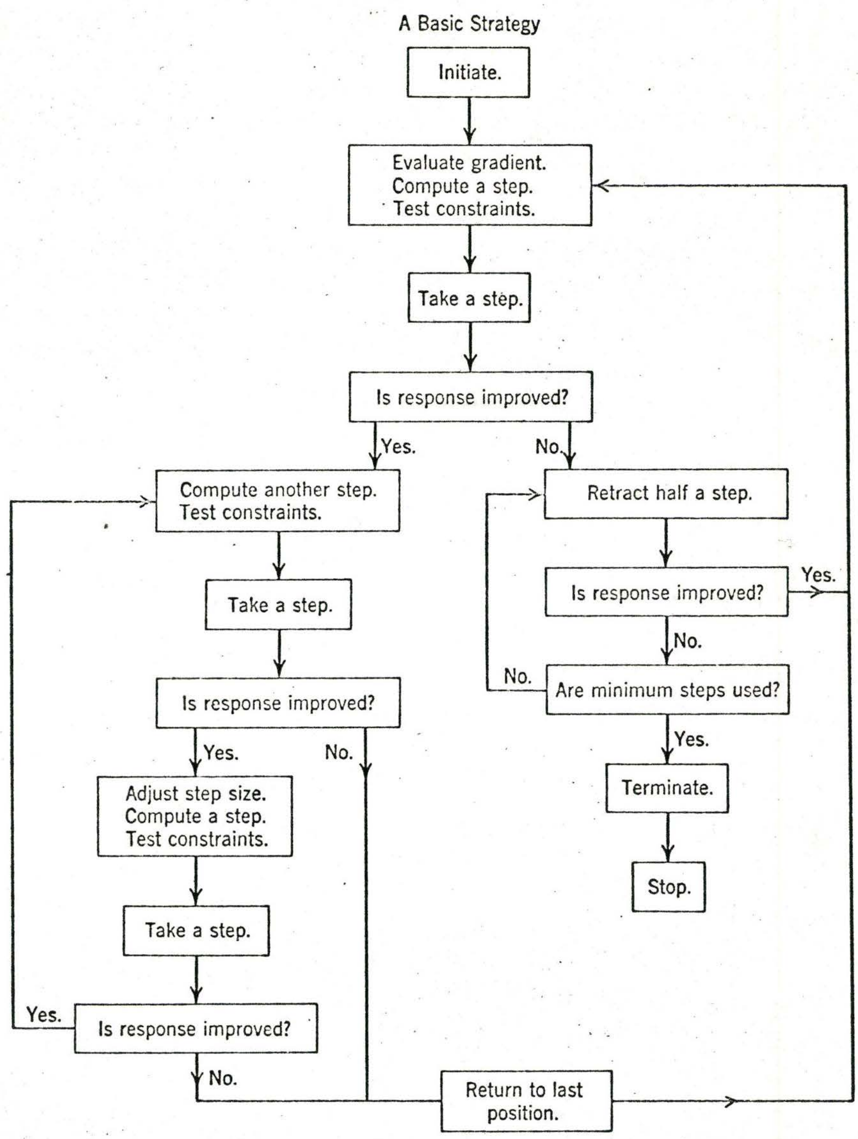


FIGURE (2.2.c)

Flow Chart of a Possible Steepest-Descent Algorithm. [10]

evaluation of the gradient at a new point and a repetition of the procedure until the extremum point is reached.

2.3 Method of Conjugate Gradients.

Practical implementation of a system using a steepest-descent adaptive controller requires extensive hardware, particularly if several parameters are to be adjusted. This results from the method itself which calls for the evaluation of the gradient at each step. For this reason, modifications have been suggested and one of them is the method of Conjugate Gradients. Equations (2.3.1) and (2.3.2) define the conjugate gradient iterative procedure.

$$\beta_i = \frac{\underline{g}_{i+1}^t \, \underline{g}_{i+1}}{\underline{g}_i^t \, \underline{g}_i} \qquad \text{.................(2.3.1)}$$

$$\underline{d}_{i+1} = -\underline{g}_{i+1} + \beta_i \, \underline{d}_i \qquad \text{..................(2.3.2)}$$

$$\underline{x}_{i+1} = \underline{x}_i + \lambda_i \, \underline{d}_i \qquad \text{.....................(2.3.3)}$$

where,

$\underline{x}_o$ = arbitrary starting point

$\underline{g}_o$ = $\underline{g}(\underline{x}_o)$, represents the gradient at $\underline{x}_o$.

$\underline{d}_o$ = $-\ \underline{g}_o$ is the initial direction.

$\underline{x}_{i+1}$ = position of the extremum of $Z(\underline{x})$ on the line through $\underline{x}_i$ in the direction of $\underline{d}_i$.

$\underline{g}_{i+1}$ = $\underline{g}(\underline{x}_{i+1})$ is the gradient at $\underline{x}_{i+1}$.

β_i = weighting factor for the previous direction.

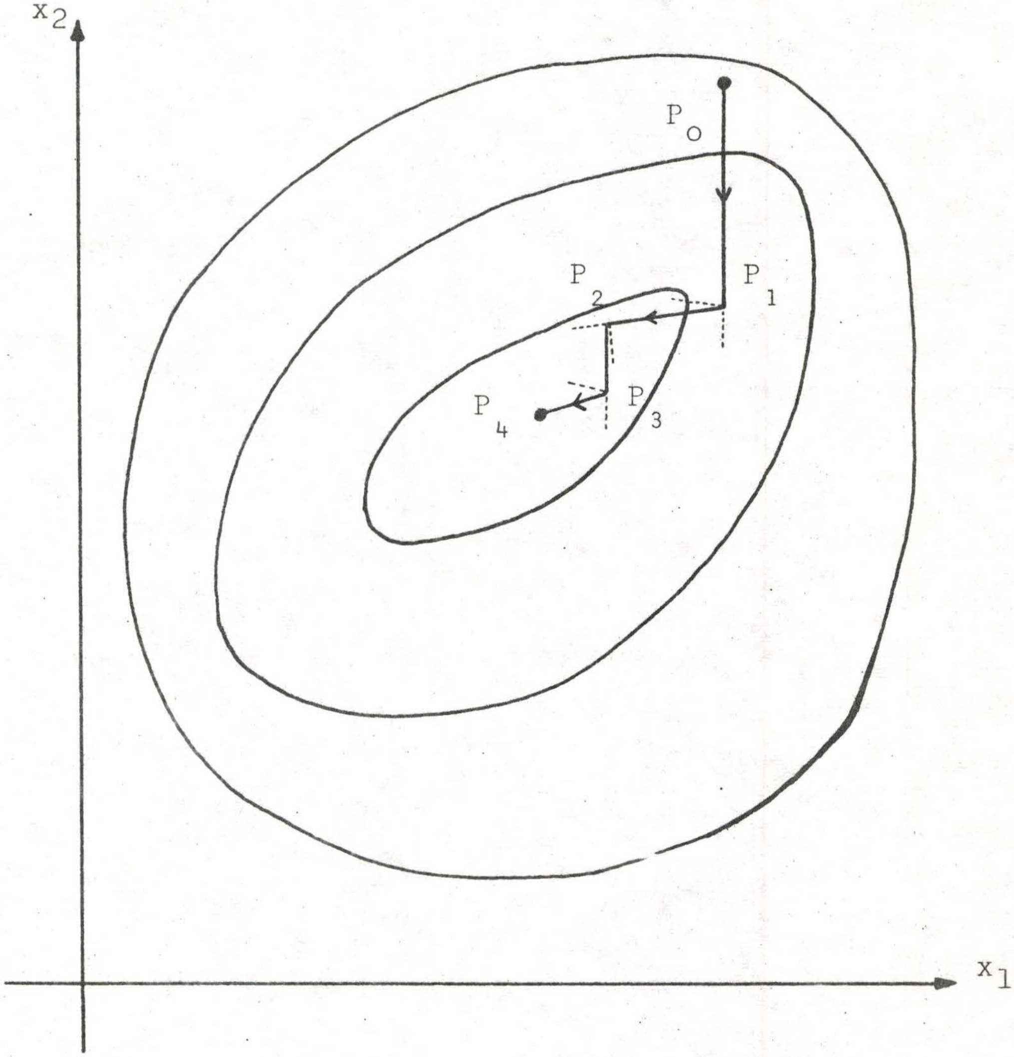


FIGURE (2.3.a)

Conjugate Gradient method for two-parameter Optimization.

In this method, the initial starting direction is determined by the gradient. However, after the position of the extremum on this line has been obtained, the next step is taken in a direction which is a linear combination of the gradient at this point and the previous one. When $\beta_i = 0$, the method of conjugate gradients reduces to that of the steepest-descent method. Figure (2.3.a) shows a two-parameter optimization procedure using the method of conjugate gradients.

2.4 Method of Parallel Tangents.

This method is based on certain global properties of ellipsoids. For cost functions with concentric ellipsoidal contours, the method of parallel tangents will locate the extremum exactly after a fixed, small number of steps. But even when the contours are not elliptical, this technique has desirable features [3].

Suppose that $Z(x_1, x_2)$ is the cost function whose extremal point is required. From any point P_o in the parameter space, proceed in a certain direction until an extremum is found at a point P_2. At P_2, progress in a direction parallel to the tangent at P_o until an extremum point P_3 is again reached. From P_3 proceed along the line P_o P_3 until an extremum is found at P_4. From P_4 move in a direction parallel to the tangent at P_3, and repeat the procedure until the true extremum point of $Z(x_1, x_2)$ is found. Figure (2.4.a) illustrates the above technique.

There are a number of variations of the method of parallel tangents [9], one of the most well known is described below.

Steepest Descent Partan

From any point P_o proceed along a polygonal line, P_o P_2 P_3 P_4 ...., for which P_k is the minimum of Z on the extended line joining it to the preceeding point, P_{k-1}. At even-numbered points proceed in the direction of steepest descent. At odd-numbered points, P_{2k+1}, proceed in the direction determined by the line joining P_{2k-2} and P_{2k+1}. Steepest descent PARTAN (PARrallel TANgents) is in fact an n-dimensional generalization of the two-dimensional procedure of two steepest descents followed by an acceleration step.

2.5. Discussion.

The method of parallel tangents locates the extremum point of a surface with elliptical contours in at most n steps, where n is the dimension of the parameter space. In practice, very few functions have elliptical contours, and further there are always experimental errors present. So, the method is iterative rather than n-step, and this is true of all the other methods also. The number of

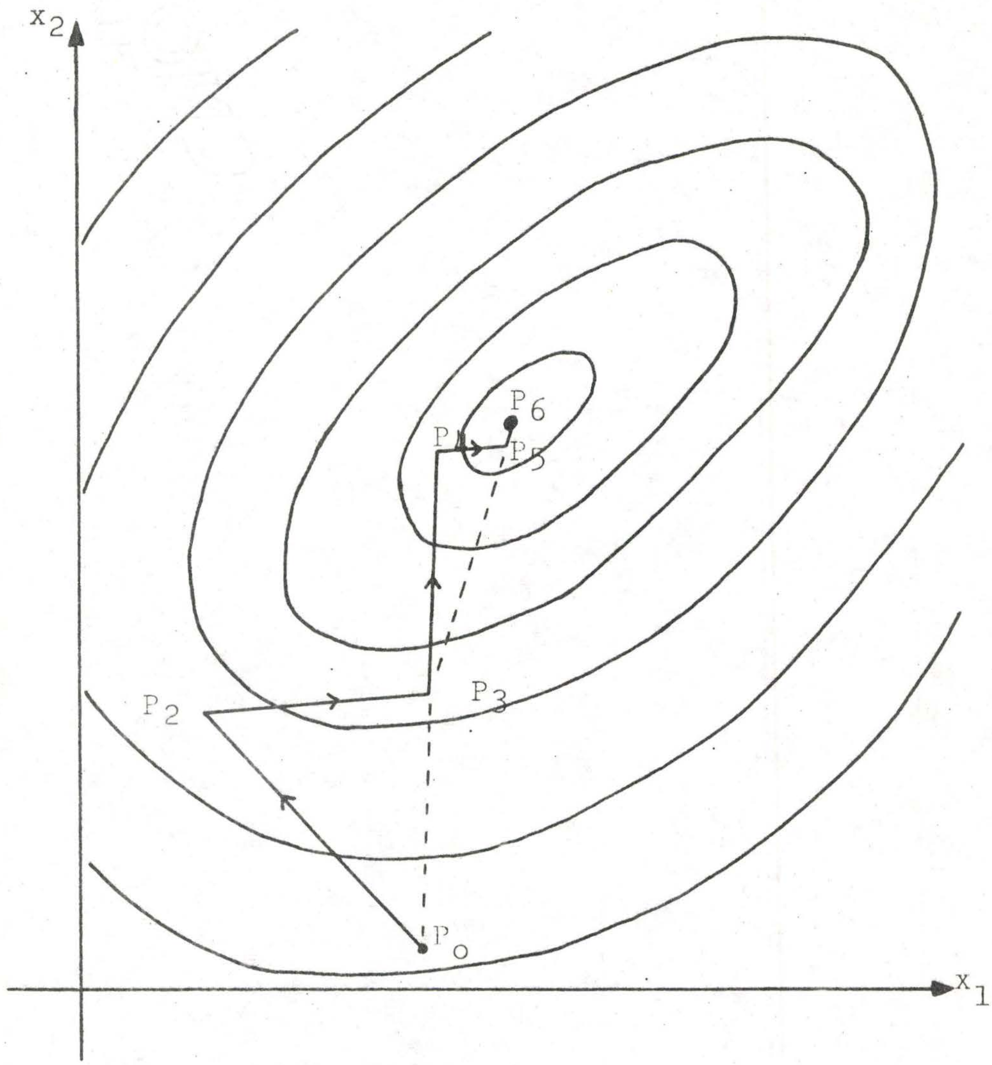


FIGURE (2.4.a)

Two-parameter optimization using the method of Parallel Tangents.

iterations required by the various methods to locate the extremum point of the cost function varies and depends very much on the shape of the contours [4], [5].

Certain practical response surface characteristics, such as ridges and valleys, often cause inefficient response patterns. Consequently, adaptive strategies must be chosen then to cope with such conditions. Often, the choice of the starting point plays an important role in avoiding such situations where progress is slow.

The methods so far developed for adaptive control without identification are far from being perfect. They all suffer from the drawback that they have no means of determining whether the extremum point reached is a local or a global one, unless the whole parameter space were searched. Search techniques are usually lengthy and time consuming. Further, because they tend to make little use of à priori data, the results obtained are often influenced by the laws of chance and probability. Also, due to the amount of time involved to search through the parameter space - a method that cannot always be applied to practical processes - they are seldom of value to practical adaptive control systems where adaptive steps usually have to be made fairly fast in order to achieve better performance.

The problem of locating an extremum point in the parameter space is analogous to that of a man searching for the bottom of a gently sloping valley in a dense fog, the searcher having at his disposal an altimeter and a compass only. The earth's surface would be analogous to the cost function surface, and the man's coordinates would correspond to a two-dimensional parameter vector $\underline{x}$. Suppose that x_1 is assigned as north, and x_2 as east in order to provide reference coordinates. One possible solution to his problem would be to measure the change in altitude for a five-step excursion north from his present position, then the change for a five-step excursion east. This would establish ∇Z and the direction of travel required to minimize altitude, Z, most rapidly for a given distance moved. The searcher might then proceed in that direction a number of steps proportional to $||\nabla Z||$ before repeating the entire process. The above technique would be that of steepest-descent. The difficulties which the searcher might encounter in his adventure would be similar to those which would be encountered in computational solutions of practical problems. Obviously, the searcher might reach the wrong minimum altitude (a local extremum) and not realize it, since he can only see the surface at his feet. If he moves too far on the basis of each measurement, he may wander aimlessly back and forth across the valley without reaching a lower altitude at any test point owing to an excessively large excursion. The conditions imposed by contraints are similar to those he would face if an unsurmountable wall suddenly blocked his path in the fog.

The methods of reaching an extremum point on a hypersurface in computer-automated systems are similar to those the searcher would use in the above example to reach the lowest point in the valley. His problem could also have been to reach the highest point of a region. The goal in adaptive control systems can be either to reach a maximum or a minimum of a certain function <u>i.e.</u> an extremum point in general.

The remainder of this thesis is devoted to the investigation of another adaptive control technique, the Vector Cost Function algorithm, which aims at extremalizing a prescribed cost function.

CHAPTER 3

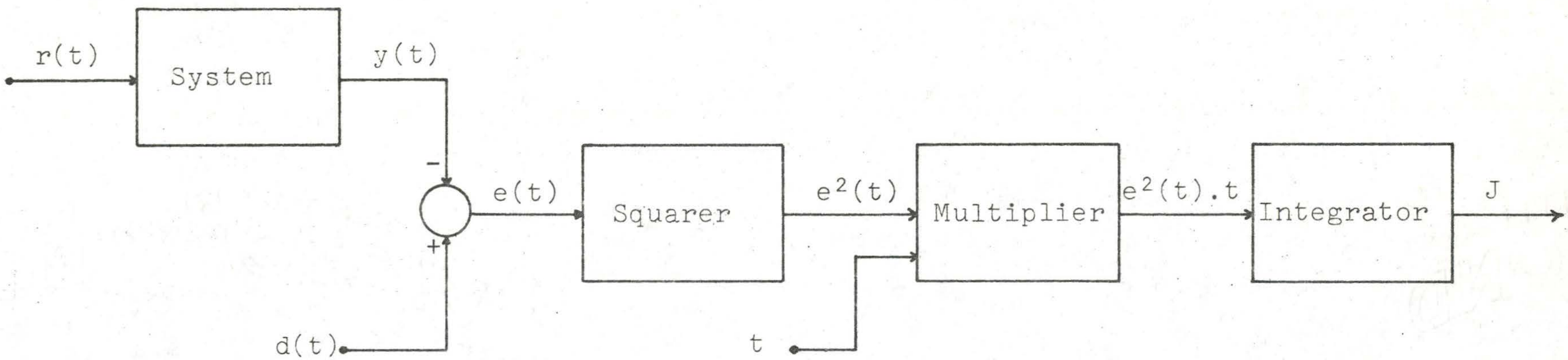
The Vector Cost Function Algorithm

The mathematical formulation of the Vector Cost Function will be developed in this chapter, and a simple example will be given to illustrate how this algorithm can be implemented on a practical system.

3.1 Cost Functions

To evaluate the performance of a system (an industrial plant, an aeroplane, a nuclear reactor......etc.) an Index of Performance - Cost Function as it is also known - is generally used. Usually this function is of an integral type which yields a scalar quantity J after a time interval T, the time over which the integration was performed, has elapsed. The most well known cost functions are the ISE (integral of squared error), the ITAE (integral of time and absolute error) and the ITSE (integral of time and squared error). A variety of other functions can also be used depending on the goal to be reached [6].

Figure (3.1.a) illustrates how the ITSE cost function can be obtained from a system. In such a system the cost



$$J = \int_{t_o}^{t_o + T} e^2(t).t.dt$$

where r(t) = system input

y(t) = system output

d(t) = desired system output

e(t) = error

J = ITSE cost function

FIGURE (3.1.a)

could be measured over intervals of time T, and hence it would be possible to determine the variation in cost with time. The value of T would usually be made a few times larger than the "time constant"* of the system itself.

The main drawbacks of such cost functions is that a scalar quantity J is obtained after a fairly long time has elapsed - any value of time comparable to the system's time constant is considered here as a long time - and further they do not yield information as to how the system's parameters should be changed in order to minimize J.

Figures (3.1.b), (3.1.c) and (3.1.d) show three configurations (open loop, forward path and feedback path configurations) which can be used for adaptive control of a plant. In the three cases f_i would be the controller parameters which would be modified in view of minimizing J. In general, the cost function can be expressed as in equation (3.1.1).

$$J = \int_{t_o}^{t_f} \phi(\underline{r}, \underline{y}, \underline{d}, t)\,dt \quad \ldots\ldots\ldots\ldots\ldots(3.1.1)$$

where, $\underline{r}(t)$ = input vector.

$\underline{y}(t)$ = plant output vector.

$\underline{d}(t)$ = desired plant output vector

t = time

$\underline{z}(t)$ = plant input vector

ϕ = a scalar function of $\underline{r}(t)$, $\underline{y}(t)$, $\underline{d}(t)$ and t.

* Here "time constant" is taken to mean the largest time constant.

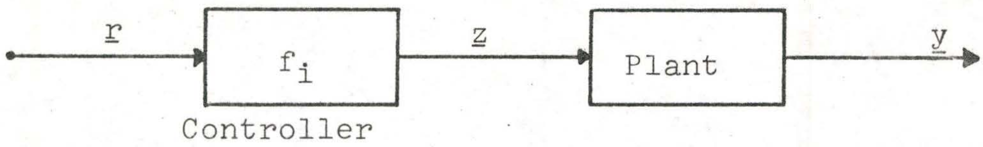


FIGURE (3.1.b) Open Loop Controller System

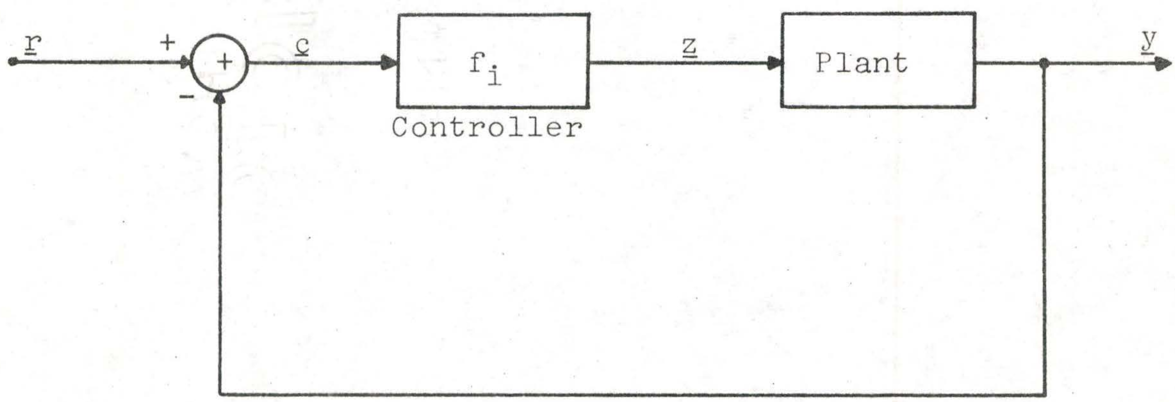


FIGURE (3.1.c) Forward Path Controller System

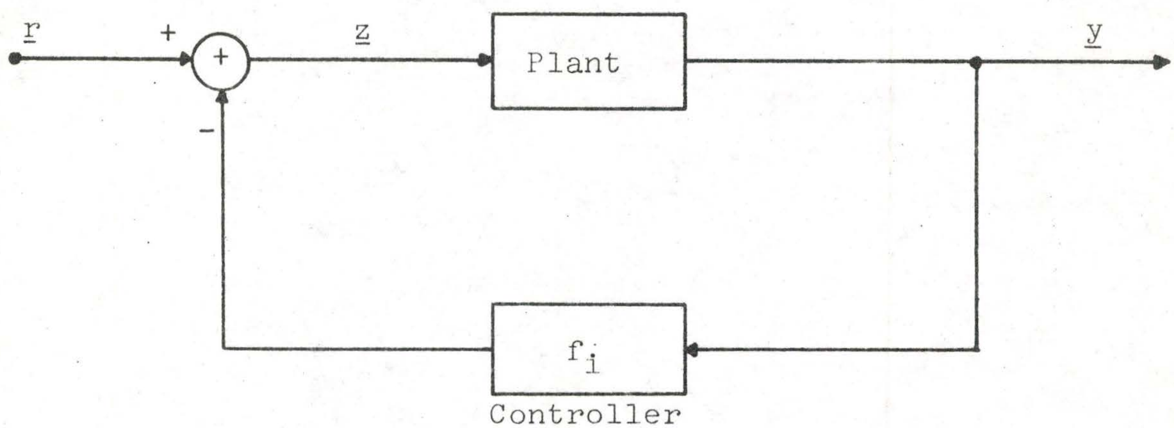


FIGURE (3.1.d) Feedback Path Controller System.

