Vector Envisionment
of
Compartmental Systems

A dissertation submitted for the degree of Master of Science.

by

George M. Coghill B.Sc.(Eng)

Department of Clinical Physics
University of Glasgow

April 1992

ABSTRACT

Vector envisionment (VE) is a qualitative modelling method, for reasoning about dynamic physical systems, based on qualitative vectors. A qualitative vector is a means of describing the shape of a continuously differentiable, monotonic, real valued, function of time. It consists of a list of elements representing the qualitative value of the function along with the qualitative value of its successive derivatives; and an envisionment is the set of all possible qualitative behaviours that a system may exhibit for a particular input. This can be contrasted with a simulation which should produce a unique behaviour. Compartmental systems are a class of dynamic system composed of a finite set of homogeneous, well mixed, lumped subsystems called compartments; the behaviour of each compartment may be described by a first order differential equation.

A linear time invariant one compartment system is analysed for all increasing monotonic input vectors and it is demonstrated that VE can find all the distinct qualitative states in which the system may exist for a given input. The analysis is performed by associating a polynomial of appropriate length with each vector and using this to interpret the results of the envisionment. From this analysis a solution space is constructed which is divided in accordance with the critical points of the system. Each qualitatively distinct region of this space is associated with a particular range of magnitudes for the initial values for the state variable. An examination of the results gives a means of overcoming the problem of chattering.

VE is applied to examples of cascaded and coupled two compartment systems for a step input. Analysis shows that all and only the valid states of the system are again produced.

ACKNOWLEDGEMENTS

I would like to express my thanks to my supervisors for all their help and encouragement throughout the course of this project: Prof J. Mckie for enabling me to commence the project and making time available to me to pursue it; Dr. J. O. Rowan for taking over when Prof. McKie retired and enabling and encouraging me to bring this thesis to completion; and Prof. C. J. van Rijsbergen for making facilities available to me in the Department of Computing Science, without which much of the work of this project would have been much harder, and for fruitful and inspiring discussions which directed my thinking.

I would also like to express my gratitude to Prof. W. Fitch and Dr. A. J. Asbury of the University Department of Anaesthesia for providing a stimulating and fruitful research environment, for their guidance on the anaesthetic aspects of this project, for listening to my ramblings and supporting me in my endeavours to finish the work.

Personal thanks are due to my former colleague Dr. W. M. Gray for all his patience and encouragement throughout the time of my working with him, for many critical discussions of my work and for his perseverance in grappling with my clumsily expressed ideas.

Thanks are also merited by my colleagues in the Intelligent Automation Laboratory, Heriot-Watt University for creating a pleasant working environment and expanding my horizons; especially Prof. R. R. Leitch who read and commented on an earlier draft of this thesis.

I would also like to thank my parents and my sister and brother in law for providing material support towards the completion of this thesis.

To my wife, Sarah, who has been my helpmeet, has had to put up with my being necessarily unsociable,and who has taken many domestic burdens of my shoulders while I carried out this research, I express my deep gratitude and love.

Above all I thank the LORD for providing the opportunity to enter such an exciting and stimulating research area.

To say a machine can think
is like saying an aeroplane can fly.

George M. Coghill.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Introduction

In the recent history of artificial intelligence there has been a focussing of attention on the ability to reason about the physical world (viewed as a Newtonian mechanism) in terms of its structure. This research effort arose from the failure of earlier attempts to reason in a general way with associational rules. The distinction between the two approaches has been distinguished by the names 'deep' and 'shallow' knowledge. The former was so named because it went deeper into the structure of the system to be reasoned about, whereas the latter remained at the surface relating input to output (behaviour), or 'cause' to 'effect' on a purely empirical basis.

The work described in this thesis seeks to apply the deeper methods in the domain of pharmacokinetics as a step towards the eventual construction of an intelligent simulator of anaesthetic drug uptake for use in teaching and control. The contribution of this work towards this goal is in providing a thorough analysis of the qualitative response of low order linear time invariant compartmental systems to a variety of monotonic input functions.

The rest of this chapter is divided into two parts: the first sets the present work in its broader context within the field of anaesthetic simulation, and the second gives an overview of the rest of the thesis.

1.2. Background

The goal of the project of which this work constitutes a very small first step, is to construct a simulation of a patient undergoing surgery under anaesthesia. This requires that both the drug uptake and response to surgical stimulus be modelled. An important use of such a simulator would be in the training of junior anaesthetists, in much the same way that a flight simulator is used to train pilots. It is possible to generate situations which will be challenging, or rare, in an environment which affords some of the benefits though none of the risks of the real situation.

Some work has already been done in these areas. Schwid (1987), for example, has developed a set of numerical models which model the physiological and pharmacological behaviour of a patient under a variety of conditions. However, only the response to drug input is simulated. Another approach (Gaba and De Anda 1988) uses a real operating theatre plus a manikin (as the patient) for maximum realism. Pharmacological behaviour is simulated, and other responses are modified in accordance with a script. The whole process is controlled by a supervisor from an observation room. The system has been used on trainees but may also be used on experts as a means of knowledge elicitation for the construction of an expert system. This procedure would fall into the category of shallow knowledge approaches.

None of these methods have attempted to simulate the response of a patient to a surgical stimulus, and there is good reason for this. The response to a surgical stimulus is very vague and has a great deal of uncertainty associated with it. Therefore, any approach used to model it would have to be able to cope with this. Qualitative reasoning, because it is relatively new and has sought to deal with the behaviour of systems without reference to the numerical values of the parameters and variables looked like a promising approach to try.

Because the attempt to simulate the response to surgical stimulus was new, and the tools (qualitative reasoning) were new it was decided to construct models of the drug uptake aspect of the project first. This was to serve two purposes: first it would result in an intelligent simulation of drug uptake and second, it would permit the testing of qualitative models against known and well validated quantitative pharmacokinetic models. Unfortunately, the problem of generating many spurious behaviours associated with qualitative reasoning manifested themselves and the scope of the project had to be narrowed. The goal now was to examine the qualitative response of compartmental models (pharmacokinetic models are an example of compartmental models). It was evident, in view of the problems associated with qualitative reasoning that an exhaustive analysis of the simplest compartmental systems was required. The results of this analysis would form a basis for proceeding to use qualitative reasoning on more complex systems. The analysis was performed exhaustively on a one compartment system; and carried out for a step input to two different kinds of two compartment system. An analysis such as this has not been carried out before and it is the aim of this thesis to show how a qualitative reasoning approach can be used to describe the qualitative response of a linear time invariant system to a variety of monotonic input functions. This is an important study, because although success in this area would not guarantee that an uncertain system, such as the response of the human body to a surgical insult, can be successfully modelled by means of qualitative reasoning techniques, failure to adequately cope with the simple systems analysed here would invalidate the approach.

1.3. Overview of the Thesis

The following is an overview of the thesis chapters.

1.3.1. Chapter 2:

This chapter introduces the definition of a compartmental system, (the discussion is limited to linear time invariant systems). The state variable form is described. The chapter closes with a discussion of the application of compartmental systems to anaesthesia; specifically the work of W. W. Mapleson.

1.3.2. Chapter 3:

This chapter introduces qualitative reasoning. The subject is briefly discussed historically to show the two strands in qualitative reasoning: the cognitive science approach and the engineering approach. The three main paradigms are summarised as a background to the method chosen, which is described in chapter 4.

1.3.3. Chapter 4:

Vector envisionment (VE) is described in this chapter. This is an implementation of the approach to qualitative reasoning developed by Morgan (1988) in his Ph.D. thesis. The basis of the approach is the qualitative vector. The vector envisionment of a dynamic system is divided into two phases. First, is the qualitative analysis. In this phase the constraints of the system are applied simultaneously in a 'generate and test' manner, to discover all the states in which the system may exist without regard to temporal ordering. The second phase applies a series of continuity rules to the set of states to discover the valid transitions between them. The resulting directed graph is the vector envisionment . The VE algorithm is described.

1.3.4. Chapter 5:

In this chapter the results of performing a vector envisionment on single compartment systems in response to the thirteen increasing monotonic input functions (plus the zero input) are analysed. The VE process results in a graph of states which require interpretation if they are to make sense. It is shown that a suitable representation for this purpose is a truncated polynomial with general co-efficients. The states generated by VE for each of the fourteen input vectors is analysed by means of this this representation: it is shown that these states give a complete coverage of the solution space with no spurious states; the axes of this space are the co-efficients of the input vector plus the state variable. However, for first order inputs, the envisionment graph does exhibit the phenomenon of chattering. A way round this, is to implement a "non return" filter which would prevent an arc being sustained against the majority flow of the envisionment graph. This filter has not been implemented in the present work. The response to complex inputs, made up from more than one vector is also discussed.

1.3.5. Chapter 6:

This chapter describes the application of the method to two compartment systems. There are several topologies of second order compartmental systems, therefore, two examples are examined: a cascaded system (one with no feedback flow between the compartments), and a coupled system (with feedback between the compartments). the method of interpreting the behaviours is extended to multivariable systems and it is shown that once again complete coverage of the solution space is obtained without spurious states being generated. Unfortunately, because the output of any compartment will be an exponential, at least one compartment will have a first order input and so chattering will always occur. However, the heuristic suggested in the previous chapter may be used to cure this. A bug in the implementation which meant that the coupled system had to be envisioned by hand is discussed.

1.3.6. Chapter 7:

In this chapter the results of the work are summarised and the conclusions reached are stated. Suggestions for future work, both in the development of VE and in its application to anaesthesia, are given.

CHAPTER 2

COMPARTMENTAL MODELLING

2.1. Introduction

The work reported in this thesis concentrates on the application domain of compartmental models. This chapter gives a brief introduction to compartmental models, what they are used for and how they are constructed. The details are taken from Godfrey (1983), which also provided the analyses against which the results of the qualitative envisionments reported in chapters 6 and 7 are validated. This book gives a comprehensive introduction to compartmental systems, and the reader is directed there for more details. The description given here is geared towards the simulation of compartmental systems rather than their analysis.

2.2. What is a Compartmental Model?

Compartmental systems are a class of dynamic system composed of a finite set of homogeneous, well mixed lumped subsystems called compartments; the behaviour of each compartment may be described by a first order differential equation (Godfrey, 1983). A definition of a compartment has been given by Atkins (1969) quoted in Godfrey (1983) as 'a quantity of a substance which has a uniform and distinguishable kinetics of transformation or transport'. This means that a compartment may represent either the transport of the same substance from one physical location to another, or the transformation of one substance into another at the same physical location.

In the most general case each first order equation has the following form, for the ith compartment of a p compartment system:

$$\frac{dx_i}{dt} = f_{i0} + \sum_{\substack{j=1 \\ j \neq i}}^{p} f_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^{p} f_{ji} - f_{0i} \qquad i = 1,2,\ldots,p \quad \text{-----(2.1)}$$

where $\frac{dx_i}{dt}$ is the rate of change of amount of material in compartment i; f_{ij} is the flow rate to compartment i from compartment j (the subscript 0 represents the environment). A system in which there is flow to or from the environment is called an open system, and if there is no such flow the system is said to be closed.

It is not certain when compartmental systems were first studied, though they were applied to biological systems as far back as 1923. Today they are used in a wide variety of fields such as ecology, chemical engineering, and pharmacokinetics (Godfrey, 1983).

Equation 2.1 represents the most general form of a compartmental model and as such is too general for practical systems. Three categories of interest are linear time invariant (LTI) systems, time varying systems and non-linear systems. Of these the category which has been studied most is LTI systems, therefore I will discuss these in more detail in the next section.

2.3. Linear Time Invariant Compartmental Systems

In a LTI system the compartmental equation becomes:

$$\frac{dx_i}{dt} = \sum_{\substack{j=1 \\ j \neq i}}^{p} k_{ij}.x_j - \sum_{\substack{j=1 \\ j \neq i}}^{p} k_{ji}.x_i - k_{0i}.x_i + u_i(t), \qquad i = 1, 2, \ldots, p$$

where the ks are known as rate constants and have the dimension of time^{-1}. The ks are non-negative. It can be seen that the flow rate from compartment i to compartment j is dependant on the amount in compartment i (the donor compartment) only; thus the system is called donor controlled. This type of system lends itself to being represented in 'state variable' form. A state is 'a set of numbers such that the knowledge of these numbers and the input function will, together with the equations describing the dynamics, give the future state and output of the system' (Godfrey, 1983).

To demonstrate, consider the general two compartment system (as shown in figure 2.1) with a single input to compartment 1. The equations are:

$$\frac{dx_1(t)}{dt} = -k_{01}.x_1(t) - k_{21}.x_1(t) + k_{12}.x_2(t) + u_1(t)$$

$$= -(k_{01} + k_{21}).x_1(t) + k_{12}.x_2(t) + u_1(t)$$

$$\frac{dx_2(t)}{dt} = k_{21}.x_1(t) - k_{02}.x_2(t) - k_{12}.x_2(t)$$

$$= k_{21}.x_1(t) - (k_{02} + k_{12}).x_2(t)$$

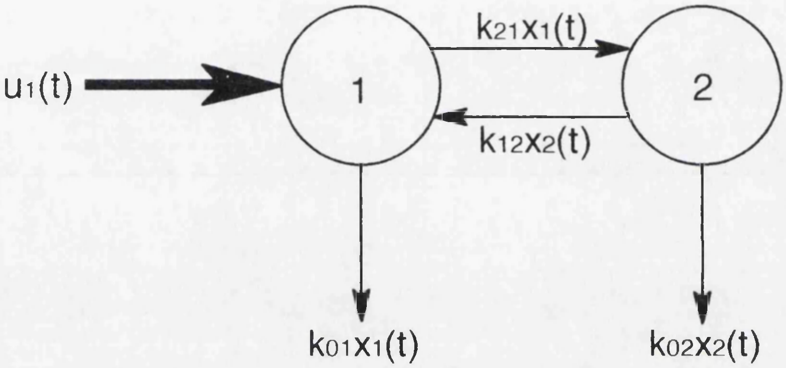

Figure 2.1 General two compartment system.

This can be written in matrix form as:

$$\begin{bmatrix} x'_1(t) \\ x'_2(t) \end{bmatrix} = \begin{bmatrix} -(k_{01} + k_{21}) & k_{12} \\ k_{21} & -(k_{02} + k_{12}) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ 0 \end{bmatrix}$$

which in matrix notation is:

$$\underline{x}' = \mathbf{A}\underline{x} + \mathbf{B}\underline{u}$$

and is known as the state equation. The output of a system is related to the state of the system by the following algebraic relationship:

$$\underline{y} = \mathbf{C}\underline{x}$$

which is known as the observation equation. The matrix **A** is the 'system matrix' and represents the structure of the system under consideration. For a p compartment system with m inputs $\underline{x}'$ and $\underline{x}$ are p × 1 vectors, $\underline{u}$ is a m × 1 vector, **A** is a p × p matrix, and **B** is a p × m matrix.

The elements of **A** are

$$a_{ij} = k_{ij} \qquad\qquad i \neq j$$

$$a_{jj} = -\sum_{\substack{i=1 \\ j \neq i}}^{p} k_{ij} - k_{0j}$$

Thus by means of the state variable approach we see that the criterion for a system to be compartmental is that all the elements of the main diagonal of the matrix **A** must be non-positive and all its other elements must be non-negative. Also

$$\left| a_{jj} \right| \geq \sum_{\substack{i=1 \\ j \neq i}}^{p} k_{ij}$$

with the equality applying if $k_{ij} = 0$. All the states of a compartmental system are non-negative. This is proved by Godfrey (1983); but it may be grasped intuitively by considering that one cannot have a bucket that is less than empty.

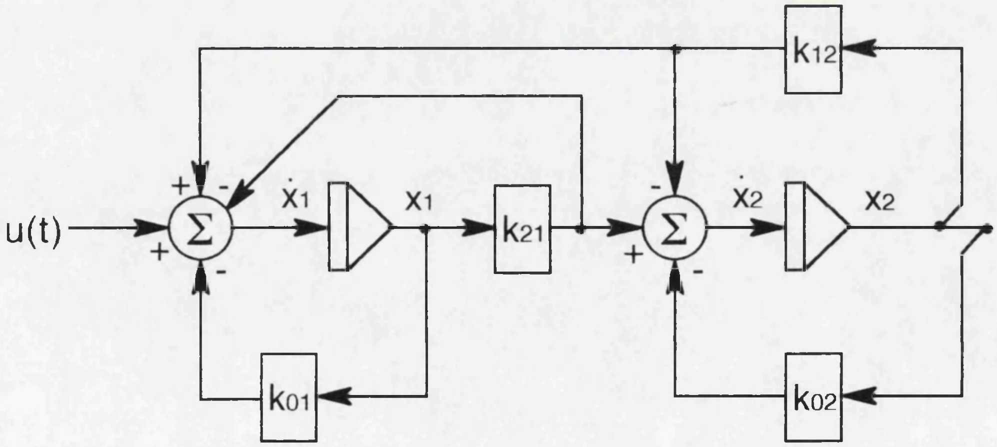

Figure 2.2 Block diagram of a two compartment system.

Compartmental systems have a form which is easily translated to an analogue computer diagram, or block diagram such as that shown in figure 2.2.

2.4. Application to Anaesthesia

Compartmental models have been used for several different aspects of the anaesthesia process: pharmacokinetics of competitive muscle relaxants (Miller 1982), pharmacodynamics of neuromuscular blocking agents (Hull, 1982) and physiological distribution of volatile anaesthetic agents. In this section I will concentrate on the third of these because it falls most closely within the original goals of the project.

An early attempt to model uptake of inert gases was carried out by Kety (1951). He lumped all the tissues into one compartment and thus got an equation which did not give a very good fit to the experimental data. Following this work Mapleson (1963), using the International Commission for Radiological Protection (ICRP) data for a standard man, worked out the parameters for a model consisting of 19 compartments. Starting from this he aggregated the compartments to find a model which would give a suitable fit to the experimental data and yet be of sufficiently low order to be manageable. The model he finally arrived at consisted of 4 compartments: a lung compartment and three tissue compartments. These represented the viscera, muscle and fat; which in terms of their capacity to dissolve anaesthetic agents were small, medium and large respectively. A circuit diagram of the passive electrical network which Mapleson used is shown in fig. 2.3 along with its compartmental representation (fig. 2.4). Here, G_a represents the ventilation of the lungs and C_a the capacity of the lungs. The G_i represent the blood flow to the tissues and the C_i the capacity of the tissues. This model was meant to serve as a teaching aid to allow students to `see' the form which the time course of drug uptake in the body takes. However, Mapleson (1984) also used it, in an equivalent hydraulic form, as an aid to explanation in writing about how different anaesthetics behaved in the body. Hence the exercise which he carried out not only produced a useful teaching tool, but also demonstrated how aggregation is used in the construction of models and also how model based reasoning is used to give insight into and explain dynamic processes.

