

University of Warwick institutional repository: <http://go.warwick.ac.uk/wrap>

**A Thesis Submitted for the Degree of PhD at the University of Warwick**

<http://go.warwick.ac.uk/wrap/4233>

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it. Our policy information is available from the repository home page.

—

# Cellular Automata and Dynamical Systems

*Leslie John Clewlow (B.Sc.)*

Department of Computer Science

University of Warwick

Coventry

November 30, 1989

## Table of Contents

2	Table of Contents
3	List of Figures and Tables
6	Acknowledgements
6	Declaration
6	Abbreviations
7	Summary
7a	Forward
8	Chapter 1: Introduction
16	Chapter 2: The General Model
28	Chapter 3: A Statistical Mechanics Approach
54	Chapter 4: Experimental Results
76	Chapter 5: Simulating Dynamical Systems with Cellular Automata
90	Chapter 6: Conclusions
98	References

## List of Figures

- 21 Figure 2.1 : Summary of **Filler**.
- 22 Figure 2.2 : Summary of **Convexer**.
- 23 Figure 2.3 : Summary of **Bouncer**.
- 23 Figure 2.4 : **Bouncer** particle and neighbourhood.
- 23 Figure 2.5 : **Bouncer** mirror.
- 24 Figure 2.6 : **Bouncer** particle reflecting off mirror.
- 25 Figure 2.7 : Summary of **Diffuser**.
- 25 Figure 2.8 : Summary of **Reverse**.
- 26 Figure 2.9 : Summary of **Twolife**.
- 30 Figure 3.1 : Statistical density function.
- 44 Figure 3.2 : Number of configurations as a function of the number of randomised cells.
- 49 Figure 3.3 : Distribution of  $\zeta$ -evolved cells.
- 54 Figure 4.1 : Example of directed graph.
- 55 Figure 4.2 : Two variable systems.
- 55 Figure 4.3 : Example behaviour of  $\dot{n}_0$ .
- 57 Figure 4.4 : Valid region of phase plane for three variable CA.
- 58 Figure 4.5 : Hypergeometric, Binomial and Normal functions.
- 58 Figure 4.6 : Hypergeometric function and Binomial and Normal errors.
- 59 Figure 4.7 : Variation of transition probability with dependent population.
- 59 Figure 4.8 : Variation of threshold probability with dependent population.



- 59 Figure 4.9 : Threshold systems summary.
- 59 Figure 4.10: Threshold systems state transition diagrams.
- 59 Figure 4.11: Threshold systems vector field diagrams.
- 59 Figure 4.12: Threshold systems phase diagrams.
- 67 Figure 4.13: Class 4 systems summary.
- 67 Figure 4.14: Class 4 systems state transition diagrams.
- 67 Figure 4.15: Class 4 vector field diagrams.
- 67 Figure 4.16: Class 4 phase diagrams.
- 71 Figure 4.17: Class 3 systems summary.
- 71 Figure 4.18: Class 3 systems state transition diagrams.
- 71 Figure 4.19: Class 3 vector field diagrams.
- 71 Figure 4.20: Class 3 phase diagrams.
- 78 Figure 5.1 : Variation of CA population genetic function.
- 78 Figure 5.2 : Error in CA population genetic function.
- 80 Figure 5.3 : Lattice evolution for simplified Lotka-Volterra CA.
- 80 Figure 5.4 : Lattice evolution for improved Lotka-Volterra CA.
- 80 Figure 5.5 : Phase diagram for simplified Lotka-Volterra model.
- 80 Figure 5.6 : Phase diagram for improved Lotka-Volterra model.
- 83 Figure 5.7 : Lattice evolution for general epidemic CA.
- 83 Figure 5.8 : Phase diagram for general epidemic model.
- 84 Figure 5.9 : Lattice evolution for recurrent epidemic CA.
- 84 Figure 5.10: Phase diagram for recurrent epidemic model.

- 86 Figure 5.11: (a) Multi-count transition probability.
- 86 Figure 5.11: (b) Product of two variable populations.
- 86 Figure 5.12: Complex population genetic CA and DS.

## Acknowledgements

The author gratefully acknowledges the support of the Science and Engineering Research Council as well as useful discussions with Mikki Larcombe, Barry Steer, Keith Langley, Tim Atherton, Roger Packwood, Neil Cresswell, Liz Wellington and many others and useful communications with Stephen Wolfram, Chris Langton and Wentian Li.

## Declaration

No part of the work in this thesis has been used before. No part of the work in this thesis is based on joint research.

## Abbreviations

ACA	Asynchronous Cellular Automata
CA	Cellular Automata
DS	Dynamical System
SCA	Synchronous Cellular Automata

## Summary

In this thesis we investigate the theoretical nature of the mathematical structures termed cellular automata.

**Chapter 1:** Reviews the origin and history of cellular automata in order to place the current work into context.

**Chapter 2:** Develops a cellular automata framework which contains the main aspects of cellular automata structure which have appeared in the literature. We present a scheme for specifying the cellular automata rules for this general model and present six examples of cellular automata within the model.

**Chapter 3:** Here we develop a statistical mechanical model of cellular automata behaviour. We consider the relationship between variations within the model and their relationship to dynamical systems. We obtain results on the variance of the state changes, scaling of the cellular automata lattice, the equivalence of noise, spatial mixing of the lattice states and entropy, synchronous and asynchronous cellular automata and the equivalence of the rule probability and the time step of a discrete approximation to a dynamical system.

**Chapter 4:** This contains an empirical comparison of cellular automata within our general framework and the statistical mechanical model. We obtain results on the transition from limit cycle to limit point behaviour as the rule probabilities are decreased. We also discuss failures of the statistical mechanical model due to failure of the assumptions behind it.

**Chapter 5:** Here a practical application of the preceding work to population genetics is presented. We study this in the context of some established population models and show it may be most useful in the field of epidemiology. Further generalisations of the statistical mechanical and cellular automata models allow the modelling of more complex population models and mobile populations of organisms.

**Chapter 6:** Reviews the results obtained in the context of the open questions introduced in Chapter 1. We also consider further questions this work raises and make some general comments on how these may apply to related fields.



## Forward

The purpose of this work was to investigate the theoretical nature of the mathematical structures termed cellular automata. These were first introduced by John von Neumann and Stanislaw Ulam in the early 1950's as models of biological systems in order to investigate the logical basis of self-reproduction. They have since been found to be applicable in a wide variety of fields from galaxy formation to crystal growth. Although much empirical work has been done on the behaviour of the various types of cellular automata and their inter-relationships, very little is known about the theoretical relationships. Some results for simple cellular automata do exist, in particular additive cellular have been extensively studied. Some workers have noted the similarity between cellular automata and statistical mechanical models and some work has been done on their relationship to Ising models. Other workers have studied the statistical behaviour of specific cellular automata models but no attempt has been made to develop a general framework.

Some of the theoretical issues which remain unresolved have recently been reviewed by Wolfram, these include,

1. What overall classification of CA can be given ?
2. What statistical quantities characterise CA behaviour ?
3. What invariants are there in CA evolution ?
4. How does thermodynamics apply to CA ?
5. How are different behaviours distributed in the space of CA rules ?

6. What is the correspondence between CA and continuous systems ?
7. What is the correspondence between CA and stochastic systems ?
8. How are CA affected by noise and other imperfections ?
9. What limit sets can CA produce ?
10. What are the connections between the computational and statistical characteristics of CA ?
11. How often does computational universality and undecidability occur ?
12. What is the nature of the infinite size or thermodynamic limit for CA ?
13. How often does computational irreducibility occur in CA ?
14. How often do computationally intractable problems occur in CA ?
15. What higher level descriptions of information processing in CA can be given ?

The purpose of this work was to begin to develop a statistical mechanical approach to the prediction of the behaviour of cellular automata and to establish statistical relationships between the various types of cellular automata in order to begin to answer some of these questions.

Firstly in Chapter 1 we review the cellular automata literature in the form of an historical overview.

In Chapter 2 we synthesise the wide variety of cellular automata structures

described in Chapter 1 into a general framework. A scheme for specifying the rules of cellular automata within this general model is introduced. Finally we present six simple examples which exhibit the four major classes of behaviour.

We use this general model in Chapter 3 to construct a statistical mechanical model of cellular automata behaviour. The statistical mechanical model leads to a dynamical systems model. We then consider variations in the cellular automata structure and the dynamical system their inter-relationships. We obtain results on the variance of the state changes, scaling of the cellular automata lattice, the equivalence of noise, spatial mixing of the lattice states and entropy, synchronous and asynchronous cellular automata and the equivalence of the rule probability and the time step of the dynamical system.

Chapter 4 is an extensive empirical study of cellular automata in the four major classes defined within our general framework. We obtain results on the transition from limit cycle to limit point behaviour as the rule probabilities are decreased. We also study differences between the behaviour of certain cellular automata and their equivalent dynamical system due to failure of the assumptions behind the statistical mechanical model.