The object is to modify f_i, the controller parameters, so that the cost function J is extremized. Because of the inherent delay present between measuring J and changing $\underline{f}$, the system might become unstable as a result.

A better cost function would be one of the vector type having as many dimensions as there are controllable quantities in ϕ, the integrand of equation (3.3.1). For example, if a cost function were defined by equation (3.1.2)

$$J(t) = \int_{t_o}^{t} (||\underline{r}||^2 + \lambda||\underline{e}||^2)\, dt \quad \ldots\ldots\ldots\ldots (3.1.2)$$

where $\underline{r}$ is an m-dimensional input vector, and $\underline{e}$ an n-dimensional error vector, then, the vector cost function should be an (m+n)-dimensional vector.

Note that in equation (3.1.2) the cost is measured continuously. If J(t) is extremized at fixed intervals of time, the resultant J would also be an extremum one.

3.2 The Vector Cost Function.

The inner product of two vectors $\underline{U}$ and $\underline{V}$ is defined as

$$<\underline{U}.\underline{V}> = \sum_{i=1}^{n} U_i V_i \quad \ldots\ldots\ldots\ldots (3.2.1)$$

where $\underline{U}$ and $\underline{V}$ are both n-dimensional vectors.

Using equation (3.2.1), define a vector $\underline{H}$ whose inner product with itself varies monotonically and is one-to-one with $\phi(\underline{r},\underline{y},\underline{d},t)$, the integrand of equation (3.1.1). Frequently, $\langle\underline{H}.\underline{H}\rangle$ would be numerically equal to ϕ. For example, if equation (3.1.2) defines a particular cost function, then $\underline{H}$ can be chosen as

$$\underline{H} = \begin{bmatrix} r_1 \\ \cdot \\ \cdot \\ \cdot \\ r_m \\ \lambda e_1 \\ \cdot \\ \cdot \\ \cdot \\ \lambda e_n \end{bmatrix}$$

where $\underline{r}$ and $\underline{e}$ are m and n vectors respectively.

Minimizing $\underline{H}$ at constant time intervals is equivalent to minimizing J at the same time intervals [7].

Define $\underline{H}'$ as being the global optimum path of $\underline{H}$ (with all constraints taken into account) which yields a global optimum value J' of the cost function J.

Further, define $\underline{\gamma} = \underline{H} - \underline{H}'$ ....(3.2.2)

In other words, $\underline{\gamma}$ gives a measure of the proximity of the system to the global optimum. Hence, minimizing the norm of $\underline{H}$ - equivalent to minimizing J - ensures that the system is driven to a local optimum. On the other hand, if the norm of $\underline{\gamma}$ is minimized, the system reaches the global optimum i.e. provided that $\underline{H}'$ can be found. It is not always possible to determine $\underline{H}'$. For example, consider the two different cost functions given by equations (3.2.3) and (3.2.4) - the symbol meanings are those of equation (3.1.1).

$$J_1 = \int_{t_0}^{t} \langle \underline{H}_1 . \underline{H}_1 \rangle dt \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.2.3)$$

$$J_2 = \int_{t_0}^{t} \langle \underline{H}_2 . \underline{H}_2 \rangle dt \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.2.3)$$

where $\underline{H}_1 = \begin{bmatrix} \underline{z} \\ \lambda(\underline{d}-\underline{y}) \end{bmatrix}$, $\underline{H}_2 = [\underline{d} - \underline{y}]$.

$\underline{H}_2$ is in fact the error between the plant output and the desired plant output. In this case, $\underline{H}_2'$ is obviously zero. But, it is not possible to find $\underline{H}_1'$ since the optimum Z is not generally known.

However, in cases where $\underline{H}'$ cannot be found, one seeks to minimize $\langle \underline{H}.\underline{H} \rangle$ instead of $\langle \underline{\gamma} . \underline{\gamma} \rangle$, and the system will be driven to a local optimum value of the cost function. It must be pointed out here that all methods of adaptive control without identification of the system do suffer from the above drawback in that one does not know whether the optimum reached is a local one or a global one. The vector cost function as defined by equation (3.2.2) does, however, yield information about the nature of the optimum, as mentioned above.

Consider the expansion of $\underline{\gamma}$ into a Taylor Series in the time domain, equation (3.2.4).

$$\underline{\gamma}(t+\Delta t) = \underline{\gamma}(t) + \frac{d\underline{\gamma}}{dt}.\Delta t + \ldots + \frac{d^n\underline{\gamma}}{dt^n}.\frac{(\Delta t)^n}{n!} + \ldots \qquad \ldots\ldots(3.2.4)$$

Define $$\underline{\gamma}^n = \frac{d^{n-1}\underline{\gamma}}{dt^{n-1}}.\frac{(\Delta t)^{n-1}}{(n-1)!} \qquad \ldots\ldots(3.2.5)$$

Hence, define the general Vector Cost Function to be

$$\underline{G} = \begin{bmatrix} \underline{\gamma}^1 \\ \underline{\gamma}^2 \\ \cdot \\ \cdot \\ \underline{\gamma}^n \end{bmatrix} \qquad \ldots\ldots(3.2.6)$$

where $\underline{G}$ is an n-dimensional vector.

To minimize $\underline{G}$ a controller with n parameters is required in order to be able to control independently the n components of $\underline{G}$.

3.3 The General Adaptor Equation and Parameter Variations.

Consider the general control system of Figure (3.3.a), where:

$\underline{c}$ = forward path controller parameter vector.

$\underline{f}$ = feedback path controller parameter vector.

$\underline{\alpha}$ = plant parameter vector.

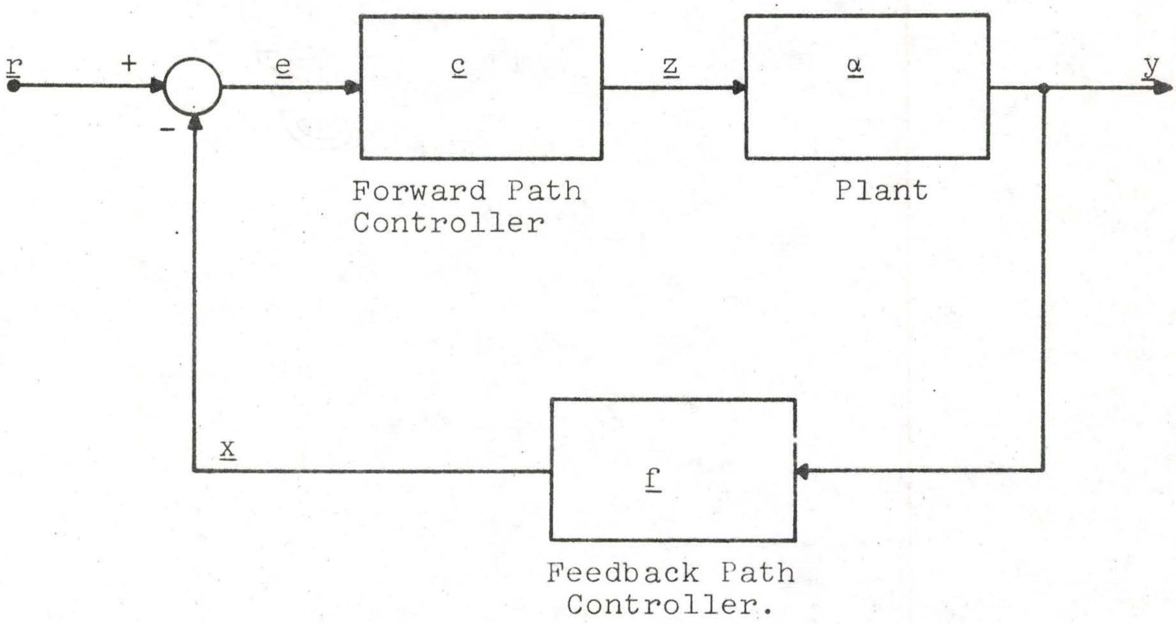

r
+
e
c
z
α
y
-
Forward Path
Controller
Plant
x
f
Feedback Path
Controller.

FIGURE (3.3.a)

The general vector cost function can be written as follows:

$$\underline{G} = \underline{G}(\underline{r},\ \underline{y},\ \underline{d},\ t,\ \underline{c},\ \underline{f},\ \underline{\alpha})$$

Expanding $\underline{G}$ into a Taylor series, equation (3.3.1) is obtained.

$$\begin{aligned}
\underline{G}(t+\Delta t) = \underline{G}(t) &+ \frac{\partial \underline{G}}{\partial t}.\Delta t + \ldots\ldots\ldots \text{(higher order terms)} \\
&\frac{\partial \underline{G}}{\partial \underline{r}}.\Delta \underline{r} + \ldots\ldots\ldots \qquad \text{"} \\
&\frac{\partial \underline{G}}{\partial \underline{d}}.\Delta \underline{d} + \ldots\ldots\ldots \qquad \text{"} \\
&\frac{\partial \underline{G}}{\partial \underline{c}}.\Delta \underline{c} + \ldots\ldots\ldots \qquad \text{"} \\
&\frac{\partial \underline{G}}{\partial \underline{f}}.\Delta \underline{f} + \ldots\ldots\ldots \qquad \text{"} \\
&\frac{\partial \underline{G}}{\partial \underline{\alpha}}.\Delta \underline{\alpha} + \ldots\ldots\ldots \qquad \text{"}
\end{aligned} \qquad \ldots\ldots\ldots\ldots(3.3.1)$$

Truncating equation (3.3.1) after the first partial derivative, the Taylor series expansion can be written as

$$\begin{aligned}
\underline{G}(t+\Delta t) = \underline{G}(t) &+ \frac{\partial \underline{G}}{\partial t}.\Delta t + \frac{\partial \underline{G}}{\partial \underline{r}}.\Delta \underline{r} + \frac{\partial \underline{G}}{\partial \underline{d}}.\Delta \underline{d} + \frac{\partial \underline{G}}{\partial \underline{c}}.\Delta \underline{c} \\
&+ \frac{\partial \underline{G}}{\partial \underline{f}}.\Delta \underline{f} + \frac{\partial \underline{G}}{\partial \underline{\alpha}}.\Delta \underline{\alpha}
\end{aligned} \qquad \ldots\ldots\ldots\ldots(3.3.2)$$

Also, $\underline{G}(t+\Delta t) = \underline{G}(t) + \Delta\underline{G}$.

Note that $\frac{\partial \underline{G}}{\partial \underline{r}}$, $\frac{\partial \underline{G}}{\partial \underline{d}}$, $\frac{\partial \underline{G}}{\partial \underline{c}}$, $\frac{\partial \underline{G}}{\partial \underline{f}}$ and $\frac{\partial \underline{G}}{\partial \underline{\alpha}}$ are sensitivity matrices with respect to the parameters $\underline{r}$, $\underline{d}$, $\underline{c}$, $\underline{f}$ and $\underline{\alpha}$ respectively. The object of the vector cost function is to make $\underline{G}(t+\Delta t) = \underline{\theta}$, where $\underline{\theta}$ is the null vector. In other words, the adaptation problem is to set $\Delta\underline{c}$ and $\Delta\underline{f}$ (and $\Delta\underline{r}$ in some cases*) to values which will cancel out the changes in $\Delta\underline{\alpha}$, $\Delta\underline{d}$ and t (and $\Delta\underline{r}$) [8]. This requires the solution of equation (3.3.3):

$$\underline{G}(t) + \frac{\partial \underline{G}}{\partial t}.\Delta t + \frac{\partial \underline{G}}{\partial \underline{r}}.\Delta\underline{r} + \frac{\partial \underline{G}}{\partial \underline{d}}.\Delta\underline{d} + \frac{\partial \underline{G}}{\partial \underline{c}}.\Delta\underline{c} + \frac{\partial \underline{G}}{\partial \underline{f}}.\Delta\underline{f} + \frac{\partial \underline{G}}{\partial \underline{\alpha}}.\Delta\underline{\alpha} = \underline{\theta} \quad \text{..........(3.3.3)}$$

In order to calculate the required changes $\Delta\underline{f}$, $\Delta\underline{c}$ (and/or $\Delta\underline{r}$) in the control parameter vectors $\underline{f}$ and $\underline{c}$ respectively, the following steps are needed:

1. Measurement of $\underline{G}(t)$
2. Calculation of $\frac{\partial \underline{G}}{\partial t}.\Delta t$ and $\frac{\partial G}{\partial \underline{d}}.\Delta\underline{d}$ from the arbitrarily defined functions $\underline{G}$ and $\underline{d}$ respectively.
3. Knowledge of $\frac{\partial \underline{G}}{\partial \underline{f}}$, $\frac{\partial \underline{G}}{\partial \underline{c}}$ (and/or $\frac{\partial \underline{G}}{\partial \underline{r}}$)
4. Estimation of $\frac{\partial \underline{G}}{\partial \underline{\alpha}}.\Delta\underline{\alpha}$

* If $\underline{r}$ is known, then $\frac{\partial \underline{G}}{\partial \underline{r}}$ and $\Delta\underline{r}$ are calculated. If however $\underline{r}$ is not known, $\frac{\partial \underline{G}}{\partial \underline{r}}$ and $\Delta\underline{r}$ have to be calculated and measured. $\underline{r}$ can also be used as a control variable in the system.

Step 1: $\underline{G}(t)$ can easily be found since it only involves the quantities $\underline{r}$, $\underline{y}$, $\underline{d}$* and t which can either be measured or calculated.

Step 2: This step follows from the fact that $\underline{G}$ and $\underline{d}$ were initially arbitrarily chosen.

Step 3: The sensitivity matrices $\frac{\partial \underline{G}}{\partial \underline{f}}$ and $\frac{\partial \underline{G}}{\partial \underline{c}}$ can only be learned as the adaptation proceeds. Initially, they have to be arbitrarily assumed and later improved by updating.

Step 4: The only possible way of obtaining the value of $\frac{\partial \underline{G}}{\partial \underline{\alpha}}.\Delta\underline{\alpha}$ is to estimate it since no attempt is made in this algorithm to identify the plant. Knowledge of the plant behaviour to parameter changes can only be acquired as time elapses.

Define, $\underline{b} = \frac{\partial \underline{G}}{\partial \underline{\alpha}}.\Delta\alpha$ ....(3.3.4)

The General Adaptor Equation for the vector cost function algorithm can therefore be written as follows:

$$\underline{G}(t+\Delta t) = \underline{G}(t) + \frac{\partial \underline{G}}{\partial t}.\Delta t + \frac{\partial \underline{G}}{\partial \underline{f}}.\Delta\underline{f} + \frac{\partial \underline{G}}{\partial \underline{c}}.\Delta\underline{c} + \frac{\partial \underline{G}}{\partial \underline{r}}.\Delta\underline{r} + \frac{\partial \underline{G}}{\partial \underline{d}}.\Delta\underline{d} + \underline{b} \quad \text{...........(3.3.5)}$$

* The Vector Cost Function assumes that the desired output is known at all instants of time.

Equation (3.3.5) will in fact be seen to be the heart of this algorithm, and will be used for predicting the changes in the controller parameters and in learning the sensitivity matrices.

3.4 Learning and Updating.

The basic idea underlying this part of the algorithm is to make calculated changes in one of the controller parameters available, and to note the resultant change in $\underline{G}$ i.e. $\Delta\underline{G}$. Then, the column of the sensitivity matrix which depends on that parameter can be updated. For example, if the change were made in f_i, then the column updated would be that of the $\frac{\partial \underline{G}}{\partial \underline{f}}$ matrix

i.e. $\left[\frac{\partial G_j}{\partial f_i}\right]_{j=1, \ldots n}$ where $\frac{\partial \underline{G}}{\partial \underline{f}}$ is an nxn matrix, and $\underline{f}$ an n-dimensional vector.

The controller parameters can be changed one by one at time intervals Δt apart, and after each change the appropriate column of the sensitivity matrix updated.

Suppose that in the general control system of Figure (3.3.f) the input vector $\underline{r}(t)$ is varying with time in an unknown fashion. What is required then is to adjust $\underline{f}$ and $\underline{c}$ so as to optimize $\underline{G}$, the vector cost function.

Figure (3.4.a) shows two possible adaptation schemes which differ in the frequencies at which adaptation of the two controller parameters is performed.

Scheme (i) adapts f_i, updates $\frac{\partial \underline{G}}{\partial \underline{f}}$, learns $\frac{\partial \underline{G}}{\partial \underline{r}}$; then adapts c_j, updates $\frac{\partial \underline{G}}{\partial \underline{c}}$, learns $\frac{\partial \underline{G}}{\partial \underline{r}}$ and starts again.

Scheme (ii) adapts f_i, updates $\frac{\partial \underline{G}}{\partial \underline{f}}$, adapts $\underline{f}$, learns $\frac{\partial \underline{G}}{\partial \underline{r}}$; then adapts c_j, updates $\frac{\partial \underline{G}}{\partial \underline{c}}$, adapts $\underline{c}$ and starts over again.

These two schemes are not unique and the particular way in which the "Learn and Adapt" steps are performed depends entirely upon the designer's choice. Thus, a great deal of freedom exists in the particular order and way that the controller parameters may be changed and the sensitivity matrices updated.

3.5 Example

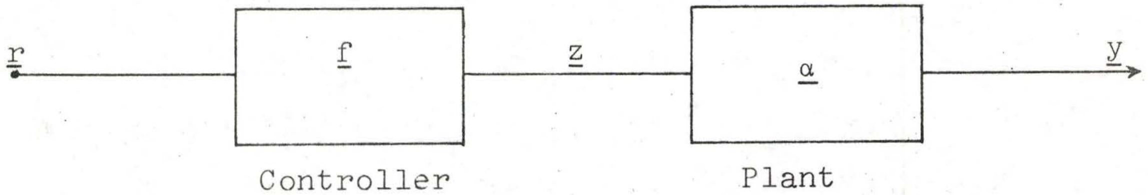


FIGURE (3.5.a) (Open Loop Controller.)

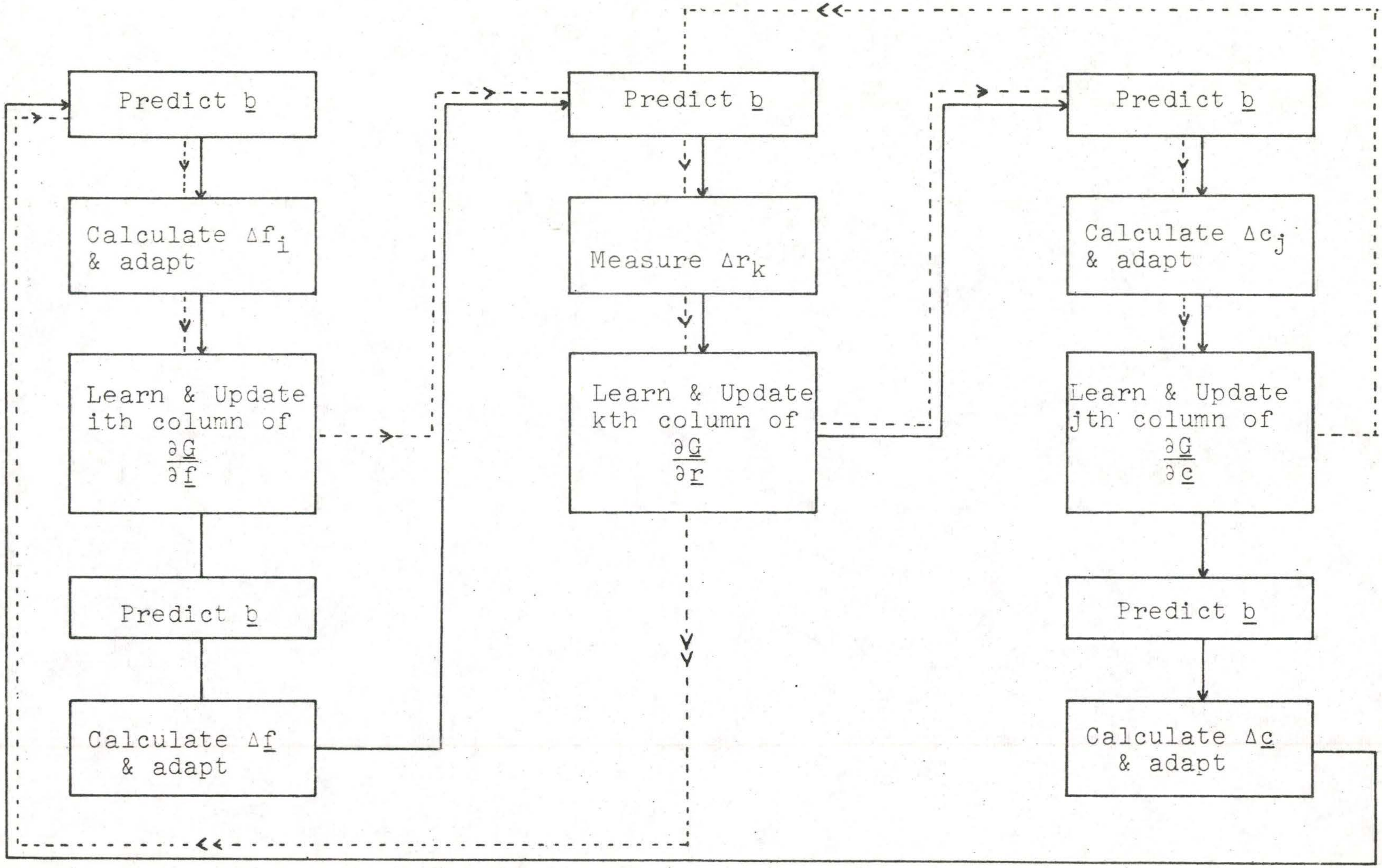


FIGURE (3.4.a)

Scheme (i) represented by the dotted line

Scheme (ii) represented by continuous line

(i) At $t = t_o$, solve for $\Delta\underline{f}$ using equation (3.5.1) below:

$$\underline{G}(t_o) + \frac{\partial \underline{G}}{\partial \underline{f}}.\Delta\underline{f} + \frac{\partial \underline{G}}{\partial \underline{d}}.\Delta\underline{d} + \frac{\partial \underline{G}}{\partial t}.\Delta t + \frac{\partial \underline{G}}{\partial \underline{r}}.\Delta\underline{r} + \underline{b} = \underline{\theta} \quad \ldots(3.5.1)$$

where $\underline{\theta}$ is the null vector, and all matrices are n x n and all vectors n-dimensional. n is the order of adaptation.

Then, change Δf_i keeping the (n-1) remaining components of $\Delta\underline{f}$ to zero.

(ii) At $t = t_o + \Delta t$, measure $\underline{G}(t_o+\Delta t)$ and hence calculate $\Delta\underline{G}$.