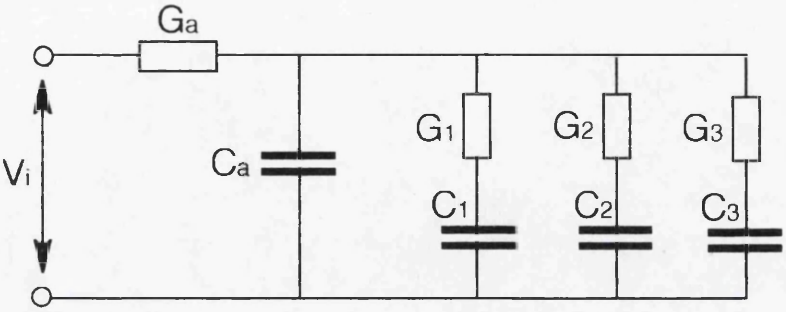

Figure 2.3 The Mapleson Analogue.

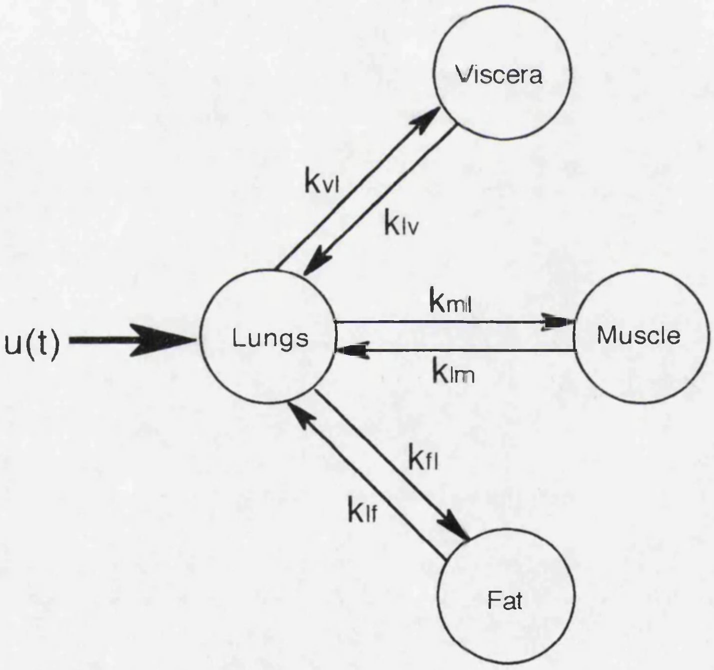

Figure 2.4 The Mapleson Analogue - compartmental form.

Mapleson's passive analogue has served as the progenitor of other models of anaesthetic drug uptake . The analogue did not incorporate the circulation time delays which occur in human physiological systems. Mapleson (1973,1978), therefore, did a thorough study of various means of representing the circulation time, and in characteristic fashion arrived at a model which balanced the competing requirements of accuracy and computational tractability. This was called `Model P' because along with the lung and tissue compartments there were separate `pools' for the arterial and venous blood which allowed the circulation time to be modelled. Model `P' has itself undergone development and has been extended to model the uptake and protein binding of the injected analgesic pethidine (Davis, 1987). It has also been used as the model in a model based control system in animal studies of automatic control of drug administration (Mapleson *et al*, 1980).

In all the Mapleson models the heart and respiration rates were held constant, which would not be the case in reality. Therefore, several workers devised models which had their roots in the passive analogue, and sought to develop it to take account of the effects of anaesthetics on cardiac output (Zwart, Smith and Beneken 1972). These approaches involved using separate models for different aspects of the system. The earlier versions were analogue two model representations (Beneken and Rideout, 1968). This resulted in a non-linear system because the blood flow to the tissues is no longer constant, and thus the concentration of drug carried to the tissues is no longer linearly related to the drug concentration in the lungs.

The latest model of this type was by Schwid (1987), is digital, and contains three separate models: a circulatory and baroreflexes model, an oxygen and carbon dioxide transport model, and a pharmacokinetic/pharmacodynamic model. These representations were designed for use in teaching and/or control. However, for teaching, interpretation of responses is still the responsibility of the user; there being no means of linguistic explanation built in. This fact makes it a ripe research area for model based reasoning, one of the goals of which is to build systems which can be used to explain the principles of physical systems based on structural descriptions of the systems themselves and their behaviours.

CHAPTER 3

QUALITATIVE MODELLING

3.1. Introduction

Qualitative reasoning, as a separate domain of intellectual enquiry is young; just over 10 years old. However, recently there has been a comprehensive review written by Cohn (1989), therefore in this chapter I will merely give a brief overview of the main thrusts of research activity to place the present work in context. If more detail is required the reader is directed to the sources themselves or to the review mentioned above.

There is a sense in which the whole field of artificial intelligence can be called qualitative reasoning in that it deals with symbolic rather than numeric tasks. However, the words `qualitative reasoning' have come to be most closely associated with a particular approach to reasoning about physical systems. The goal is to be able to reason about a system by using knowledge of its structure, the relationship existing between variables, or the processes involved in the system.

Variables are (in the most abstract case) represented by a mapping of the real number line onto its sign, giving the set of values [+ 0 -]. Normal arithmetic operations can be applied among these quantities as shown in fig. 3.1. It should be noted that since only the sign of the variables is used, ambiguity arises when quantities of opposite sign are summed: it would require knowledge of the relative magnitude of the quantities (an ordinal scale) to resolve the ambiguity. It is because of this that the set of quantities is sometimes written as [+ 0 - ?], with [?] representing either `don't know' or `don't care'. Since the quantities used are abstractions of the real number line, if a system is represented by a qualitative version of its structure (e.g. a qualitative differential equation) then a one to many relationship exists between the qualitative and

B \ A	+	o	-	?
+	+	+	?	?
o	+	o	-	?
-	?	-	-	?
?	?	?	?	?

Add(A,B)

B \ A	+	o	-	?
+	?	-	-	?
o	+	o	-	?
-	+	+	?	?
?	?	?	?	?

Sub(A,B)

B \ A	+	o	-	?
+	+	o	-	?
o	o	o	o	o
-	-	o	+	?
?	?	o	?	?

Mult(A,B)

B \ A	+	o	-	?
+	+	o	-	?
o	x	x	x	x
-	-	o	+	?
?	?	o	?	?

Div(A,B)

Figure 3.1 Qualitative arithmetic operations.

the quantitative versions. That is, an ordinary differential equation may only be represented by one qualitative differential equation, but a qualitative differential equation may represent many ordinary differential equations (Struss, 1988a).

There are, at root, two ways of viewing qualitative reasoning: 1) the cognitive science approach and 2) the engineering view. In the former qualitative reasoning is seen as an attempt to use the models to reason about physical systems in the same way as humans. Hence finding the correct solution (or even the best solution) may not be of paramount importance because humans make mistakes (regularly). On the other hand, from the engineering viewpoint the goal is to build a reasoning system which gives the correct answer whenever possible, so that it can be used with confidence in situations where having the right answer can be important (e.g. safety critical systems).

3.2. Paradigms

In the study of artificial intelligence, the ability to make general statements about `things' has always been an aim (McCarthy 1987), and qualitative reasoning arose out of this desire. The history of qualitative reasoning can be divided into three periods:

Prehistory -- work done prior to 1979

Early history -- work done between 1979 and 1986

Steps to Maturity -- work done since 1986

During the pre-historic period the problems were formulated (McCarthy and Hayes 1969), and theoretical solutions proposed (Hayes 1978,1985) , though little was done towards implementing the ideas. An exception to this was de Kleer (1975) who designed and built a system (NEWTON) to reason about kinematics in a roller coaster world. The system was designed to work at different levels of quantitation depending on what information was available (qualitative reasoning was tried first and if that failed then quantitative reasoning was tried by means of MACSYMA).Though limited in its application this research was, I think, the precursor of the engineering approach; and hence though both the theoretical work of Hayes and the more practical work of de Kleer had the same end in view their different approaches have resulted in the present division between the cognitive science and engineering views.

The second period saw the emergence of three major paradigms for qualitative reasoning: ENVISION (de Kleer 1984), Qualitative Process Theory (Forbus 1984) and Qualitative Simulation (Kuipers 1986). These have been so influential that a great deal of the subsequent research has consisted in trying to improve and develop one or more of them. The final period has seen an explosion of research interest in qualitative reasoning and though much of it has been spent trying to overcome the problems in the first generation of qualitative reasoning systems, some new approaches are now being implemented (Raimon 1986, Morgan 1988, Weld 1990, Wiegand 1991 and Shen 1991).

In the rest of this section I will briefly describe the three main paradigms of qualitative reasoning, because of their influence on subsequent research.

3.2.1. ENVISION

In ENVISION (de Kleer 1984) a physical system is viewed as a device. A device is made up of components which are joined together by conduits, materials can be operated on by components and can be transferred between components via conduits. Components have associated with them confluences and qualitative states. A confluence is a qualitative version of an ordinary differential equation which uses just the signs of the variables. There are separate equations for the variables and the first derivatives. A confluence has a different meaning from its quantitative counterpart in that an equation of the form:

$$[A] \oplus [B] = [0]$$

(where $\oplus$ is the symbol for qualitative addition) must be interpreted to mean that the result of the qualitative addition of [A] and [B] must include [0] as a solution.

The qualitative state of a component refers to the 'operating regions' in which it can exist. Thus a valve may have three states: closed, working, and fully open. Each state will have a set of confluences associated with it which dictate how the component will behave in that state. A change of state occurs when one or more variables of a device exceeds some boundary value which defines the different states of a component. Conduits merely allow the passage of materials without affecting the material in any way. The model of a coupled tank would require three components (two tanks and a pipe), two conduits (notional connectors between the pipe and tanks) and one material (water, say). A device can be represented as a graph, with the components as nodes and the conduits as edges.

If a device is perturbed then an envisionment ensues. An envisionment is the generation of all the possible behaviours which may result from a particular perturbation. This term has come to be distinguished from 'simulation' which is the attempt to find a unique behaviour as the response to a perturbation. (There is also a partial envisionment which is the finding of some, but not all, of the possible behaviours). There are two types of behaviour associated with this envisionment: intrastate behaviour and interstate behaviour. The former is the set of qualitative values that a variable can take and still satisfy the confluences of the state; and the latter refers to the number of transitions which are possible between two equilibrium states.

3.2.2. Qualitative Process Theory (QPT)

Qualitative Process Theory (Forbus 1984) is an effort to implement a representation of commonsense reasoning about the physical world along the lines suggested by Hayes (1978). The only mechanism deemed to exist is the *process*'; which consists of changes in an object over time. These changes may consists of changes of the properties of an object or the creation and destruction of objects.

An *individual* in QPT is an object in the theory. For instance, a bucket, and some water, are individuals. An *individual view* consists of a set of individuals, a set of preconditions which assess the applicability of the individual view, and a set of relations. For example, some water in a bucket would satisfy the individual view 'contained liquid'. Processes contain the same information as individual views plus a set of influences or proportionalities between the variables of the individuals, and a set of initial conditions. An individual may have several influences and proportionalities acting on it, so any change which occurs in it will be the sum of the possible effects. The output of QPT is an envisionment, though the quantity space can contain more values than +, o, and -. However, all the values a variable can take must be specified *a priori.*

The approach is reminiscent of medieval philosophy (with materials 'containing' heat). However, this would not be considered a problem by Forbus because he is seeking to work in Naive Physics domain (Hayes 1978) which has a cognitive science viewpoint; seeking to reason in the same way as humans.

3.2.3. Qualitative Simulation (QSIM)

QSIM was developed in the context of medical expert diagnosis as a knowledge engineering tool. Kuipers designed it as a component based method with the components represented as variables linked by constraints (Kuipers 1984). The models can be constructed from textbook descriptions and existing models (Kuipers 1986,1987a), or from analysis of interviews with domain experts by means of knowledge elicitation techniques (Kuipers 1987b). The method of model construction parallels the method for constructing analogue models of engineering systems, which makes it the closest of the three approaches to standard engineering simulations. This contradicts those who consider QSIM to be a purely mathematical abstraction of ordinary differential equations (Cohn 1989).

3.3. The Structure of QSIM

Kuipers (1986) was the first to attempt to give a formal treatment of the design of a qualitative reasoning system. In his paper several definitions and proofs of the different parts which go to make up QSIM were given. Many of these are applicable to any of the qualitative reasoning paradigms; but one feature of QSIM which distinguishes it from the other approaches is that it can discover new landmarks in the quantity space, thus increasing the number of qualitatively distinct values which may be used. The landmarks found in this way are a a partially ordered set, because, while it is possible to describe the ordinal relationship between two values of the same sign, it is not possible to order the magnitudes of positive and negative quantities.

The purpose of QSIM is to derive a description of system/model behaviour from a given description of its structure. As noted above, a structural description in Kuipers' terms consists of a set of parameters related by constraints. In this context a constraint is the qualitative (symbolic) representation of a mathematical relationship and a parameter is a continuously differential real valued function of time; or what he calls a `reasonable function'. The qualitative state of such a function at a time point or in an interval is described by the pair <qval qdir>, where qval is the qualitative value of the function, and is ordinally related to the landmarks of the quantity space. The qdir is the direction of change (the sign of the derivative) of the function. Thus qval is of variable resolution and qdir is of fixed, low resolution.

QSIM generates the behaviours of the parameters of the system by progressing alternatively from time points to intervals and from intervals to time points. The only time points generated in the course of a simulation are those which correspond to something interesting happening to the behaviours (the `distinguished time points'). These occur when one of the parameters of the system arrives at an existing landmark, or a new landmark is found for that parameter.

The constraints of the system are then used individually to filter this set for assignments to the parameters which are inconsistent with the constraints, known as constraint filtering. After this, constraints are used in pairs, if they share a parameter, to check that the values of the shared parameter, though consistent with each constraint are also consistent with both together; called pairwise consistency filtering. A global interpretation is `an assignment of a transition to each parameter of the system. That is, each variable is assigned one of its remaining values, consistent with the constraints, to form a set. The number of such assignments which can be made gives the total number of successor states which can exist after one cycle of the algorithm. After this, several global filters have to be applied to a global interpretation before it can be passed as valid.

3.4. Problems

Kuipers was the first to demonstrate that there would be branching problems and the creation of spurious behaviours in using QSIM for qualitative reasoning. The main reason which he gives for the problems is that the simulation progresses and filters states locally. That is, the only qualitative states examined in a transition are the present state and its immediate successor. Therefore, if QSIM branches at a time point, and, at a later time, one of the parameters is in a state of the same form, QSIM will not remember that it has `been here before' and will branch again and again until the computer memory fills up. The problem exists for any qualitative reasoning system in which the reasoning is done locally. Kuipers also points out that numerical simulation is a local procedure. However, because the variables are bound to real numbers, branching does not occur and the problem does not manifest itself.

To overcome this problem Kuipers and his fellow workers have tried a number of approaches: ignoring irrelevant distinctions between states and application of higher order derivatives to the system variables (Kuipers and Chui 1987), studying the phase plane trajectories (Lee and Kuipers 1988, Struss 1988b) and incorporation of incomplete quantitative knowledge (Kuipers and Berleant 1988).

Most qualitative reasoning systems suffer from problems with branching and the creation of spurious states. Struss (1988a) has done a thorough mathematical analysis of why this is/must be so. The method utilised in the present work is vector envisionment which seeks to overcome the problems of qualitative reasoning by giving a global picture. The approach is described in the next chapter.

CHAPTER 4

VECTOR ENVISIONMENT

4.1. Introduction

Vector Envisionment (VE) is an implementation of the qualitative reasoning approach described by Morgan (1988) in his thesis. In the third chapter of the thesis he defines a method for the qualitative modelling of dynamic systems by means of qualitative vectors; the elements of which can take values from the [+ 0 - ?] set. In this chapter I will summarise those parts of Morgan's approach relevant to the modelling of linear time invariant compartmental systems.

4.2. Qualitative Vectors

A qualitative vector is a means of describing the shape of a continuously differentiable, monotonic, real function (of time in the present context). It consists of a list of elements representing the qualitative value of the function along with the qualitative value of its successive derivatives (the QSIM representation <qval qdir> can be considered as a two element vector). The more elements a vector has, the more information it contains about the function it represents. To illustrate, consider the following:

For $a,t > 0$; if we examine $x=at^2$, then:

$$x=at^2 \leftrightarrow [+]$$
$$x'=2at \leftrightarrow [+]$$
$$x''= 2a \leftrightarrow [+]$$
$$x'''=0 \leftrightarrow [0]$$
$$\vdots$$

where ↔ should be read as "has the qualitative value".

The first differentiation above shows that the sign of the direction of change of the quadratic function is positive. The second derivative of a function is an indicator of its curvature, which in this case is positive. Thus the curve shape of a quadratic is concave upwards. A qualitative vector which gives the shape of this function can be constructed in this way, consisting of the sign of the value, first and second derivatives; which in this case gives $x = at^2 \leftrightarrow [+ + +]$. While a vector may have as many elements as desired there is a pragmatic reason for truncating it after three elements (for an input function at least), namely, a three element vector gives the curvature of a function and that is all that is usually perceived `with the naked eye'. This gives nine possible monotonic curve shapes as shown in figure 4.1.

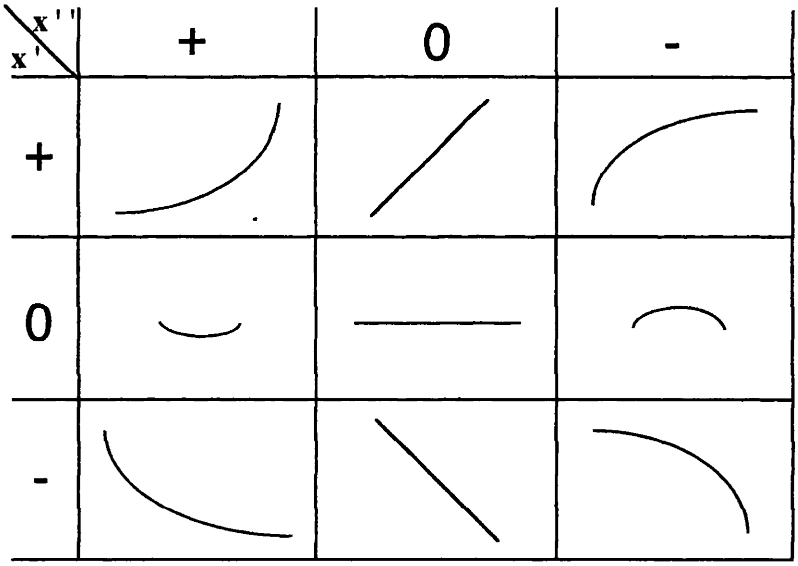

Figure 4.1 The nine possible curve shapes of a three element vector.

Time is always represented as an interval. That is, the time over which a vector describes a function is always a temporal interval and never a time point. If there is a zero in the vector then the interval can be considered 'infinitesmal'.

4.3. Vector Algebra

The sign algebra of qualitative vectors is an extension of the sign algebra of qualitative scalars discussed in chapter three, since a scalar can be thought of as a vector with all its derivatives equal to zero. In this section only addition and subtraction of two vectors, and multiplication by a scalar are described. This is because these are the only ones required to envision a linear time invariant system. More complex calculations arise in multiplying two vectors or inverting a vector. The interested reader is to Morgan (1988) for details of these operations.