In Chapter 5 we develop an application of the statistical mechanical model to population genetics. We study some established population models and show it may be most useful in the field of epidemiology. This approach leads us to generalise the cellular automata model further to allow modelling of populations of organisms which are spatially mobile.

The concluding Chapter 6 contains a summary of the results obtained and how they relate to the open questions identified by Wolfram. We discuss new questions and prob-

lems that this work raises, certain conclusions are drawn and future directions for this work are proposed.



## 1. Introduction

Towards the end of the 1940's and into the early 1950's John von Neumann began to consider the problem of the logical basis of self-reproduction in biological systems. He identified five distinct types of system in which self-reproduction processes may be constructed. These were kinematic, cellular, oscillation-threshold-fatigue, continuous and probabilistic[1]. However he only considered the first two in any detail. The classification was based on the form of fundamental processes which the environment could support. These fundamental processes would then be used to synthesise a more complex process whose action would be to construct a copy of itself using copies of the fundamental processes "floating" freely in the environment. For example the kinematic environment was considered to contain switches, delays, kinematic elements, cutting elements, fusing elements, rigid elements and sensing elements. The switches and delays allowed the construction of computational elements (it can be shown that Turing machines can be constructed from just these elements). The kinematic element could move other elements around under control of a computational element. The cutting element could separate elements while the fusing element could join them together. The rigid elements would be used to form supporting structures for the other elements and the sensing element could identify the other elements and signal this information to the computational element.

Von Neumann quickly realised that the kinematic environment left the method of interaction of the processes too vaguely defined to allow a useful analysis. Furthermore any attempt to define the interactions rigorously would lead to an unmanageably complex set of processes. He therefore began to work with a cellular environment in which space consisted of discrete cells. Each cell could be in one of a small set of states and the whole

lattice of cells changed state synchronously in discrete time steps. Every cell had the same set of simple rules for state transitions which were conditional on the states of the cells in a small region around that cell. The motivation for this system was partly von Neumann's ideas for the design and construction of a general purpose electronic computing instrument and partly from suggestions by Stanislaw Ulam. During the next decade many workers began to consider discrete systems similar to this as the basis for models of various biological processes. These systems were given various names such as cellular automata or cellular spaces, tessellation automata or tessellation spaces, iterative arrays and homogeneous structures. The different nomenclature partially reflected variations on the original structure which von Neumann introduced. For example tessellation structures tended to have more general and computationally complex automata at each lattice point. Iterative arrays were allowed to have specialised processing elements at each point and homogeneous structures was a term coined by Victor Aladyev[2] to describe the whole group of systems of this type. From now on we will refer to them as cellular automata (CA) since we will be mainly concerned with systems similar to those of von Neumann.

At the time of his death in 1957 von Neumann had essentially completed the design of a cellular automaton which was computation and construction universal[1]. That is given the necessary instructions in the form of an area of the lattice in a specific set of states, it could perform any computation or alternatively simulate any Turing Machine. It could also construct, given the instructions as above, in a quiescent region of the lattice any other cellular automaton including itself.

Throughout the 1960's two main strands of research occurred. Firstly, the application of CA to the modelling of biological processes. This work was mainly concerned with the growth and development of living organisms as well as continuing the work which von



Neumann began on reproduction[3-9]. The second main area of work was concerned with the computational properties of iterative arrays. This included for example the generation of prime numbers and image processing[10-13].

Then in 1970 John Conway discovered the CA known as LIFE[14-16]. It soon became apparent that this CA was very special and in 1974 it was shown that LIFE was capable of universal computation[17]. This was done by showing that structures existed which could represent signals, clocks and the logical operations of AND, OR and NOT and that these could be made to interact in a way necessary to embed a computing structure on the lattice.

During the 1970's further applications of CA appeared, in particular in the modelling of crystal growth and also in galaxy formation[18-25]. This reflected the emergence of a general interest in self-organisation in complex systems by Haken and others[26-32]. At about this time the first attempt to model the behaviour of a CA using a stochastic dynamical system was published by Schulman and Seiden[33]. Wolfram also suggested a similar approach in an extensive review of work on CA in 1983[34].

In 1983, between March 7th and 11th an Interdisciplinary Workshop on Cellular Automata was held at Los Alamos, New Mexico. In this Tommaso Toffoli and Gerard Vichniac both presented papers suggesting that CA could provide an alternative and complementary approach to the modelling of physical processes[35,36]. This approach was subsequently used to provide CA models of hydrodynamic systems[37-40]. However the analogy was not seen as providing a means of predicting the evolution of CA in general. This is the approach which we take and we hope to show that it provides a very powerful tool in the problem of understanding the behaviour of CA. Furthermore, it demonstrates a new approach to the modelling of physical phenomena with CA.

We define a CA as a regular  $d$ -dimensional lattice of  $N$  cells  $c_i$ <sup>1</sup>. Each cell may take one of  $k$  possible values. All cells on the lattice are updated simultaneously in discrete time steps according to a local rule  $F$  which depends on the values of a small set of  $r$  cells  $R_i$ , these are called the neighbourhood cells as they are normally the cells having the shortest path distance along the lattice connections. Each cell can therefore be considered to be a simple automaton or finite state machine. With this definition of the cell we have a lattice of Moore Machines. A Moore Machine is defined as an ordered quintuple  $MO = (I, O, S, f, \phi)$ , where  $I, O$  are finite alphabets called the *input* and *output* alphabets,  $S$  is a finite alphabet called the *set of states*,  $f$  is a function mapping  $S \times I$  into  $S$  and  $\phi$  is a function mapping  $S$  into  $O$  [41]. Furthermore  $I, O$  and  $S$  are normally considered identical. However, we could consider each cell to be a more complex automaton. For example a Mealy machine, which is defined as an ordered quintuple  $ME = (I, O, S, f, \phi)$ , where  $I, O, S$  and  $f$  are as before, but  $\phi$  is a function mapping  $S \times I$  into  $O$  [41]. Here we see that the internal state of the cell is not the same as the state which its neighbours see. As noted above, the level of complexity of the cells determined what generic name it was given.

The value of a cell  $c_i$  at position  $i$  in the lattice at time  $t+1$  is given by,

$$c_i^{(t+1)} = F [\{c_j^{(t)} : j \in R_i\}] \quad 1.1$$

The lattice is normally considered to be square, that is four-connected for a planar lattice, although other lattice arrangements have been studied. In particular hexagonal lattices have been used extensively to construct CA models of hydrodynamic systems [38, 40].

The rule is normally specified by listing the state to be taken by a given cell for each possible configuration of its neighbourhood  $R_i$ .

---

<sup>1</sup> We work with a finite region of  $N$  cells embedded in an infinite lattice which is in the zero state.

As already noted, although they are very simple systems, CA are capable of apparently complex behaviour including generation of self-similar or fractal patterns, chaotic behaviour and universal computation. Direct mathematical analysis is in general not possible as the behaviour of the CA cannot in general be determined from its specification. Although for certain classes of CA, simplifying assumptions can be made which allow an analytical solution to be derived. For example for additive CA in which the future state is simply the sum of the value of the states in its neighbourhood (modulo the number of states per cell  $k$ ) an algebraic solution based on the products of polynomials has been found[42].

There are still many unresolved questions concerning CA. Some of these are as follows,

1. What overall classification of CA can be given ?
2. What statistical quantities characterise CA behaviour ?
3. What invariants are there in CA evolution ?
4. How does thermodynamics apply to CA ?
5. How are different behaviours distributed in the space of CA rules ?
6. What is the correspondence between CA and continuous systems ?
7. What is the correspondence between CA and stochastic systems ?
8. How are CA affected by noise and other imperfections ?



9. What limit sets can CA produce ?
10. What are the connections between the computational and statistical characteristics of CA ?
11. How often does computational universality and undecidability occur ?
12. What is the nature of the infinite size or thermodynamic limit for CA ?
13. How often does computational irreducibility occur in CA ?
14. How often do computationally intractable problems occur in CA ?
15. What higher level descriptions of information processing in CA can be given ?

These questions and others are discussed by Wolfram in[43]. We will be addressing the question of CA classification, the statistical properties of CA, the distribution of CA behaviour in the rule space, the correspondence between CA and continuous and stochastic systems and the effect of noise.