Then, update the ith column of the sensitivity matrix $\frac{\partial \underline{G}}{\partial \underline{f}}$ using equation (3.5.2).

$$\frac{\partial G_j}{\partial f_i}.\Delta f_i = \Delta G_j - \frac{\partial G_j}{\partial t}.\Delta t - b_j - \sum_{k=1}^{n}\left(\frac{\partial G_j}{\partial d_k}.\Delta d_k + \frac{\partial G_j}{\partial r_k}.\Delta r_k\right)$$

for $j = 1 \ldots\ldots\ldots n$ $\qquad\ldots\ldots\ldots\ldots(3.5.2)$

There are n elements to be updated in the ith column of the $\frac{\partial \underline{G}}{\partial \underline{f}}$ matrix, and so j takes successive values of 1 to n.

The next step after estimating $\underline{b}$ could be to change the (i + 1)th element of $\underline{f}$ and update the (i + 1)th row of the sensitivity matrix at an interval of time Δt later. Or, all the elements of $\underline{f}$ could be changed simultaneously after the updating phase in order to make better use of the recent updating; then, go on to change the (i+1)th element of $\underline{f}$ and update the (i+1)th row of the sensitivity matrix, and start all over again. The first method of adaptation would resemble scheme (i) (Figure 3.4.1) and the second scheme (ii).

In the example, the above schemes are probably the only two suitable ones to use. Any others would be a combination of these two and would probably be not as efficient since there is only one controller in the system and $\underline{r}$ has been assumed fixed.

CHAPTER 4

The Vector Cost Function Implementation

This chapter considers the implementation of the Vector Cost Function algorithm upon three different configurations, and the methods of prediction and updating which were used. The complete systems were simulated on digital computers (IBM 7040 and CDC 6400) using the Fortran IV compiler.

Three different system configurations were investigated which had:

(a) a feedback path controller.

(b) a forward path controller.

(c) an open loop controller.

These general configurations were tested since the Vector Cost Function method makes no mention of the type of configuration and the particular controller which should be used. Theoretically, any configuration using any controller should lead to an adaptive system which would yield an optimum value of the cost function, either a local optimum or a global one depending on whether $\underline{H}$ or

$\underline{\gamma}$ was used. It will be seen later in chapter five that this is not always the case since stability and other factors have to be taken into account.

4.1 Feedback Path Controller System

This configuration appears to be an attractive one, for the properties of feedback have long been established. But, it must be remembered that feedback introduces another loop in the system on top of the adaptive loop itself. The system, apart from being more sensitive to changes in the parameters of the feedback loop, could become more oscillatory or even totally unstable as a result.

Figure (4.1.a) shows the feedback configuration which was used here. The state-space formulation of the plant is given by equation (4.1.1.) below:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -0.5 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} [z] \quad \ldots\ldots (4.1.1)$$

where y_1 and y_2 represent the system's state-variables. The feedback-path controller output is given by equation (4.1.2).

$$x = f_1 |e_1| + f_2 \left|\frac{de_1}{dt}\right| \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.1.2)$$

$$\underline{e} = \underline{d} - \underline{y} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.1.3)$$

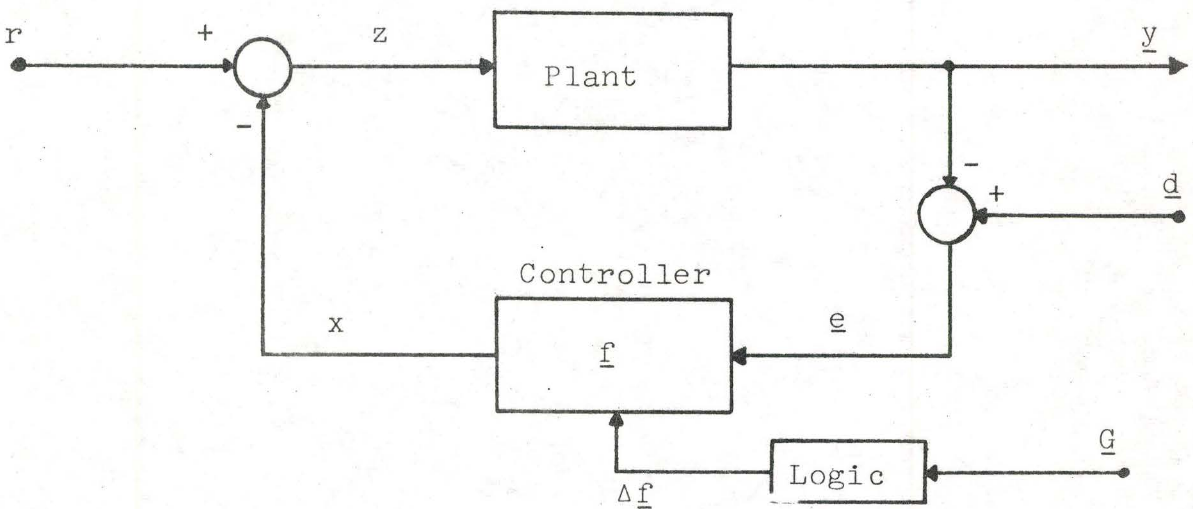


FIGURE (4.1.a)

Feedback Path Controller System

r = system input

z = plant input

x = controller output

y = plant output

d = desired plant output

e = error

f = controller parameter

G = vector cost function

The quantities r, z and x are scalars and $\underline{d}$, $\underline{y}$, $\underline{f}$ and $\underline{e}$ are two-dimensional vectors. The initial conditions imposed on the system were the following:

$$\underline{f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \underline{y} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \underline{d} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and } r = 1.$$

r and $\underline{d}$ were kept at their initial values throughout the response, and so the system had therefore to respond to a unit step input. The value of the sensitivity matrix was arbitrarily chosen to be:

$$\frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The value of the prediction vector $\underline{b}$ was initially taken to be zero. This was probably the best choice that could be made considering the fact that no knowledge of the plant should be available initially.

Adaptive changes were made at every 0.01-second time intervals though values of $\underline{G}$ were measured at every 0.001 seconds. This will be more thoroughly explained later in section (4.4) of this chapter.

The vector cost function was selected as shown by equation (4.1.4). This cost function is basically equivalent to the ISE index of performance. Other indices of performance could also have been chosen. This particular one places more emphasis on large errors than on small ones. On the other hand, had the ITAE criterion been chosen, more emphasis would have been placed on long duration errors and transients.

$$\underline{H} = \underline{d}-\underline{y} \qquad \text{(i)}$$

Hence $$\underline{H}' = \underline{\theta} \qquad \text{(ii)}$$

$$\underline{G} = \underline{H}-\underline{H}' = \underline{d}-\underline{y} \qquad \text{(iii)} \qquad \ldots\ldots\ldots\ldots\ldots (4.1.4)$$

The ISE criterion was selected because it was the simplest one to handle. Further $\underline{H}'$ in that case was easily found. Had the ITAE criterion or any other criterion involving a time weighting factor been selected, another term in the Adaptor Equation would have resulted, i.e. $\frac{\partial \underline{G}}{\partial t}.\Delta t$. Also, less emphasis would have been placed on the initial error. Consequently, the system would have taken longer to adapt to the desired output had the system been driven initially by the controllers along a trajectory which would have increased the initial error. Because of the initial arbitrariness of the choice of the parameters, what is required is a cost function which will drive the system in the right direction in a fairly short time. The ISE criterion does provide the system with such a means and was therefore judged useful here.

With the set of parameters chosen and the controller used, the Adaptor Equation reduces to equation

(4.1.5).

$$\underline{G}(t + \Delta t) = \underline{G}(t) + \frac{\partial \underline{G}}{\partial \underline{f}} . \Delta \underline{f} + \underline{b} \quad \text{.............(4.1.5)}$$

Note that here the vector $\underline{b}$ is still used though the plant does not contain variable parameters. This will be discussed later in section (4.4) of this chapter. Also note that the controller sensitivity matrix $\frac{\partial \underline{G}}{\partial \underline{f}}$ is of dimension 2 x 2 because the controller parameter vector $\underline{f}$ is two-dimensional.

4.2 Forward Path Controller System

In this configuration, the controller is inserted in the forward path of the system and in series with the plant. Its output is used as the forcing function to the plant to be controlled. Usually the feedback signal is either the plant output itself or the error between the desired plant output and the actual plant output. Of course, any other signal can also be used but the above two are the most common ones.

Once again, the input to the controller does not need to be of a specific nature and the system should be driven towards an optimum of the cost function whatever the input function. But, as will be pointed out later, it was found from the work carried out that this is not always the case, for the present form of the Vector Cost

Function technique does not take into account the major problem of stability.

The feedback signal chosen for this system was the plant output itself because the other types were found less satisfactory and more prone to instability. The forward path controller system used is shown in Figure (4.2.a). The state-space formulation of the plant is given by equation (4.2.1), and that of the controller by equation (4.2.2).

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -0.5 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} [z] \quad \text{.....(4.2.1)}$$

$$z = \langle \underline{f} . \underline{q} \rangle \quad \text{........................(4.2.2)}$$

$$\underline{q} = \underline{r} - \underline{y} \quad \text{........................(4.2.3)}$$

The initial conditions imposed upon the system are given below

$$\underline{y} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \underline{d} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \underline{r} = \begin{bmatrix} 1.1 \\ 0 \end{bmatrix} \text{ and } \underline{f} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\text{..........(4.2.4)}$$

The sensitivity matrix chosen is given by equation (4.2.5) and the initial value of $\underline{b}$ was again taken to be zero.

$$\frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{..................(4.2.5)}$$

The cost function used was again the ISE criterion

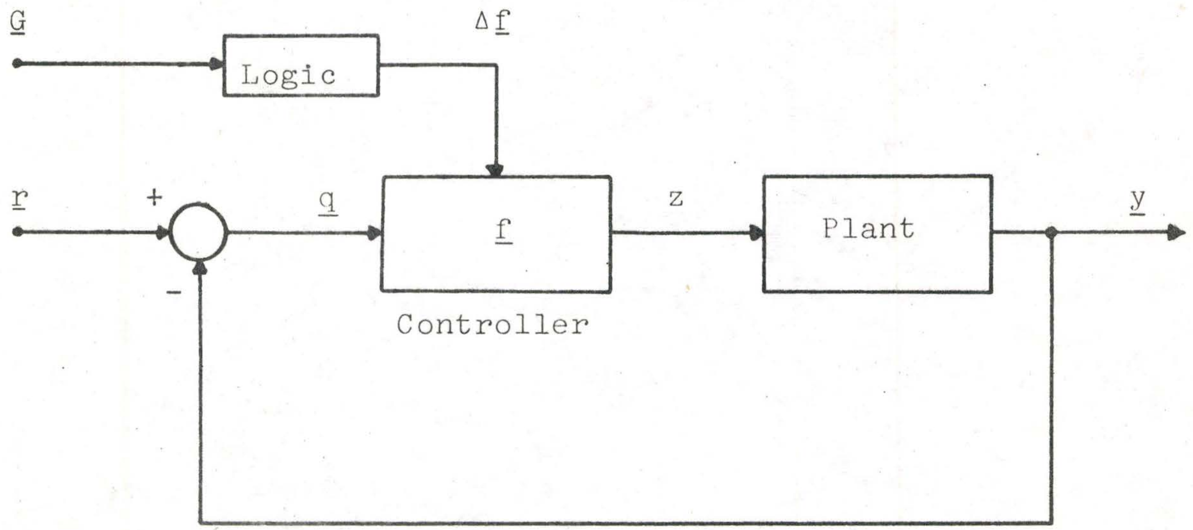


FIGURE (4.2.a)

Forward Path Controller System

r = system input

z = plant input

y = plant output

q = controller input

f = controller parameter

G = vector cost function

and reduces to equation (4.2.6) below which is identical to equation (4.1.4) previously obtained in section (4.1).

$$\underline{H} = \underline{d} - \underline{y} \qquad \text{(i)}$$

$$\underline{H}' = \underline{\Theta} \qquad \text{(ii)} \qquad \ldots\ldots\ldots\ldots\ldots(4.2.6)$$

$$\underline{G} = \underline{H} - \underline{H}' = \underline{d} - \underline{y} \quad \text{(iii)}$$

The system input vector $\underline{r}$ and the desired plant output vector $\underline{d}$ were kept unchanged during the response at their initial values given above. Once more, the Adaptor equation reduced to equation (4.1.5) of section (4.1). Adaptation was carried out at 0.01-second time intervals and again $\underline{G}$ was measured at every 0.001 seconds.

4.3 Open Loop Controller System

The last one of the three configurations considered was of the open loop controller type. Here, the controller is placed in series with the plant and preceeding it. As in the case of the forward path controller system (Figure (4.2.a)), the output of the controller, is the forcing function to the plant but in this case the controller input is a constant function instead of being a variable one. Figure (4.3.a) illustrates this type of adaptive control configuration.

This configuration has long been used in connection with adaptive control systems and offers the advantage that it is relatively simple to analyze, i.e. the effect of the controller on the plant can be directly assessed.

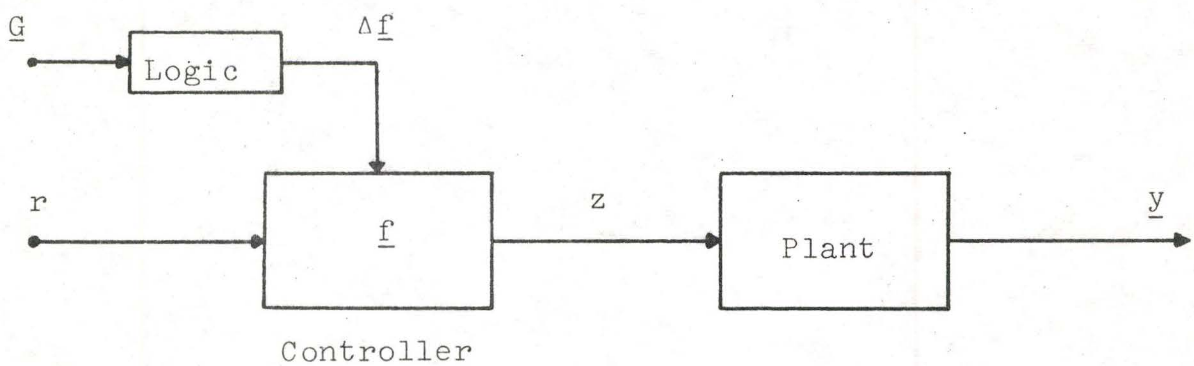


FIGURE (4.3.a)

Open Loop Controller System.

r = controller input

z = plant input

y = plant output

f = controller parameter

G = vector cost function

d = desired output

In the case of the other systems, the effect of feedback usually makes the analysis more complex.

The state-space representation of the plant to be controlled is given by equation (4.3.1) below.

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -4 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} [z] \quad \text{..............(4.3.1)}$$

The controller output which is also the forcing function to the plant is represented by equation (4.3.2)

$$z = f_1 r + x \qquad \text{(i)}$$

$$\text{...........(4.3.2)}$$

where, $\dot{x} = -f_2 x + f_1 r$ (ii)

In the above case x, r and z are scalar quantities and $\underline{f}$, $\underline{G}$ and $\underline{y}$ are two-dimensional vectors. The initial values of the parameters of the system are as given by equations (4.3.3) and (4.3.4)

$$\underline{y} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \underline{d} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \underline{f} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } r = 1 \text{ ......(4.3.3)}$$

$$\frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{..................(4.3.4)}$$

The initial estimate of the vector $\underline{b}$ was again taken to be zero as explained earlier in section (4.1).

Adaptive steps were made at every 0.001-second time intervals, and ten measurements of $\underline{G}$ were taken between each adaptive step.

As for the other systems, the cost function was selected to be the ISE criterion, i.e. equation (4.3.5)

$$\begin{aligned} \underline{H} &= \underline{d} - \underline{y} && \text{(i)} \\ \underline{H}' &= \underline{\theta} && \text{(ii)} \qquad \ldots\ldots\ldots(4.3.5) \\ \underline{G} &= \underline{H} - \underline{H}' = \underline{d} - \underline{y} && \text{(iii)} \end{aligned}$$

The input r to the system and the desired plant output vector were kept constant throughout the response which meant that the system had in fact to respond to a step input. So, the adaptor equation once more reduced to equation (4.3.6).

$$\underline{G}(t + \Delta t) = \underline{G}(t) + \frac{\partial \underline{G}}{\partial \underline{f}}.\Delta\underline{f} + \underline{b} \ \ldots\ldots\ldots\ldots(4.3.6)$$

The results obtained for the systems of sections (4.1), (4.2) and (4.3) are presented in the next chapter and the values of the graphs plotted are given in Appendix (II).

4.4 The Prediction Term

[A] The vector $\underline{b}$ forms part of the adaptor equation (section (3.3), equ: (3.3.5)) and its estimated values are used at time intervals Δt apart. Careful choice of the method used to estimate $\underline{b}$ from past data gathered is required and this choice would usually be dictated by the type of statistics that $\underline{\alpha}$ is likely to obey, the storage space available in the computer used in the adaptive loop, and so on.

An arbitrary value is initially chosen for $\underline{b}$ and afterwards better estimates are made as more is learnt about the system. In the adaptor equation, the calculation of the controller parameter change $\Delta \underline{f}$ requires a value of $\underline{b}$, and this same value is used at a time interval Δt later to update the controller sensitivity matrix, as explained in chapter three (sections (3.4) and (3.5)). Hence, this vector plays an important role in calculating the required changes in the controller parameters, and in updating the controller sensitivity matrix.

For the adaptive systems considered in this thesis (Figures (4.1.a), (4.2.a) and (4.3.a)), the vector $\underline{b}$ was divided into two parts according to equation (4.4.1).

$$\underline{b} = \underline{b}' + \underline{b}'' \quad ....(4.4.1)$$

Here the vector $\underline{b}'$ represents the estimated future state of the system at the next adaptive time interval Δt, and is calculated from past data stored in the computer memory. The vector $\underline{b}''$ is used to take into account unforseen plant parameter changes and any errors arising from the calculations.

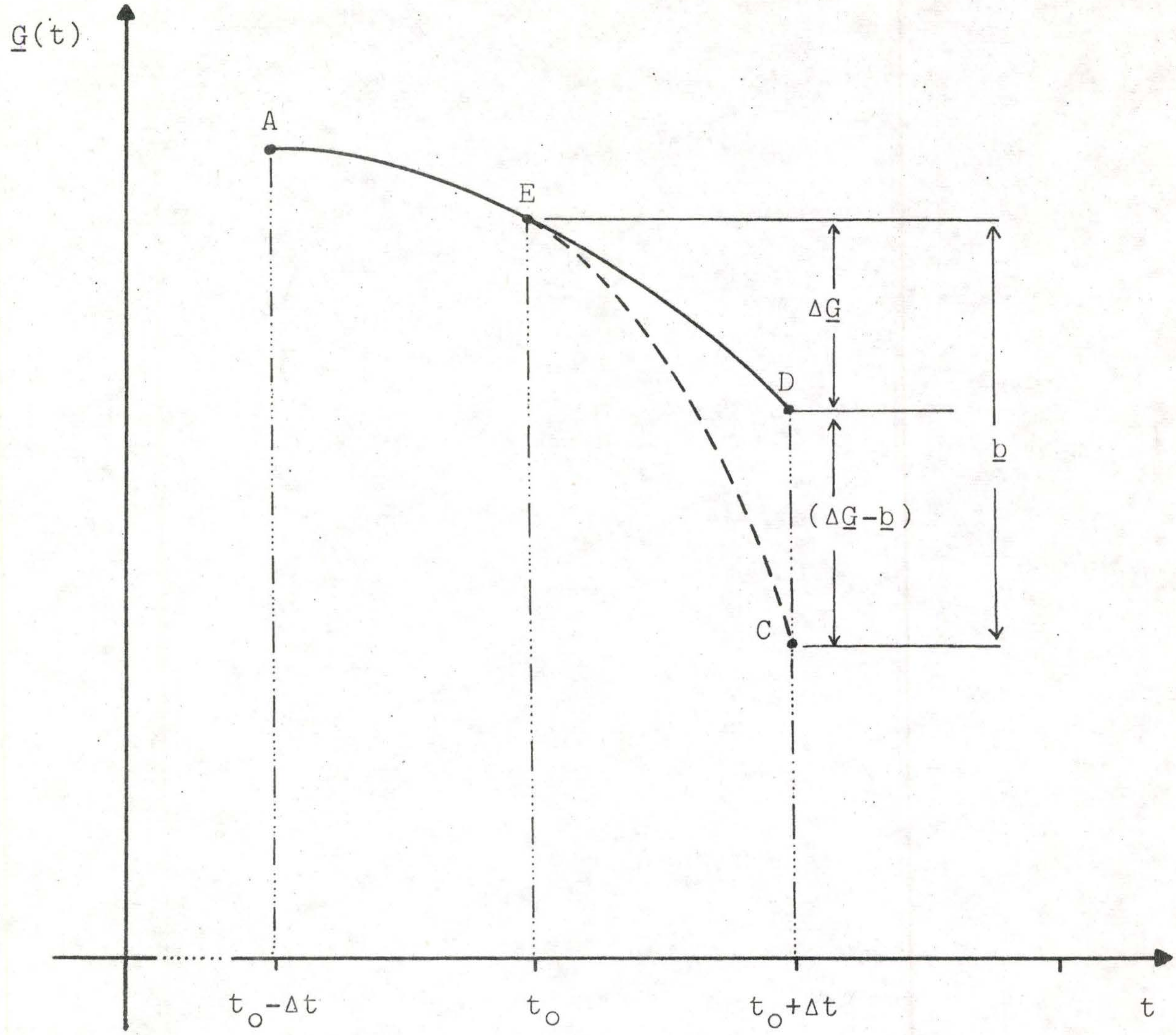


FIGURE (4.4.a)

A represents the value (measured) of $\underline{G}$ at $t = t_o-\Delta t$

E " " " " " " " $t = t_o$

C " " " (predicted) " " $t = t_o+\Delta t$

D " " " (measured) " " $t = t_o+\Delta t$

$DC = \Delta\underline{G} - \underline{b}$

A diagram illustrating the meaning of the vector $\underline{b}$ is given in Figure (4.4.a).

Changes in the controller parameters were made at time intervals Δt apart in the adaptive systems investigated. To obtain a better estimate of $\underline{b}'$, the cost function vector $\underline{G}$ was measured at every $0.1\Delta t$ interval. Consequently, between any two adaptive steps, ten values of $\underline{G}$ were obtained and from them the vector $\underline{b}'$ was estimated. No other past values of $\underline{G}$ or $\underline{b}'$ were stored and used for the calculation since these were judged adequate. Further, because of the time interval between adaptive steps was small, no significant error was made by estimating the value of the vector $\underline{b}''$ as zero.

[B] Evaluation of $\underline{b}'$

From the values of $\underline{G}$ obtained between successive adaptive steps, the gradient of $\underline{G}$, i.e. $\frac{d\underline{G}}{dt}$, was calculated for each $0.1\Delta t$ time interval. The point s_a of Figure (4.4.b) was obtained by linear extrapolation from the initial slope s_1 to the final slope s_{10}. Again by extrapolating the final slope s_{10} and the one $0.1\Delta t$ before, i.e. s_9, the point s_b - which is Δt away from s_{10} - was obtained.

s_c, the predicted slope at a time interval Δt away from s_{10}, was obtained from equation (4.4.2).