When two vectors are added or subtracted, the elements of the resulting vector are the sum or difference of the corresponding elements of the original vectors.

The effect of multiplying a vector by a scalar depends on the sign of the scalar. If the scalar is positive, the vector remains unchanged, if it is negative, all the elements of the vector change their sign, and if it is zero, all the elements of the vector become zero as well.

4.4. Qualitative Integration

In order to simulate or envision a dynamic system, some form of integration must be available in the formalism. To this end Morgan defines an integration rule. Since each vector element is the derivative of the element immediately preceding it in the vector, integration basically involves shifting each element one place to the right. This gives the integral value of all but the first vector element which is calculated by other means. The Integration Rule may be stated as follows:

For a three element vector with elements d^0, d^1, and d^2; let the integral of this have elements $d^0{}_I$, $d^1{}_I$, and $d^2{}_I$ where

$$d^2{}_I = d^1$$

$$d^1{}_I = d^0$$

$$d^0{}_I = d^0 + d^1 + \text{constant of integration}$$

The expression for d^{0}_{I} is stated but its derivation is not explained in Morgan, although it appears to be a qualitative version of Euler integration with an additional constant of integration. Unfortunately, this rule does not always produce a valid function as the integral if the constant of integration is any value other than [?]. For example, Morgan applies the rule to a sine wave with an undefined constant of integration and claims that the result represents a cosine wave. However, because the constant of integration is [?] the first element of each vector will also be '?'. This may represent a cosine, but disambiguation would have to be done on other grounds. On the other hand, if the constant of integration is defined as [o] then a definite, completely spurious, function is calculated. Since Euler integration does not require an additional constant of integration and I can only conclude, from the above observations that it was included, and left undefined, to avoid the spurious waveform. This example is shown in figure 4.2 and Table 1. For this reason, and others which will be dealt with later, I leave the first element of the integral undefined and resolve the resulting ambiguity be means of the algebraic relationships in the system being envisioned; this is similar to the way originally suggested by Morgan (1987)

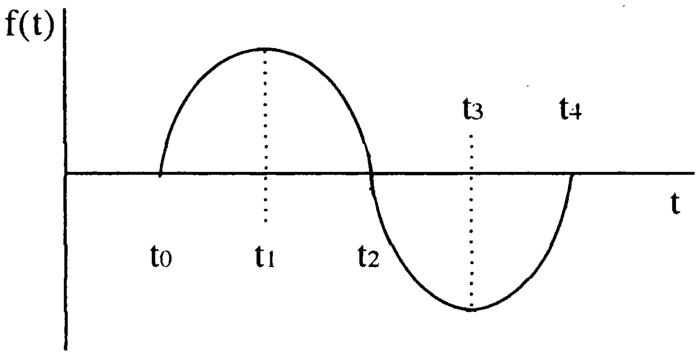

Figure 4.2 Sinusoidally varying function.

time t	sinusoid	integral	qualitative integration (c.o.i. = [?])	qualitative integration (c.o.i. = [o])
t_0	[o + o]	[- o +]	[? o +]	[+ o +]
$t_0 - t_1$	[+ + -]	[- + +]	[? + +]	[+ + +]
t_1	[+ o -]	[o + o]	[? + o]	[+ + o]
$t_1 - t_2$	[+ - -]	[+ + -]	[? + -]	[? + -]
t_2	[o - o]	[+ o -]	[? o -]	[- o -]
$t_2 - t_3$	[- - +]	[+ - -]	[? - -]	[- - -]
t_3	[- o +]	[o - o]	[? - o]	[- - o]
$t_3 - t_4$	[- + +]	[- - +]	[? - +]	[? - +]
t_4	[o + o]	[- o +]	[? o +]	[+ o +]

Table 4.1 Vector history for a sinusoid and its integral.

4.5. Vector Envisionment of Dynamic Systems

The method of qualitative reasoning developed by Morgan was for generating the qualitative behaviour of dynamic physical systems. The model used to represent the system is a qualitative abstraction of those used in numerical or analogue simulations.

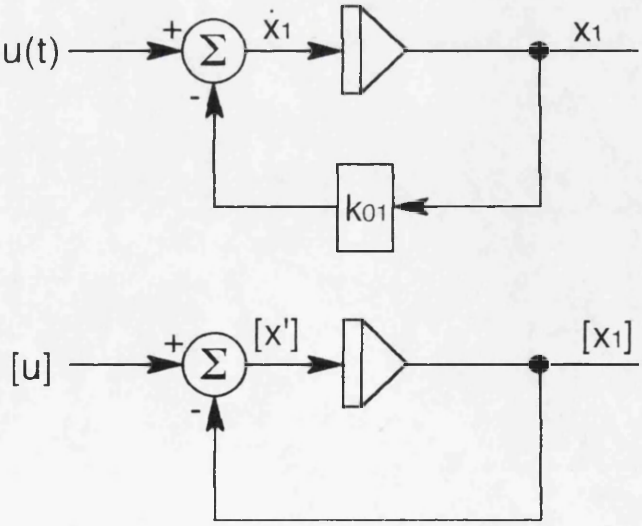

Figure 4.3 Quantitative and qualitative block diagrams of a one compartment system.

This is illustrated by figure 4.3 which shows the quantitative and qualitative block diagrams of a one compartment system. Vector Envisionment is a two stage process consisting of Qualitative Analysis and Transition Analysis. The former generates all the valid states in which the systems may exist for a given input, and the latter works out the ways in which the states can be joined together to form valid histories. These two processes will now be described in more detail.

4.5.1. Qualitative Analysis

In VE the qualitative analysis is implemented (as described by Morgan (1988)) as a generate and test algorithm. The values for the derivatives of the state variables are hypothesized, (the 'test variables') and these, along with the input, are used to generate values for the other variables of the system by application of the constraints. The values thus assigned are tested for consistency, also by means of the constraints. If the values are consistent the process proceeds; if they are not, new values are hypothesized for the test variables and the process starts over again.

Vectors are built up element by element. The first element of the test variables is generated and the variables tested for consistent value assignment. If successful the next element is added; and so on until a consistent state is found with vectors of the chosen length. This state is stored as a valid state of the system and the process continues until no further valid states can be found. If a test vector does not yield a consistent value assignment then no further elements are added to that vector and the next value of the present test vector element is tried. This results in a pruned depth first search of the tree of all possible system states, yielding all the legal states that the system may exist in for given input vector. Once all the legal states have been found the next step is to discover the ways the ways in which they can be joined to form a history: this is the role of Transition Analysis.

4.5.2. Transition Analysis

Transition Analysis consists of a set of continuity rules which define the legal transitions which can occur between states. They are as follows (Morgan 1988):

Continuity Rule 1

Elements can only change to the opposite sign by a transition through zero.

Continuity Rule 2

If any element d^n has the same value value in the succeeding vector $d^n{}_{succ}$, its lower order element must behave consistently, as defined by:

$$d^{n-1}{}_{succ} = 0 \text{ or } d^{n-1}, \text{ if } d^{n-1} = (\text{minus } d^n)$$

$$d^{n-1}{}_{succ} = d^{n-1} + d^n, \text{ otherwise.}$$

Each element, d^n, of the vector is the derivative of the element d^{n-1}. This rule uses this fact to ensure that no element changes in one direction if its derivative indicates that it is stationary or changing in the other direction.

Continuity Rule 3

Any vector with element $d^n = 0$ and an element d^{n+1} will have predecessor and successor vectors with the corresponding element values as follows:

If $d^{n+1} \neq 0$, then

predecessor: $d^n{}_{pred} = (\text{minus } d^{n+1}) \text{ or } 0$

$d^{n+1}{}_{pred} = d^{n+1} \text{ or } 0$

successor: $d^n{}_{succ} = d^{n+1}{}_{succ} = d^{n+1} \text{ or } 0$

If $d^{n+1} = 0$ then,

predecessor = successor with $d^n = 0$ and $d^{n+1} = ?$

This rule ensures that the critical points of a response are connected to curves of the correct shape. It differs slightly from Morgan's description which only dealt with the case where $d^1 = 0$ and caused some ambiguity in the response to a ramp input when $d^0 = 0$. These are all the rules necessary for the envisionment of linear time invariant systems and figure 4.4 shows the valid transitions which result for any vector element pair. The algorithm which I have developed is described in the next section.

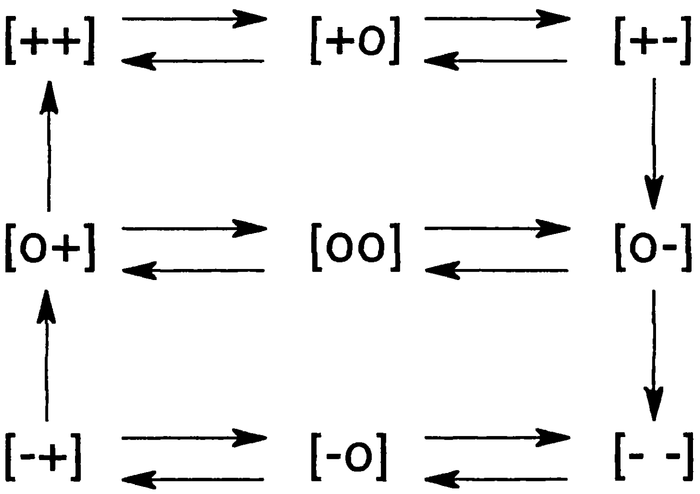

Figure 4.4 Pairwise transitions.

4.6. The VE Algorithm

The structures used in VE are (as in QSIM) qualitative differential equations. I have therefore implemented a version of the qualitative differential equation structure developed by Kuipers for storing and inputing qualitative differential equations to QSIM.

A qualitative differential equation is a LISP structure with the following fields:

variable: a list of the variables of the system

constraints: a list of the constraints holding between the system variables.

positive-variables: a list (possibly empty) of variables which must be non-negative. this reflects the fact that state variables in a compartmental system must be non-negative.

input: gives the name of the variable which is the input. The instantiation of this variable implicitly defines the length of vector to be used.

test-variables: a list of variables used by the `generate-and-test` part of the envisionment to generate test values which allow the envisionment to proceed.

print-variables: a list of variables whose values are printed as output.

The algorithm follows, with QUEUE being the list (+ o -), the elements of which are assigned as values to the test variables and STATES being the list of valid state-vectors of the system.

Step 1 Assign the next value from QUEUE to the first element of the test variable(s).

Step 2 Apply constraints to get values for the other variables.

Step 3 Test final variable values for consistency with constraints. If it is consistent, go on to *Step 4*, otherwise return to *Step 1*.

Step 4 Assign values to other elements of the test variables; apply constraints and check for consistency in a depth first manner until a set of vectors of the same length as the input vector is obtained.

Step 5 Expand any vector elements with value '?' into '+', `0' or '-' and check for consistency.

Step 6 Add consistent qualitative states to the list STATES.

Step 7 Return to Step 4 for the same value of the first element of the test variables till all possible subsequent element values have been tried.

Step 8 Return to Step 1 and repeat until all values of the first element of the test variables have been tried.

Step 9 Go through the list STATES applying continuity rule 1 to the present member of STATES with each other member.

Step 10 As for Step 9, but with continuity rule 2

Step 11 As for Step 9, but with continuity rule 3

This algorithm produces a total envisionment, for any given input, from which a particular behaviour may be selected under certain conditions of relative variable quantitation. This is dealt with in subsequent chapters. This algorithm works for one compartment systems and cascaded multicompartment systems. However, it produces spurious states for coupled compartmental systems. This due to a bug in the implementation which will be dealt with when I discuss the response of a coupled system.

4.7. Summary

In this chapter I have described the methodology developed by Morgan (1988) for describing the qualitative behaviour of dynamic physical systems. The reasons for describing functions as vectors consisting of the qualitative value of the function plus its successive derivatives have been given. Morgan's rules for Qualitative Analysis and Transition Analysis have been stated; the former consist of qualitative arithmetic relationships and an integration rule, and the latter consist of continuity rules for connecting the qualitative states generated by qualitative analysis. I have argued that this approach should be modified to leave the first element of the integral vector undefined after application of the integration rule. Because the integration rule will be used along with the other rules to describe the behaviour of dynamic systems the undefined vector element can be assigned a value from other system considerations. An algorithm, incorporating this modification, which can be used to generate all the qualitative states in which a system may exist (for a given input) and all the state transitions that can occur has been described. The output of the algorithm is an envisionment of a system. The algorithm has been implemented and was used to conduct the experiments discussed in the next two chapters.

CHAPTER 5

APPLICATION TO SINGLE COMPARTMENT SYSTEMS

5.1. Introduction

In chapter 4 of his thesis, Morgan gives an analysis of the qualitative response of a parallel R-C circuit to a step change in the input current. The initial voltage across the capacitor (initial condition) is left undefined and the distinct qualitative behaviours are shown to depend on the value of the initial condition. Morgan states that the initial condition was left undefined to show the generality of the approach. However, it is an argument of this thesis that V.E. only gives valid qualitative states if the initial value is left undefined. That is, if one tries to specify the initial value spurious states result. To demonstrate this, the R-C circuit with a step change to the input is analysed with the initial value specified.

It is an aim of qualitative reasoning to be able to predict, qualitatively, the response of physical systems in which the system parameters are unknown or uncertain. However, confidence in the ability of the technique to model such systems cannot be had without a demonstration that qualitative reasoning can adequately model existing well defined systems for all initial conditions: giving valid predictions of behaviour. Success in such an experiment would not guarantee success in modelling uncertain systems; but failure would highlight the inadequacy of the approach. Little attention has been paid to this aspect of qualitative reasoning research. As a step in this direction I have analysed the qualitative response of a one-compartment system to all relevant single vector inputs. I also give a coherent means of interpretation of the envisionments produced based on polynomial expansion of analytic solutions.

A three element vector represents a monotonic function. However, sometimes it is desired to represent non-monotonic functions, and this can be achieved by concatenating monotonic function portions together over time providing the concatenation is valid according to the continuity rules. One reason for doing an exhaustive analysis of a one compartment system is that compartmental models in general are systems of one-compartment models joined together; therefore, if a one-compartment model can be shown to handle any arbitrary shape of input function then so can any order of cascaded LTI system.

5.2. A Parallel R-C Circuit

It was stated in chapter 4 that there were good reasons for preferring to view the operation of Morgan's methodology as an envisionment rather than as a simulation. These reasons will now be examined by means of the analysis of the response of a parallel R-C circuit (shown schematically in figure 5.1) to a step change in the input current (represented by the qualitative vector [+ o o]). This system is chosen for two reasons: it is the example used by Morgan and it is analogous to a one compartment system with the exception that the state variable is allowed to take on negative values. The problems I wish to highlight do manifest themselves if only positive values are allowed but they are more marked if negative values are permitted as well.

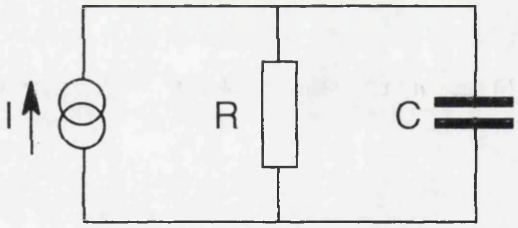

Figure 5.1 A parallel R-C circuit.

In his analysis Morgan states that the initial condition is left undefined to show the generality of the approach; however, examination of the results below will indicate that to avoid the generation of spurious states the initial condition must be left undefined. The first order ordinary differential equation describing this system is:

$$v'(t) = -k.v(t) + b.i(t)$$

where k = 1/RC, the rate constant, and b = 1/C. The qualitative form of this is:

$$[v\_prime] = [-] \otimes [v] \oplus [i]$$

which can be re-arranged to give the constraint:

$$[v\_prime] \oplus [v] = [i] \qquad \text{-----} \ (5.1)$$

Table 5.1 shows all the outcomes for the 27 possible values of [v_prime] (the test variable). This table summarises the qualitative analysis and is reproduced from Morgan (1988). The vectors for [v] in the third column of the table are generated by applying the integration rule to the corresponding vector for [v-prime]. The constant of integration is undefined in this case. It can be seen that there are three possible values for [v] which satisfy the constraint (equation 5.1) in row 7, whereas only one satisfies the constraint in rows 14 and 21. (It should be noted that a value of [? ? ?] for [I] is a valid solution because it 'contains' the value [+ o o]).

Applying the continuity rules to the states found by qualitative analysis yields the total envisionment shown in table 5.2 which represents the behaviours shown in figure 5.2. The fact that VE produces all valid states in which a linear time invariant system may exist will be dealt with in detail later; for the purposes of the present discussion an intuitive treatment will be given.

The response of a first order system to a step input will be an exponential. If the initial condition is zero then it will be a saturating response (i.e. increasing towards the steady state, concave downwards) from zero to the steady state value. If the initial condition is greater than zero there are three possible behaviours: a saturating curve if the initial condition is less than the steady state, a straight line if it is equal to the steady state, and a decaying curve if it is greater than the steady state. If the initial condition is less than zero, the response will be a saturating curve from the initial value, through zero, and up to the steady state. These behaviours, which are all shown on figure 5.2 are the only ones possible: therefore the envisionment finds all and only the valid responses of the system to a step input. (The circled numbers in figure 5.2 refer to the corresponding states and transitions in table 5.2).

row	[v_prime]	[v]	[I] = [+ o o]	Possible values for [v]
1	+ + +	? + +	? + +	
2	+ + o	? + +	? + +	
3	+ + -	? + +	? + ?	
4	+ o +	? + o	? + +	
5	+ o o	? + o	? + o	
6	+ o -	? + o	? + -	
7	+ - +	? + -	? ? ?	- + - o + - + + -
8	+ - o	? + -	? ? -	
9	+ - -	? + -	? ? -	
10	o + +	? o +	? + +	
11	o + o	? o +	? + +	
12	o + -	? o +	? + ?	
13	o o +	? o o	? o +	
14	o o o	? o o	? o o	+ o o
15	o o -	? o o	? o -	
16	o - +	? o -	? - ?	
17	o - o	? o -	? - -	
18	o - -	? o -	? - -	
19	- + +	? - +	? ? +	
20	- + o	? - +	? ? +	
21	- + -	? - +	? ? ?	+ - +
22	- o +	? - o	? - +	
23	- o o	? - o	? - o	
24	- o -	? - o	? - -	
25	- - +	? - -	? - ?	
26	- - o	? - -	? - -	
27	- - -	? - -	? - -	

Table 5.1 The qualitative analysis of the R-C circuit.

However, consider now the states generated by qualitative analysis if the constant of integration is set to positive (these are shown in table 5.3). It can be seen that the possible vectors for [v] generated are the same as for the undefined condition, though the vectors in column 3 of table 5.3 differ from those of table 5.1 (because the constant of integration in the integration rule is now specified as positive). Thus two spurious states (and on application of the continuity rules, two spurious behaviours) are generated.

State	[v]	[v_prime]	Successors
1	+ + -	+ - +	nil
2	0 + -	+ - +	1
3	- + -	+ - +	2
4	+ 0 0	0 0 0	nil
5	+ - +	- + -	nil

Table 5.2 A total envisionment for a step change.