Empirical studies by Wolfram[44] have identified four qualitative classes with properties analogous to those of continuous dynamical systems. From a typical initial state Class-1 evolve to a homogeneous final state, analogous to a fixed point. Class-2 evolve to a set of periodic states, analogous to a limit cycle. Class-3 exhibit chaotic, aperiodic sequences of states, analogous to chaotic or “strange” attractors. Finally, Class-4 exhibit complicated localised and propagating structures and it is thought that this class may be generically capable of universal computation. However as Langton[45] has pointed out the more natural ordering for these classes is 1, 2, 4, 3. This follows the progression from

the least complex limit point behaviour to the most complex chaotic behaviour. Class 4 then emerges as a transitory and consequently rare stage between class 2 and class 3. Wolfram has also introduced the concept of legal rules. These are based on two restrictions, firstly a state consisting entirely of zeroes which Wolfram terms the quiescent state, must remain in the quiescent state. This is based on the idea that the zero state represents empty space and thus this quiescence condition is required to prevent instantaneous propagation of non-zero site values. The second restriction is that the rules should be completely symmetric, that is a symmetric configuration will remain symmetric and this guarantees isotropy and homogeneity. In one dimension class 1 and 2 become progressively less common as  $k$  and  $r$  increase, class 3 become more common and class 4 slowly less common[44]. In two dimensions class 3 are overwhelmingly the most common and class 4 are very rare[43]. Some class 4 CA which have appeared in the literature are listed below,

$k=3$	$r=3$	1D	S.Wolfram, Physica D 10, 1, 1984
$k=18$	$r=3$	1D	A.R.Smith, J.ACM 18, 339, 1971
$k=2$	$r=5$	1D	S.Wolfram, Physica D 10, 1, 1984
$k=2$	$r=5$	2D	E.R.Banks, MIT Report TR81, 1971
$k=2$	$r=9$	2D	John Conways "Game of Life"
$k=2$	$r=4$	2D	N.Margolus, Physica D 10, 81, 1984

In Chapter 2 we develop a general framework encompassing the most important variations in cellular automata structure which have appeared in the literature and which exhibits all of the classes of behaviour described above. We introduce a rule specification scheme for this general model and present six simple examples.

In Chapter 3 we use this framework to develop a statistical mechanical model of cellular automata behaviour. We then study the relationship between variations in cellular automata structure and the relationship to continuous dynamical systems. We obtain results on the variance of the state changes, scaling of the cellular automata lattice, the equivalence of noise, spatial mixing of the lattice states and entropy, synchronous and asynchronous cellular automata and the equivalence of the rule probability and the time step of a dynamical system.

In Chapter 4 we perform an extensive empirical study of cellular automata within our general framework. We obtain results on the transition from limit cycle to limit point behaviour as the rule probabilities are decreased and note discrepancies between the behaviour of certain cellular automata and their equivalent dynamical system.

In Chapter 5 we develop the relationship between cellular automata and continuous dynamical systems further. We study this relationship in the context of established population models and show it may be useful in the field of epidemiology. We also generalise the cellular automata model further to allow modelling of mobile populations of organisms.

Finally in Chapter 6 we summarise the results obtained in the context of the open questions identified by Wolfram and introduced in this Chapter. Certain conclusions are drawn from this and further work to be done is suggested.



## 2. The General Model

The standard method of specifying the evolutionary rules for a CA is to list the state which a cell should take for each possible configuration of its neighbourhood. However with this type of rule specification the size of the rule space rapidly becomes unmanageably large as the number of states per cell and especially the size of the neighbourhood increases.

The number of possible rules is given by,

$$k^{(k^r)} \quad 2.1$$

Since the number of configurations of a neighbourhood with  $k$  possible states per cell is  $k^r$  and for each of these configurations there is a choice of  $k$  states for the centre cell to take.

It can be seen from this that the number of possible rules becomes large very quickly because the number of possible configurations of the neighbourhood increases exponentially with the size of the neighbourhood. It is therefore impractical to study the complete rule space of any but the simplest CA. One must therefore resort to selecting rules at random in the expectation that these will be typical[46,47] or develop a rule specification scheme which reduces the rule space to a more manageable size. The latter approach is taken here.

A convenient way to specify the rule for a given CA, first introduced by Wolfram[34], is to specify the state the centre cell should take in the next time step, which we will call the *output state*, for every possible configuration of the neighbourhood. If we assign each configuration of the neighbourhood a unique number  $v$  and let the *output state* be  $f[v]$

then the rule becomes a vector of output states  $f$ . The unique value  $v$  can be calculated by assigning each cell in the neighbourhood  $R_i$  a unique position in an  $r$ -digit  $k$ -ary number, each configuration will then correspond to a unique  $k$ -ary number. In the same way the vector of output states  $f[v]$  can be represented by a unique number or code  $F_c$ ,

$$F_c = \sum_v f[v]k^v \quad 2.2$$

Note however that this is only unique for a given  $k$ ,  $r$  and dimension  $d$ .

In order to reduce the size of the rule space we need to reduce the number of configurations which a neighbourhood with a given  $k$  and  $r$  will have. This can be achieved by labeling configurations by the number of cells in each state, this type of rule is termed *totalistic* by Wolfram. The number of configurations will now be the number of ways to distribute the  $r$  neighbourhood cells amongst the  $k$  states. This is equivalent to the number of distinguishable arrangements of  $r$  circles and  $k-1$  lines[48]. We therefore have the number of configurations as the binomial coefficients,

$$\binom{k+r-1}{r} \quad 2.3$$

A more general form of this rule is to specify the neighbourhood by the current state of the cell and the number of cells in each state in the neighbourhood not including the cell itself. This type is termed *outer totalistic* by Wolfram and a classic example of a CA with this type of rule is John Conways "Game of Life"[49]. The number of configurations for this type of rule will be,

$$k \binom{k+r-1}{r} \quad 2.4$$

Note that here the value of  $r$  is one less than for equation 2.3 since the centre cell is no

longer included in the neighbourhood.

Packard and Wolfram[46] have determined that *outer totalistic* CA exhibit all four of the qualitative classes described in section 1 and thus we will lose no generality if we use rules specified in this way.

Outer totalistic rules are closely related to threshold functions which are found in neural networks and many other types of control system. Additionally, the basic logical functions in computers of AND, OR and NOT are fundamentally threshold functions. Consider an multiple-input logic element with inputs  $I_i : i = [1..N]$  and output  $O$  then AND can be computed as  $O = (\sum_i I_i > N-1)$  and OR can be computed as  $O = (\sum_i I_i > 0)$  (NAND and NOR simply require reversing the logical comparisons). So we have identified a very general rule specification scheme.

Some simple results exist for *totalistic*, *outer totalistic* and *threshold* CA rules. Vichniac[36] has investigated 2D threshold CA with  $k=2$ , he points out that these rules always lead to growth of clusters of either the zeroes or ones. The limiting behaviour depends critically on the initial concentration  $\rho$  of ones. There are two distinct types of limiting behaviour, for the first type the growth of the clusters is limited for all  $\rho$ . Above a critical value of  $\rho_c$  the ones form a connected path or percolation structure with islands of zeroes. Below this value the zeroes form the connected path. This behaviour only occurs for rules with odd  $r$  such that an exact symmetry exists between the neighbourhood configurations leading to a zero and one. With the second type of behaviour the clusters are convex confined. That is growth of the clusters occurs by filling of convex regions of the cluster. Above a critical density  $\rho_c$  the clusters always join before becoming completely convex thus leading to eventual nucleation of the lattice to zeroes or ones.



Below this critical density the clusters stop growing before they merge, this is however a meta-stable state and on infinitely large lattice there will be a finite probability of a cluster large enough to continue growing indefinitely thus leading to eventual nucleation of the lattice. Wolfram[43] has noted that *totalistic* rules are discrete approximations of functions of a continuous parameter and as the parameter is increased regular periodic (Class 1/2) exhibit period doubling until eventually chaotic behaviour (Class 3) occurs (Class 4 appears as an intermediate phase between Class 1/2 and Class 3).

Another facet of CA which may be chosen freely is the geometry of the lattice. However, although different lattices may impose certain preferred directions for patterns generated by the CA[46], this will simply be due to the restrictions on the possible shapes of the neighbourhoods caused by the discrete nature of the lattice. In many cases suitable choice of the neighbourhood will make two CA with different lattices equivalent. As Packard and Wolfram[46] point out square, triangular and hexagonal lattices with nearest cell neighbourhoods are special cases of the  $r=9$  symmetric neighbourhood on a square lattice. We will therefore work mainly with this von Neumann neighbourhood.

Further to allow the CA to be completely general we must allow the rules to be probabilistic so that for a given neighbourhood configuration a cell has a certain probability of changing to the output state. This is important for two reasons, firstly it allows simulation of noise on the lattice and secondly by reducing the probability of each *output state* equally the rate of evolution of the CA can be effectively slowed. Both these points will become important later.

Finally, the configuration should also include the past state of the cell so that we can construct rules which are reversible. Note that the inclusion of the past state of a cell in the rule scheme is a necessary but not sufficient condition for reversibility of a CA rule set.