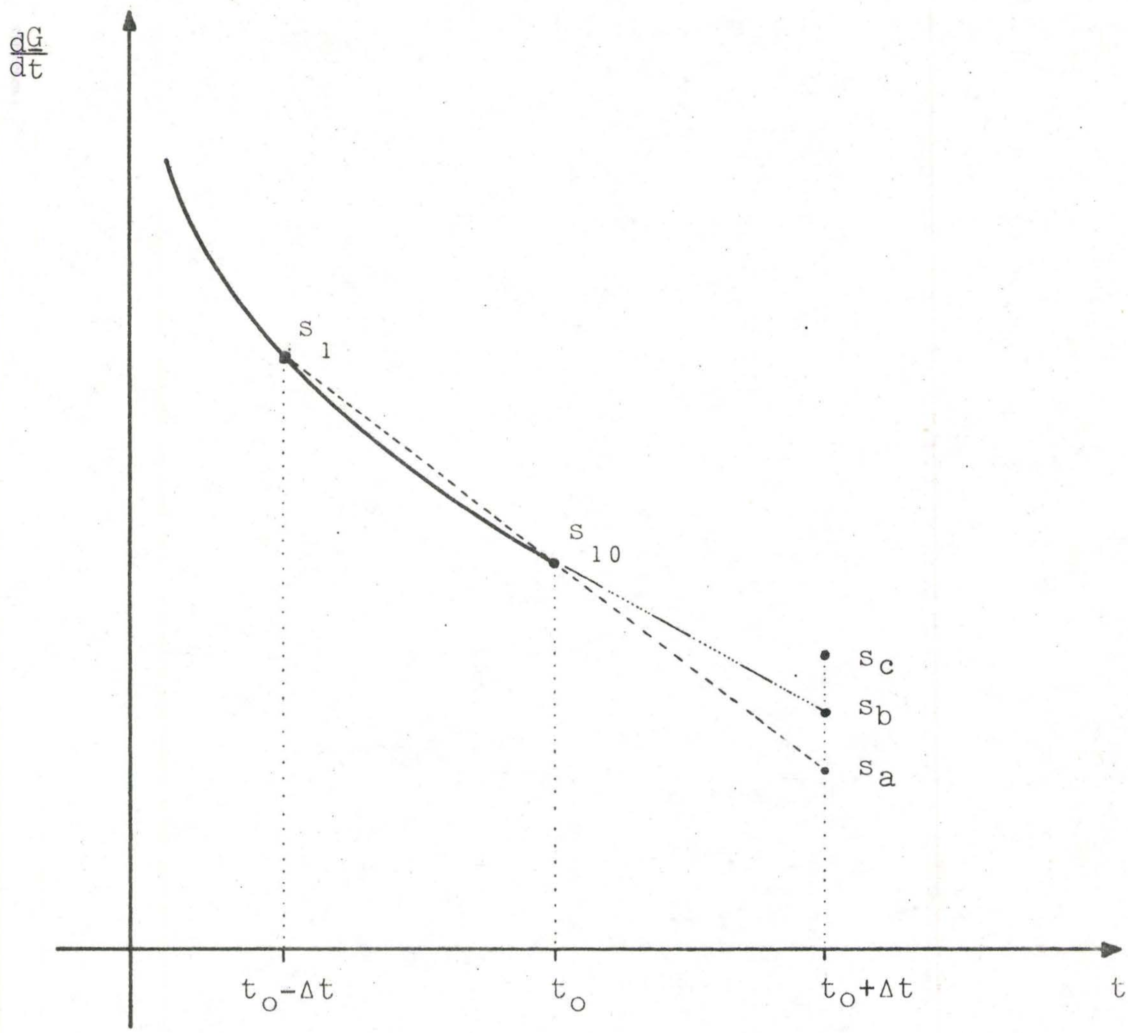


FIGURE (4.4.b)

Estimation of b'

$$s_c = s_b - (s_a - s_b) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.4.2)$$

$$s_a = s_{10} + (s_{10} - s_1) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.4.3)$$

$$s_b = s_{10} + \frac{(s_{10} - s_9).\Delta t}{0.1\Delta t} \quad \ldots\ldots\ldots\ldots\ldots (4.4.4)$$

The vector $\underline{b}'$, the estimated state of the system, was then calculated by assuming that the gradient of $\underline{G}$ varied linearly from the point s_{10} to the point s_c.

For on-line systems, the shorter the computation time of the operations on the computer, the larger is the maximum possible frequency of adaptation. Here there has to be a compromise between adaptation speed and maximum error which can be tolerated. The slower the speed of adaptation, the larger the adaptive time-lag, i.e. information gathered at time t_o is only used at time $t_o+\Delta t$, and hence the time lag is always Δt in this system.

More sophisticated methods* of estimating the future state of the system could have been used, but the one described was found accurate enough for this type of response**. It must be remembered that adaptation to any sudden input must be made fairly rapidly in practice if an improved response is to be obtained. On the other hand, for systems with slowly varying inputs or parameters, less

* The Lagrange interpolation and extrapolation method is one of them.

** see appendix (III)

frequent adaptation and greater accuracy of prediction are required. There is also more time available to perform the computation because of the slower adaptive rate, and so sophisticated and accurate methods can be used to a great advantage. For a step input response, transients often hide the true nature of the response, and so more accurate methods would not in general mean great improvement in the system's performance.

4.5 Learning and Updating

The highest frequency at which one can update the controller sensitivity matrix is determined by Δt, the time interval between adaptive steps. One such method could be to change each controller parameter in turn, and after each adaptive step update the appropriate column of the matrix. In other words, the whole matrix can be updated in not less than $n\Delta t$, where n represents the dimension of the controller parameter vector. This holds, of course, if there is only one controller in the system. If not, it will take longer than $n\Delta t$ to completely update it, and the frequency of updating will depend on how many more adaptive steps are required to learn and update the remainder of the sensitivity matrices.

The controller sensitivity matrix $\frac{\partial \underline{G}}{\partial \underline{f}}$ was updated every $2\Delta t$ interval of time because of the adaptive scheme chosen* for the systems considered. The method used is given by equation (4.5.1).**

$$\left[\frac{\partial G_j}{\partial f_i}\right]_N = \left[\frac{\partial G_j}{\partial f_i}\right]_O \times (1-W) + \left[\frac{\partial G_j}{\partial f_i}\right]_C \times W \quad ..(4.5.1)$$

$$(j = 1, \ldots n)$$

where i = ith column of $\frac{\partial \underline{G}}{\partial \underline{f}}$

j = jth row of $\frac{\partial \underline{G}}{\partial \underline{f}}$

W = arbitrary weighting coefficient.

Each column of the matrix was updated according to that equation. The particular way in which the updating takes place is completely arbitrary, as pointed out earlier in chapter three (section 3.4). Equation (4.5.1) is a simple method of updating by which more or less emphasis can be placed on the newly calculated values of the sensitivity coefficients depending on the value of W. This parameter had the following values for the systems considered.

Feedback path controller system: W = 1

Forward path controller system : W = 0.4

Open Loop controller system: W = 0.2

* see section (4.7)

** The subscripts N, O and C refer to the New, Old and Calculated values of $\frac{\partial G_j}{\partial f_i}$ respectively.

From equation (4.5.1) it is apparent that if the calculated values of the matrix elements are identical to the old ones, then there is no change in the sensitivity matrix. The response can be changed depending on the method used in updating $\frac{\partial \underline{G}}{\partial \underline{f}}$, but this aspect was not investigated here.

4.6 Constraints Imposed.

In all practical systems, there is a limit to the values that certain parameters - gain, speed, current.... etc. - can have. These limits are usually dictated by the geometry and design requirements of the system. On the other hand, simulated systems can assume almost any parameter values (as large as the computer can handle) without disastrous consequences such as breakdown in a practical system. Because of this very fact, constraints must be imposed on the maximum and/or minimum values that certain variables of simulated systems can have so that the results obtained do bear a practical and physical meaning, which is, of course, the aim of simulation work.

In all the systems considered, it was found necessary and desirable to impose constraints on the maximum and/or minimum values that certain system parameters could take. The purpose of this was two-fold; firstly, the results obtained would be of practical significance, and secondly, it was found necessary to impose these constraints in order that the resulting systems be stable. The latter is discussed thoroughly in the next chapter.

Limits were therefore placed on the maximum absolute value that f_i, the controller parameters, could assume. Also, the maximum and minimum permissible changes in f_i, i.e. Δf_i, were set. The former was to avoid too large transients to occur in the system, and the latter to prevent the elements of the controller sensitivity matrix, $\frac{\partial \underline{G}}{\partial \underline{f}}$, to become very large or even infinite as the case would be if Δf_i were very small or zero upon updating that matrix.

The values of these constraints for the three system configurations investigated are given in table (4.6.1) below.

	$\lvert f_i \text{ max.}\rvert$	$\lvert f_i \text{ max.}\rvert$	$\lvert f_i \text{ min.}\rvert$
Feedback Path Controller system	10	0.1	10^{-8}
Open Loop Controller system	5	0.1	10^{-8}
Forward Path Controller system	10	0.1	10^{-8}

For a practical system, there would have to be other constraints such as the maximum permissible input to the plant, the maximum rate of change of the plant output and so on. Here these were not considered necessary because the unconstrained variables did not exceed any reasonable values.

4.7 Adaptive Scheme

In chapter three (section 3.4) it was shown that various schemes of adaptation could be chosen, and that this was entirely at the designer's choice. The three adaptive configurations considered here used the same scheme for the order of adaptation, updating and prediction. Figure (4.7.a) illustrates the scheme adopted. Second order adaptation was selected for the systems, and so the cycle time required to completely update the controller sensitivity matrix was $4\Delta t$. This follows from the adaptive scheme of Figure (4.7.a).

Here one controller parameter was changed, i.e. f_i, and at a time Δt later the ith column of the controller sensitivity matrix updated. Then, in order to make full use of the recent updating, both controller parameters were changed, and the process was repeated again.

Note that the vector $\underline{b}$ is estimated at every Δt time interval irrespective of whether it was $\underline{f}$ or f_i which was altered. Had the vector $\underline{b}''$ been taken into account, then it could have been estimated just after $\underline{f}$ was changed by substituting all the values of the other parameters in the adaptor equation. For adaptive systems with fairly low frequencies of adaptation, it is advisable to estimate $\underline{b}''$ along with $\underline{b}'$ also.

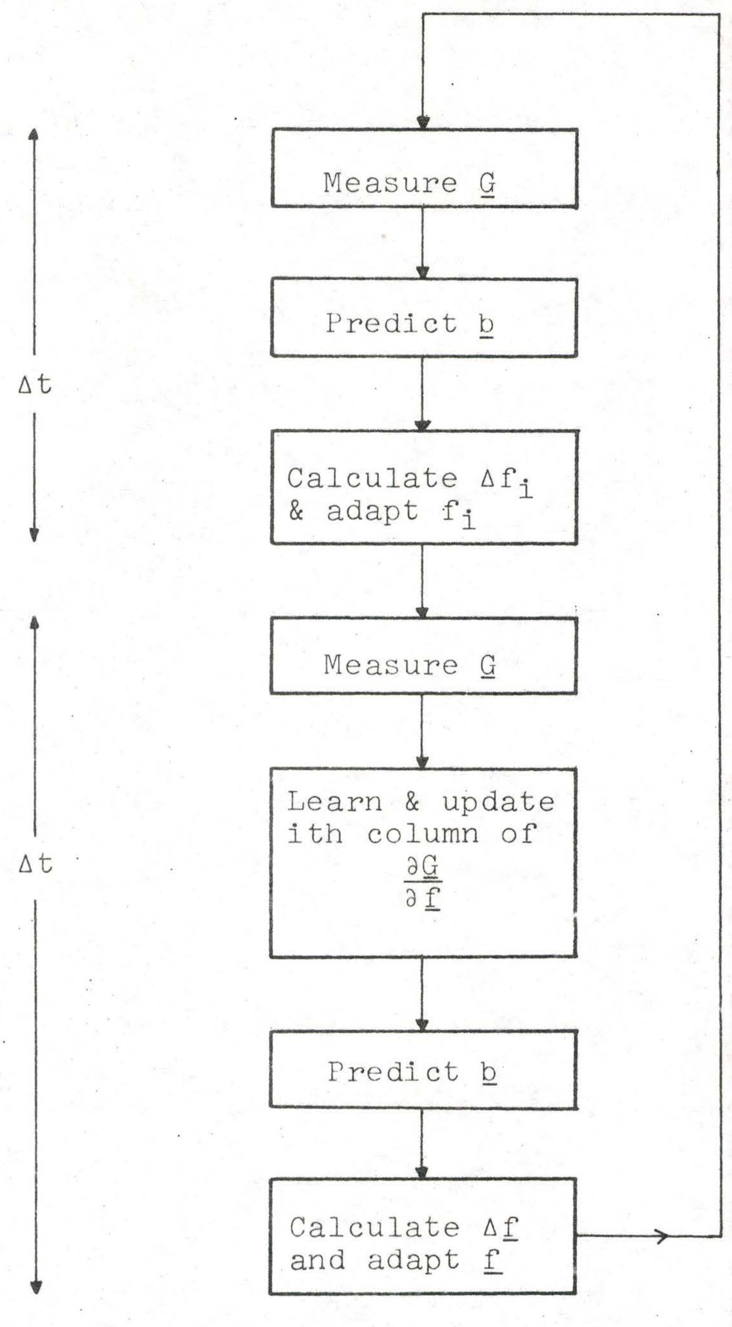


FIGURE (4.7.a)

(Adaptive Scheme)

N.B. Measurements of $\underline{G}$ are spaced at time intervals Δt apart. The cycle time to go through the steps shown above is therefore $2\Delta t$.

CHAPTER 5

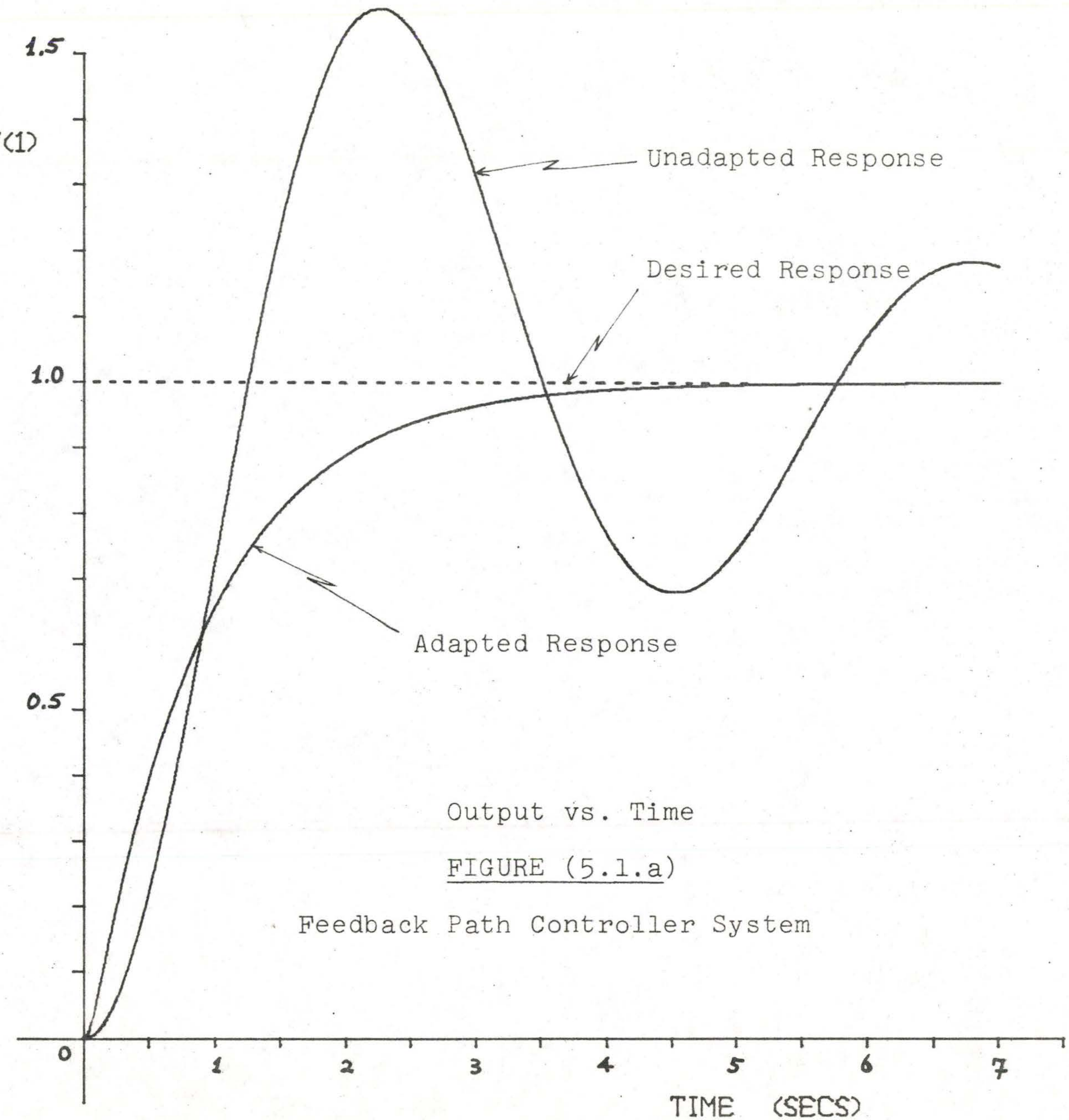
The Vector Cost Function Results

Chapter five contains the results obtained for the systems investigated, and also partial conclusions regarding the effectiveness of the algorithm chosen. The graphs plotted are: output vs. time, phase-plane diagram, cost function vs. time. A comparison of the results obtained with different controller configurations is shown in appendix (IV).

5.1 Feedback Path Controller System

The results plotted here are those of the system described in section (4.1), and where the plant output vector is required to move from state $\underline{y}^t = [0, 0]$ to state $\underline{y}^t = [1, 0]$ so as to minimize J, the integral cost function.

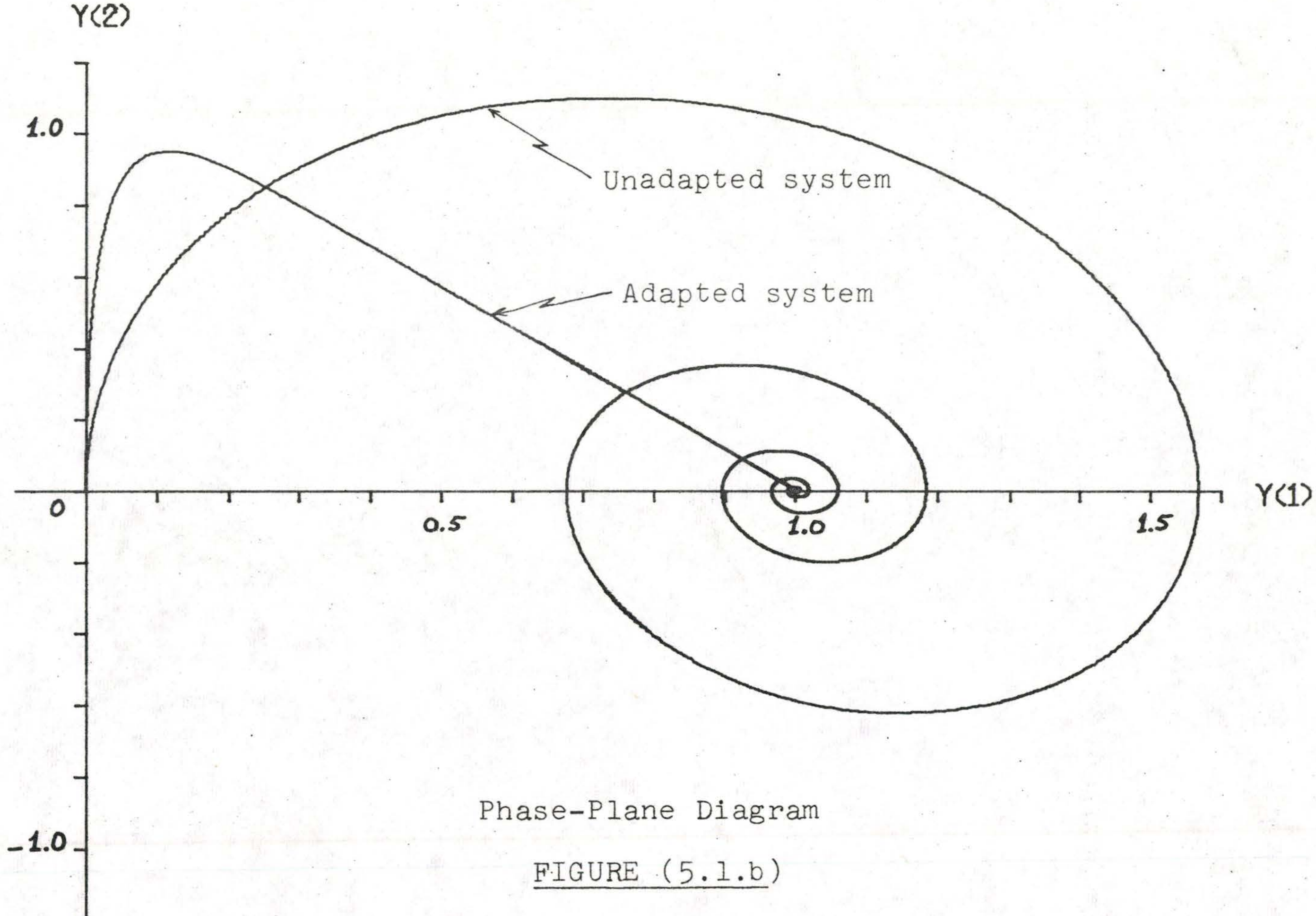
The adapted and unadapted plant responses are shown in Figure (5.1.a) on the next page. It is immediately obvious from these responses that the algorithm adapts well, and in a fairly short time too. The response of the unadapted system is very underdamped and has a first overshoot of 56.8 per cent, and then lies within five per cent



Output vs. Time

FIGURE (5.1.a)

Feedback Path Controller System



Phase-Plane Diagram

FIGURE (5.1.b)

Feedback Path Controller System

of the final value after 11.71 seconds. On the other hand, the adapted system's response shows no overshoot and has a settling time of only 2.71 seconds. Further, after 7.00 seconds the system has adapted to an error of less than four parts in ten thousand, whereas the unadapted system has not reached such a value after 25.0 seconds.

This is a very good result considering the fact that adaptation to a step function is required and that the algorithm had no knowledge of the plant before hand. Also, the initial values of the parameters were completely arbitrary, and not optimum by any means. Figure (5.1.a) shows conclusively that the system did learn fairly quickly the effect of the controller parameter changes, and did change them in the right direction in spite of all transients present and especially those after discrete adaptive steps.

On considering the phase-plane diagrams, Figure (5.1.b), for both systems, it is to be noted that the adapted system's output speed does not exceed that of the unadapted one. This is due to a number of reasons. Firstly, equal weight was assigned to the output and its time derivative in so far as adaptive changes were concerned. This meant that the output position and its time derivative were considered equally important. Secondly, the maximum controller parameter magnitude was limited - here it was

10 - and so the controller output, once the parameters had reached their maximum values, depended entirely upon the forcing function to the controller. Thirdly, the controller input was the error between the desired and the actual plant outputs, and this was a time decreasing function with a final value of zero. Consequently, since both the forcing function and the controller parameters were bounded, the output of that unit was bounded too. This implies that the plant output speed could not rise above a certain limit, unless the plant were itself extremely non-linear. It will be pointed out later that such effects are, in fact, desirable.

During the response the cost was measured, and Figure (5.1.c) shows the variation in cost with time for both adapted and unadapted systems*. As is expected in this case, both curves tend to a constant value and the cost of the adapted system has a final value of 0.501 and that of the unadapted system is 1.12. This is a 55.3 per cent improvement in cost which is fairly good in view of the fact that no knowledge of the plant was assumed; partial knowledge is acquired as time proceeds, and this is later used to greater efficiency in calculating the adaptive changes necessary, but initially the plant appears as a black box to the controller.