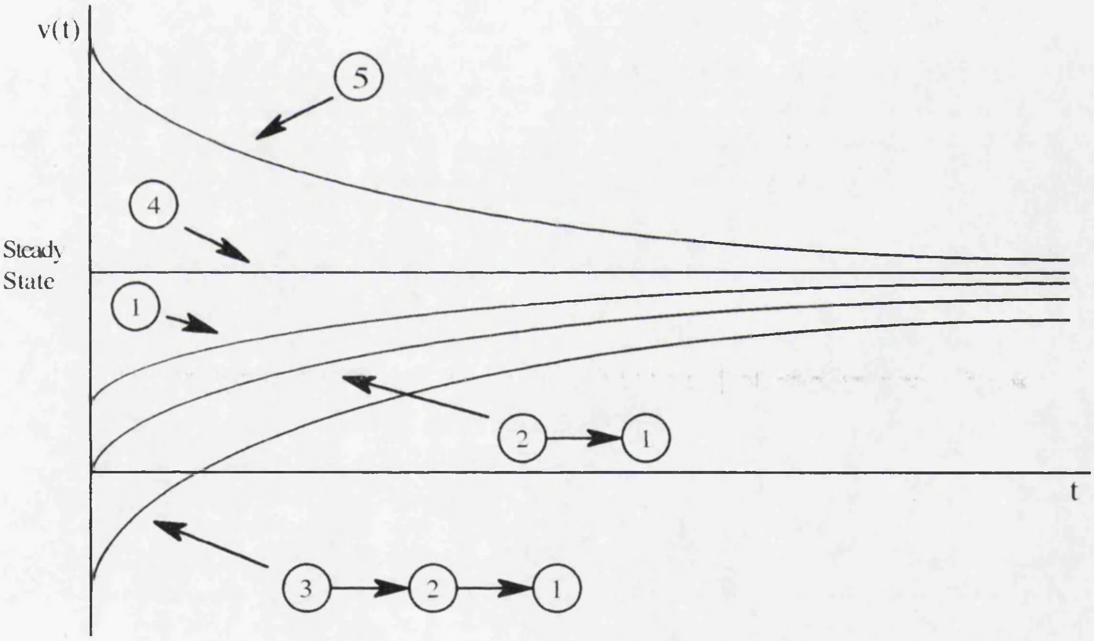

Figure 5.2 The possible responses to an input step change.

The situation is not much improved with zero constant of integration. Here there should only be one behaviour, but three are found (if one ignores the fact that

[+ + -] can exist as a behaviour on its own). This is shown in table 5.4. And with a negative constant of integration there are the same number of states generated as with zero and therefore at least one spurious state, as shown in table 5.5.

Of course the spurious states can be eliminated by means of heuristics, but this would defeat the purpose of having a separate integration rule as defined by Morgan. Therefore, I conclude that ambiguities should be treated as friends, not enemies, and an envisionment used which will seek to generate all the possible behaviours of the system.

row	[v_dot]	[v]	[I] = [+ o o]	Possible values for [v]
1	+ + +	+ + +	+ + +	
2	+ + o	+ + +	+ + +	
3	+ + -	+ + +	+ + ?	
4	+ o +	+ + o	+ + +	
5	+ o o	+ + o	+ + o	
6	+ o -	+ + o	+ + -	
7	+ - +	? + -	? ? ?	- + - o + - + + -
8	+ - o	? + -	? ? -	
9	+ - -	? + -	? ? -	
10	o + +	+ o +	+ + +	
11	o + o	+ o +	+ + +	
12	o + -	+ o +	+ + ?	
13	o o +	+ o o	+ o +	
14	o o o	+ o o	+ o o	+ o o
15	o o -	+ o o	+ o -	
16	o - +	? o -	? - ?	
17	o - o	? o -	? - -	
18	o - -	? o -	? - -	
19	- + +	? - +	? ? +	
20	- + o	? - +	? ? +	
21	- + -	? - +	? ? ?	+ - +
22	- o +	? - o	? - +	
23	- o o	? - o	? - o	
24	- o -	? - o	? - -	
25	- - +	? - -	? - ?	
26	- - o	? - -	? - -	
27	- - -	? - -	? - -	

Table 5.3 Result of qualitative analysis with a positive initial condition.

row	[v_dot]	[v]	[I] = [+ o o]	Possible values for [v]
1	+ + +	+ + +	+ + +	
2	+ + o	+ + +	+ + +	
3	+ + -	+ + +	+ + ?	
4	+ o +	+ + o	+ + +	
5	+ o o	+ + o	+ + o	
6	+ o -	+ + o	+ + -	
7	+ - +	? + -	? ? ?	- + - o + - + + -
8	+ - o	? + -	? ? -	
9	+ - -	? + -	? ? -	
10	o + +	+ o +	+ + +	
11	o + o	+ o +	+ + +	
12	o + -	+ o +	+ + ?	
13	o o +	o o o	o o +	
14	o o o	o o o	o o o	
15	o o -	o o o	o o -	
16	o - +	- o -	- - ?	
17	o - o	- o -	- - -	
18	o - -	- o -	- - -	
19	- + +	? - +	? ? +	
20	- + o	? - +	? ? +	
21	- + -	? - +	? ? ?	+ - +
22	- o +	- - o	- - +	
23	- o o	- - o	- - o	
24	- o -	- - o	- - -	
25	- - +	- - -	- - ?	
26	- - o	- - -	- - -	
27	- - -	- - -	- - -	

Table 5.4 Result of qualitative analysis with zero initial conditions.

5.3. A Truncated Polynomial Representation

The qualitative vectors represent an abstraction of real, continuously differentiable functions of time. Therefore, to interpret the responses predicted by VE an analytical analogue of the qualitative vectors would be helpful. A useful analytical function is a polynomial series of the form:

$$x(t) = a_0 + a_1.t + a_2.t^2, \quad a_0, a_1, a_2 > 0 \qquad \text{-----(5.2)}$$

A version of this quadratic can be found for any three element vector, with the a's being coefficients. For example, a constant function is represented by the vector [+ o o] which in turn is interpreted as

$$x(t) = a_0, \quad a_0 > 0$$

similarly a stepped ramp is [+ + o] which is interpreted as:

$$x(t) = a_0 + a_1.t\,, \qquad a_0, a_1 > 0$$

and equation 5.2 above is an interpretation of [+ + +].

row	[v_dot]	[v]	[I] = [+ o o]	Possible values for [v]
1	+ + +	? + +	? + +	
2	+ + o	? + +	? + +	
3	+ + -	? + +	? + ?	
4	+ o +	? + o	? + +	
5	+ o o	? + o	? + o	
6	+ o -	? + o	? + -	
7	+ - +	? + -	? ? ?	- + - o + - + + -
8	+ - o	? + -	? ? -	
9	+ - -	? + -	? ? -	
10	o + +	? o +	? + +	
11	o + o	? o +	? + +	
12	o + -	? o +	? + ?	
13	o o +	- o o	- o +	
14	o o o	- o o	- o o	
15	o o -	- o o	- o -	
16	o - +	- o -	- - ?	
17	o - o	- o -	- - -	
18	o - -	- o -	- - -	
19	- + +	? - +	? ? +	
20	- + o	? - +	? ? +	
21	- + -	? - +	? ? ?	+ - +
22	- o +	- - o	- - +	
23	- o o	- - o	- - o	
24	- o -	- - o	- - -	
25	- - +	- - -	- - ?	
26	- - o	- - -	- - -	
27	- - -	- - -	- - -	

Table 5.5 Result of qualitative analysis with negative initial conditions.

The utility of this representation can be seen by considering two functions, at t=0, which both have [+ + +] vectors for t>0. The polynomial $x(t) = a_0 + a_1.t + a_2.t^2$ is positive at t=0. However, one may wish to represent a function which is `curved up', but is zero at t=0, such as $x(t) = a.t^2$. This function has the vector [+ + +] for t>0, but at t=0 it gives:

$$\begin{aligned} x&=at^2 \leftrightarrow [o] \\ x'&=2at \leftrightarrow [o] \\ x''&= 2a \leftrightarrow [+] \\ x'''&=0 \leftrightarrow [o] \\ &\vdots \end{aligned}$$

which gives the vector [0 0 +]. So I have [0 0 +] at t=0 and [+ + +] when t>0. But the sequence: [0 0 +] → [+ + +] breaks continuity rule 3. The only way to get from [0 0 +] to [+ + +] is via [0 + +] giving the sequence:

$$[0\ 0\ +] \rightarrow [0\ +\ +] \rightarrow [+\ +\ +]$$

Unfortunately, this gives the vectors for $x(t) = a.t^2$ at t=0 and t>0 respectively, joined together by a vector which does not represent $x(t) = a.t^2$ at any time. A way round this, which gives a consistent representation of the vectors is to make use of the fact that in Vector Envisionment time is always represented by an interval, and so a sequence of vectors is a concatenation of time intervals (perhaps infinitesimal). I can therefore take the beginning of each interval to be (notionally) t=0 for the vector in that interval and represent each element of the vector by a function which gives the `value' of that element when differentiated the requisite number of times. Thus a three element vector, $[d^0\ d^1\ d^2]$ can be represented by three functions:

$$d^0 = a_0$$
$$d^1 = a_1.t$$
$$d^2 = a_2.t^2$$

allowing the vector to be represented by the analytical function $x(t) = a_0 + a_1.t + a_2.t^2$.

So, returning to the example
[0 0 +] can be interpreted by $a_2.t^2$
[0 + +] can be interpreted by $a_1.t + a_2.t^2$
[+ + +] can be interpreted by $a_0 + a_1.t + a_2.t^2$
giving the temporal sequence

$$a_2.t^2 \rightarrow a_1.t + a_2.t^2 \rightarrow a_0 + a_1.t + a_2.t^2$$

which satisfies the continuity rules.
(it should be noted that the a_i will not in general have the same numeric value in successive intervals).

This representation has the added advantage that vectors representing critical points, such as [+ 0 -], which cannot be described by any other functions, can be described in this way, and thus, the response evoked by them in an envisionment can be interpreted.

5.4. Analysis of the Response of a One Compartment System

A one compartment system is shown in figure 5.3. The ordinary differential equation describing it is:

$$x'(t) = -k_{01}.x(t) + b.u(t) \qquad \text{---- (5.3)}$$

where k_{01} is the elimination rate constant, x(t) is the state variable and b is the bio-availability of the input substance.The observation equation is

$$y(t) = cx(t)$$

where y(t) is the observed value of x(t) and c is the gain. In what follows b and c are assumed equal to 1 and the state variable is discussed directly. This prevents clogging up the interpretation of the qualitative analysis. The following qualitative equation is obtained from equation 5.3:

$$[x\_dot] = [-] \otimes [x] \oplus [u]$$

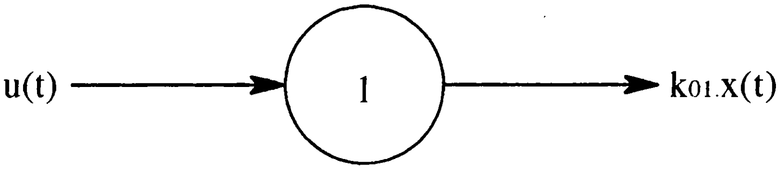

Figure 5.3 A one compartment system.

Vector envisionments have been obtained for this system for 14 possible input vectors. These are the vectors where the variable values are either positive, or zero with the first non-zero derivative positive. The results of the envisionment are shown in tables 5.7 to 5.11 and Appendix B (the output from VE is shown in Appendix A). The responses are shown as four element vectors because the relationship between x and x' permits the third element of x' to be added on to end of the x vector. To allow a consistent labelling of the states the four element vectors are therefore labelled as shown in table 5.6. The result of VE is a set of vectors joined together to form a graph. This result needs interpretation. An intuitive analysis of the response to a step change has already been given but now a more formal analysis of the general case will be considered. The interpretations of the results are shown in column 3 of the tables, and the input function is represented by the polynomial

$$u(t) = u_0 + u_1.t + u_2.t^2$$

state	vector	state	vector
1	+ + + +	22	+ - o +
2	+ + + o	23	+ - o o
3	+ + + -	24	+ - o -
4	+ + o +	25	+ - - +
5	+ + o o	26	+ - - o
6	+ + o -	27	+ - - -
7	+ + - +	28	o + + +
8	+ + - o	29	o + + o
9	+ + - -	30	o + + -
10	+ o + +	31	o + o +
11	+ o + o	32	o + o o
12	+ o + -	33	o + o -
13	+ o o +	34	o + - +
14	+ o o o	35	o + - o
15	+ o o -	36	o + - -
16	+ o - +	37	o o + +
17	+ o - o	38	o o + o
18	+ o - -	39	o o + -
19	+ - + +	40	o o o +
20	+ - + o	41	o o o o
21	+ - + -		

Table 5.6 State vector labels.

The basis of the interpretation is that the behaviour of a state variable over a time interval in response to a forcing function is dependent only on its value at the start of the time interval; and not on how the value was reached. (Details of the state variable approach to system modelling may be found any of the standard textbooks such as Franklin, Powell & Emami-Naeini (1991)). It follows from this that any qualitative state (which will represent the behaviour over a time interval), generated by the qualitative analysis phase, may begin a history. In interpreting the vectors one seeks to match values of the state variable with qualitative values of the vectors. Since the only qualitative landmark being used is [o] interpretations may only be calculated for initial values of the state variable at critical points of the system (that is, where one or more of the derivatives is zero). For vectors without zeros the initial values of the state variable will be between two critical point values. In the tabulated results the values of the state variable are recorded from largest to smallest.

state	vector	interpretation
21	+ - + -	$x(0) > 0$
41	o o o o	$x(0) = 0$

Table 5.7 Envisionment for [o o o] input vector.

The interpretation of the results is performed as follows: at $t = 0$, equation 5.3 , with $b = 1$, becomes :

$$x'(0) = -k_{01}.x(0) + u(0) \qquad \text{---(5.4)}$$

The first critical point occurs when $x'(0) = 0$ (corresponding to the second vector element being zero) in which case one has:

$$0 = -k_{01}.x(0) + u(0)$$

giving

$$x(0) = \frac{u(0)}{k_{01}}$$

Differentiating equation 5.4 gives

$$\begin{aligned} x''(0) &= -k_{01}.x'(0) + u'(0) \\ &= -k_{01}.(-k_{01}.x(0) + u(0)) + u'(0) \qquad \text{----(5.5)} \\ &= k^2{}_{01}.x(0) - k_{01}.u(0) + u'(0) \end{aligned}$$

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}}$
14	+ o o o	$x(0) = \frac{u_0}{k_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}}$
34	o + - +	$x(0) = 0$

Table 5.8 Envisionment for a [+ o o] input vector.

Figure 5.4 Transition graph for [+ o o] input vector.

The second critical point (corresponding to the third vector element being zero) occurs when $x''(0) = 0$, therefore:

$$0 = k^2{}_{01}.x(0) - k_{01}.u(0) + u'(0)$$

from which one gets

$$x(0) = \frac{u(0)}{k_{01}} - \frac{u'(0)}{k^2_{01}}$$

Differentiating again gives:

$$\begin{aligned} x'''(0) &= -k_{01}.x''(0) + u''(0) \\ &= -k_{01}.(k^2_{01}.x(0) - k_{01}.u(0) + u'(0)) + u''(0) \quad \text{--(5.6)} \\ &= -k^3_{01}.x(0) - k^2_{01}.u(0) + k_{01}.u'(0) + u''(0) \end{aligned}$$

The third critical point occurs when x'''(0) = 0, therefore

$$0 = -k^3_{01}.x(0) - k^2_{01}.u(0) + k_{01}.u'(0) + u''(0)$$

from which one gets

$$x(0) = \frac{u(0)}{k_{01}} - \frac{u'(0)}{k^2_{01}} + \frac{u''(0)}{k^3_{01}}$$

Hence the value of the state variable at the beginning of the time period over which any particular vector is valid is defined from the rate constant of the system and the successive derivatives of the input.

This analysis defined the initial value of the system at each critical point; however, a qualitative vector consists of several elements not all of which are zero but which must each have a consistent interpretation. That they do can be seen from an examination of the envisionment tables. Because there is only one landmark in the quantity space, the state variable will have a value in any interpretation which will be greater than, less than, or equal to the value it had at the particular critical point for which the vector element in question was zero. As an example, consider state 16 in table 5.10. The input vector is [+ - +] in this case, and is therefore represented by the expression $u(t) = u_0 - u_1.t + u_2.t^2$. State 16 is the vector [+ o - +].

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}}$
12	+ o + -	$x(0) = \frac{u_0}{k_{01}}$
3	+ + + -	$\frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} < x(0) < \frac{u_0}{k_{01}}$
5	+ + o o	$x(0) = \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}}$
30	o + + -	$x(0) = 0,\ x'(0) > \frac{u_1}{k_{01}}$
32	o + o o	$x(0) = 0,\ x'(0) = \frac{u_1}{k_{01}}$
34	o + - +	$x(0) = 0,\ x'(0) < \frac{u_1}{k_{01}}$

Table 5.9 Envisionment for the [+ + o] input vector.

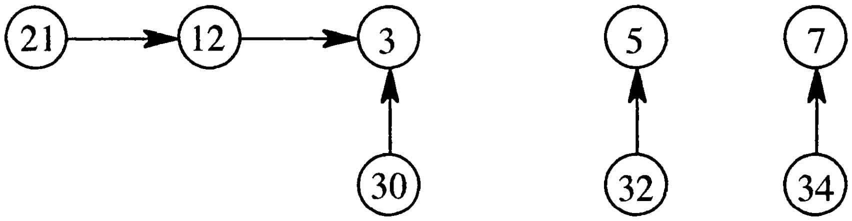

Figure 5.5 Transition graph for [+ + o] vector input.

The first element is [+], which indicates that x(0) > 0.

The second element is [o], which has already been shown to give:

$$x(0) = \frac{u(0)}{k_{01}}$$

which for the present input becomes:

$$x(0) = \frac{u_0}{k_{01}}$$

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
20	+ - + o	$x(0) = \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
19	+ - + +	$\frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} < x(0) < \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
22	+ - o +	$x(0) = \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
25	+ - - +	$\frac{u_0}{k_{01}} < x(0) < \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
16	+ o - +	$x(0) = \frac{u_0}{k_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}}$
34	o + - +	$x(0) = 0$

Table 5.10 Envisionment for the [+ - +] input vector.

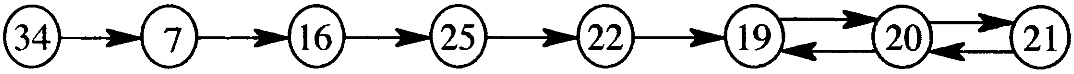

Figure 5.6 Transition graph for the [+ - +] vector input.