We must also require that each possible pair of neighbourhood configuration and past state has a unique future state. A simple type of rule which guarantees reversibility was suggested by Fredkin and Toffoli[50] and can be written as follows,

$$c_i^{(t+1)} = F [\{c_j^{(t)}:j \in R_i\}] - c_i^{(t-1)} \quad 2.5$$

The subtraction is performed modulo the number of states  $k$ . This type of rule will always be reversible even if the function  $F []$  is not. Also note that rules constructed in this way are also time reversal invariant, that is by exchanging two consecutive global configurations we can obtain the reverse evolution by application of the same rule. This is not the case for all reversible rules, for instance the left-shift rule  $x_i^{t+1} = x_{i+1}^t$ , is easily reversible. That is it is easy to obtain the reverse evolution, but it is not obtained by reversing two consecutive global states but by applying the right shift rule  $x_i^{t+1} = x_{i-1}^t$ .

The rules for the CA described above can be specified as follows,

$$s_i \& \{C_k(\{s_k:k \in R_i\})\} \rightarrow s_o \mid \text{Pr} = p \quad 2.6$$

which means that a cell in state  $s_i$  with the set of conditions  $\{C_k(s_k)\}$  on the set of states  $\{s_k\}$  in its neighbourhood  $R_i$  satisfied changes to state  $s_o$  in the next time step with probability  $p$ . The conditions  $C_k$  will be written as either  $\alpha_k s_k$  meaning  $\alpha_k$  cells in state  $s_k$  or  $>\alpha_k s_k$  meaning greater than  $\alpha_k$  cells in state  $s_k$ . We will call the complete list of rules the *rule set*.

As already noted a CA with the structure outlined above is capable of exhibiting all four of the qualitative classes of behaviour which have been identified by Wolfram. In order to demonstrate the typical behaviour of CA under this general scheme a number of CA systems were constructed which made use of the properties introduced above. These

were as follows,

<b>Filler</b>	Fills a convoluted outline with a given state.
<b>Convexer</b>	Generates minimal convex bounding region for a shape.
<b>Bouncer</b>	Simulates particles bouncing off mirrors.
<b>Diffuser</b>	Diffuses shapes into a uniform distribution.
<b>Reverse</b>	A reversible Cellular Automaton.
<b>Twolife</b>	Two competing “Game of Life” systems.

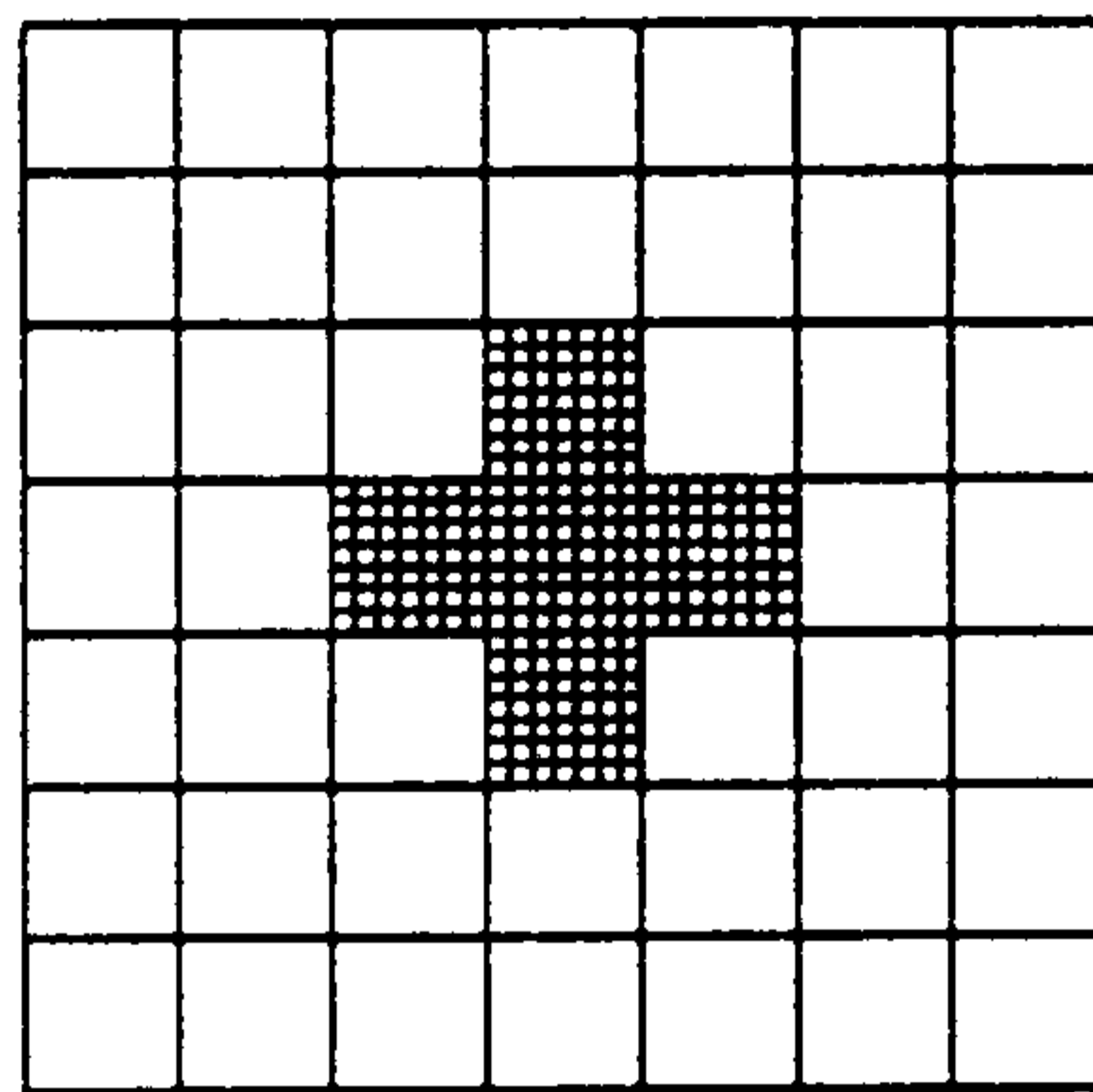
We will now study these systems in more detail.

### **Filler**

This CA is summarised in Figure 2.1. It is an implementation of the simplest filling algorithm the “flood fill”, the rules simply specify that any cell in state  $s_0$  which has a cell in state  $s_1$  in its neighbourhood should change to state  $s_1$ . In this way any cell in state  $s_1$  will cause all cells which are connected to it in the four orthogonal directions, that is by movements lying within the neighbourhood, by cells in state  $s_0$  to be changed to state  $s_1$ . We may call this connection condition *neighbourhood connected*. This CA is in fact a threshold CA and the rule set could be simplified to the following,

Figure 2.1: Summary of Filler.

Three states ( $s_0, s_1, s_2$ ).



neighbourhood

$$s_0 \& \{(1s_1)\} \rightarrow s_1$$

$$s_0 \& \{(2s_1)\} \rightarrow s_1$$

$$s_0 \& \{(3s_1)\} \rightarrow s_1$$

$$s_0 \& \{(4s_1)\} \rightarrow s_1$$



in fact a threshold CA and the rule set could be simplified to the following,

$$s_0 \& \{(\geq 1 s_1)\} \rightarrow s_1$$

This CA can be placed in class 2 since it will always reach the limiting state when all neighbourhood connected cells in state  $s_0$  have been converted to  $s_1$ .

### **Convexer**

This CA is summarised in Figure 2.2. The rule set for this CA simply marks cells in state  $s_0$  which have 4, 5, 6, 7 or 8 cells in state  $s_1$  in their neighbourhood with states  $s_2$  to  $s_6$  respectively. This highlights the denser regions of the lattice in terms of  $s_1$ , which will in general be correlated with the local concavity of regions of state  $s_1$ . The states  $s_2$  to  $s_6$  are simply converted to state  $s_1$ . We therefore obtain a repetitive two-stage process which will fill out all concave regions of state  $s_1$ . The first stage of the process is of course superfluous to the operation of the algorithm and the rule set could be simplified to the following threshold CA,

$$s_0 \& \{(\geq 4 s_1)\} \rightarrow s_1$$

As for **Filler** this CA will eventually reach a stable state when all concave regions have been filled and so this CA is class 2. This CA is related to those studied by Vichniac mentioned earlier and the comments on their limiting behaviour apply to this CA.