* Values of all graphs shown in this thesis are found in appendix (II).

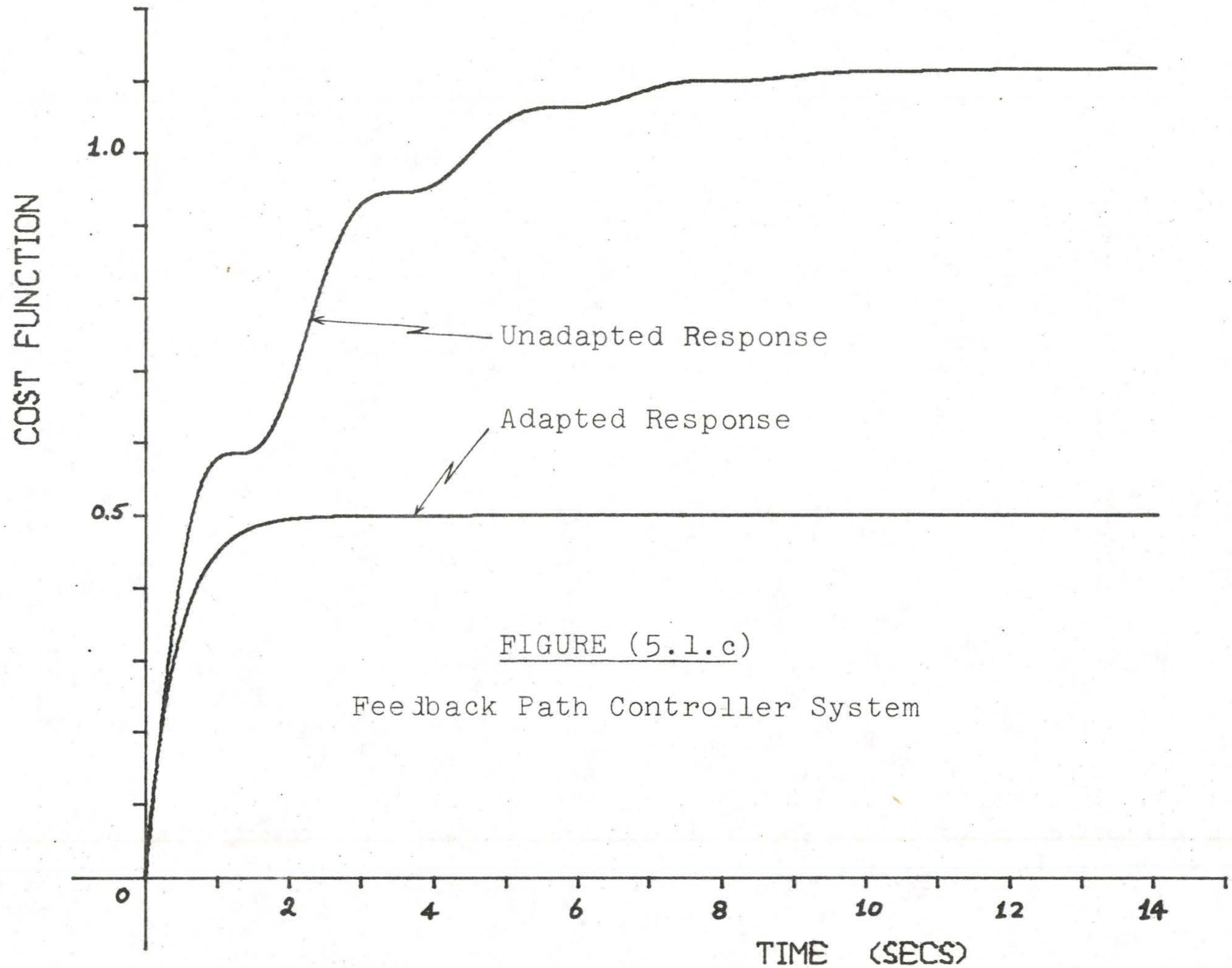


FIGURE (5.1.c)

Feedback Path Controller System

In the later sections of this chapter [(5.4), (5.5) and (5.6)] some important aspects of the Vector Cost Function are investigated using the above feedback configuration.

5.2 Forward Path Controller System

These results are those obtained from the system of section (4.2). This configuration is the one which yielded the least satisfactory results since they cannot be considered truly adaptive. The reason is that the input vector value is $\underline{r}^t = [1.1,0]$ and the maximum controller parameter magnitude is 10.0. Hence, in the final steady-state, the value of $\underline{z}$, the controller input vector function, is given by $\underline{z}^t = [0.1,0]$, because of the configuration chosen. Therefore, f_1 has to assume its maximum value in order that the plant output reaches the desired output value.

Figure (5.2.a) shows the response of the plant in the adapted and unadapted modes. It was found necessary to set the controller parameter limit to 10.0 in order that the system's response be a stable one. In view of the constraints imposed, this system cannot be considered as being totally adaptive. But, on the other hand, it does show the property of being able to change the controller parameters in the right direction so as to minimize $\underline{G}$, the vector cost function, and hence drive the system

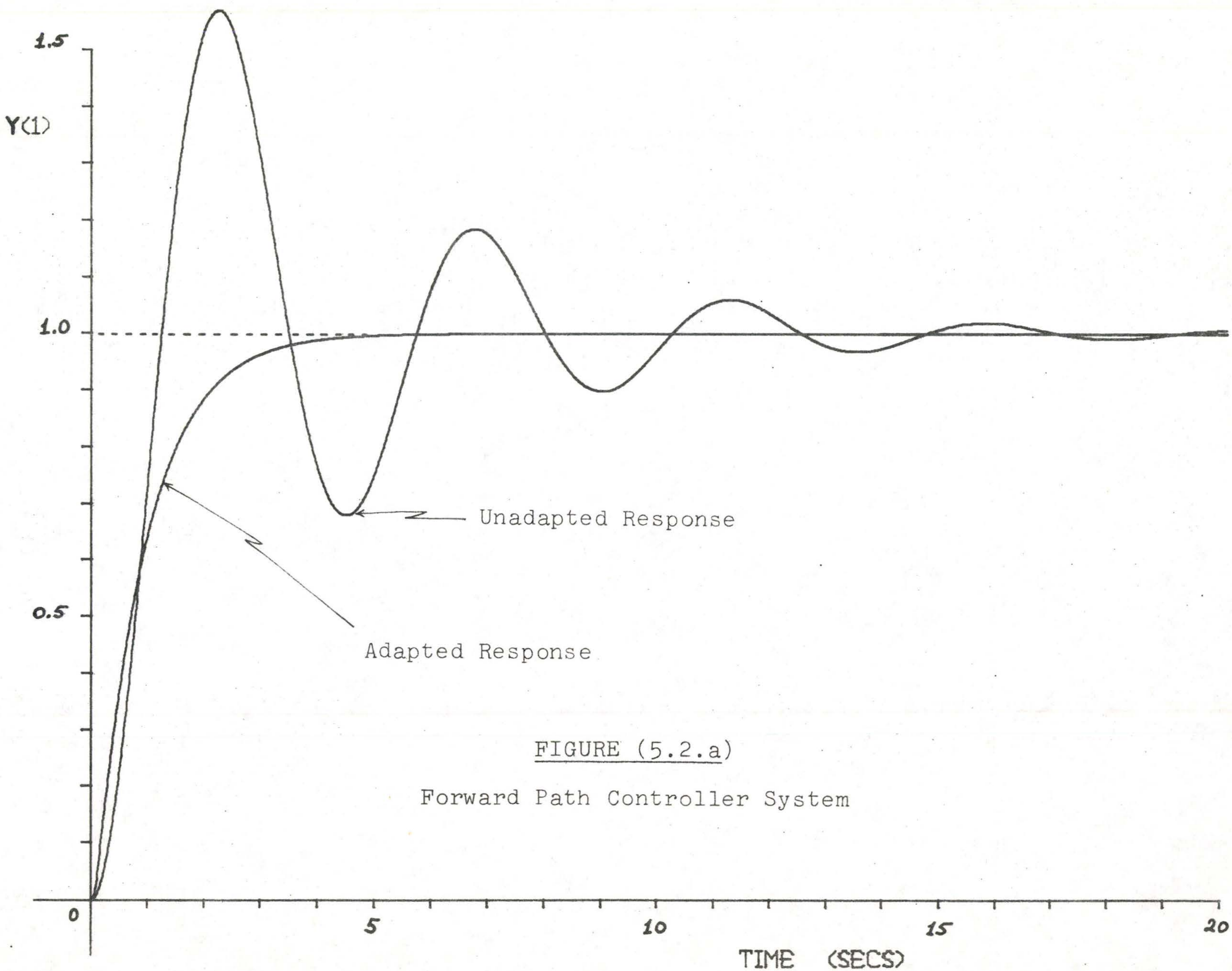
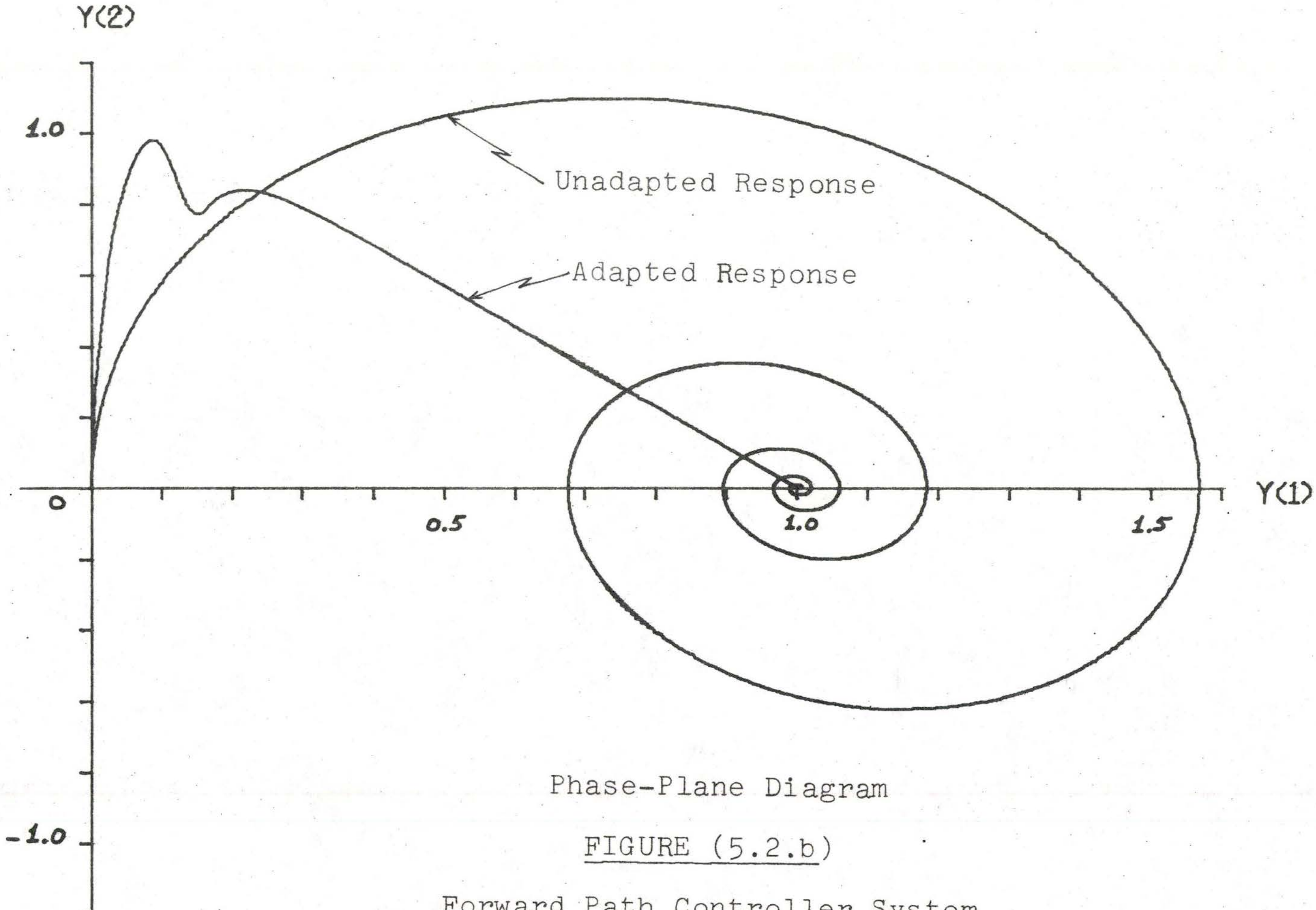


FIGURE (5.2.a)

Forward Path Controller System

towards the desired value. This is a very important property which cannot be overlooked especially in the vector cost function algorithm, for it reflects the ability of the algorithm to learn the sensitivity of the plant to controller parameter changes and hence to minimize $\underline{G}$. In other words, it is the criterion which determines whether the system is adaptive at all. These results have been given here since they do show this very property of the vector cost function.

The phase-plane diagram of Figure (5.2.b) indicates an interesting feature of the adaptive system. In the last chapter, it was pointed out that the methods of estimating the future state of the system, i.e. estimating $\underline{b}'$, and of learning and updating the matrices, here $\frac{\partial \underline{G}}{\partial \underline{f}}$, were arbitrary and depended upon the control engineer's choice. The valley occuring in the adapted system's phase-plane diagram between the output values of 0.1 and 0.2 is a consequence of this arbitrariness in choice. It occurs because of the specific prediction and updating schemes used were simple and did not involve the storage of numerous past values in order to calculate the new ones. In other words, this basically means that the system, though it had been adapting the plant towards the desired output, had no real means of knowing this fact had it not been for the controller sensitivity matrix. But, after slowing down, it does readjust itself and



Phase-Plane Diagram

FIGURE (5.2.b)

Forward Path Controller System

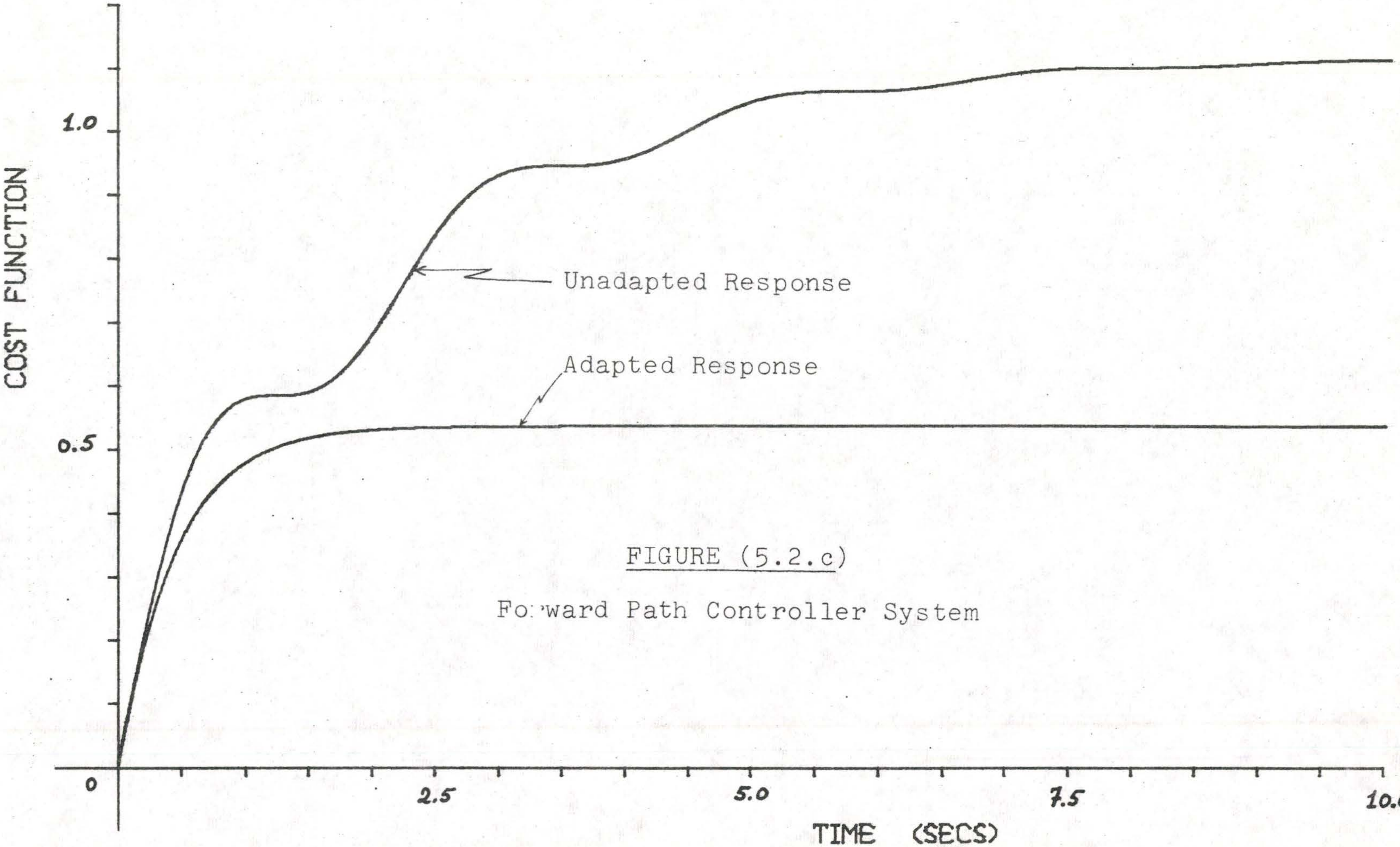


FIGURE (5.2.c)

Forward Path Controller System

drives the output towards the desired value.

It cannot be guaranteed that another method for estimating the future state of the system and updating the controller sensitivity matrices would result in the system being driven smoothly and more efficiently towards the desired state. In fact, it will be seen in the next section that the same scheme used gives a different type of response for the open loop configuration. The best methods can only be deduced if some knowledge of the plant is available or arrived at by previous runs using the adaptation scheme.

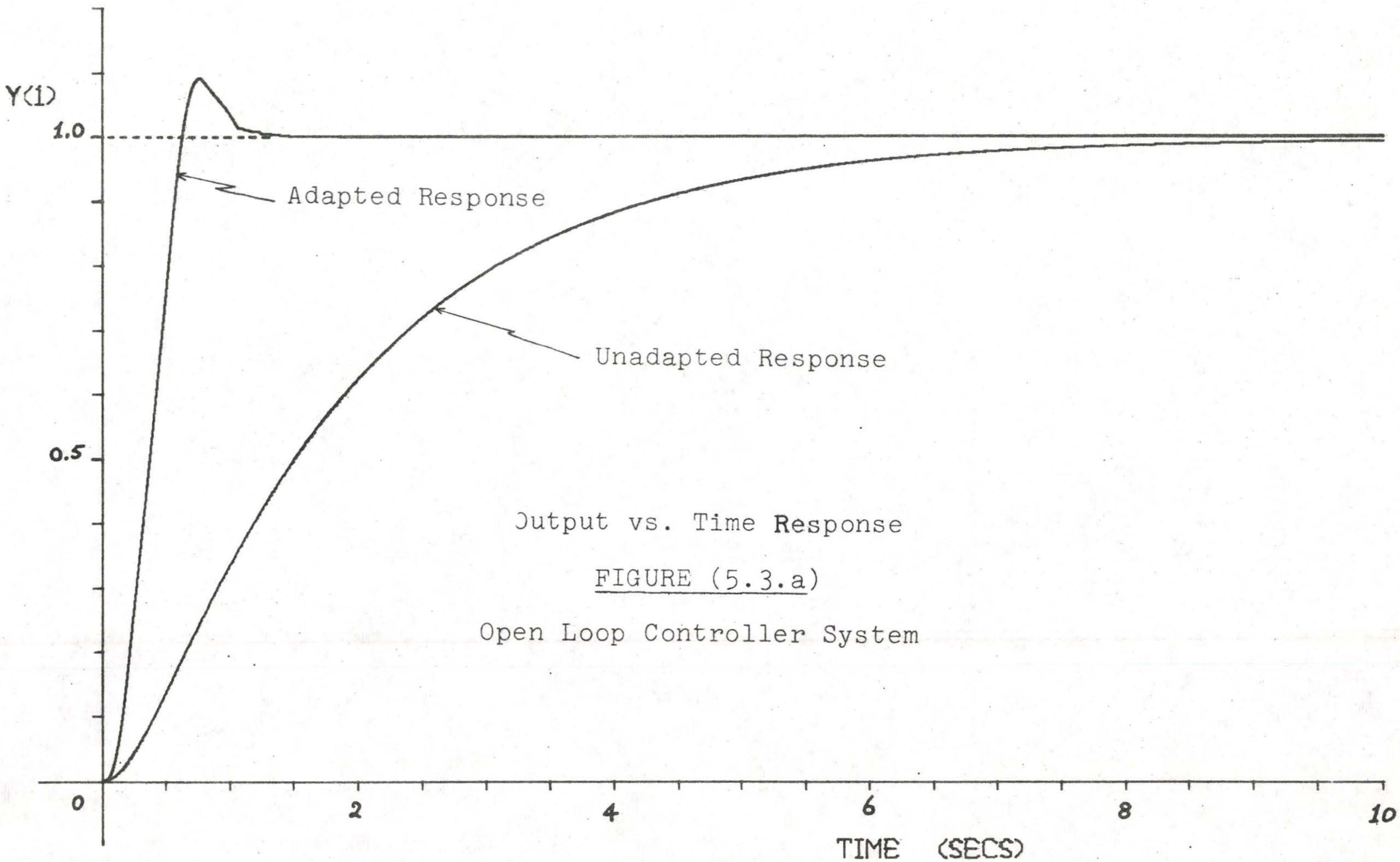
Finally, Figure (5.2.c) shows the difference in cost resulting from both the adapted and unadapted systems. The final value of the cost is 0.538 for the adapted plant and this is a 58.2 per cent improvement from that of the unadapted plant.

5.3 Open Loop Controller System

The results presented in this section are those of the system of section (4.3). Again here the system had to respond to a step input i.e. move from state $\underline{y}^t = [0,0]$ to state $\underline{y}^t = [1,0]$.

It is fairly common to find adaptive control systems of this type in which the controller is placed in series with the plant and preceeding it. This configuration is attractive in the sense that it is simpler than the feedback one. As can be seen from Figure (5.3.a), it yielded a very good result. After 0.934 seconds the plant output has settled to within five per cent of the final value, and after 1.49 seconds the error between the plant output and the desired output is less than four parts in ten thousand. Contrary to the response of the other two systems, the open loop configuration yielded a single overshoot of nine per cent before quickly settling to the desired state.