The expression for the second derivative (third vector element) is given by equation 5.5:

$$\begin{aligned} x''(0) &= k^2_{01}.x(0) - k_{01}.u(0) + u'(0) \\ &= k^2_{01}.x(0) - k_{01}.u_0 - u_1 \end{aligned}$$

and substituting for x(0) in this gives:

$$\begin{aligned} x''(0) &= k^2_{01}.\frac{u_0}{k_{01}} - k_{01}.u_0 - u_1 \\ &= -u_1 \end{aligned}$$

Therefore $x''(0) = -u_1$, which is less than zero and matches the value of element 3 of the vector, which is [-].

One also has (equation 5.6):

$$\begin{aligned} x'''(0) &= -k^3_{01}.x(0) - k^2_{01}.u(0) + k_{01}.u'(0) + u''(0) \\ &= -k^3_{01}.x(0) + k^2_{01}.u_0 + k_{01}.u_1 + 2.u_2 \end{aligned}$$

and substituting for x(0) here gives

$$\begin{aligned} x'''(0) &= -k^3_{01}.\frac{u_0}{k_{01}} + k^2_{01}.u_0 + k_{01}.u_1 + 2.u_2 \\ &= k_{01}.u_1 + 2.u_2 \end{aligned}$$

Therefore $x'''(0) = k_{01}.u_1 + 2.u_2$, which is positive and matches the fourth element of the vector (which is [+]).

Thus it has been demonstrated that the values of all the vector elements of this state are consistent. This analysis can be applied to any vector in any of the envisionment tables. The interpretation of the states in each of the envisionment tables is exhaustive: there is no vector which does not have an interpretation and no interpretation without a vector. Thus VE generates all and only the valid qualitative states of the system for any particular input function.

The analysis given in this section is based on the Taylor expansion approach to the system identification problem for non-linear compartmental systems (Godfrey 1988).

5.5. The State Space

The previous analysis of the results obtained has concentrated on their interpretation. In the present section this will be built upon to show how a state (or solution) space can be constructed from the interpretations. This space has as axes the initial value of the state variable plus the value and derivatives of the input. The space is also divided into regions by the critical values of the system. The more complex spaces will be built up to by examining a simpler space first and developing from there.

Consider the interpretation of the response to a step input as recorded in table 5.8. All the critical points appear in a single vector (state 14) which is interpreted as:

$$x(0) = \frac{u_0}{k_{01}} \quad \text{----} \; (5.7)$$

which is the equation of a straight line relating x(0) and u_0, with gradient $1/k_{01}$ (the time constant of the system). This relationship is shown in figure 5.8. It should also be noted that because this is a step function the value of u_0 will be constant for any single envisionment, and the trajectory representing the behaviour of the system will be a horizontal straight line headed towards the steady state line from either the left or right, depending on the initial conditions. The steady state line would be reached at t = ∞. The d^ns in figures 5.8, 5.9 and 5.10 represent the vector elements which are zero at the line, or plane, pointed to. The circled numbers are the state numbers from the envisionment tables. The two states marked on the x(0) axis in figure 5.8 represent the responses if u_0 were zero (the responses to a [o o o] vector).

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}}$
12	+ o + -	$x(0) = \frac{u_0}{k_{01}}$
3	+ + + -	$\frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} < x(0) < \frac{u_0}{k_{01}}$
6	+ + o -	$x(0) = \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}}$
9	+ + - -	$\frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} - \frac{u_2}{k^3_{01}} < x(0) < \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}}$
8	+ + - o	$x(0) = \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} - \frac{u_2}{k^3_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} - \frac{u_2}{k^3_{01}}$
34	o + - +	$x(0) = 0,\ x'(0) > \frac{u_1}{k_{01}} + \frac{u_2}{k^2_{01}}$
35	o + - o	$x(0) = 0,\ x'(0) = \frac{u_1}{k_{01}} + \frac{u_2}{k^2_{01}}$
36	o + - -	$x(0) = 0,\ \frac{u_1}{k_{01}} < x'(0) < \frac{u_1}{k_{01}} + \frac{u_2}{k^2_{01}}$
33	o + o -	$x(0) = 0,\ x'(0) = \frac{u_1}{k_{01}}$
30	o + + -	$x(0) = 0,\ x'(0) < \frac{u_1}{k_{01}}$

Table 5.11 Envisionment for the [+ + -] input vector.

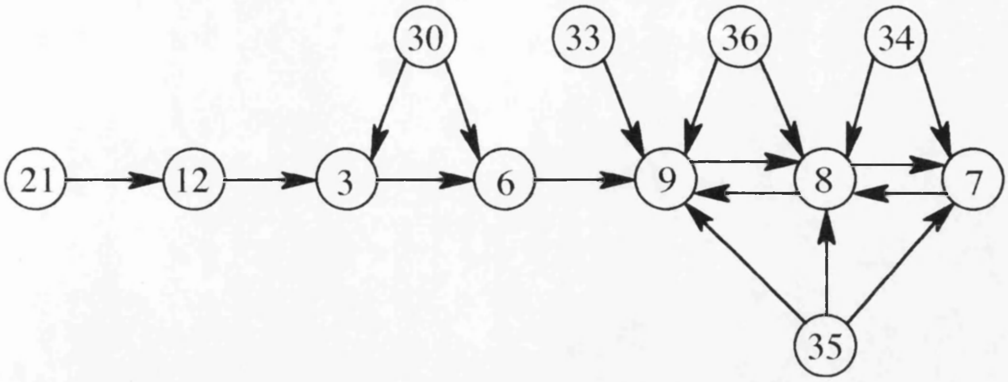

Figure 5.7 Transition graph for the [+ + -] input vector.

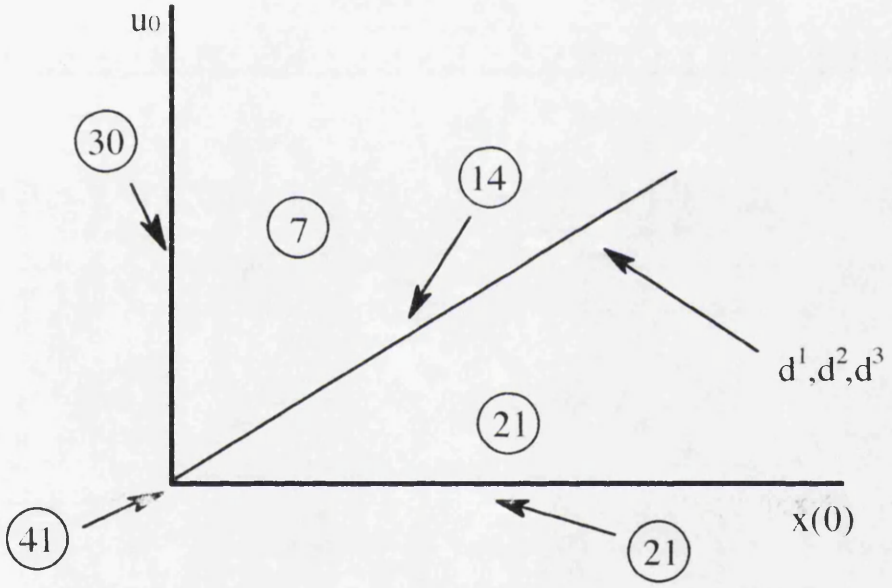

Figure 5.8 The state space for a step input.

To continue the development, consider now the interpretation of a stepped ramp input, [+ + o]. Here, as shown in table 5.9, the first critical point is state 12 and the relationship still represents a straight line through the origin. However, all the critical points of the higher derivatives appear in the vector: state 5, which is interpreted as:

$$x(0) = \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} \quad \text{---- (5.8)}$$

Since u_1 is constant this can be represented as a straight line relating x(0) and u_0, which has gradient $1/k_{01}$ (as before) but intersects the u_0 axis at u_1/k_{01}. Thus for any value of u_1 (constant) the solution space can be drawn in two dimensions with the critical point boundary lines running in parallel. If u_1 is zero, equation 5.8 reduces to equation 5.7 and the situation becomes that depicted in figure 5.8. As u_1 increases the point of intersection increases and the two parallel lines get further apart. If a third axis, the u_1 axis, is added to the solution space the result is as shown in figure 5.9. From this diagram it can be seen that any two dimensional slice perpendicular to the u_1 axis will consist of two parallel lines. In figure 5.9 d^1 points to the boundary representing the critical value of the first derivative and d^2,d^3 point to the boundary representing the critical values of the higher derivatives. This diagram, plus the table of results, highlights the fact that as the number of non zero elements which the input vector contains increases, so the state space evolves by the boundaries representing the critical points of the system "unfolding" (or separating).

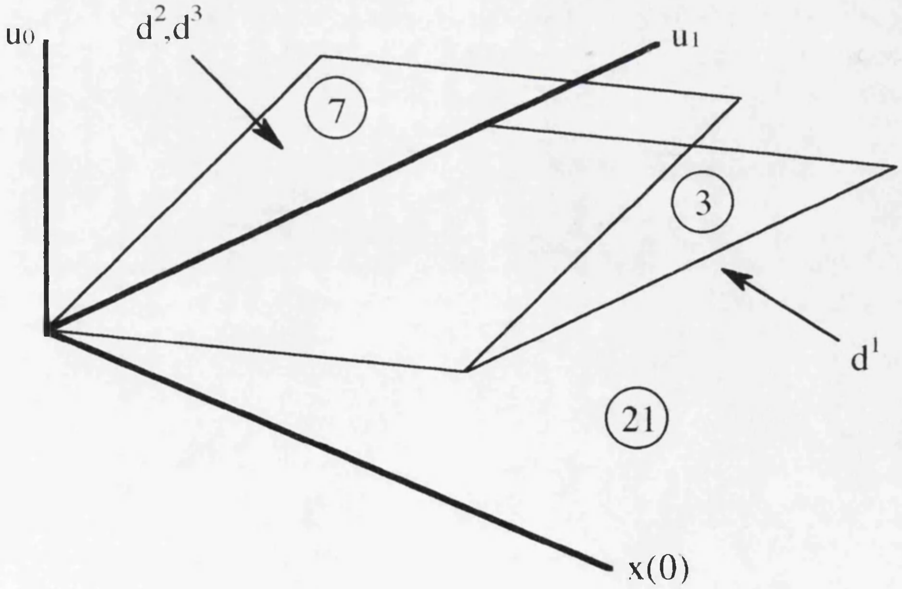

Figure 5.9 The state space for a stepped ramp input.

Continuing this evolution leads to the diagram of figure 5.10. The complete solution space for a three element input vector would require four dimensions to display it (the three axes shown plus the u_2 axis). However, figure 5.10 can be viewed as a three dimensional 'slice' of this four dimensional space (with constant u_2). Here, d^1 once again points to the the boundary representing the critical value of the first derivative; d^2 points to the critical point boundary of the second derivative, and d^3 to that for the third derivative. Reasoning by analogy from the solution space for the ramp input discussed above, if u_2 zero then the boundary pointed to by d^3 will be coincident with that pointed to by d^2 (as in figure 5.9). As u_2 is increased the d^2 and d^3 planes separate, which can be displayed as a series of 'slices' (each one similar to figure 5.10) with the d^2 and d^3 planes successively further apart. Figure 5.10 shows the form of the solution space for any three element input vector with a positive third element; the form of the solution space for the case where the third element is negative may be similarly constructed. The state labels have been omitted from this diagram in the interests of clarity.

5.6. Response to Complex Input Vectors

The responses studied so far have been for single monotonic input vectors. However, in some cases these may not be sufficient to represent the desired input. For example a ramp starting from zero cannot be represented by a single vector (and the case of $x(t) = a.t^2$ has already been mentioned). Vectors may be joined together in accordance with the continuity rules, and the envisionment from the complex vector will consist of all the states from the first vector input along with those states from the second and subsequent vector inputs which are reachable from the states of the first vector input (or immediately previous vector) by application of the continuity rules. As a simple example consider the response to a ramp with value [o] at t = 0.

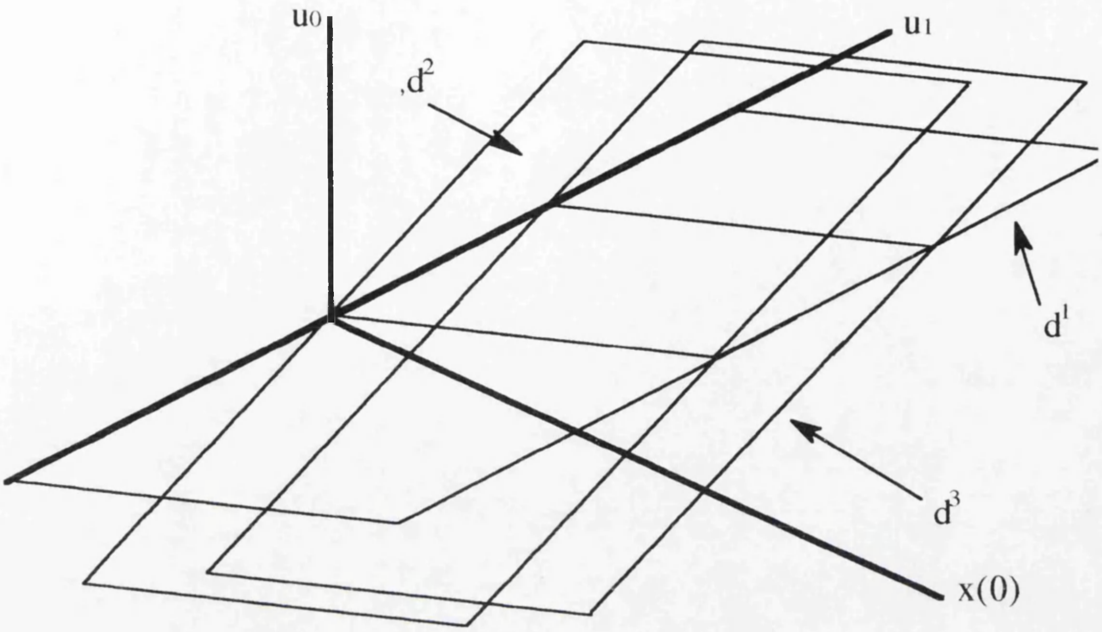

Figure 5.10 The state space for a three element vector.

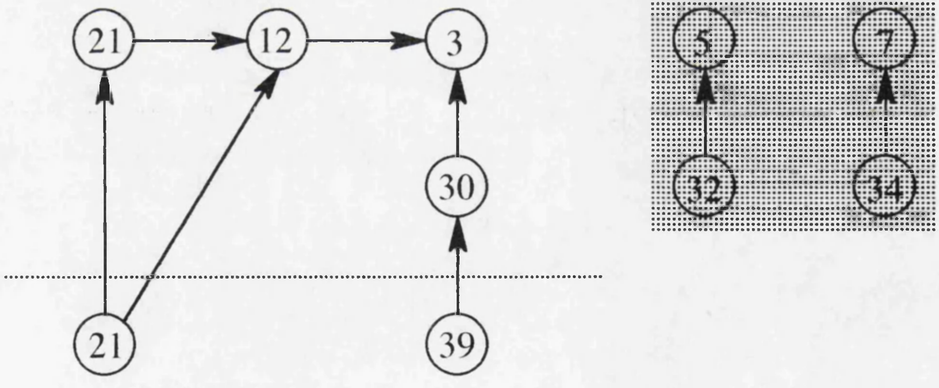

Figure 5.11 Envisionment for the complex input [o + o] → [+ + o].

A ramp commencing from [o] will be represented by the vector [o + o]. However, this vector in instantaneous and must immediately transition to [+ + o]. As soon as this happens the responses must also transit from those for [o + o] to those for

[+ + o], and this is shown in figure 5.11. The states drawn below the horizontal line in figure 5.11 are the envisionment graph for the [o + o] (from envisionment table B.7) vector input, and those states above the line are the envisionment graph for the [+ + o] vector input (figure 5.5). It can be seen that now, instead of there being three possible distinct behaviours when the value of the state variable is zero (as there was for the [+ + o] input vector alone) there is only one possible behaviour (the eliminated states are shown shaded). However, if the initial value of x is positive then there are two possible valid transitions (shown in figure 5.11 by the two arrows leaving state 21 below the line). The conditions for each of these to occur can be considered by remembering that everything in VE is viewed as an interval; thus one may take a slice through figure 5.9 parallel to the x,u_0 plane and explode the diagram at the origin as shown in figure 5.12. From this it can be seen that while both transitions are valid, the transition 21 → 12 is a special case, when the state variable is 'only just' positive. This type of bifurcation in the behaviours also occurs in some response to single input vectors, for example [+ + -] (figure 5.7). The VE algorithm has not been implemented to envision complex input vectors but the discussion of this section has been included for completeness.

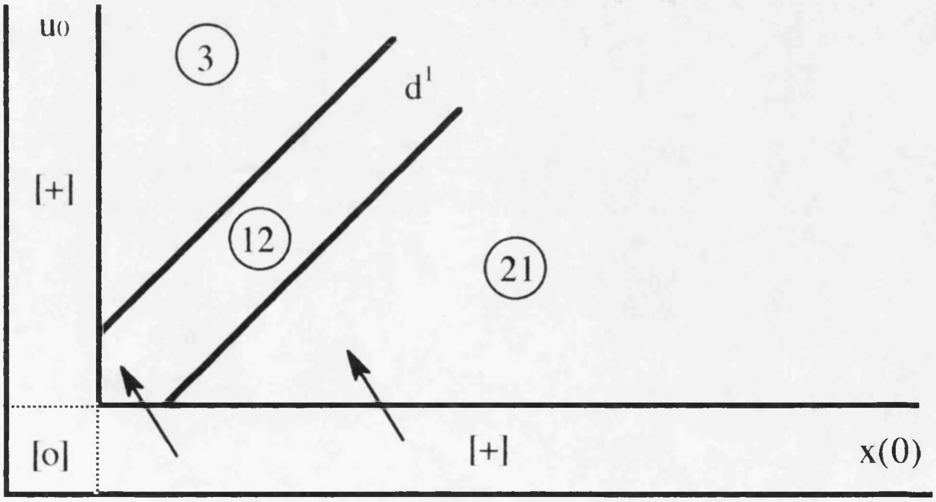

Figure 5.12 Exploded view of the origin for a ramp input.

5.7. Chattering

Chattering is the aimless oscillation about the highest derivative of the system. All qualitative reasoning schemes have exhibited chattering and a great deal of effort has been expended in attempting to eliminate (or at least reduce) the spurious behaviours caused by it. It arises when the input vector is not sufficiently long to allow the valid transitions between states to occur in one direction only (an example of this is seen in the response to the [+ - +] input, figure 5.6). Because the oscillation has the [o] value as a pivot, it would cease if the last two elements of this vector were [o o] (from the transition diagram, figure 4.4). The structure of the system is such that the input is equal to the qualitative sum of the state variable and its derivative (equation 5.1). Also, each element of a vector is the derivative of the element immediately preceding it. Therefore, the qualitative sum of the last two elements of the four element vectors, used to represent the state variable in this chapter, must equal the qualitative value of the last element of the input vector. The only way these last two elements can be zero is if the last element of the input vector has the qualitative value [o].