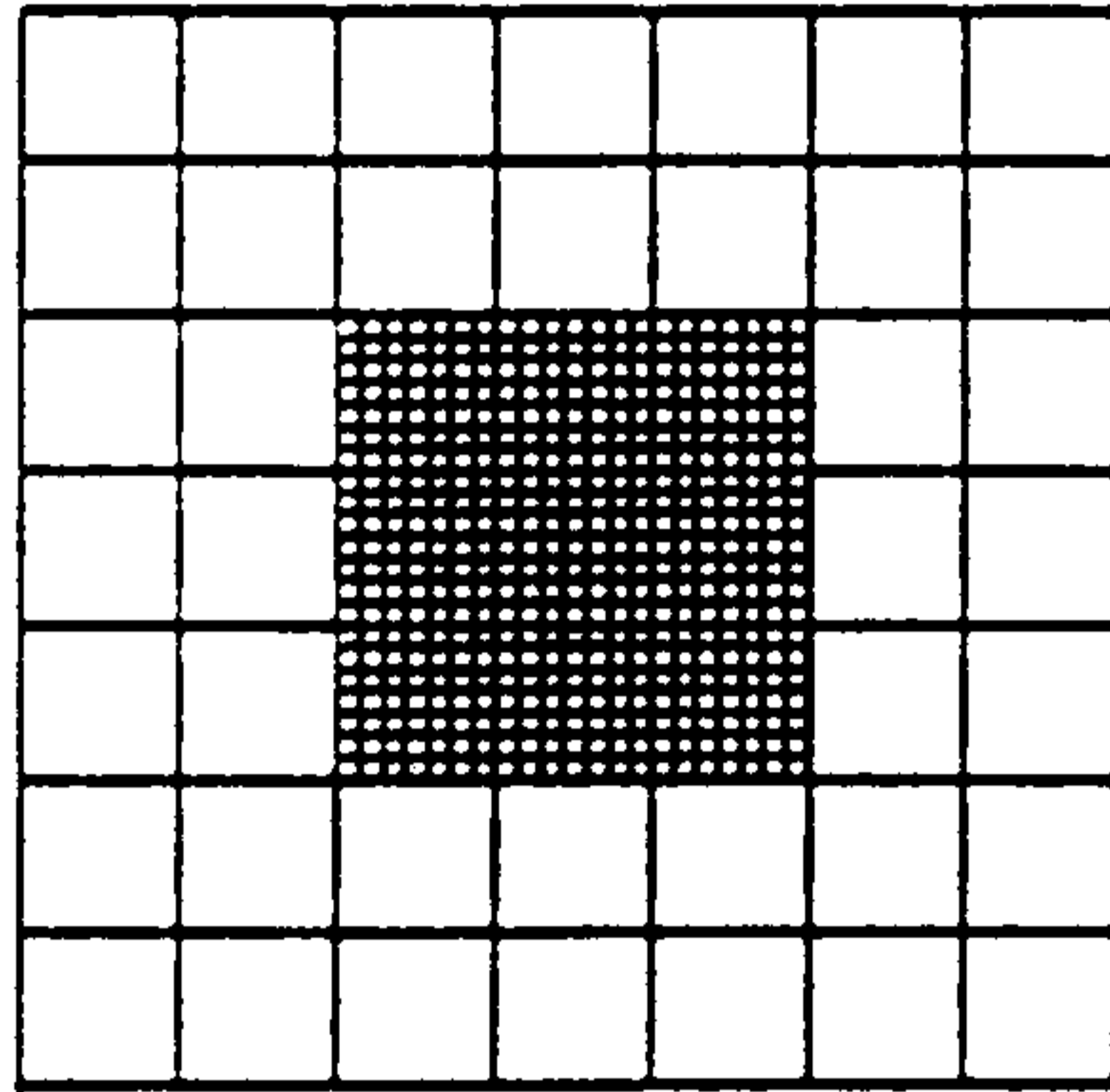
### **Bouncer**

This CA is summarised in Figure 2.3. This CA was developed to demonstrate how a set of rules within our general framework can be developed to simulate a physical process. In this example the process is the reflection of a point particle off an infinitely hard



Figure 2.2: Summary of Convexer.

Eight states ( $s_0, \dots, s_7$ ).



neighbourhood

$$s_0 \& \{(4s_1)\} \rightarrow s_2$$

$$s_0 \& \{(5s_1)\} \rightarrow s_3$$

$$s_0 \& \{(6s_1)\} \rightarrow s_4$$

$$s_0 \& \{(7s_1)\} \rightarrow s_5$$

$$s_0 \& \{(8s_1)\} \rightarrow s_6$$

$$s_0 \& \{(9s_1)\} \rightarrow s_7$$

$$s_2 \rightarrow s_1$$

$$s_3 \rightarrow s_1$$

$$s_4 \rightarrow s_1$$

$$s_5 \rightarrow s_1$$

$$s_6 \rightarrow s_1$$

$$s_7 \rightarrow s_1$$

surface or “mirror”.

In this CA empty space will be represented by state  $s_0$ . We now require a particle which will travel in the direction it is “pointing”. It should be as simple as possible so that the rules for the reflection off a mirror are simple to construct. The particle should be reversible in the sense that if two consecutive time steps are reversed the particle will travel in the opposite direction. To make the reflection clear the particle should travel diagonally across the lattice so that it does not reflect directly back along its path, the mirror surfaces will therefore be horizontal and vertical. The other possibility would be for the particle to travel horizontally and vertically and for the mirror surfaces to be diagonal to the lattice but the former option was thought to be more appropriate. A simple particle which fulfills these constraints within our CA framework and the neighbourhood which is required are shown in Figure 2.4.

The rules for propagation of this particle are therefore,

$$\begin{aligned} s_0 \& \{(s_0^{(t-1)}) \& (2s_1)\} \rightarrow s_1 \\ s_1 \& \{(s_0^{(t-1)}) \& (1s_1)\} \rightarrow s_1 \\ s_1 \& \{(s_1^{(t-1)}) \& (1s_1)\} \rightarrow s_0 \\ s_0 \& \{(s_1^{(t-1)}) \& (2s_1)\} \rightarrow s_0 \end{aligned}$$

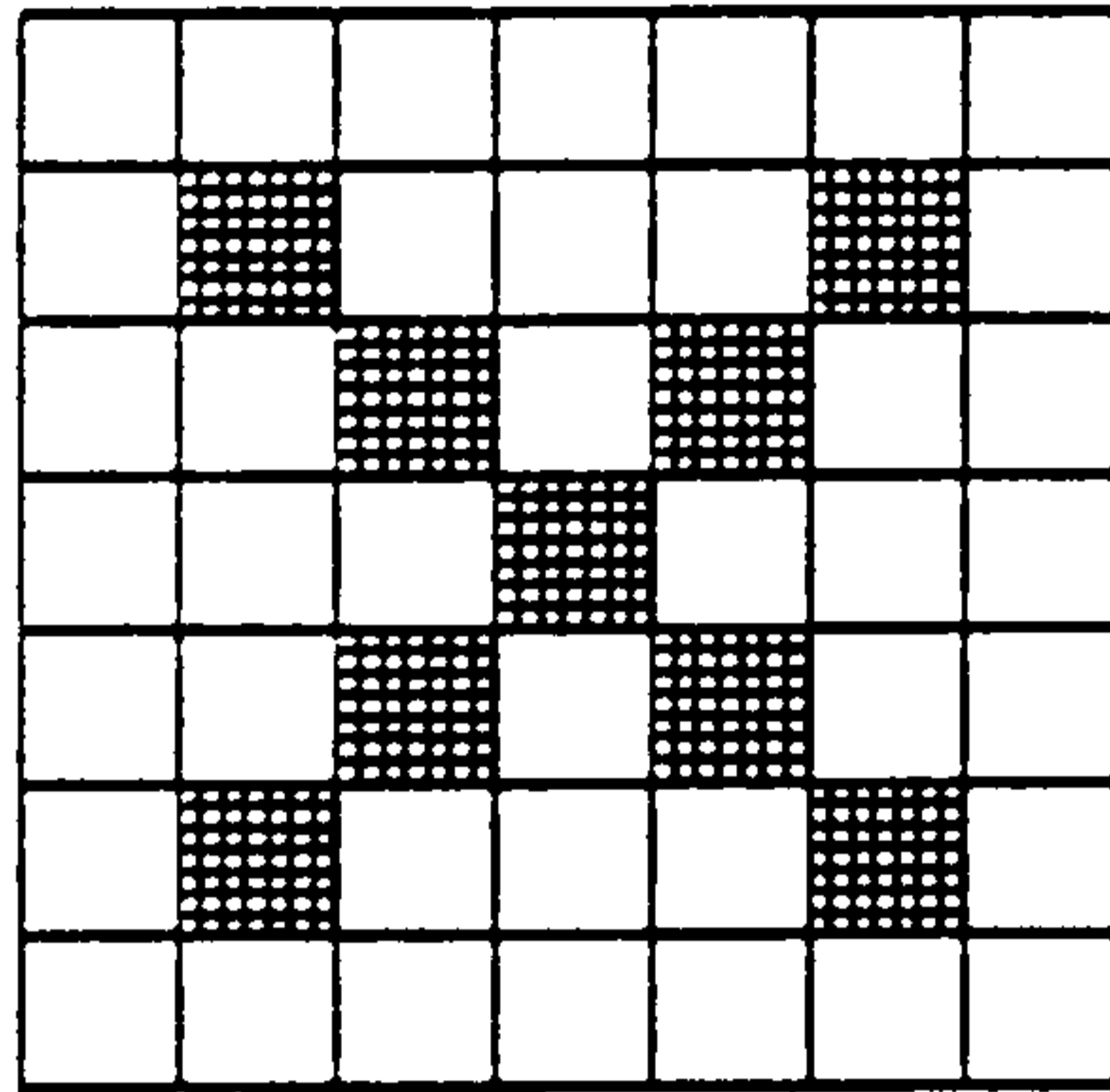
The mirror will consist of cells in state  $s_2$  and this state will not be used for any other purpose. This makes the rule for stability of the mirrors very simple,

$$s_2 \rightarrow s_2$$

The shape of the mirrors is chosen so that there is as much overlap as possible between the mirror and the neighbourhood of the particle reflecting from it (see Figure 2.5).