The unadapted plant response is very overdamped and takes about 5.5 seconds to be within five per cent of the final value and 17.4 seconds to be within an error of four parts in ten thousand. Of the configurations considered, the open loop controller was the one which drove the plant the fastest to the desired output state. This is attributed to the fact that the controller input was a constant for this system, whereas those of the forward path and feedback path controller systems were the errors arising from the difference between the desired output state and the plant output state, and consequently in the latter case these functions were decreasing functions of time. With a constant forcing function to the open loop controller, the controller output reaches a larger value which is maintained at this point if necessary. Since the other



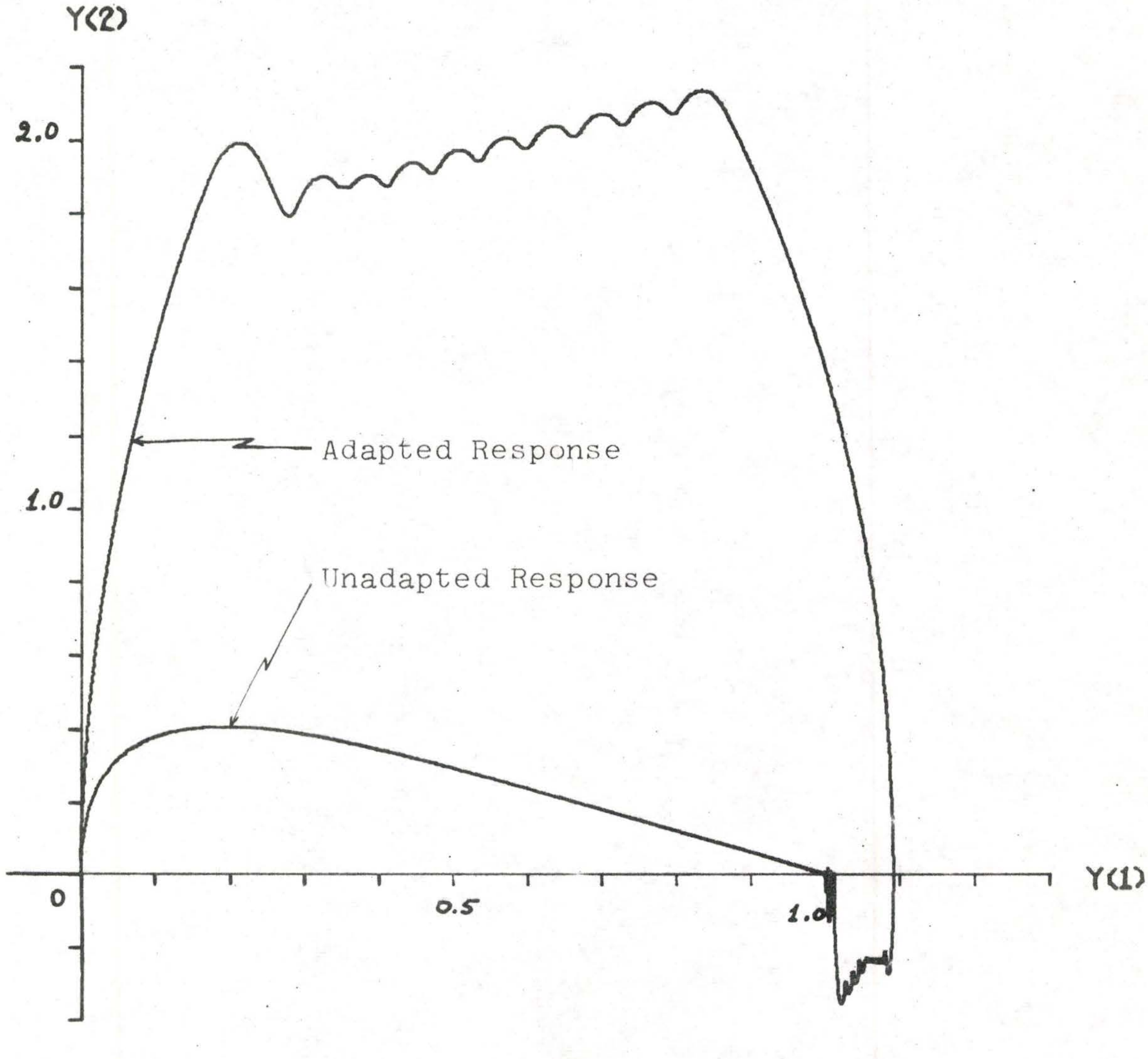
Output vs. Time Response

FIGURE (5.3.a)

Open Loop Controller System

two systems had time decreasing controller forcing functions and since the maximum permissible changes in the controller parameters were limited, it seems that the increments in the controller parameters could not compensate sufficiently for the decreasing input in order to maintain the controller output at a steady level. The result was that the controllers in the feedback systems had less influence upon the systems with such forcing functions as the plant output approached the desired state. This is evident from the fact that when the error is zero for these configurations, the controller output is zero whatever the values of the controller parameters. Of course, this is only true for the controllers chosen and their respective forcing functions. Had an integrator been used in the controller, then the above would not hold.

Figure (5.3.b) shows the phase-plane trajectories obtained for the open loop adaptive system and the unadapted plant response. The latter is a well known type of phase-plane trajectory, whereas the former is rather unusual. The methods used for estimating the vector $\underline{b}'$ and updating the sensitivity matrix had probably important roles to play here. The estimation of $\underline{b}'$ was made on the last ten measurements of $\underline{G}$, but more important was the updating probably. The greater the



Phase Plane Diagram

FIGURE (5.3.b)

Open Loop Controller System

number of stored values of $\frac{\partial \underline{G}}{\partial \underline{f}}$, the more accurate is the estimation of the sensitivity coefficients with respect to the controller parameters and the less fluctuations there should be in them.

The phase-plane diagram of the adapted response indicates that as the system moved from state $\underline{y}_o$ (where $\underline{y}_o^t = [0,0]$) to state $\underline{y}_f$ (where $\underline{y}_f^t = [1,0]$), the updating was taking place in such a way as to speed up the plant output towards the final desired state. Then came a certain rate of output change which was too large for equal weighting of the output and its time derivative - equal weighting was assigned to the output and its time derivative in the open loop system. Updating then took place in such a manner as to prevent the system from speeding too much. Between the initial acceleration and the final deceleration, there is a region where the speed fluctuates about a more or less constant value. The constant speed level results from the equal weighting of the state variables, but the fluctuations are probably due to the arbitrary prediction and updating methods selected.

This means that two conflicting efforts were taking place at the same time: the output attempting to reach the desired value of 1.0, while the rate of

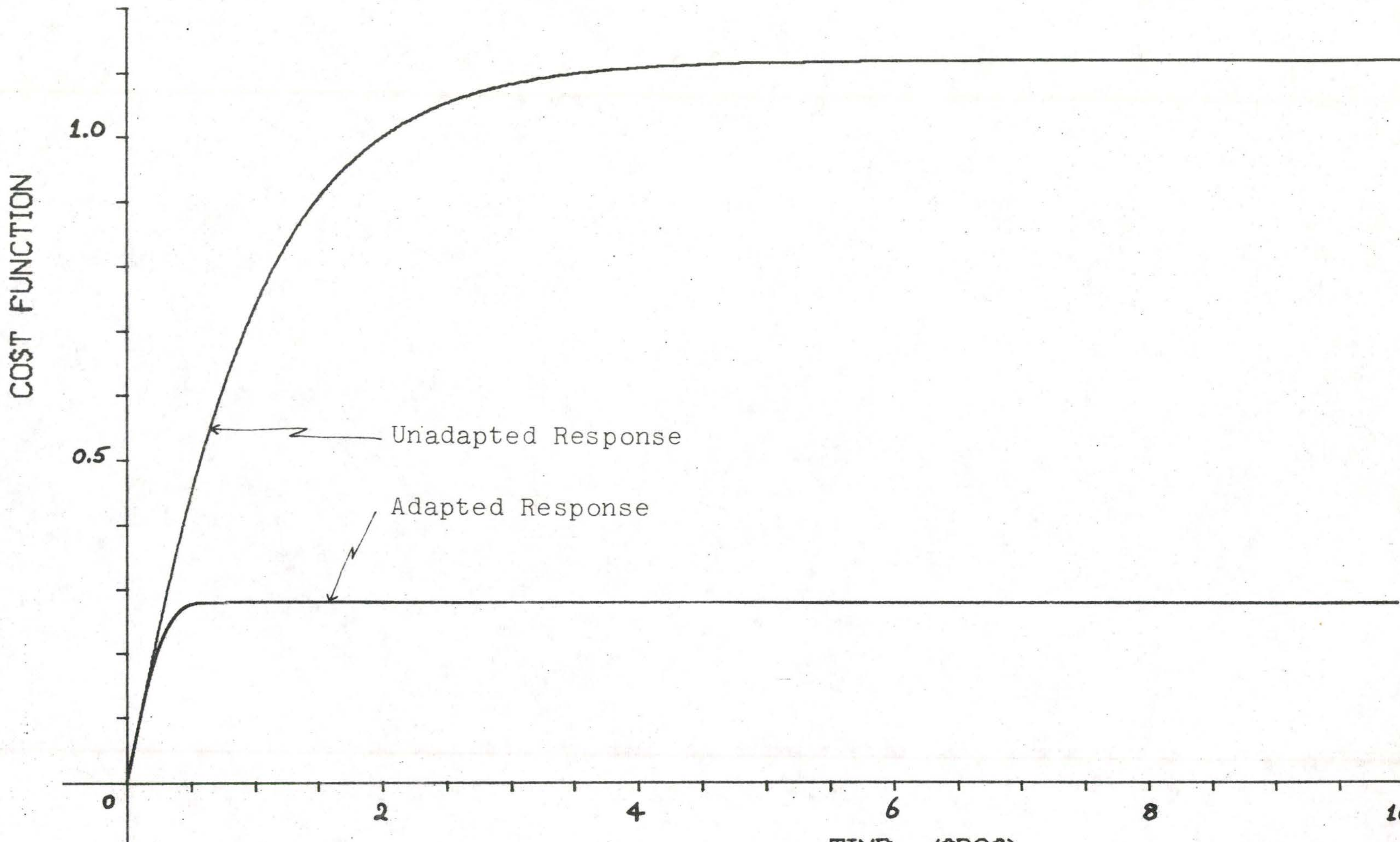


FIGURE (5.3.c)

Open Loop Controller System

change of the output was trying to stay at zero or near it. Hence, when the output and its time derivative have about equal weight, small fluctuations result. In the end, when the final output state has been nearly approached, the system decelerates until it finally reaches the desired value. It is to be noted also that the same phenomenon occurs after the maximum overshoot has taken place and the system is trying once more to reach the desired output state. Near the desired state there are some more speed fluctuations, but these could be due to the fact that $\underline{G}$ is very small then and that errors could consequently have a dominating effect.

The resulting improvement in the cost for the open loop system is apparent from the Output vs. Time response, and Figure (5.3.c) confirms this fact. The final value of the cost function for the adapted system is three to four times less than that of the unadapted plant response. This is indeed a very good result.

5.4 Effect of Constraints on the Controller Parameters

The following sections show the effects of the arbitrary initial parameters upon the adapted response of the plant, and also the effect of some other parameters of the vector cost function. The feedback path controller system as described in section (4.1) was the configuration used for the investigation.

(A) Effect of $|f_i \text{ max.}|$

The effect of varying the maximum allowable controller parameter magnitude was investigated using the feedback controller system described in section (4.1). The initial conditions imposed upon the system were identical to those described in that section except for $|f_i \text{ max.}|$ which was made to take up different values.

Figure (5.4.a) shows the responses obtained for values of $|f_i \text{ max.}|$ equal to 100 and 1000, and also that of the unadapted plant. The response due to the larger value of the maximum controller parameter magnitude shows that the system is being driven more quickly to the desired state. This is to be expected since a large $|f_i \text{ max}|$ allows a larger controller output. Hence, the plant input can in turn be made larger so that the system can speed up faster to the desired state.

But, the controller parameter magnitude cannot

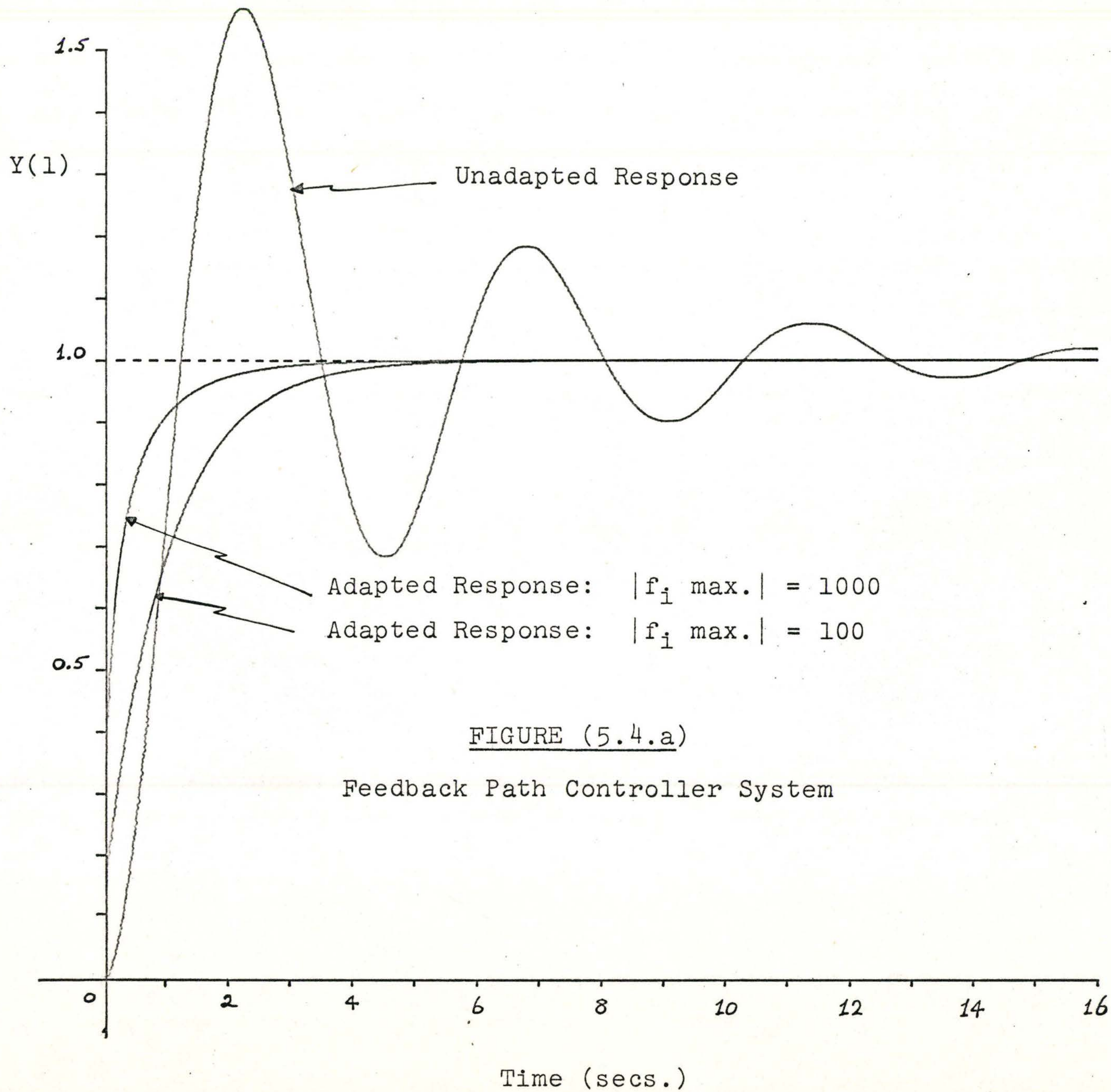


FIGURE (5.4.a)

Feedback Path Controller System

be increased indefinitely, for eventually the resulting system would become unstable. Here too, stability is of prime importance and the $|f_i \text{ max.}|$ that can be used would depend on the plant which is being adapted.

(B) Effect of $|\Delta f_i \text{ max.}|$

There are numerous ways in which constraints can be imposed upon the changes made in $\underline{f}$, the controller parameter vector. In chapter four a limit was placed upon the maximum and minimum changes which could be made in f_i. Here, the effect of placing a different constraint on $|\Delta f_i \text{ max.}|$ was investigated.

The change applied to the controller parameters was made to be two per cent of the calculated changes, and any change smaller in magnitude than 10^{-8} was not made.

i.e.

$$\underset{\text{(applied)}}{\Delta f_i} = \frac{2}{100} \underset{\text{(calculated)}}{\Delta f_i}$$

and, if $\Delta f_i < 10^{-8}$; $\Delta f_i = 0$.

The system investigated was that of section (4.1) with identical initial conditions to those described in that section. Figure (5.4.b) shows the adapted and unadapted responses obtained. On comparing this adapted plant response to the one of Figure (5.1.a), it is obvious

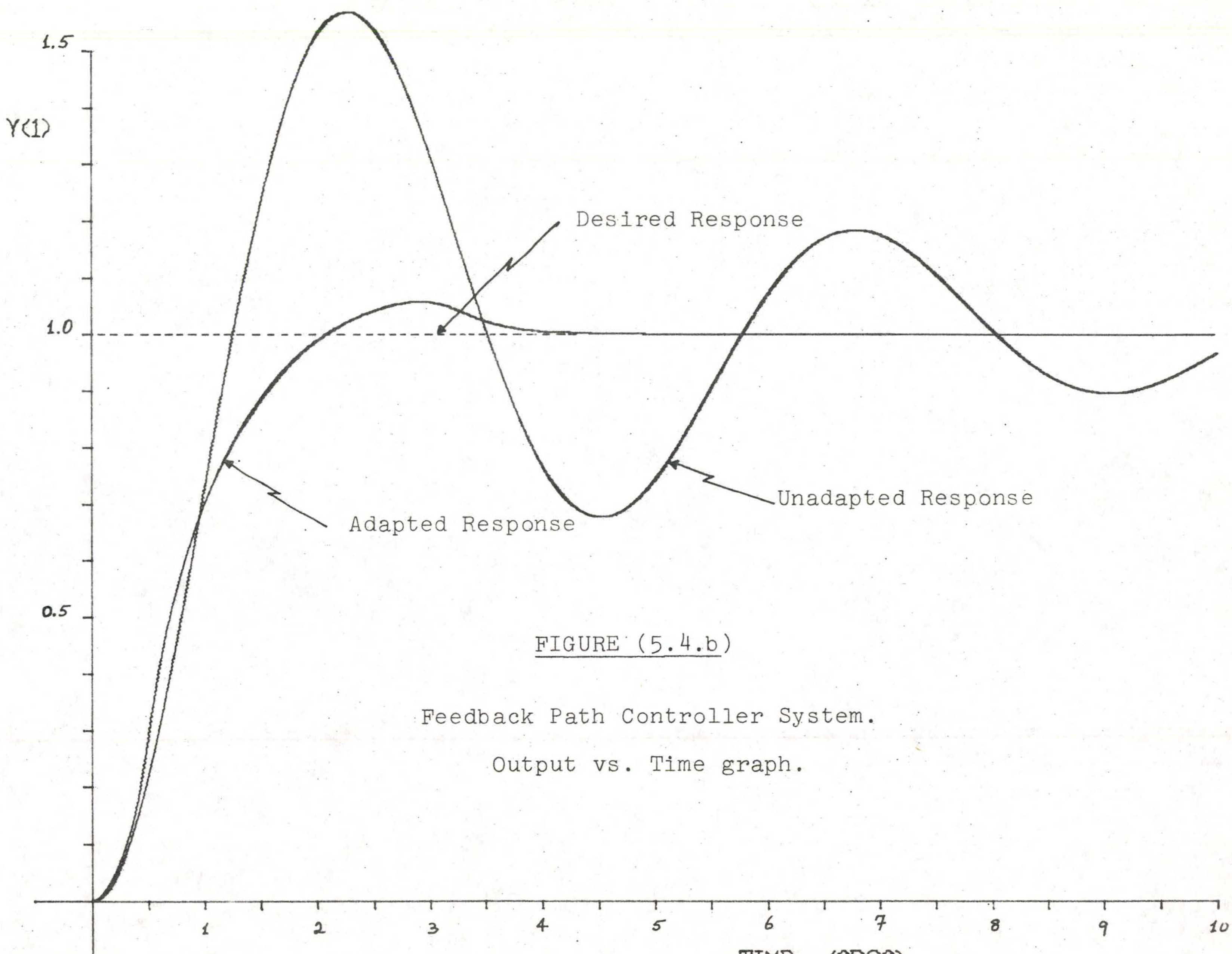


FIGURE (5.4.b)

Feedback Path Controller System.

Output vs. Time graph.

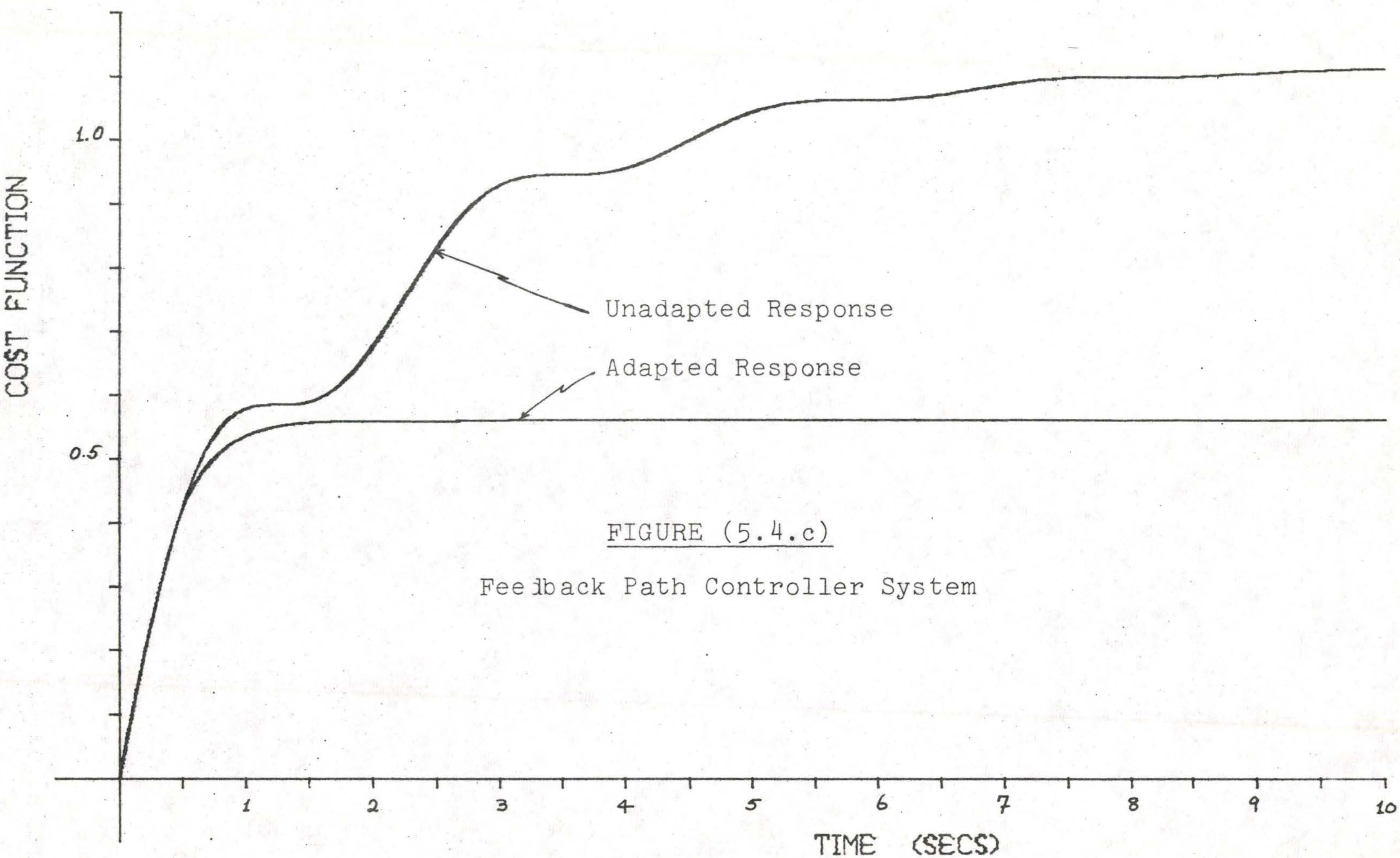


FIGURE (5.4.c)

Feedback Path Controller System

that here too the adapted response can be altered by varying the way in which changes are made to the controller parameter vector $\underline{f}$. In fact, the response of Figure (5.4.b) is seen to reach the desired state faster than the one of Figure (5.1.a), and further it has an overshoot. This improvement is due to the larger applied changes which were initially made in the controller parameters by the above technique. Hence, the system is forced to move faster through the larger plant input which results, and, after an overshoot of about six per cent, it finally settles to the desired plant output state. Figure (5.4.c) shows the resulting value of the cost obtained.