This result also indicates the minimum length of input vector required to envision the system. Strictly speaking, one only needs a two element input vector to represent a step change. However, humans are used to thinking of the curvature of functions; therefore, pragmatically, three elements are used as the minimum length of vector. A similar result was obtained by Wiegand (1991). From this argument it follows that inputs representing exponentials cannot give non-chattering responses because no vector of finite length can represent an exponential and also have a zero as the last element. A possible solution to this is to recognise that for dissipating linear time invariant compartmental systems, such as those considered here, the responses are well behaved and once one of the fixed boundaries has been crossed in one direction, the behaviour will not double back on itself and recross it in the opposite direction. This is a heuristic solution.

5.8. Summary

In this chapter the qualitative response of a single compartment system to the 14 input vectors whose value is either positive, or zero with the first non-zero derivative positive have been analysed.

A simple parallel R-C circuit was used to demonstrate the advantages of leaving the value of the integral vector undefined when performing qualitative analysis. When this value was specified in accordance with the integration rule, the qualitative analysis was found to generate spurious states. It was argued that ambiguities should be treated as friends rather than enemies; and used to discover all the states in which the system may exist, for any particular input.

The output from an envisionment is a set of states (generated by qualitative analysis) connected together to form a graph representing the behaviours of the system (generated by transition analysis). There is no indication in the graph as to what conditions are required for a particular behaviour to pertain; therefore the states in the graph need to be interpreted. Qualitative vectors represent continuously differentiable, real valued functions of time and to aid interpretation an analytical analogue of the vectors, in the form of a polynomial series with general coefficients was used.

The behaviour of a state variable over a time interval is dependent only on its value at the beginning of the interval and not on how that value was reached. Therefore, each vector in the envisionment graph may be considered to start a behaviour, for some initial condition of the system. To interpret the vectors the differential equation describing the system was analysed at t=0. The critical points of the system occur when one, or more, of the derivatives of the state variable is equal to zero. If one successively differentiates the differential equation describing the system, and sets the corresponding derivatives to zero at t=0, an initial value for the state variable can be calculated which would start the system behaviour from that critical value. Each element of a qualitative vector has a value from the set {+ o -}, with zero as the only landmark. A vector therefore represents a critical value of the system if one, or more, of the elements of the vector representing the state variable is zero. By comparing the vectors for critical points with the value of the state variable obtained from analysing the system at t=0 an interpretation for each of these vectors is obtained. Because the only other values an element can take are '+' and '-' the interpretation of non critical point vectors is an initial value for the state variable which is either greater than or less than its value at the critical point. This analysis was carried out for every vector in the envisionment graph. It was thereby shown that each vector has a valid interpretation and no interpretation is without a corresponding vector. Thus VE finds all and only the valid states of a single compartment system. The form of the input vector was represented by an appropriate polynomial.

From these results a solution space was constructed. The axes of this space are the initial value of the state variable and the co-efficients of the input polynomial. It was shown that the expression for the critical point values of the state variable could be given in a form which related the axes of the solution space.

The response of a one compartment system to complex inputs (that is, inputs consisting of more than one vector) was discussed, and it was demonstrated that the envisionment graph consists of those vectors in the individual envisionment graphs which adhere to the continuity rules when the transition from one input vector to the next takes place.

The problem of chattering was also discussed. It was argued that an envisionment without chattering will be obtained if an input vector in which the last element is [o] is used. A vector representing an exponential cannot meet this criterion because it would consist of an infinite series of non-zero elements. To overcome this problem a 'non-return filter' which only retains those transitions in the envisionment which correspond to the majority direction was suggested. This solution is based on a knowledge of the behaviour of a single compartment linear time invariant system.

CHAPTER 6

APPLICATION TO TWO COMPARTMENT SYSTEMS

6.1. Introduction

While single compartment models are useful as representations of whole body uptake and elimination kinetics, it will often be the case that a larger number of compartments is required to model the system of interest. Indeed, one of the aims of this projects was to realise a qualitative model of the Mapleson analogue, which is a four compartment system. In this chapter a further step towards that goal is described.

The principles discussed in the last chapter are extended to multicompartment systems and applied to two examples of two compartment systems. There are several topologies possible for a two compartment system. However, here the discussion is restricted to systems with a single input to the first compartment and elimination only from the second compartment. The only difference between the two systems described is that in one the flow between the compartments is unidirectional (a cascaded system); whereas in the other the flow is bidirectional (a coupled system). These are shown in figures 6.1 and 6.2. The responses to a step input are examined for these two models, though the approach can be used for all the inputs described previously.

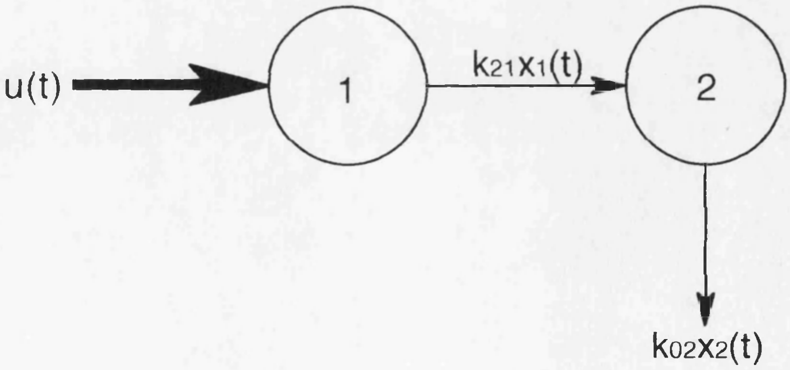

Figure 6.1 A two compartment cascaded system.

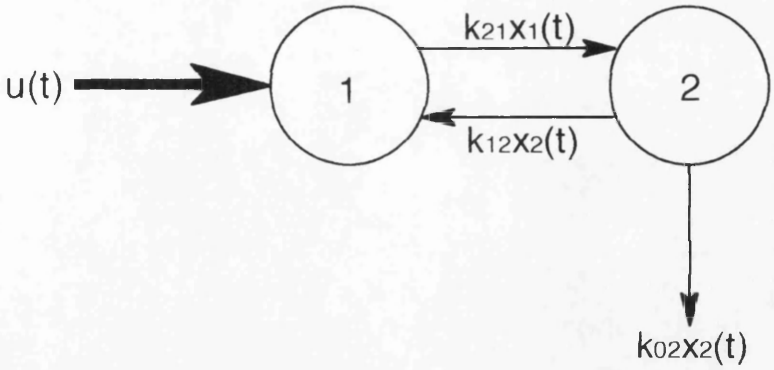

Figure 6.2 A two compartment coupled system.

6.2. A Description of Multicompartment Systems

In chapter two it was seen that a two compartment system could be described either as two differential equations or in a state vector representation. The latter form is a compact means of representing any order of compartmental system and will therefore be used in this section to indicate how the methods described in the previous chapter can be extended to multicompartment systems.

The state equation for a general compartmental system is written as:

$$\underline{x}' = \mathbf{A}\underline{x} + \mathbf{B}\underline{u}$$

where **A** contains the rate constants of the system, **B** is assumed to be the identity matrix and $\underline{u}$ is the input to the system. Once again the observation equation is omitted for ease of presentation.

This equation can be differentiated to give:

$$\begin{aligned} \underline{x}'' &= \mathbf{A}\underline{x}' + \mathbf{B}\underline{u}' \\ &= \mathbf{A}(\mathbf{A}\underline{x} + \mathbf{B}\underline{u}) + \mathbf{B}\underline{u}' \\ &= \mathbf{A}^2\underline{x} + \mathbf{A}.\mathbf{B}\underline{u} + \mathbf{B}\underline{u}' \end{aligned}$$

and differentiating again gives:

$$
\begin{aligned}
\underline{x}''' &= \mathbf{A}\underline{x}'' + \mathbf{B}\underline{u}'' \\
&= \mathbf{A}(\mathbf{A}^2\underline{x} + \mathbf{A.B}\underline{u} + \mathbf{B}\underline{u}') + \mathbf{B}\underline{u}'' \\
&= \mathbf{A}^3\underline{x} + \mathbf{A}^2\mathbf{.B}\underline{u} + \mathbf{A.B}\underline{u}' + \mathbf{B}\underline{u}''
\end{aligned}
$$

A comparison of these equations and equations 5.4, 5.5, and 5.6 will reveal their similarities. However, because each vector on the left hand side of the above equations will contain more than one state derivative, it is only rarely that any of them will be a zero vector. Therefore, the critical points of the system have to be examined by setting each element of the selected derivative vector to zero individually and examining the associated equation. This is equivalent to differentiating each individual equation that made up the original state equation, and setting the result equal to zero. This is what will be done in the sequel. The result is state space for the qualitative envisionment, the axes of which are the state variables and the elements of the qualitative input vector (the natural extension of the single compartment case).

A full analysis of the most general case would be too complex and could not be represented graphically (even for the two compartment case). Therefore, two examples of qualitative responses to a step input will be analysed: first a cascaded system, and then a coupled system.

6.3. A Two Compartment Cascaded System

The equations governing a two compartment system (figure 6.1) are:

$$x'_1(t) = -k_{21}.x_1(t) + u_0 \qquad \text{---- (6.1)}$$

and

$$x'_2(t) = k_{21}.x_1(t) - k_{02}.x_2(t) \qquad \text{---- (6.2)}$$

which yields the following qualitative equations:

$$[x1\_dot] = [-] \otimes [x1] \oplus [uo]$$

and

$$[x2\_dot] = [x1] \Theta [x2]$$

No.	State Tuple	x_1	x_3	No.	State Tuple	x_1	x_3
1	21,21	+ - + -	+ - + -	14	7,6	+ + - +	+ + o -
2	21,20	+ - + -	+ - + o	15	7,9	+ + - +	+ + - -
3	21,19	+ - + -	+ - + +	16	7,8	+ + - +	+ + - o
4	21,22	+ - + -	+ - o+	17	7,7	+ + - +	+ + - +
5	21,25	+ - + -	+ - - +	18	21,34	+ - + -	o + - +
6	21,16	+ - + -	+ o - +	19	14,34	+ o o o	o + - +
7	21,7	+ - + -	+ + - +	20	7,34	+ + - +	o + - +
8	14,21	+ o o o	+ - + -	21	7,35	+ + - +	o + - o
9	14,14	+ o o o	+ o o o	22	7,36	+ + - +	o + - -
10	14,7	+ o o o	+ + - +	23	7,33	+ + - +	o + o -
11	7,21	+ + - +	+ - + -	24	7,30	+ + - +	o + + -
12	7,12	+ + - +	+ o + -	25	34,39	o + - +	o o + -
13	7,3	+ + - +	+ + + -	26	34,21	o + - +	+ - + -

Table 6.1 The envisionment of a step input to a cascaded system.

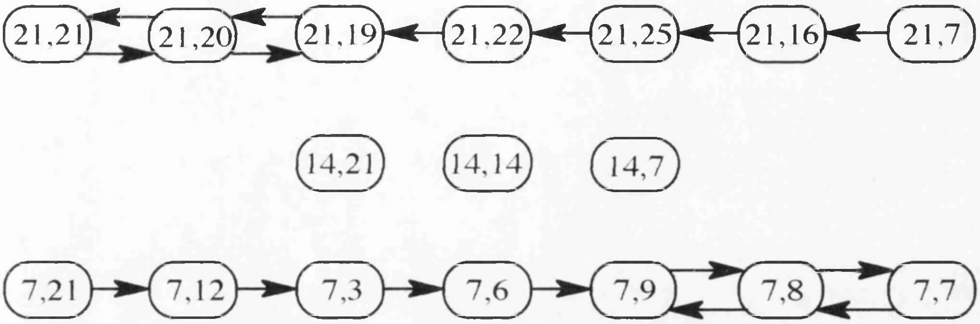

Figure 6.3 Transition graph for a step input to a cascaded system.

The result of the envisionment of a step input is shown in table 6.1. These results require interpretation and once again this is carried out by means of examining the critical points at t=0 (though the interpretations are not shown in the table due to the size of the table and their complexity). Looking first at equation 6.1 and setting $x'_1(0)$ to zero gives:

$$0 = -k_{21}.x_1(0) + u_0$$

$$\Rightarrow x_1(0) = \frac{u_0}{k_{21}}$$

similarly equation 6.2 becomes:

$$0 = k_{21}.x_1(0) - k_{02}.x_2(0)$$

$$\Rightarrow x_2(0) = \frac{k_{21}}{k_{02}}.x_1(0)$$

at steady state $x'_1(0) = x'_2(0) = 0$, in which case

$$x_2(0) = \frac{k_{21}}{k_{02}} \cdot \frac{u_0}{k_{21}}$$

$$\Rightarrow \quad x_2(0) = \frac{u_0}{k_{02}}$$

Differentiating equation 6.1 at t=0 gives:

$$\begin{aligned} x_1''(0) &= -k_{21}.x'_1(0) \\ &= -k_{21}(-k_{21}.x_1(0) + u_0) \\ &= k^2{}_{21}.x_1(0) - k_{21}.u_0 \end{aligned}$$

This becomes a critical point when $x_1''(0) = 0$

$$0 = k^2{}_{21}.x_1(0) - k_{21}.u_0$$

which once again implies

$$x_1(0) = \frac{u_0}{k_{21}}$$

This is the expected result because the only thing affecting the first compartment is the input; thus its response will be the same as that of a single compartment system to a step change. Therefore one need analyze only the second and third critical points for the second compartment to achieve a complete analysis.

Now differentiating equation 6.2 at t=0 gives:

$$
\begin{aligned}
x_2''(0) &= k_{21}.x'_1(0) - k_{02}.x'_2(0) \\
&= k_{21}.(-k_{21}.x_1(0) + u_0) - k_{02}.(k_{21}.x_1(0) - k_{02}.x_2(0)) \\
&= -k_{21}.(k_{21} + k_{02}).x_1(0) + k_{21}.u_0 + k^2_{02}.x_2(0)
\end{aligned}
$$

Now setting $x_2''(0) = 0$ and re-arranging one gets:

$$x_2(0) = \frac{k_{21}}{k^2_{02}}.(k_{21} + k_{02}).x_1(0) - \frac{k_{21}}{k^2_{02}}.u_0$$

Differentiating again:

$$
\begin{aligned}
x_2'''(0) &= k_{21}.x''_1(0) - k_{02}.x''_2(0) \\
&= k_{21}.(k^2_{21}.x_1(0) - k_{21}.u_0) \\
&\quad - k_{02}.(-k_{21}.(k_{21} + k_{02}).x_1(0) + k_{21}.u_0 + k^2_{02}.x_2(0)) \\
&= k_{21}.(k^2_{21} + k_{02}.k_{21} + k^2_{02}).x_1(0) \\
&\quad - k^3_{02}.x_2(0) - k_{21}.(k_{21} + k_{02}).u_0
\end{aligned}
$$

Now setting $x'''_2(0) = 0$ and re-arranging:

$$x_2(0) = \frac{k_{21}}{k^3_{02}}.(k^2_{21} + k_{02}.k_{21} + k^2_{02}).x_1(0) - \frac{k_{21}}{k^3_{02}}.(k_{21} + k_{02}).u_0$$

Thus, once again, one has a set of relationships which can be drawn in a state space as shown in figure 6.4. In figures 6.4 and 6.7 the circled numbers refer to the first and fifth columns of tables 6.1 and 6.2. The d_{ij} in figures 6.4, 6.5 and 6.7 refer to the system critical points of the j^th^ element of the i^th^ state variable vector. All the critical points of x_1 lie on a vertical straight line because, as noted above, the only thing affecting this state variable is the input. Also, the steady state vector is the point where all the critical point lines are co-incident. The spatial relationship of these lines can be analysed by examining the critical points when x_2 is zero.

For the first critical point of x_2

$$x_2(0) = \frac{k_{21}}{k_{02}} . x_1(0)$$

therefore $x_1 = 0$ if $x_2 = 0$

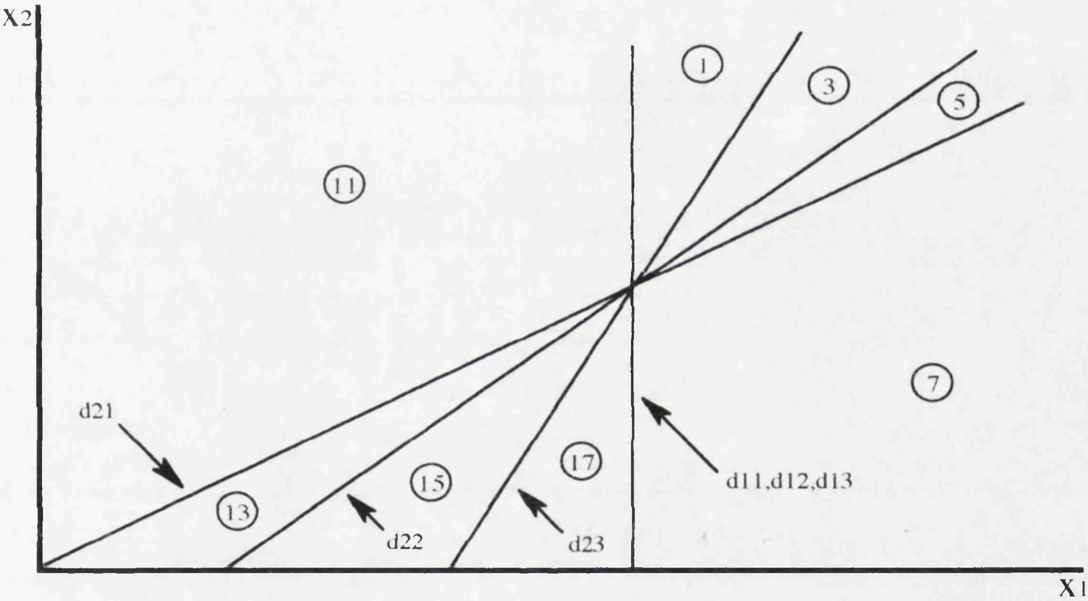

Figure 6.4 The State space of the cascaded system for constant u_0.

For the second critical point of x_2

$$x_1(0) = \frac{u_0}{(k_{21} + k_{02})}$$

which is greater than zero.

For the third critical point of x_2

$$x_1(0) = \frac{(k_{21} + k_{02}).u_0}{(k^2_{21} + k_{02}.k_{21} + k^2_{02})}$$

which is greater than the value for the second critical point.

Also, for all the critical points of x_1

$$x_1(0) = \frac{u_0}{k_{21}}$$

which is greater than any of the critical points for x_2. Therefore, the critical point lines of the state space will always have the spatial relationship shown in figure 6.4 for a step input to a cascaded system.