Figure 2.3: Summary of Bouncer.

Six states ( $s_0, \dots, s_5$ ).



neighbourhood

$$s_z \rightarrow s_0$$

$$s_2 \rightarrow s_2$$

$$s_2 \& \{(s_2^{(t-1)}) \& (1s_3)\} \rightarrow s_5$$

$$s_5 \& \{(s_2^{(t-1)})\} \rightarrow s_2$$

$$s_0 \& \{(s_0^{(t-1)}) \& (2s_1)\} \rightarrow s_1$$

$$s_1 \& \{(s_0^{(t-1)}) \& (1s_1)\} \rightarrow s_1$$

$$s_1 \& \{(s_1^{(t-1)}) \& (1s_1)\} \rightarrow s_0$$

$$s_0 \& \{(s_1^{(t-1)}) \& (2s_1)\} \rightarrow s_0$$

$$s_1 \& \{(s_0^{(t-1)}) \& (4s_2)\} \rightarrow s_3$$

$$s_1 \& \{(s_1^{(t-1)}) \& (1s_2)\} \rightarrow s_4$$

$$s_4 \& \{(s_1^{(t-1)}) \& (1s_3)\} \rightarrow s_4$$

$$s_0 \& \{(s_0^{(t-1)}) \& (1s_3)\} \rightarrow s_1$$

$$s_0 \& \{(s_0^{(t-1)}) \& (1s_4)\} \rightarrow s_0$$

$$s_3 \& \{(s_1^{(t-1)}) \& (4s_2)\} \rightarrow s_3$$

$$s_3 \& \{(s_3^{(t-1)}) \& (4s_2)\} \rightarrow s_0$$

$$s_0 \& \{(s_0^{(t-1)}) \& (1s_5)\} \rightarrow s_0$$

Figure 2.4: Bouncer particle and neighbourhood.

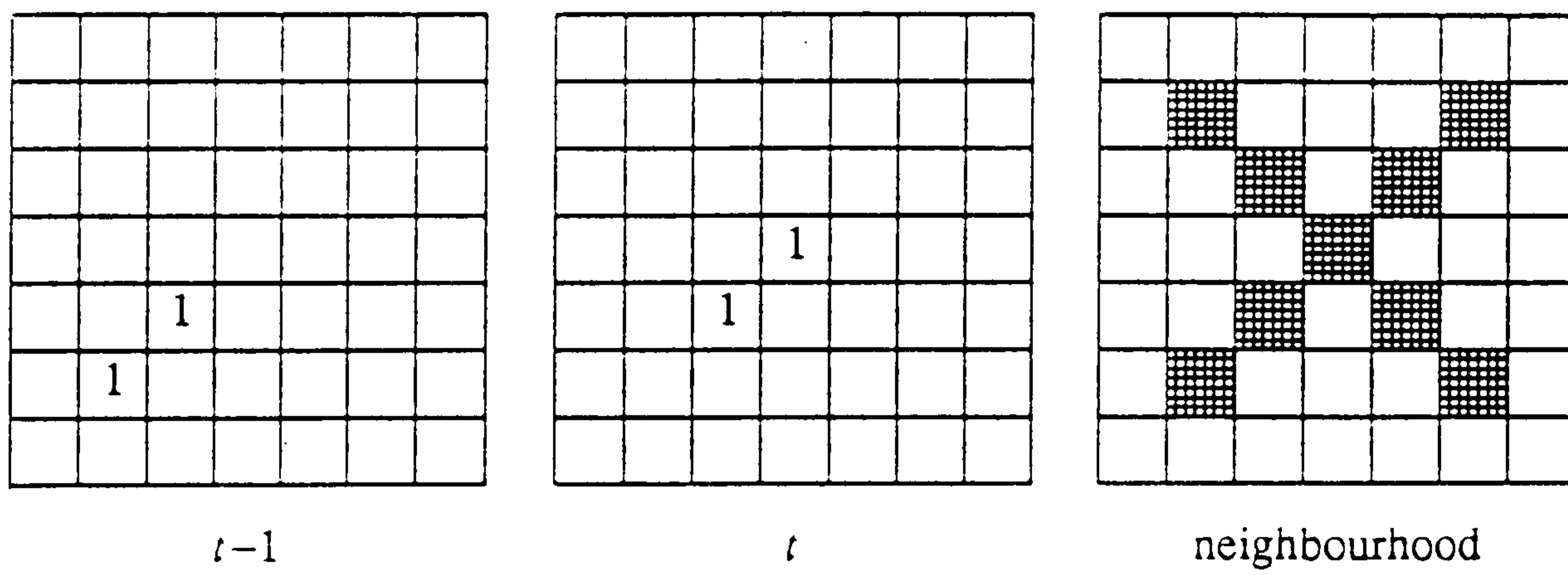
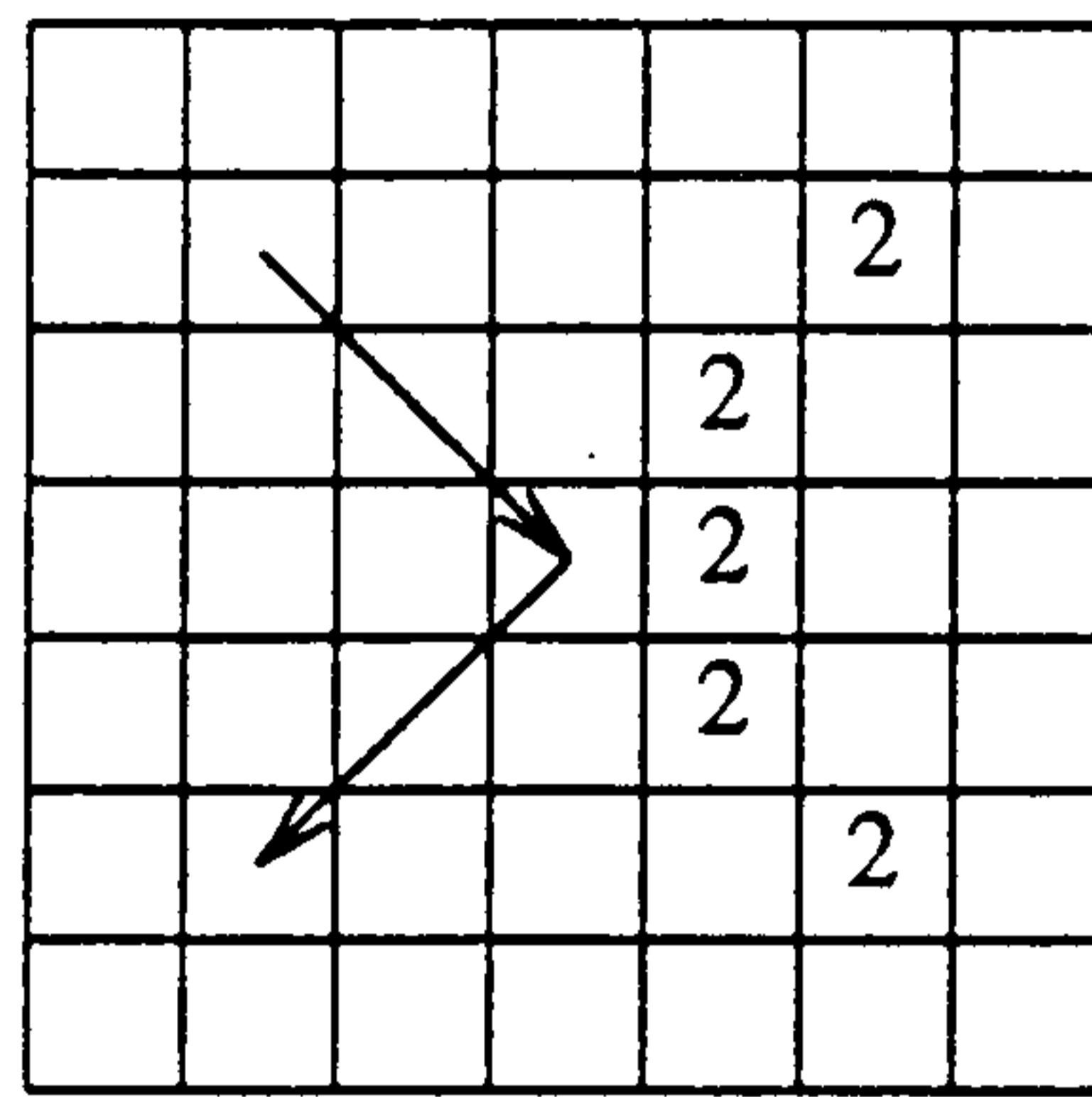