This particular way of applying changes in $\underline{f}$ is probably better than the one described earlier in this chapter, since the applied change bears a direct relation to the one calculated from the adaptor equation. In fact the controller sensitivity matrix was found to be more accurate than that obtained using the other set of constraints, and consequently the response was improved. Other types of constraints can also be imposed on $\Delta\underline{f}$ depending on the goal to be reached.

5.5 Effect of initial controller matrix.

In order to use the Vector Cost Function algorithm to control the response of an unidentified plant, one has to assume arbitrarily the values of certain parameters of the Adaptor Equation. Then, as the system evaluates the effect of the controller parameters upon the plant response, these initial values are modified using the learning process described in chapter three. One of the important parameters initially assumed is the controller sensitivity matrix.

As the response proceeds, the controller parameters are altered and so is the sensitivity matrix. The appropriate weighting is given to each controller parameter, and the calculated changes in the parameters become more accurate as the number of adaptive steps increase.

The system used to investigate the effect of the initial controller sensitivity matrix was the one described in section (4.1). The values of the initial parameters were identical to those described except for $\frac{\partial \underline{G}}{\partial \underline{f}}$ and W. The applied change in $\underline{f}$ was made as described in section (5.4), (B).

Figure (5.5.a) shows how the response of the plant is affected using two different initial controller sensitivity matrices, $\underline{\underline{A}}$ and $\underline{\underline{B}}$.

$$W = 0.65$$

$$\underline{\underline{A}} = \begin{bmatrix} 0.2 & 0.01 \\ 0. & -5.0 \end{bmatrix} \qquad \underline{\underline{B}} = \begin{bmatrix} 1.0 & 0.04 \\ -0.03 & -1.0 \end{bmatrix} \qquad \text{..........(5.5.1)}$$

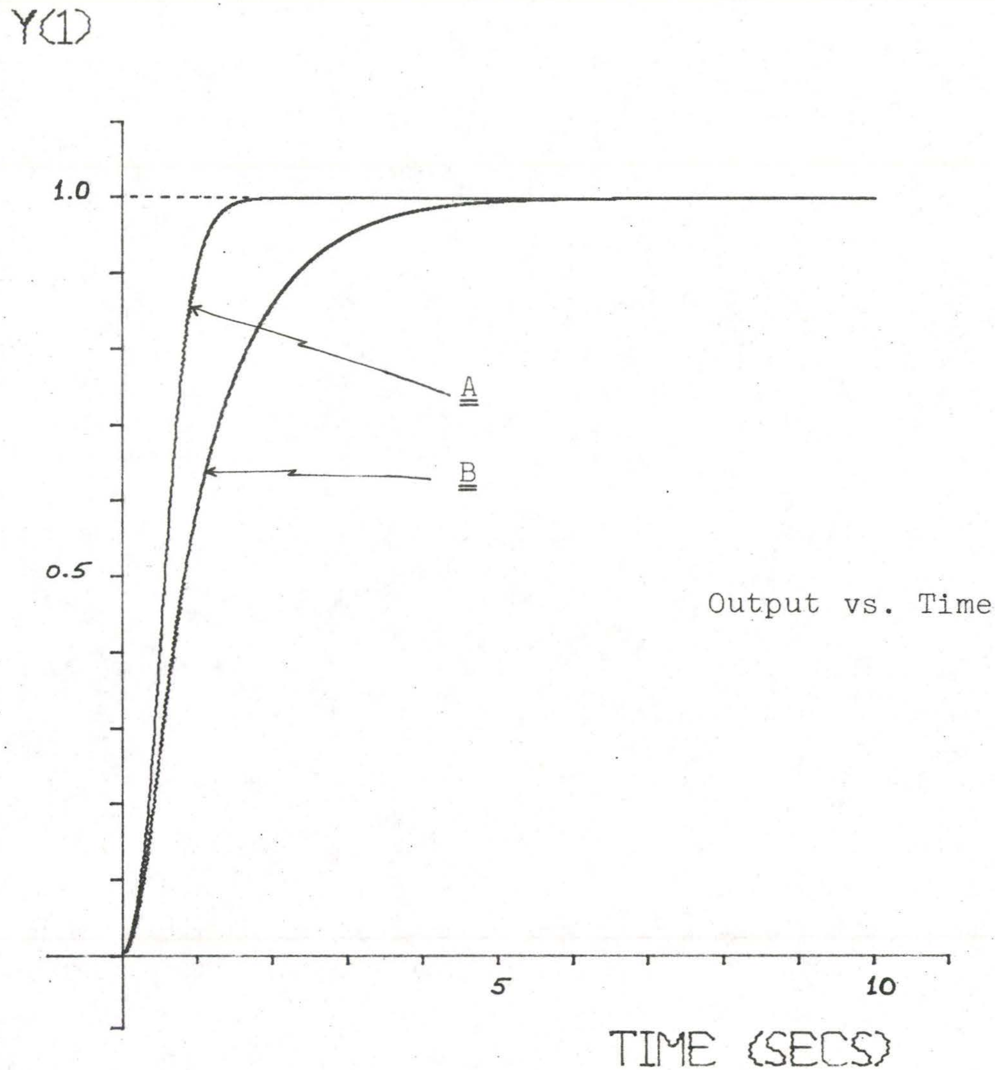


FIGURE (5.5.a)

Feedback Path Controller System

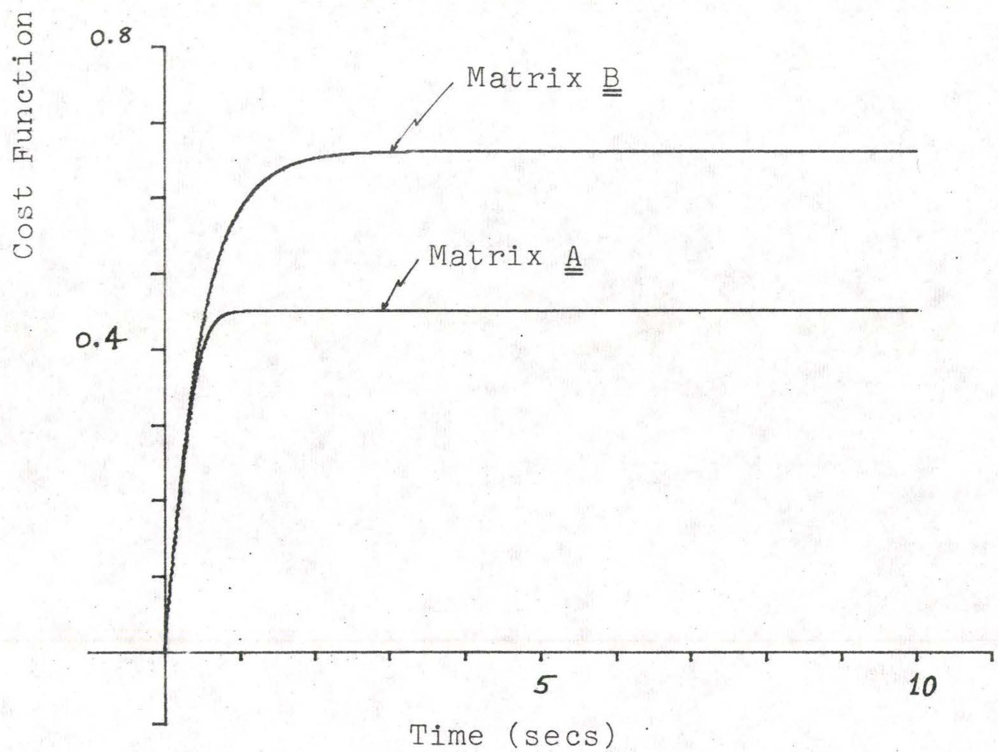


FIGURE (5.5.b)

Feedback Path Controller System

Though there is a difference resulting from the initial arbitrariness in both responses, the final value of the cost function is still better than that of the unadapted plant. Figure (5.5.b) shows the resulting variation in cost. The final value of the cost function of the unadapted plant is 1.12 as compared to 0.452 and 0.663 for the adapted systems using the matrices $\underline{\underline{A}}$ and $\underline{\underline{B}}$ respectively.

There exists an optimum controller sensitivity matrix for this system, but in order to evaluate it, full knowledge of the plant must be acquired. Then, working backwards from the final desired state to the initial state, one would eventually be able to evaluate it. But no such method can be used here since the whole purpose of the algorithm is to adapt the plant without having to identify it.

As a result, there are a large number of responses that can be obtained using various initial arbitrary matrices. The final value of the controller sensitivity matrix is not the optimum initial value of that matrix for the system. This is because here adaptation to a transient is required <u>i.e.</u> a step. In this case, the value of the matrix sensitivity coefficients are continually being changed in order to bring the system to the desired state. The weighting given to the various con-

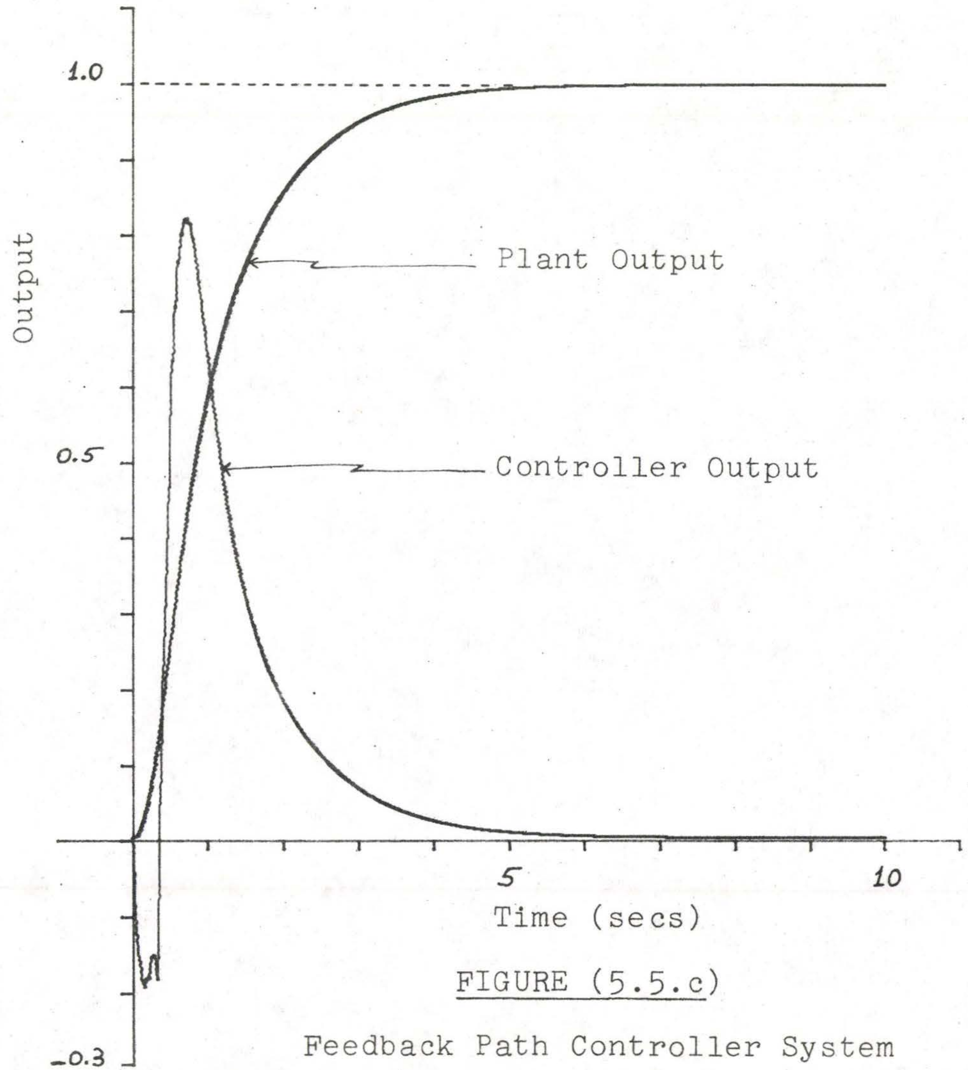


FIGURE (5.5.c)

Feedback Path Controller System

troller parameters during the initial part of the response, when the system is required to speed up to the desired state, is not the same as that in the final part of the response when the system must slow down to the final desired state. Figure (5.5.c) illustrates this point. To speed up the system initially, positive feedback is applied and later to slow it down negative feedback is applied - matrix $\underline{\underline{B}}$ was used as the initial value of the controller sensitivity matrix.

The next section is a discussion of the various points investigated in this chapter, and also some other factors by which the adapted response could be modified.

5.6 Discussion

(a) There are numerous ways by which the adapted response of a system using the Vector Cost Function algorithm could be modified. Also, there are many techniques which can be used to update the controller matrices and to estimate the vector $\underline{b}$. This is indeed a great freedom which the control engineer can take advantage of. On the other hand, the performance of the system depends to a great extent on the choice made. In this thesis, the objective was not to find the methods which could be used to obtain the best possible response, but rather to investigate the various aspects of the above mentioned algorithm.

As mentioned in section (5.5), the initial value of the controller sensitivity matrix is one of the factors which affect the transient response of the adapted plant. For such responses, the controller matrix is in fact a trajectory in parameter space originating from the initial arbitrary value to a final point. There is an infinite variety of starting points because of the inherent initial arbitrariness of the algorithm, and all the trajectories later converge to the same point in space. In practice, because of errors in measurements and the limited accuracy obtainable, the final value of the matrix would be a set of points rather than a single one.

In the case of adaptation to plant parameter variations and external disturbances, the final value of the controller sensitivity matrix would be the optimum starting value. The problem in this case is to find the appropriate value of the controller parameter vector at each time interval Δt which will compensate for the variations occuring within the system. This reasoning assumes that the properties of the disturbances do not change drastically with time, but are stationary or slowly varying.

Another factor which affects the response is the initial choice of the controller parameters. Here again the engineer can improve the system's performance by using knowledge acquired from previous runs, or by using initial values which are most suitable if no knowledge of the plant is yet available. For example, a desired choice for the initial values of the feedback path controller parameters is zero. In this case, the arbitrariness in choice of these parameters does not affect the system's response initially since the feedback signal would then be zero; later, better starting values can be used based on the experience acquired previously.

The constraints imposed upon the controller parameter vector $\underline{f}$ and the applied $\Delta\underline{f}$ also alter the system's performance as previously discussed in section (5.4). Here the choice of these constraints has to be made in the light of stability considerations. The initial choice should be made so as to minimize the risk of the system running away to infinity. Afterwards they can be relaxed and improved as more knowledge of the system is acquired.

In the present form of the Vector Cost Function, the values of the controller parameters are calculated by predicting the value of $\underline{G}$ at the next adaptive step to be $\underline{\theta}$, the null vector. By predicting $\underline{G}(t + \Delta t)$ to be a fraction of $\underline{G}(t)$, it would thus be possible to apply smaller changes in the controller parameters if these had previously been found to be too large.

$$\underline{G}(t + \Delta t) = \xi \underline{G}(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.6.1)$$

where, ξ = fraction of $\underline{G}(t)$ predicted for $\underline{G}(t + \Delta t)$

In the proposed form of the algorithm, the value of ξ is zero. An extra design parameter can therefore be used by making ξ greater than zero.

(b) The last point worth mentioning concerns the values of the coefficients of the controller sensitivity matrix. The Adaptor Equation for the systems considered is given by equation (4.1.5) in section (4.1). Thus at a time Δt after a single controller parameter has been changed, one column of the sensitivity matrix can be learnt. Then, the updated value of the matrix can be used to calculate the required changes in the controller parameters.

Appendix (III) contains some typical values as calculated from the algorithm for the systems described in sections (4.1), (4.2) and (4.3), also some other values for the systems described in sections [5.4 (B)] and (5.5).

It is apparent that the values of $\Delta \underline{f}$ calculated from the adaptor equation are too large. Hence, the use of constraints previously described. If such changes were made in the system, instability would result. The large values of the calculated controller parameter changes are in fact due to the small values of the controller matrix sensitivity coefficients which evaluate the system to be less sensitive to the controller parameters than it actually is.

But, it must be pointed out that the constraint on $\underline{f}$ which is described in section [(5.4), (B)] enables the system to calculate the sensitivity matrix more acurately than the other one [described in section (4.6)] does. Stability is an important factor which cannot be overlooked when using the Vector Cost Function algorithm. A criterion which would guarantee the system's stability at each adaptive step would be of tremendous help in overcoming some of the problems associated with this technique; unfortunately, the present stability criteria available cannot be used to guarantee the stability of an adaptive system "without identification".

CHAPTER 6

Conclusions

The conclusions presented here are both of a general and of a particular nature, but they do reflect upon some of the important deductions which were arrived at pertaining to adaptive control strategies using the Vector Cost Function method.

The adaptor equation cannot be applied as it stands in its present form to adapt systems to given transient inputs without the use of constraints, for the resulting systems would almost invariably be unstable. The calculated changes in the control parameters are too large, and unrealistic values of the parameters are soon arrived at. It is therefore necessary to impose constraints on both Δf_i, the change in the controller parameters, and f_i, the value of the controller parameters. In cases where Δf_i is very small or zero, it is necessary to avoid updating the system then. Otherwise, the elements of the sensitivity matrix will suddenly become very large or even infinite. The above also implies that when the error between the actual plant output and the desired plant output does not exceed a certain minimum value, adaptation can still be carried out but not updating since Δf_i will again be fairly small.

A very important feature of the algorithm is that if $\underline{\gamma}$ can be found (see section 3.2), the system will then be definitely driven to a global extremum instead of a possible local one by minimizing $<\underline{\gamma}.\underline{\gamma}>$ in lieu of $<\underline{H}.\underline{H}>$. Other algorithms used for self-optimizing control systems do not have this means of distinguishing between a global and a local extremum. On the other hand, it is not always possible to determine $\underline{H}'$, the optimum trajectory of the system; but, by suitable choice of the cost function, $\underline{H}'$ can be found or deduced usually. A cost function which will always yield $\underline{H}'$ is the ISE (integral of squared error) criterion; here, it is obvious that $\underline{H}'$ is always zero.

Another advantage of this algorithm is that it is relatively simple to implement. The most time consuming operations are those of estimating $\underline{b}$, updating the controller sensitivity matrices and solving the adaptor equation. These can easily be programmed, and a special purpose digital computer used instead of a general purpose one if necessary. The amount of computer memory needed will largely depend on the storage space used in storing past data values, especially those of the vector $\underline{b}$ and of the sensitivity matrices. Also, it will depend on the order of the controller i.e. the number of parameters which have to be changed in the system.

The vector cost function method does change the parameters of the controller in such a manner as to optimize the performance of the plant. This may appear to be a trivial statement, but it must be remembered that the initial values of the controller parameters are completely arbitrary. In other words, it may happen that the values chosen for the controller will drive the system initially in such a manner that would increase $\underline{G}$. But, the algorithm does correct for this as soon as it starts learning about the system and in fact changes the matrix sensitivity coefficients to the correct sign, and finally drives the system towards an extremum value of the cost function. In the case of other adaptive control algorithms without identification, this problem does not exist since the gradient of the cost function is first evaluated and the parameters then changed according to the gradient of the cost function surface; finally, an extremum is arrived at in an iterative fashion. The vector cost function method does not evaluate the gradient directly at the start, but rather, it learns about it as further adaptive steps are taken. So, as can be deduced, if learning methods provide the algorithm with the wrong information, the system would not extremalize the cost function.

This algorithm also attempts to approximate a bang-bang controller for the systems investigated. What was desired was that the systems move from one point in the state space to another one in such a way as to minimize $\underline{G}$. It is well known that a bang-bang controller is the one which will minimize the cost function best in this case. The controllers used in the systems - though they were not of the bang-bang type and were also each different in nature - did perform in a similar fashion [see Figure (5.5.c)]. Of course, much of their performance would depend on how far from optimal were the initial values selected. One would expect that the closer to the optimal are the initial parameter values, the closer to a bang-bang system will the adaptive response be.

The stability and response of the overall system are affected by such factors as the weighting of the past values of parameters with respect to the latest ones. The better the estimation of a variable at a future time, the greater the number of its past values which are required. Further, greater accuracy of prediction implies longer computational time. Accuracy of prediction and speed of adaptation are therefore conflicting requirements. Also, for a better estimate more computer memory is required due to the amount of data which has to be stored. But, as in any other field, here too there is a compromise which has to be reached. In the present work it was found that prediction of the future state of the system to three or four significant figures was sufficient. In practice it is very rare to be able to work to higher accuracies than that, because of errors in measurement. At each adaptive step, the change in the output state was noticed in the second significant figure usually, and so predicting to four significant figures was considered adequate.

Since real time simulation was not used in this work, the length of the computational operations between successive adaptive steps could take as long as required. In practice, the compromise between computational time and frequency of adaptation must always be kept in mind when designing an on-line adaptive control system.

Some of the difficulties of the vector cost function method can be partially overcome by a proper choice of the initial parameter values. Some knowledge of the plant is always available in practice and this can be used to advantage in choosing the initial conditions which have to be set arbitrarily. Moreover, once these values are learnt for a given plant, they can be used profitably in future.

The algorithm is well suited for on-line optimization of plants. Unlike other methods of adaptive control "without identification" where small changes in the controller parameters have to be made individually in order to calculate the gradient of the cost function surface at a particular point, the values of the parameters in the case of the vector cost function are found by measurements while the system is operating and without having to introduce small parameter disturbances. Further, for adaptation to transient response, this method is expected to perform better than others because of its inherent speed and ease of computation.

A great deal of freedom exists in choosing the methods which are to be used to update and learn the controller sensitivity matrices, and in choosing the system configuration. Further, the particular way and order in which the controllers are changed, the plant parameter variations learnt, the controller matrices updated and so on, are entirely at the control engineer's choice. No method working satisfactorily for one plant can be guaranteed to work as efficiently for another, but here too the final decision remains with the designer.

Of the three types of configurations investigated, the feedback path controller system was found to be the most satisfactory, though the minimum cost was obtained with the open loop controller system. The forward path controller and the open loop controller systems were found to be more prone to instability than was the feedback path controller system. The best results with the latter system were obtained when the controller input was the error between the desired and the actual plant outputs i.e. a time decreasing function. In such a case, the controller output was zero when the plant was in the desired state, and so the effect of the errors which were made by the arbitrary predicting and updating schemes used was consequently minimized.

The vector cost function offers a new approach to the field of adaptive control without identification. A learning mechanism is used to update the matrices of the system and which in turn are responsible for driving the system to an extremum value of the cost function. Hence, the learning methods used have to be sufficiently accurate to provide these matrices correctly.

Finally, stability is one of the serious problems encountered with this algorithm. But, by proper choice of the controller configuration, the initial parameter values and the constraints imposed, the adapted response can be made to remain in a stable space. Using a time decreasing input to the controller in the feedback path may not drive the system as fast as another input to its desired final state, but it will certainly ensure a greater stability and avoid such oscillations as were encountered with the open loop controller system.