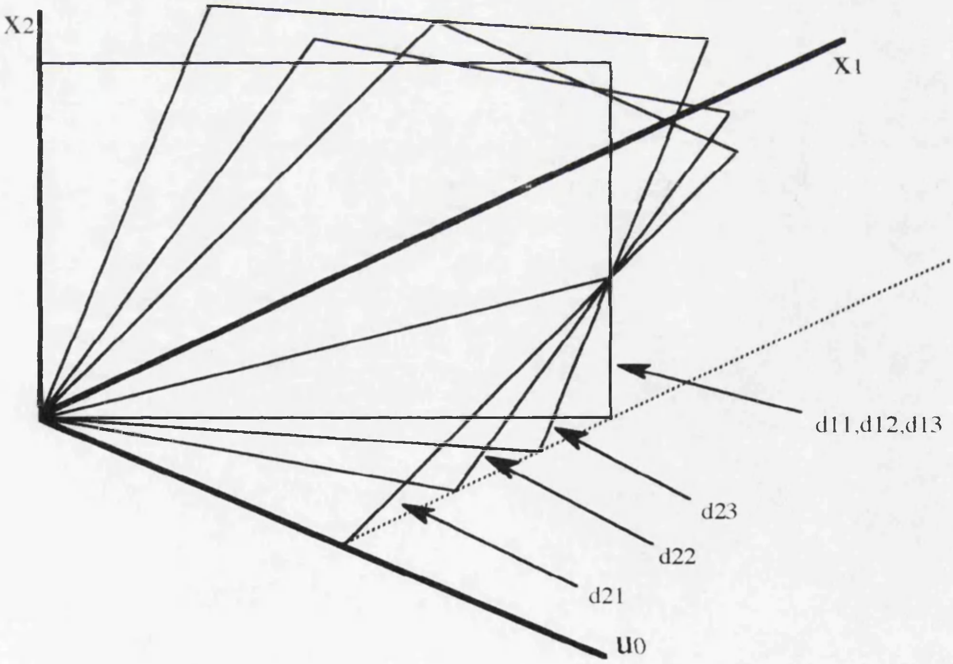

Figure 6.5 Full state space for a step input to a cascaded system.

Figure 6.4 is the envisionment state space for a particular value of u_0; that is, it is a slice through the full state space shown n figure 6.5. This shows how the solution space is divided for all values of u_0. The state transition graph for the system is as shown in figure 6.3. It may be seen from this that the chattering behaviour has appeared even for a step input. This is because the output from the first compartment is the input to the second, and will have the form of an exponential. As was discussed in the previous chapter, and exponential must be represented by a vector of infinite length in which the signs of the elements alternate. Therefore any finite length approximation will not end in a zero. Since this is the criterion for chattering not to occur in the envisionment there is no adequate length of input vector for which chattering will not occur in a multicompartment system. However, because the envisionment graph gives a global picture, a 'no return' filter like that suggested in chapter 5 could be employed to prevent chattering from occurring.

6.4. A Two Compartment Coupled System

The equations governing a two compartment coupled system (figure 6.2) are:

$$x'_1(t) = -k_{21}.x_1(t) + k_{12}.x_2(t) + u_0 \quad \text{----} \ (6.3)$$

and

$$x'_2(t) = k_{21}.x_1(t) - (k_{02} + k_{12}).x_2(t) \quad \text{----} \ (6.4)$$

This yields the following qualitative equations (where [x12] is the qualitative version of $k_{12}.x_2(t)$ and [x02] is the qualitative counterpart of $k_{02}.x_2(t)$):

$$[x1\_dot] = [-] \otimes [x1] \oplus [x12] \oplus [uo] \quad \text{----} \ (6.5)$$

and

$$[x2\_dot] = [x1] \ \Theta \ [x12] \ \Theta \ [x02] \quad \text{----} \ (6.6)$$

No.	State Tuple	x_1	x_2	No.	State Tuple	x_1	x_2
1	7,21	+ + - +	+ - + -	20	8,7	+ + - o	+ + - +
2	16,21	+ o - +	+ - + -	21	9,7	+ + - -	+ + - +
3	25,21	+ - - +	+ - + -	22	6,7	+ + o -	+ + - +
4	22,21	+ - o +	+ - + -	23	3,7	+ + + -	+ + - +
5	19,21	+ - + +	+ - + -	24	12,7	+ o + -	+ + - +
6	20,21	+ - + o	+ - + -	25	21,7	+ - + -	+ + - +
7	21,21	+ - + -	+ - + -	26	21,34	+ - + -	o + - +
8	21,20	+ - + -	+ - + o	27	12,34	+ o + -	o + - +
9	21,19	+ - + -	+ - + +	28	3,34	+ + + -	o + - +
10	21,22	+ - + -	+ - o +	29	6,34	+ + o -	o + - +
11	21,25	+ - + -	+ - - +	30	9,34	+ + - -	o + - +
12	21,16	+ - + -	+ o - +	31	8,34	+ + - o	o + - +
13	14,14	+ o o o	+ o o o	32	7,34	+ + - +	o + - +
14	7,12	+ + - +	+ 0 + -	33	7,35	+ + - +	o + - o
15	7,3	+ + - +	+ + + -	34	7,36	+ + - +	o + - -
16	7,6	+ + - +	+ + o -	35	7,33	+ + - +	o + o -
17	7,9	+ + - +	+ + - -	36	7,30	+ + - +	o + + -
18	7,8	+ + - +	+ + - o	37	34,39	o + - +	o o + -
19	7,7	+ + - +	+ + - +	38	34,21	o + - +	+ - + -

Table 6.2 Envisionment of a step input to a coupled system.

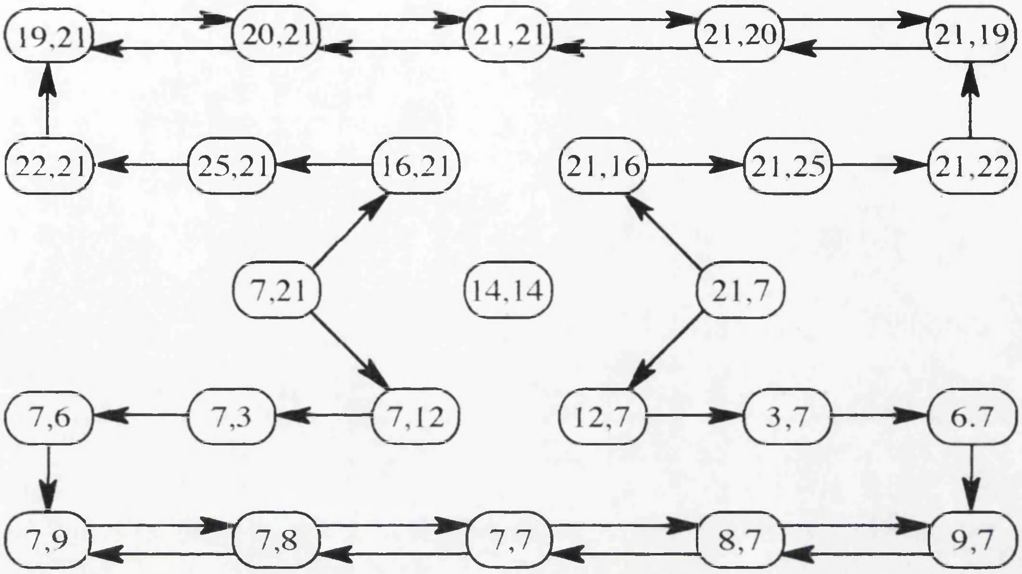

Figure 6.6 Transition graph for a step input to a coupled system.

The results of the envisionment are shown in table 6.2 and the transition graph is shown in figure 6.6. The interpretation of the results is carried out in exactly the same way as for the cascaded system. The state space is shown in figure 6.7; from which it may be seen that the critical points of the first compartment are no longer co-linear. This is because there are now two flows into compartment 1: the system input and the feedback flow from compartment 2.

The results discussed in this section were calculated by hand because a bug in the present implementation causes a plethora of spurious states to be generated. The reason for the spurious states being generated can be seen if equation 6.5 is re-arranged to give the two qualitative equations:

$$[uo] = [x1\_dot] \oplus [x1] \Theta [x12] \quad ---- \quad (6.7)$$

and

$$[x2\_dot] = [x1] \Theta [x12] \Theta [x02] \quad ---- \quad (6.8)$$

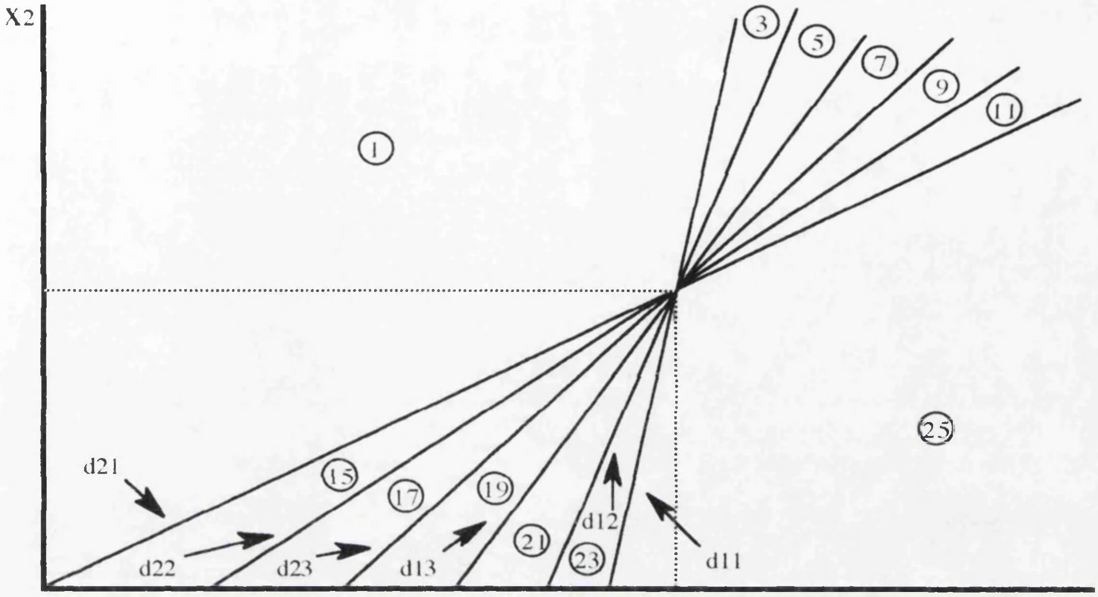

Figure 6.7 State space for a coupled system with constant u_0.

Here, the expression [x1] Θ [x12] appears in both equations. However, V.E. does not recognise this and allows this expression to take values in one equation which are excluded by the other. For example, consider the state tuple <6,7> (state number 22 in table 6.2); substituting these vectors into equations 6.7 and 6.8 gives:

$$\begin{bmatrix}+\\o\\o\end{bmatrix} = \begin{bmatrix}+\\o\\+\end{bmatrix} \oplus \begin{bmatrix}+\\+\\o\end{bmatrix} \Theta \begin{bmatrix}+\\+\\-\end{bmatrix}$$

and

$$\begin{bmatrix}+\\-\\+\end{bmatrix} = \begin{bmatrix}+\\+\\o\end{bmatrix} \Theta \begin{bmatrix}+\\+\\-\end{bmatrix} \Theta \begin{bmatrix}+\\+\\-\end{bmatrix}$$

Examining the second row of these sums one can note that in the first sum [+] Θ [+] must be [o]. If this is substituted into the second sum one gets [o] Θ [+] which is [-]. However, if this connection is not recognised then the result of [+] Θ [+] Θ [+] in the second sum could be [+], [o] or [-]; which is the result that V.E. tries to produces. This is also the reason why the two appearances of the state variable in equation 6.6 are kept separate. If they were not then the minus in the second row of the second sum would not be calculated and the state would be missed.

6.5. Summary

In this chapter the method described for the interpretation of the envisionment of a single compartment system were applied to two compartment systems. Two examples were used to illustrate the technique: a cascaded system and a coupled system. Each one was analysed in response to a step input. The interpretation of the envisionment of each system is performed by applying the methods of chapter 5 to each of the equations in the system description individually. The had to be done because the critical points for the two compartments do not necessarily appear together.

The solution space for each system was constructed and the location of each state shown. It was also demonstrated that the critical point lines in the solution space must always have the same spatial relationship.

Chattering behaviour appeared in the envisionment graphs of both systems. This was because at least one compartment in each system had an exponential input which always causes chattering. However, the solution proposed in chapter 5 is also applicable here.

CHAPTER 7

SUMMARY, CONCLUSIONS AND FUTURE WORK

7.1. Summary

The work described in this thesis concerns the application of qualitative reasoning techniques to compartmental modelling, the goal being to construct "intelligent" versions of such systems. This aim is a subgoal of the original project specification, which was to construct an intelligent simulation of an anaesthetised patient undergoing surgery.This proved to be too large an undertaking, so the scope of the project work was narrowed to be a thorough analysis of the qualitative behaviour of linear time invariant compartmental systems. To place this work in its general context and discuss the distinction between qualitative simulation and envisionment, compartmental modelling and qualitative reasoning are briefly described.

The method used for this research: Vector Envisionment (VE) (Morgan 1988) is described. A qualitative vector consists of the signs of the value and successively higher derivatives of a monotonic function to be represented. A working length of three elements is used here. VE is divided into two phases: qualitative analysis, which finds all the states in which the system may exist; and transition analysis, which finds the valid transitions between the states. The resulting digraph is the envisionment. It is shown that an attempt at qualitative simulation will produce spurious behaviours where an envisionment will not; thus justifying the use of envisionment in this project. It is also shown that to interpret the qualitative vectors an appropriate representation is a truncated polynomial with general co-efficients.

The method is applied to examples of one and two compartment systems and it is demonstrated that the states generated give complete coverage of a solution space whose axes are the co-efficients of the input vector plus the state variables of the system. Unfortunately, chattering behaviour is found to occur for first order inputs because such vectors do not terminate, and finite length is a condition of non-chattering behaviour. However, it is suggested that because the envisionment gives a global picture such chattering can be overcome by means of a 'non return' filter which only sustains the transition arcs if they are in the majority direction.

7.2. Conclusions

From the analysis presented in this thesis the following conclusions can be drawn:

7.2.1. Vector envisionment provides a method of modelling linear time invariant compartmental systems qualitatively which gives complete coverage of the solution space, and finds all and only those states in which a system may validly exist for any particular monotonic input vector This has been shown exhaustively for single compartment systems and for two examples of two compartment systems.

7.2.2. The validity of the states in the envisionment and their location in the solution space can be interpreted by examining the critical points of the system at t=0 and representing each vector by a truncated polynomial with general co-efficients.

7.2.3. Complex inputs can be handled by connecting the envisionment graphs of the simple inputs (of which the complex input is made up) in accordance with the continuity rules.

7.2.4. The problem of chattering can be overcome by means of a "non-return" filter. This means that for linear time invariant systems VE would give all and only those behaviours of the system which were valid (a fundamental goal of qualitative reasoning).

7.3. Suggestions for future work

This thesis has dealt with only a small part of the potential areas in which VE might be utilised. The following are some suggestions for the next steps required to extend the approach.

The first thing which should be tackled is to correct the bug in the implementation of VE so that it will handle coupled systems correctly. Following this it is a priority to implement the "non-return" filter, which is essential to the correct operation of the qualitative reasoning paradigm.

Next come those actions which should be performed to round off the work in regard to compartmental systems. A more detailed analysis of two compartment systems (all the different topologies) and an analysis of three compartment systems should be carried out. From this I would expect a pattern to emerge from which a general law for the envisionment for linear time invariant compartmental systems could be discerned. Such a law, or structure, would be a very important tool in constructing systems for teaching or control. Once this was done the Mapleson analogue (a fourth order mamillary system) should be straightforward to envision.

Non-linear models have not been tackled in this project, but represent an important class of compartmental systems. Therefore, the application of VE to these systems should be examined.

REFERENCES

Atkins, G. L. (1969) *Multicompartment Models for Biological Systems*, London: Methuen.

Beneken, J. E. W. & Rideout, V. C. (1968) The use of multiple models in cardiovascular system studies: Transport and perturbation methods. *IEEE Transactions on Bio-medical Engineering*, **BME-15**, 281-189.

Cohn, A. G. (1989) Approaches to qualitative reasoning. *Artificial Intelligence Review*, **3**, 177-232.

Davis, N. R. (1987) *The Pharmacokinetics of Injected Analgesics*, Ph.D. Thesis, University of Wales.

de Kleer, J (1975) *Qualitative and quantitative reasoning in classical mechanics*. MIT AI Laboratory, **TR-51**

de Kleer, J. (1984) A qualitative physics based on confluences. *Artificial Intelligence*, **24**, 7-83.

Forbus, K. D. (1984) Qualitative Process Theory. *Artificial Intelligence*, **24**, 85-186.

Franklin, G. F. Powell, J. D. & Emami-Naeini A (1991) *Feedback Control of Dynamic Systems (Second Edition)*, Addison-Wesley.

Gaba, D. M. & De Anda, A. (1988) A Comprehensive Anaesthesia Simulator Environment: Re-creating the Operating Room for Research and Training. *Anesthesiology*, **69**, 387-394.

Godfrey, K. R. (1983) *Compartmental Models and their Application,*
London: Academic Press.

Hayes, P. J. (1979) The naive physics manifesto. In *Expert Systems in the Microelectronic Age*, ed. Michie, D. Pp. 242-270 Edinburgh University Press.

Hayes, P. J. (1985) The second naive physics manifesto. In *Formal Theories of the Commonsense World*, ed. Hobbs, J. R. & Moore R. C. Pp. 1-36
Norwood, NJ: Ablex Publishing Corp.

Hull, C. J. (1982) Pharmacodynamics of Non-depolarizing Neuromuscular Blocking Agents. *British Journal of Anaesthesia*, **54**, 169-182.

Kety, S. S. (1951) The theory and application of the exchange of inert gas at the lungs and tissues. *Pharmacological Review*, **3**, 1-41.

Kuipers, B. J. (1984) Commonsense reasoning about causality: Deriving behavior from structure. *Artificial Intelligence*, **24**, 169-203.

Kuipers, B. J. (1986) Qualitative simulation. *Artifiacial Intelligence*, **29**, 289-338.

Kuipers, B. J. (1987a) Qualitative simulation as causal explanation. *IEEE Transactions on Systems Man and Cybernetics*, **SMC-17**, 432-444.

Kuipers, B. J. (1987b) New reasoning methods for artifcial intelligence in medicine. *International Jounal of Man-Machine Studies*, **26**, 707-718.

Kuipers, B. J. & Chui, C. (1987) Taming intractable branching in qualitative simulation. In *Proceedings of IJCAI-87*, Pp. 23-28 Milan, Italy.

Kuipers, B. J. & Berleant, D. (1988) Using incomplete quantitatve knowledge in qualitative reasoning. In *Proceedings of AAAI-88*, Pp. 324-329 Saint Paul, Minn.

Lee, W. W. & Kuipers, B. J. (1988) Non-intersection trajectories in qualitative phase space: A global constraint for qualitative simulation. In *Proceedings of AAAI-88*, Pp. 286-291 Saint Paul, Minn.

McCarthy, J. (1987) Generality in artificial intelligence. In *The ACM Turing Award Lectures the First Twenty Years, 1966 to 1985*, 257-267, ACM Press.

McCarthy, J. & Hayes, P. J. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, ed Michie, D. Elsevier.

Mapleson, W. W. (1963) An electric analogue for uptake and exchange of inert gases and other agents. *Journal of Applied Physiology*, **18**, 197-204.