Figure 2.5: **Bouncer mirror.**



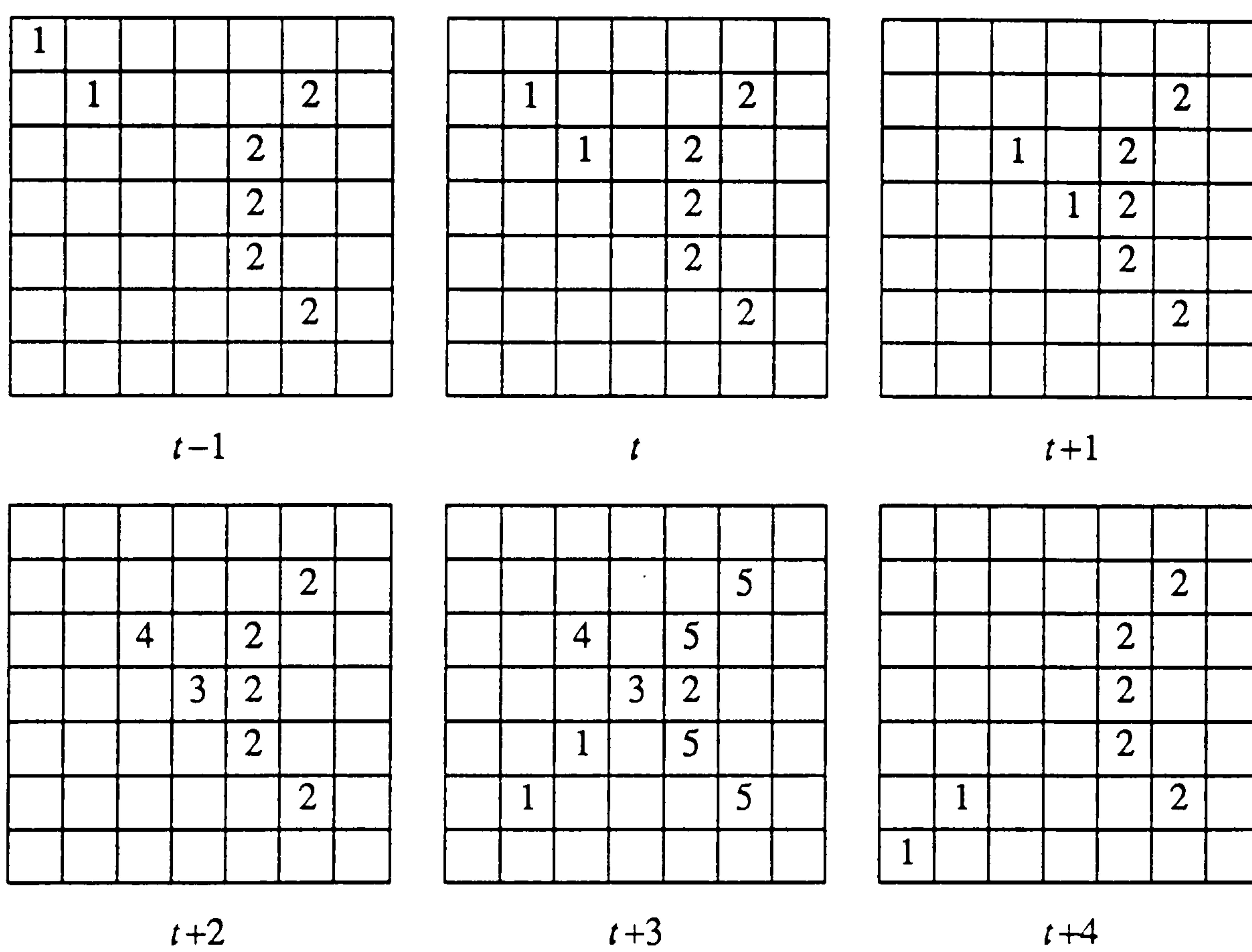
When the particle hits the mirror it must be deactivated to prevent it propagating into the mirror. This is done by converting the head of the particle into state  $s_3$  and the tail into state  $s_4$ . The mirror must also be deactivated to prevent any spurious reflection effects, so the active parts of the mirror are converted to state  $s_5$ . We can now construct the reflected particle by using the marked head of the particle in state  $s_3$  as a reference point. Finally we must delete the marked cells and return the mirror to its working state. If we now list the full rule set so that the highest priority rule occurs at the bottom of the list we obtain the following,

$$\begin{aligned}
& s_x \rightarrow s_0 \\
& s_2 \rightarrow s_2 \\
& s_2 \& \{ (s_2^{(t-1)}) \& (1s_3) \} \rightarrow s_5 \\
& s_5 \& \{ (s_2^{(t-1)}) \} \rightarrow s_2 \\
& s_0 \& \{ (s_0^{(t-1)}) \& (2s_1) \} \rightarrow s_1 \\
& s_1 \& \{ (s_0^{(t-1)}) \& (1s_1) \} \rightarrow s_1 \\
& s_1 \& \{ (s_1^{(t-1)}) \& (1s_1) \} \rightarrow s_0 \\
& s_0 \& \{ (s_1^{(t-1)}) \& (2s_1) \} \rightarrow s_0 \\
& s_1 \& \{ (s_0^{(t-1)}) \& (4s_2) \} \rightarrow s_3 \\
& s_1 \& \{ (s_1^{(t-1)}) \& (1s_2) \} \rightarrow s_4 \\
& s_4 \& \{ (s_1^{(t-1)}) \& (1s_3) \} \rightarrow s_4 \\
& s_0 \& \{ (s_0^{(t-1)}) \& (1s_3) \} \rightarrow s_1 \\
& s_0 \& \{ (s_0^{(t-1)}) \& (1s_4) \} \rightarrow s_0 \\
& s_3 \& \{ (s_1^{(t-1)}) \& (4s_2) \} \rightarrow s_3 \\
& s_3 \& \{ (s_3^{(t-1)}) \& (4s_2) \} \rightarrow s_0 \\
& s_0 \& \{ (s_0^{(t-1)}) \& (1s_5) \} \rightarrow s_0
\end{aligned}$$

This gives the sequence of events in Figure 2.6.

This CA is obviously periodic if we set up mirrors on the lattice so that the particle follows a closed path. However in the case of a random lattice the CA will normally reach a

Figure 2.6: Bouncer particle reflecting off mirror.





limit point where only mirror states ( $s_2$ ) exist. Since the CA can behave periodically and will usually have a non-quiescent limiting state we may place this CA in class 2.

### Diffuser

This CA is summarised in Figure 2.7. This CA simulates the diffusion of solid regions of state  $s_1$ . This is done by making the probability of a cell in state  $s_0$  changing to state  $s_1$  proportional to the number of cells in state  $s_1$  in its neighbourhood. Similarly, cells in state  $s_1$  change to  $s_0$  based on the number of cells in state  $s_0$ . In this way a cell in state  $s_1$  which is in a dense  $s_1$  region will tend to stay in that state and a cell in state  $s_0$  in a dense  $s_1$  region will tend to change to  $s_1$ . Since the diffusion of particles is simulated probabilistically, the CA will not conserve the numbers of cells in state  $s_1$  exactly although it will do so on average. In fact, if we consider the time series of the number of particles after each individual cell evolution, we obtain a random walk with zero drift and standard deviation,

$$2\sqrt{(\log_e(N/n_1))^{-3}+(\log_e(N/n_0))^{-3}} \quad 2.7$$

Where  $N$  is the total number of cells on the lattice and  $n_0$  and  $n_1$  are the number of cells in states  $s_0$  and  $s_1$ . See Chapter 3 for the theoretical basis of this result. We can place this CA in class 3, because it will never reach a limiting or periodic state, but will evolve chaotically forever.