The following suggestions concerning further work can be made:

1. Investigation of the efficiency of this method to problems concerning plants operating in their steady state, and where large parameter variations and/or external disturbances occur which affect the desired performance.

2. Implementation of a stability algorithm which would guarantee the final stability of the overall system.

3. Comparison of this method with other methods of adaptive control "without identification".

REFERENCES

[1] Mishkin, E., and Braun, L., "Adaptive Control Systems", McGraw-Hill Book Co., New York, 1961.

[2] Wilde, D.J., "Optimum Seeking Methods", Prentice Hall, 1964.

[3] Buehler, R.J., Shah, B.V., and Kempthorne, O., "Method of Parallel Tangents", Chemical Engineering Progress Symposium Series, Vol.60, 1964.

[4] Iqleem, M.J., "On the Feasibility of Adaptive Control without Identification", M.Eng. Thesis, McMaster University, 1967.

[5] Davis, R.H., Roberts, P.D., "Methods of Conjugate Gradients applied to self-adaptive digital control systems", Proc. I.E.E., Vol.115, No.4, April 1968.

[6] Shinners, S.M., "Control System Design", John Wiley and Sons, New York, 1964.

[7] Temple, V.A.K., "Nth Order Self-Adapting Control Systems", M.Eng. Thesis, McMaster University, 1967.

[8] Sinha, N.K., and Temple, V.A.K., "A Vector Cost Function for Adaptive Systems", Proc. of the 6th Allerton Conf. on Circuit & Systems Theory, Oct. 1968. pp. 700-701.

[9] Sinha, N.K., and Iqleem, M.J., "Adaptive Control Without Complete Plant Identification", Proc. of the fifth Allerton Conf. on Circuit and System Theory, 1967, pp. 201-210.

[10] Chestnut, H. , "Systems Engineering Tools", John Wiley & Sons, New York, 1965, p 473.

List of Symbols

a = a scalar quantity

$|m|$ = the absolute value of m

$\underline{u}$ = a vector $\underline{u}$

$||\underline{u}||$ = the norm of $\underline{u}$

$\langle \underline{v} . \underline{u} \rangle$ = the inner product of $\underline{v}$ with $\underline{u}$

c_i = the ith element of $\underline{c}$

$\underline{y}^t$ = the transpose of $\underline{y}$

$\frac{\partial x}{\partial v}$ = the partial derivative of x with respect to v

$\underline{\underline{A}}$ = a matrix $\underline{\underline{A}}$

$\underline{G}$ = the Vector Cost Function

t = time

$\dot{c}$ = $\frac{dc}{dt}$, the derivative of c with respect to t

∇Z = the gradient of Z

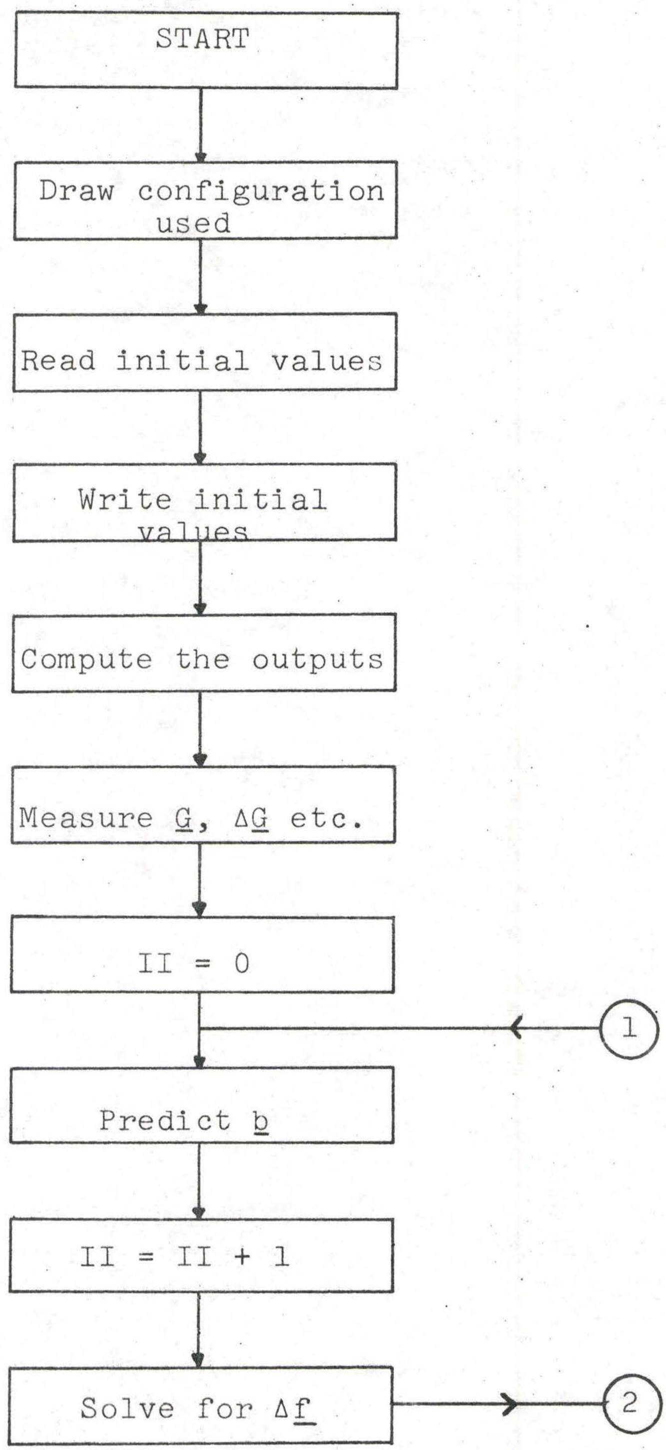
$x > y$ = x greater than y

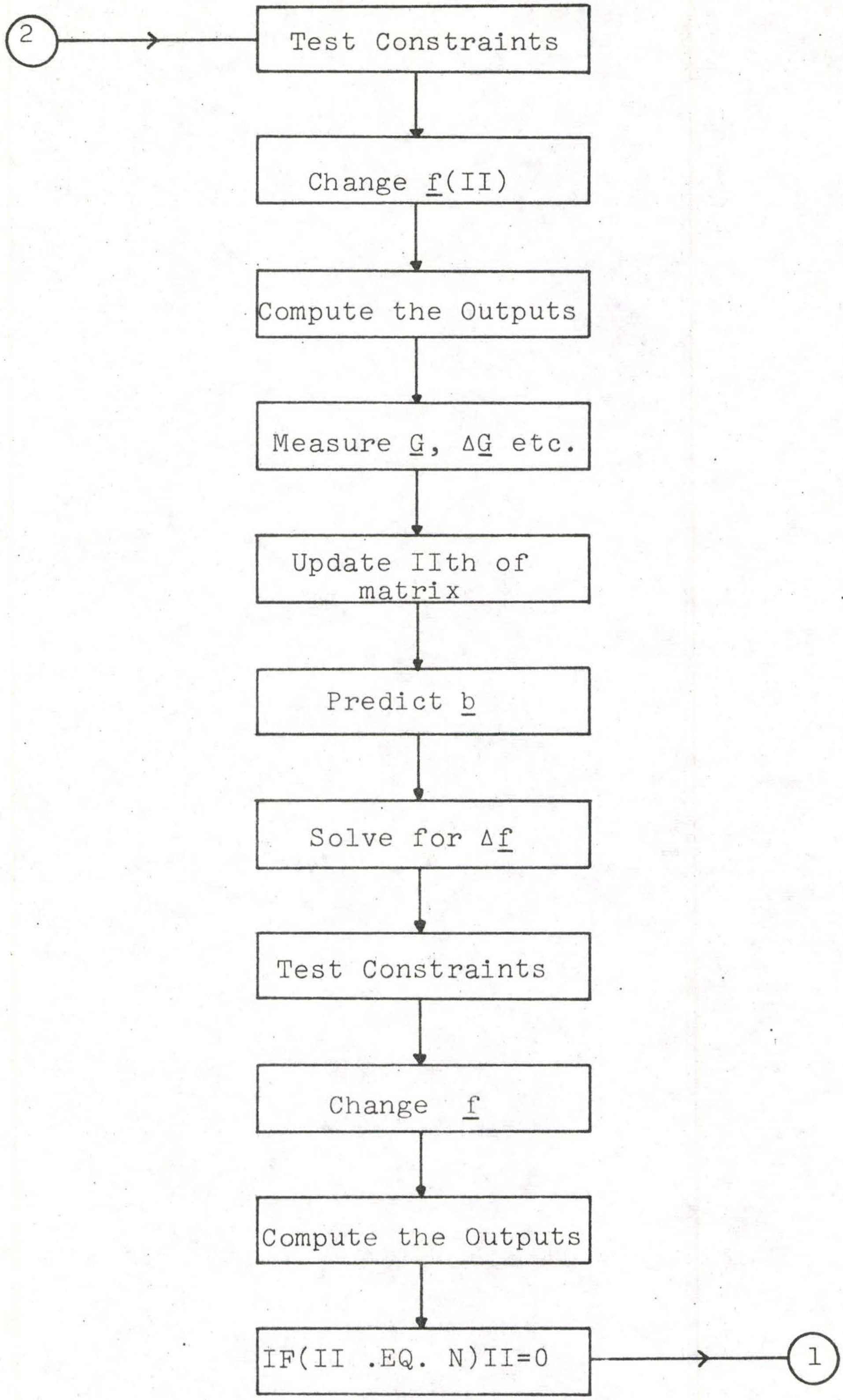
$m < n$ = m less than n

$\Delta\rho$ = a finite increment in ρ

APPENDIX (I)

General program outline chart.




2
Test Constraints
Change f(II)
Compute the Outputs
Measure G, ΔG etc.
Update IIth of matrix
Predict b
Solve for Δf
Test Constraints
Change f
Compute the Outputs
IF(II .EQ. N)II=0
1

APPENDIX (II)

TABLE OF VALUES FOR GRAPHS PLOTTED

Feedback Path Controller System

[Fig. (5.1.a)]

(I) Output vs. Time

Unadapted Plant		Adapted Plant	
Time (secs)	Output [y(1)]	Time (secs)	Output [y(1)]
0.	0.	0.	0.
0.5	0.2211	0.5	0.3871
1.0	0.7238	1.0	0.6527
1.5	1.2325	1.5	0.8032
2.0	1.5300	2.0	0.8885
2.5	1.5368	2.5	0.9368
3.0	1.3144	3.0	0.9642
3.5	1.0078	3.5	0.9797
4.0	0.7657	4.0	0.9885
4.5	0.6765	4.5	0.9935
5.0	0.7444	5.0	0.9963
5.5	0.9057	5.5	0.9979
6.0	1.0713	6.0	0.9988
6.5	1.1700	6.5	0.9993
7.0	1.1748	7.0	0.9996
7.5	1.1043		
8.0	1.0052		
8.5	0.9258		

Time (secs)	Output [y(1)]	Time (secs)	Output [y(1)]
9.0	0.8954		
9.5	0.9161		
10.0	0.9679		
10.5	1.0218		
11.0	1.0545		
12.0	1.0346		
15.0	1.0066		
18.1	0.9891		
20.0	1.0056		

(II) Phase-Plane Diagram

[Fig.(5.1.b)]

Unadapted Plant		Adapted Plant	
y(1)	y(2)	y(1)	y(2)
0.	0.	0.	0.
0.0036	0.1180	0.0022	0.2572
0.0191	0.2687	0.0103	0.5317
0.1013	0.5866	0.0382	0.8216
0.2376	0.8355	0.1114	0.9528
0.5066	1.0555	0.2046	0.8957
0.7238	1.1012	0.3513	0.7367
1.0040	1.0311	0.5005	0.5675
1.3505	0.7114	0.7004	0.3404
1.5688	-0.0034	0.9016	0.1118
1.3543	-0.5565	0.9806	0.0220

Unadapted Plant		Adapted Plant	
y(1)	y(2)	y(1)	y(2)
1.1552	-0.6263	0.9988	0.0013
0.7460	-0.3076	0.9996	0.0004
0.6771	-0.0287		
0.6851	0.1021		
0.7393	0.2499		
0.9128	0.3562		
1.1300	0.2034		
1.1711	0.1003		
1.1676	-0.1020		
1.0510	-0.2026		
1.0014	-0.1098		
1.0006	0.1078		

III Cost Function vs. Time

[Fig.(5.1.c)]

Time (secs)	Cost	Time (secs)	Cost
0.	0.	0.	0.
0.2	0.1945	0.1	0.0966
0.3	0.2824	0.2	0.1707
0.4	0.3604	0.3	0.2437
0.5	0.4267	0.4	0.2960
1.0	0.5803	0.5	0.3377
2.0	0.6780	0.8	0.4185
3.0	0.9308	1.0	0.4487
4.0	0.9578	1.5	0.4843

Unadapted Plant		Adapted Plant	
Time (secs)	Cost	Time (secs)	Cost
5.0	1.0465	2.7	0.5000
6.0	1.0646	4.0	0.5011
8.0	1.1019	5.0	0.5011
10.0	1.1141	7.0	0.5011
12.0	1.1179		
20.5	1.1200		

Forward Path Controller System

[Figure (5.2.a)]

Adapted System*

(I) Output vs. Time

Time (secs)	Output [y(1)]
0.	0.
0.03	0.0016
0.2	0.1205
0.5	0.3608
0.7	0.4907
1.0	0.6378
1.5	0.7948
2.0	0.8837
3.0	0.9627
4.0	0.9880
5.0	0.9961
6.0	0.9987
7.0	0.9996
8.0	0.9998
9.0	0.9999
10.0	1.0000
12.9	1.0000

* The unadapted plant values for the output, phase-plane diagram and the cost function are identical to those of (I), (II) and (III) respectively for the feedback path controller system.

(II) Phase-Plane Diagram.

[Figure (5.2.b)]

y(1)	y(2)
0.	0.
0.0016	0.1308
0.0056	0.2744
0.0127	0.4328
0.0454	0.8469
0.0763	0.8632
0.1021	0.9575
0.1370	0.7898
0.1525	0.7723
0.2013	0.8413
0.4029	0.6779
0.6655	0.3800
0.8500	0.1090
0.9526	0.0538
0.9952	0.0054
0.9999	0.0000
1.0000	0.0000

(III) Cost Function vs. Time

[Figure (5.2.c)]

Time (secs)	Cost
0.	0.
0.1	0.0992
0.3	0.2555
0.6	0.3950
1.0	0.4805
2.0	0.5322
3.0	0.5375
4.0	0.5381
5.0	0.5381
6.0	0.5381
10.0	0.5381

Open Loop Controller System

[Figure (5.3.a)]

(I) Output vs. Time.

Unadapted System		Adapted System	
Time (secs)	Output [y(1)]	Time (secs)	Output [y(1)]
0.	0.	0.	0.
0.5	0.1369	0.1	0.0332
1.0	0.3347	0.2	0.1707
1.5	0.4999	0.3	0.3588
2.0	0.6262	0.4	0.5515
3.0	0.7918	0.5	0.7548
4.0	0.8841	0.6	0.9544
5.0	0.9355	0.7	1.0715
7.0	0.9800	0.76	1.0916
10.0	0.9965	1.0	1.0342
15.0	0.9998	1.2	1.0077
20.0	0.9999	1.5	1.0004

(II) Phase-Plane Diagram

[Figure (5.3.b)]

Unadapted System		Adapted System	
y(1)	y(2)	y(1)	y(2)
0.	0.	0.	0.
0.0033	0.1066	0.0012	0.1086
0.0143	0.2016	0.0032	0.2011

Unadapted System		Adapted System	
y(1)	y(2)	y(1)	y(2)
0 0652	0.3443	0.0140	0.5080
0.1171	0.3911	0.0496	1.0097
0.1895	0.4066	0.1001	1.4584
0.2932	0.3859	0.2119	1.9940
0.3568	0.3610	0.4598	1.9240
0.5027	0.2879	0.5595	2.0030
0.6494	0.2049	0.6500	2.0261
0.7713	0.1339	0.7632	2.1081
0.8867	0.0663	0.9986	1.3875
0.9800	0.0117	1.0644	0.7625
0.9952	0.0028	1.0916	-0.0003
		1.0668	-0.2429

(III) Cost Function vs. Time

[Figure (5.3.c)]

Time (secs)	Cost	Time (secs)	Cost
0.	0.	0.	0.
0.5	0.4481	0.1	0.0981
1.0	0.7391	0.2	0.1806
1.5	0.9070	0.3	0.2348
2.0	1.0013	0.4	0.2648
4.0	1.1086	0.6	0.2794
6.0	1.1189	0.9	0.2809
9.0	1.1200	1.3	0.2812
10.0	1.1200	1.5	0.2812

Feedback Path Controller System

[Figure (5.4.b)]

Adapted System* with Δf_i (applied) = 2% of Δf_i (calculated)

Output vs. Time

Time (secs)	Output [y(1)].
0.	0.
0.3	0.0918
0.5	0.2771
0.7	0.4998
0.9	0.6397
1.1	0.7410
1.5	0.8822
2.0	0.9879
3.0	1.0570
4.0	1.0051
5.0	1.0004
6.0	1.0001
7.0	1.0000

* The values of the unadapted system are those of table (I) for the previous feedback path controller system.

(II) Cost Function vs. Time

[Figure (5.4.c)]

Time (secs)	Cost
0.	0.
0.1	0.0919
0.3	0.2809
0.5	0.4175
0.8	0.5086
1.0	0.5364
1.5	0.5580
2.0	0.5601
3.0	0.5619
4.0	0.5629
5.0	0.5629
6.0	0.5629
7.0	0.5629

Feedback Path Controller System

[Figure (5.5.a)]

Output vs. Time

(a) System using matrix $\underline{\underline{A}}$		(b) System using matrix $\underline{\underline{B}}$	
Time (secs)	Output	Time (secs)	Output
0.	0.	0.	0.
0.31	0.1377	0.33	0.1151
0.41	0.2450	0.59	0.3198
0.55	0.4387	1.01	0.5874
0.69	0.6378	1.69	0.8011
0.87	0.8220	3.01	0.9506
1.01	0.9048	4.09	0.9852
1.51	0.9922	4.55	0.9913
2.05	0.9999	5.55	0.9973
2.09	1.0000	6.45	0.9990

$$\underline{\underline{A}} = \left.\frac{\partial \underline{G}}{\partial \underline{f}}\right|_{t=0} = \begin{bmatrix} 0.2 & 0.01 \\ 0. & -5.0 \end{bmatrix}$$

$$\underline{\underline{B}} = \left.\frac{\partial \underline{G}}{\partial \underline{f}}\right|_{t=0} = \begin{bmatrix} 1.0 & 0.04 \\ -0.03 & -1.0 \end{bmatrix}$$

Cost Function vs. Time

[Figure (5.5.b)]

(a) System using matrix $\underline{\underline{A}}$

Time (secs)	Cost
0.	0.
0.11	0.1087
0.23	0.2183
0.53	0.4009
0.95	0.4506
1.33	0.4521
2.77	0.4521

(b) System using matrix $\underline{\underline{B}}$

Time (secs)	Cost
0.	0.
0.11	0.1089
0.33	0.3043
0.65	0.4894
1.01	0.5849
1.55	0.6380
3.03	0.6620
5.99	0.6630

Output vs. Time

[Figure (5.5.c)]

The values of the plant response are those of Figure (5.5.a) for the system using the initial matrix $\underline{\underline{B}}$. The values of the feedback path controller output for this system are given below

Time.	Controller Output.
0.	0.
0.07	-0.1396
0.17	-0.1970
0.23	-0.1800
0.35	-0.1241
0.39	0.1013

Time	Controller Output
0.45	0.3292
0.55	0.6558
0.73	0.8231
0.99	0.6475
1.35	0.3923
1.75	0.2397
2.21	0.1461
2.59	0.0989
3.33	0.0456
4.15	0.0186

APPENDIX (III)

TYPICAL VECTOR COST FUNCTION VALUES

(a) Open Loop Controller System

t = 0.183

$$\text{Predicted } \underline{b} = \begin{bmatrix} -1.70976 \times 10^{-3} \\ -1.03122 \times 10^{-2} \end{bmatrix}$$

$$\text{Measured } \underline{G} = \begin{bmatrix} 8.6171 \times 10^{-1} \\ -1.7142 \end{bmatrix}$$

$$\text{Measured } \Delta\underline{G} = \begin{bmatrix} -1.7091 \times 10^{-3} \\ -1.0111 \times 10^{-2} \end{bmatrix}$$

$$\frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} -3.004 \times 10^{-5} & 5.879 \times 10^{-6} \\ -1.999 \times 10^{-3} & 4.673 \times 10^{-5} \end{bmatrix}$$

$$\underline{f} = \begin{bmatrix} 4.50 \\ 4.60 \end{bmatrix}$$

$$\text{Calculated } \Delta\underline{f} \text{ change} = \begin{bmatrix} -4.864 \times 10^{3} \\ -1.714 \times 10^{5} \end{bmatrix}$$

Δf_2 change applied $= 1.0 \times 10^{-1}$

Δf_1 change applied $= 0.$

t = 0.184

$$\text{Predicted } \underline{b} = \begin{bmatrix} -1.71976 \times 10^{-3} \\ -1.01161 \times 10^{-2} \end{bmatrix}$$

$$\text{Measured } \underline{G} = \begin{bmatrix} 8.5999 \times 10^{-1} \\ -1.7243 \end{bmatrix}$$

$$\text{Measured } \Delta\underline{G} = \begin{bmatrix} -1.71920 \times 10^{-3} \\ -1.01158 \times 10^{-2} \end{bmatrix}$$

$$\frac{\partial G_2}{\partial f_1} \text{ (learnt)} = 5.562 \times 10^{-6}$$

$$\frac{\partial G_2}{\partial f_2} \text{ (learnt)} = 2.810 \times 10^{-6}$$

$$\text{Updated } \frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} -3.004 \times 10^{-5} & 5.816 \times 10^{-6} \\ -1.999 \times 10^{-3} & 3.795 \times 10^{-5} \end{bmatrix}$$

$$\text{Calculated } \Delta\underline{f} = \begin{bmatrix} -4.069 \times 10^{3} \\ -1.685 \times 10^{5} \end{bmatrix}$$

$$\Delta\underline{f} \text{ Change applied} = \begin{bmatrix} -1.0 \times 10^{-1} \\ -1.0 \times 10^{-1} \end{bmatrix}$$

(b) Feedback Path Controller System of Section [(5.4),(B)].

Some typical values of the controller sensitivity matrix:

$$t = 0.72 \quad , \quad \frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} 0.1444 & -110.4 \\ -0.2403 & 198.9 \end{bmatrix}$$

$$t = 2.40 \quad , \quad \frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} -0.0064 & 10.42 \\ -0.0347 & 6.396 \end{bmatrix}$$

$$t = 6.72 \quad , \quad \frac{\partial \underline{G}}{\partial \underline{f}} = \begin{bmatrix} -8.185 & 0.0065 \\ 21.96 & -0.0142 \end{bmatrix}$$

<u>APPENDIX (IV)</u>

Comparison of different configurations of adaptive systems

Configuration	Settling Time (secs)	Final Cost	Percentage Overshoot
(a) Feedback path Controller System (section 4.1)	7.00	0.501	0.
(b) Forward path Controller System	9.00	0.538	0.
(c) Open Loop Controller System	1.49	0.281	9.16
(d) Feedback path Controller System (section(5.4),B)	4.98	0.563	5.82
(e) Feedback path Controller System (section 5.5), (Matrix A)	2.57	0.452	0.
(f) Feedback path Controller System (section 5.5), (Matrix B)	9.07	0.663	0.