Mapleson, W. W. (1973) Circulation-time models of the uptake of inhaled anaesthetics and data for quantifying them. *British Journal of Aneasthesia*, **45**, 319-333.

Mapleson, W. W. (1973) Circulation-time models of the uptake of inhaled anaesthetics. (letter), *British Journal of Aneasthesia*, **50**, 731.

Mapleson, W. W. (1984) Pharmacokinetics of inhaled anaesthetics. In *Pharmacokinetics of Anaesthetics*, ed. Prys-Roberts, C. & Hug, C. C. Ch. 5 Oxford: Blackwells.

Mapleson, W. W. Chilcoat, R. T. Lunn, J. N. Blewett, M. C. Khatib, M. T. & Willis, B. A. (1980) Computer assistance in the control of depth of anaesthesia. *British Journal of Anaesthesia*, **52**, 234.

Miller , R. D. (1982) Pharmacokinetics of competitive muscle relaxants. *British Journal of Anaesthesia*, **54**, 161-167.

Morgan, A. J. (1987) Predicting the behaviour of dynamic systems with qualitative vectors. In *Advances in Artificial Intelligence*, ed.Hallam, J. & Mellish C. Pp. 81-95.

Morgan, A. J. (1988) *The Qualitative Behaviour of Dynamic Physical Systems*, Ph.D. Thesis, Cambridge University.

Raiman, O. (1986) Order of magnitude reasoning. In *Proceedings of AAAI-86*, Pp. 100-105 Philadelphia, PA.

Schwid, H. A. (1987) A flight simulator for general anaesthesia training. *Computers and Biomedical Research*, **20**, 64-75.

Shen, Q (1991) *Fuzzy Qualitative Simulation and Diagnosis of Continuous Dynamic Systems*, Ph.D. Thesis, Heriot-Watt University.

Struss, P. (1988a) Mathematical aspects of qualitative reasoning, *Artificial Intelligence in Engineering*, **3**, -170

Struss, P. (1988b) Global filter for qualitative behaviors. In *Proceedings of AAAI-88*, Saint Paul, Minn, 275-280.

Weld, D. S. (1990) Exaggeration. *Artificial Intelligence*, **43**, 311-368.

Wiegand, M. E. (1991) *Constructive Qualitative Simulation of Continuous Dynamic Systems*, Ph.D. Thesis, Heriot-Watt University.

Zwart, A. Smith, N. T. & Beneken, J. E. W. (1972) Multiple model approach to uptake and distribution of halothane: The use of an analogue computer. *Computers and Biomedical Research*, **5**, 228-238.

APPENDIX A

RESULTS OF VECTOR ENVISIONMENT

This appendix contains the ouput from VE for all the input vectors to a one compartment system, plus the response of a cascaded two compartment system to a step input. The state numbers in the 'successors' column refer to the number of the state label(s) in column 1.

```
 Generating states ......
16 states generated.
 Checking continuity .....
 Input:  (+ + +)
====
State        X[T]          X-DOT          Successors
  1          (+ + +)       (+ + +)          (3)
  2          (0 + +)       (+ + +)          (1 3)
  3          (+ + +)       (+ + 0)          (1 5)
  4          (0 + +)       (+ + 0)          (1 3 5)
  5          (+ + +)       (+ + -)          (3)
  6          (0 + +)       (+ + -)          (3 5)
  7          (+ + 0)       (+ 0 +)          (1)
  8          (0 + 0)       (+ 0 +)          (1)
  9          (+ + -)       (+ - +)          (7)
  10         (0 + -)       (+ - +)          (7 9)
  11         (+ 0 +)       (0 + +)          (1 3)
  12         (+ 0 +)       (0 + 0)          (1 3 5)
  13         (+ 0 +)       (0 + -)          (3 5)
  14         (+ - +)       (- + +)          (11 12 15)
  15         (+ - +)       (- + 0)          (11 12 13 14 16)
  16         (+ - +)       (- + -)          (12 13 15)
NIL
```

```
Generating states .....
 14 states generated.
 Checking continuity .....
 Input:  (+ - -)
 ====
  State         X[T]          X-DOT          Successors
   1            (+ + -)       (+ - +)          (3 7 8)
   2            (0 + -)       (+ - +)          (1 3)
   3            (+ + -)       (+ - 0)          (1 5 7 8 9)
   4            (0 + -)       (+ - 0)          (1 3 5)
   5            (+ + -)       (+ - -)          (3 8 9)
   6            (0 + -)       (+ - -)          (3 5)
   7            (+ 0 -)       (0 - +)          (12 13)
   8            (+ 0 -)       (0 - 0)          (12 13 14)
   9            (+ 0 -)       (0 - -)          (13 14)
    10          (+ - +)       (- + -)          (11)
    11          (+ - 0)       (- 0 -)          (14)
    12          (+ - -)       (- - +)          (13)
    13          (+ - -)       (- - 0)          (12 14)
    14          (+ - -)       (- - -)          (13)
NIL
```

```
 Generating states .....
 8 states generated.
 Checking continuity .....
 Input:  (+ - +)
 ====
  State         X[T]          X-DOT          Successors
   1            (+ + -)       (+ - +)          (3)
   2            (0 + -)       (+ - +)          (1)
   3            (+ 0 -)       (0 - +)          (8)
   4            (+ - +)       (- + +)          (5)
   5            (+ - +)       (- + 0)          (4 6)
   6            (+ - +)       (- + -)          (5)
   7            (+ - 0)       (- 0 +)          (4)
   8            (+ - -)       (- - +)          (7)
 NIL
```

```
Generating states .....
 6 states generated.
 Checking continuity .....
 Input:  (+ - 0)
 ====
  State          X[T]           X-DOT        Successors
   1             (+ + -)        (+ - +)         (3)
   2             (0 + -)        (+ - +)         (1)
   3             (+ 0 -)        (0 - +)         (6)
   4             (+ - +)        (- + -)         NIL
   5             (+ - 0)        (- 0 0)         NIL
   6             (+ - -)        (- - +)         NIL
NIL
```

```
 Generating states .....
 12 states generated.
 Checking continuity .....
 Input:  (+ + -)
 ====
  State          X[T]           X-DOT        Successors
   1             (+ + +)        (+ + -)         (3)
   2             (0 + +)        (+ + -)         (1 3)
   3             (+ + 0)        (+ 0 -)         (9)
   4             (0 + 0)        (+ 0 -)         (9)
   5             (+ + -)        (+ - +)         (7)
   6             (0 + -)        (+ - +)         (5 7)
   7             (+ + -)        (+ - 0)         (5 9)
   8             (0 + -)        (+ - 0)         (5 7 9)
   9             (+ + -)        (+ - -)         (7)
   10            (0 + -)        (+ - -)         (7 9)
   11            (+ 0 +)        (0 + -)         (1)
   12            (+ - +)        (- + -)         (11)
 NIL
```

```
Generating states .....
8 states generated.
Checking continuity .....
Input:  (+ + 0)
====
State           X[T]            X-DOT           Successors
  1             (+ + +)         (+ + -)           NIL
  2             (0 + +)         (+ + -)           (1)
  3             (+ + 0)         (+ 0 0)           NIL
  4             (0 + 0)         (+ 0 0)           (3)
  5             (+ + -)         (+ - +)           NIL
  6             (0 + -)         (+ - +)           (5)
  7             (+ 0 +)         (0 + -)           (1)
  8             (+ - +)         (- + -)           (7)
NIL
```

```
Generating states .....
8 states generated.
Checking continuity .....
Input:  (+ 0 -)
====
State           X[T]            X-DOT           Successors
  1             (+ + -)         (+ - +)           NIL
  2             (0 + -)         (+ - +)           NIL
  3             (+ + -)         (+ - 0)           NIL
  4             (0 + -)         (+ - 0)           NIL
  5             (+ + -)         (+ - -)           NIL
  6             (0 + -)         (+ - -)           NIL
  7             (+ 0 0)         (0 0 -)           NIL
  8             (+ - +)         (- + -)           NIL
NIL
```

```
Generating states .....
6 states generated.
Checking continuity .....
Input:  (+ 0 +)
====
State          X[T]           X-DOT          Successors
  1            (+ + -)        (+ - +)          NIL
  2            (0 + -)        (+ - +)          NIL
  3            (+ 0 0)        (0 0 +)          NIL
  4            (+ - +)        (- + +)          NIL
  5            (+ - +)        (- + 0)          NIL
  6            (+ - +)        (- + -)          NIL
NIL
```

```
Generating states .....
4 states generated.
Checking continuity .....
Input:  (+ 0 0)
====
State          X[T]           X-DOT          Successors
  1            (+ + -)        (+ - +)          NIL
  2            (0 + -)        (+ - +)          (1)
  3            (+ 0 0)        (0 0 0)          NIL
  4            (+ - +)        (- + -)          NIL
NIL
```

```
Generating states .....
2 states generated.
Checking continuity .....
Input:  (0 + -)
====
State          X[T]           X-DOT          Successors
  1            (0 0 +)        (0 + -)          NIL
  2            (+ - +)        (- + -)          NIL
NIL
```

```
Generating states .....
6 states generated.
Checking continuity .....
Input:  (0 + +)
====
State         X[T]          X-DOT        Successors
  1           (0 0 +)       (0 + +)        NIL
  2           (0 0 +)       (0 + 0)        NIL
  3           (0 0 +)       (0 + -)        NIL
  4           (+ - +)       (- + +)        NIL
  5           (+ - +)       (- + 0)        NIL
  6           (+ - +)       (- + -)        NIL
NIL
```

```
Generating states .....
2 states generated.
Checking continuity .....
Input:  (0 + 0)
====
State         X[T]          X-DOT        Successors
  1           (0 0 +)       (0 + -)        NIL
  2           (+ - +)       (- + -)        NIL
NIL
```

```
Generating states .....
4 states generated.
Checking continuity .....
Input:  (0 0 +)
====
State         X[T]          X-DOT        Successors
  1           (0 0 0)       (0 0 +)        NIL
  2           (+ - +)       (- + +)        NIL
  3           (+ - +)       (- + 0)        NIL
  4           (+ - +)       (- + -)        NIL
NIL
```

```
Generating states .....

2 states generated.

Checking continuity .....

Input:  (0 0 0)
====
 State          X[T]          X-DOT        Successors
   1            (0 0 0)       (0 0 0)         NIL
   2            (+ - +)       (- + -)         NIL
NIL
```

```
Generating states .....

26 states generated.

Checking continuity .....

Input:  (+ 0 0)
====

State      X1[T]       X1-DOT      X2[T]       X2-DOT      Successors
  1        (+ + -)     (+ - +)     (+ + +)     (+ + -)       (3)
  2        (+ + -)     (+ - +)     (0 + +)     (+ + -)       (1 3)
  3        (+ + -)     (+ - +)     (+ + 0)     (+ 0 -)       (9)
  4        (+ + -)     (+ - +)     (0 + 0)     (+ 0 -)       (9)
  5        (+ + -)     (+ - +)     (+ + -)     (+ - +)       (7)
  6        (+ + -)     (+ - +)     (0 + -)     (+ - +)       (5 7)
  7        (+ + -)     (+ - +)     (+ + -)     (+ - 0)       (5 9)
  8        (+ + -)     (+ - +)     (0 + -)     (+ - 0)       (5 7 9)
  9        (+ + -)     (+ - +)     (+ + -)     (+ - -)       (7)
  10       (+ + -)     (+ - +)     (0 + -)     (+ - -)       (7 9)
  11       (+ + -)     (+ - +)     (+ 0 +)     (0 + -)       (1)
  12       (0 + -)     (+ - +)     (0 0 +)     (0 + -)       NIL
  13       (+ + -)     (+ - +)     (+ - +)     (- + -)       (11)
  14       (0 + -)     (+ - +)     (+ - +)     (- + -)       (11 13)
  15       (+ 0 0)     (0 0 0)     (+ + -)     (+ - +)       NIL
  16       (+ 0 0)     (0 0 0)     (0 + -)     (+ - +)       (15)
  17       (+ 0 0)     (0 0 0)     (+ 0 0)     (0 0 0)       NIL
  18       (+ 0 0)     (0 0 0)     (+ - +)     (- + -)       NIL
  19       (+ - +)     (- + -)     (+ + -)     (+ - +)       (21)
  20       (+ - +)     (- + -)     (0 + -)     (+ - +)       (19)
  21       (+ - +)     (- + -)     (+ 0 -)     (0 - +)       (26)
  22       (+ - +)     (- + -)     (+ - +)     (- + +)       (23)
  23       (+ - +)     (- + -)     (+ - +)     (- + 0)       (22 24)
  24       (+ - +)     (- + -)     (+ - +)     (- + -)       (23)
  25       (+ - +)     (- + -)     (+ - 0)     (- 0 +)       (22)
  26       (+ - +)     (- + -)     (+ - -)     (- - +)       (25)
NIL
```

APPENDIX B

This appendix contains the envisionment tables for all the input vectors to a one compartment system not shown in the main text of this thesis.

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}} + \frac{u_2}{k^3_{01}}$
20	+ - + o	$x(0) = \frac{u_0}{k_{01}} + \frac{u_2}{k^3_{01}}$
19	+ - + +	$\frac{u_0}{k_{01}} < x(0) < \frac{u_0}{k_{01}} + \frac{u_2}{k^3_{01}}$
16	+ o - +	$x(0) = \frac{u_0}{k_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}}$
34	o + - +	$x(0) = 0$

Table B.1 Envisionment for the [+ o +] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > 0$
39	o o + -	$x(0) = 0$

Table B.2 Envisionment for the [0 + -] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
20	+ - + o	$x(0) = \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
19	+ - + +	$\frac{u_0}{k_{01}} < x(0) < \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
10	+ o + +	$x(0) = \frac{u_0}{k_{01}},\ x'(0) = 0,\ x''(0) > \frac{2u_2}{k_{01}}$
11	+ o + o	$x(0) = \frac{u_0}{k_{01}},\ x'(0) = 0,\ x''(0) = \frac{2u_2}{k_{01}}$
12	+ o + -	$x(0) = \frac{u_0}{k_{01}},\ x'(0) = 0,\ x''(0) < \frac{2u_2}{k_{01}}$
3	+ + + -	$\frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}} < x(0) < \frac{u_0}{k_{01}}$
2	+ + + o	$x(0) = \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
1	+ + + +	$\frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} < x(0) < \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
4	+ + o +	$x(0) = \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}} - \frac{u_1}{k^2_{01}}$
34	o + - +	$x(0) = 0,\ x'(0) > \frac{u_1}{k_{01}}$
31	o + o +	$x(0) = 0,\ x'(0) = \frac{u_1}{k_{01}}$
28	o + + +	$x(0) = 0,\ \frac{u_1}{k_{01}} - \frac{u_2}{k^2_{01}} < x'(0) < \frac{u_1}{k_{01}}$
29	o + + o	$x(0) = 0,\ x'(0) = \frac{u_1}{k_{01}} - \frac{u_2}{k^2_{01}}$
30	o + + -	$x(0) = 0,\ x'(0) < \frac{u_1}{k_{01}} - \frac{u_2}{k^2_{01}}$

Table B.3 Envisionment for the [+ + +] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}}$
15	+ o + -	$x(0) = \frac{u_0}{k_{01}}$
9	+ + - -	$\frac{u_0}{k_{01}} - \frac{u_2}{k^3_{01}} < x(0) < \frac{u_0}{k_{01}}$
8	+ + - o	$x(0) = \frac{u_0}{k_{01}} - \frac{u_2}{k^3_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}} - \frac{u_2}{k^3_{01}}$
34	o + - +	$x(0) = 0, \ x'(0) > \frac{u_2}{k^2_{01}}$
32	o + o o	$x(0) = 0, \ x'(0) = \frac{u_2}{k^2_{01}}$
30	o + + -	$x(0) = 0, \ x'(0) < \frac{u_2}{k^2_{01}}$

Table B.4 Envisionment for the [+ o -] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > 0$
39	o o + -	$x(0) = 0$

Table B.5 Envisionment for the [o + o] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
24	+ - o -	$x(0) = \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
27	+ - - -	$\frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}} < x(0) < \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
26	+ - - o	$x(0) = \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
25	+ - - +	$\frac{u_0}{k_{01}} < x(0) < \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
16	+ o - +	$x(0) = \frac{u_0}{k_{01}},\ x'(0) = 0,\ x''(0) < -\frac{2u_2}{k_{01}}$
17	+ o - o	$x(0) = \frac{u_0}{k_{01}},\ x'(0) = 0,\ x''(0) = -\frac{2u_2}{k_{01}}$
18	+ o - -	$x(0) = \frac{u_0}{k_{01}},\ x'(0) = 0,\ x''(0) > -\frac{2u_2}{k_{01}}$
9	+ + - -	$\frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}} < x(0) < \frac{u_0}{k_{01}}$
8	+ + - o	$x(0) = \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}} + \frac{u_2}{k^3_{01}}$
34	o + - +	$x(0) = 0,\ x'(0) > -\frac{u_1}{k_{01}} + \frac{u_2}{k^2_{01}}$
35	o + - o	$x(0) = 0,\ x'(0) = -\frac{u_1}{k_{01}} + \frac{u_2}{k^2_{01}}$
36	o + - -	$x(0) = 0,\ x'(0) < -\frac{u_1}{k_{01}} + \frac{u_2}{k^2_{01}}$

Table B.6 Envisionment for the [+ - -] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
23	+ - + o	$x(0) = \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
25	+ - + +	$\frac{u_0}{k_{01}} < x(0) < \frac{u_0}{k_{01}} + \frac{u_1}{k^2_{01}}$
16	+ o - +	$x(0) = \frac{u_0}{k_{01}}$
7	+ + - +	$0 < x(0) < \frac{u_0}{k_{01}}.$
34	o + - +	$x(0) = 0$

Table B.7 Envisionment for the [+ - o] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > \frac{u_2}{k^2_{01}}$
20	+ - + o	$x(0) = \frac{u_2}{k^2_{01}}$
19	+ - + +	$0 < x(0) < \frac{u_2}{k^2_{01}}$
40	o o o +	$x(0) = 0$

Table B.8 Envisionment for the [o o +] input vector.

state	vector	interpretation
21	+ - + -	$x(0) > -\frac{u_1}{k_{01}^2} + \frac{2u_2}{k_{01}^3}$
20	+ - + o	$x(0) = -\frac{u_1}{k_{01}^2} + \frac{2u_2}{k_{01}^3}$
19	+ - + +	$0 < x(0) < -\frac{u_1}{k_{01}^2} + \frac{2u_2}{k_{01}^3}$
37	o o + +	$x(0) = 0,\ x'(0) = 0,\ x'' > \frac{2u_2}{k_{01}}$
38	o o + o	$x(0) = 0,\ x'(0) = 0,\ x'' = \frac{2u_2}{k_{01}}$
39	o o + -	$x(0) = 0,\ x'(0) = 0,\ x'' < \frac{2u_2}{k_{01}}$

Table B.9 Envisionment for the [o + +] input vector.