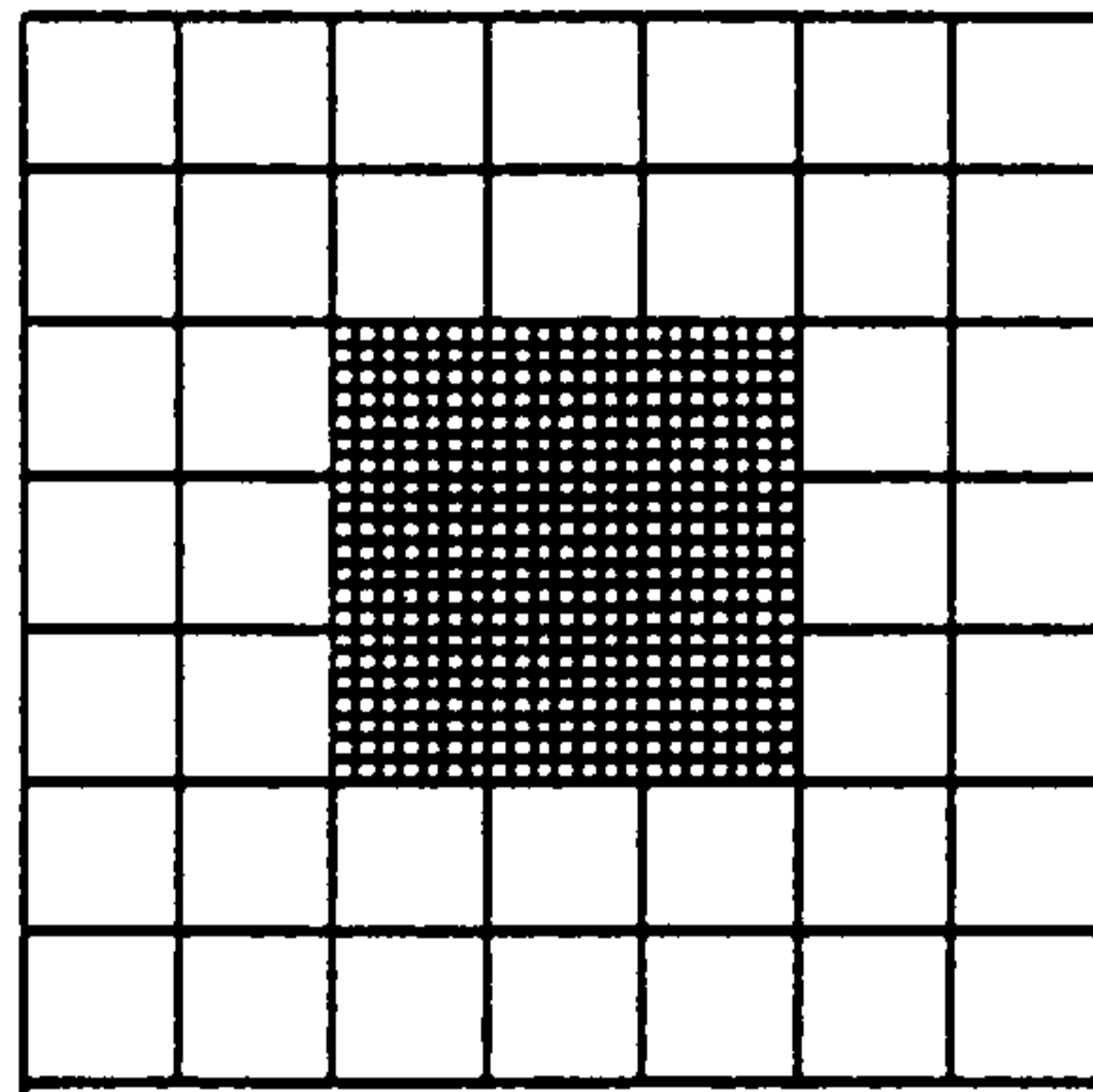
### Reverse

This CA is summarised in Figure 2.8. This CA utilises the past state property of the general rule scheme to construct a reversible CA. That is if two consecutive states of the lattice are time order reversed then the CA will evolve through the previous global states



Figure 2.7: Summary of Diffuser.

Two states ( $s_0, s_1$ ).



neighbourhood

$$s_0 \& \{(1s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.1$$

$$s_0 \& \{(2s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.2$$

$$s_0 \& \{(3s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.3$$

$$s_0 \& \{(4s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.4$$

$$s_0 \& \{(5s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.5$$

$$s_0 \& \{(6s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.6$$

$$s_0 \& \{(7s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.7$$

$$s_0 \& \{(8s_1)\} \rightarrow s_1 \mid \text{Pr} = 0.8$$

$$s_1 \& \{(1s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.1$$

$$s_1 \& \{(2s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.2$$

$$s_1 \& \{(3s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.3$$

$$s_1 \& \{(4s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.4$$

$$s_1 \& \{(5s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.5$$

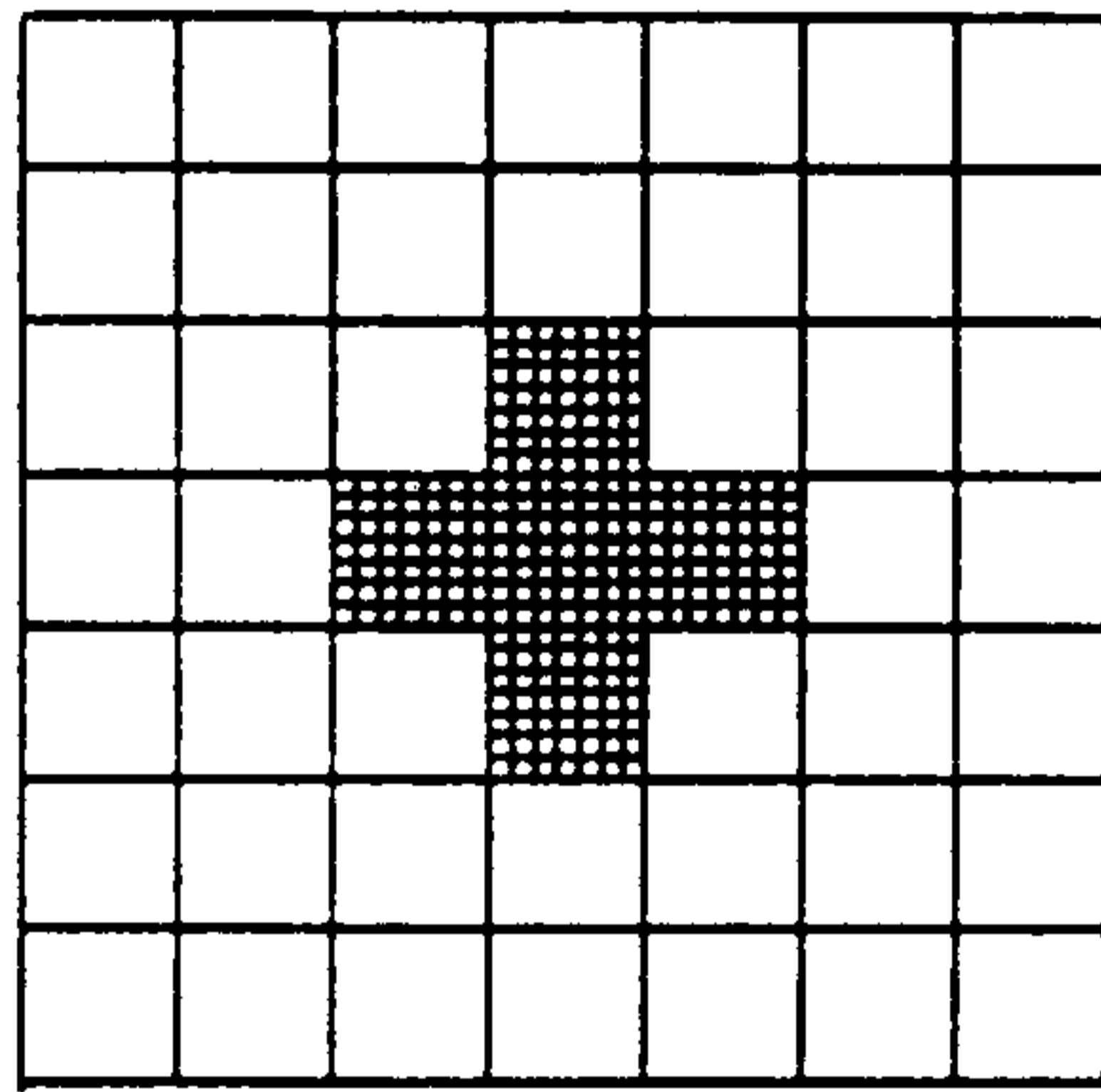
$$s_1 \& \{(6s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.6$$

$$s_1 \& \{(7s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.7$$

$$s_1 \& \{(8s_0)\} \rightarrow s_0 \mid \text{Pr} = 0.8$$

Figure 2.8: Summary of Reverse.

Two states ( $s_0, s_1$ ).



neighbourhood

$$s_x \& \{(s_0^{(t-1)})\} \rightarrow s_0$$

$$s_x \& \{(s_0^{(t-1)}) \& (2s_1)\} \rightarrow s_1$$

$$s_x \& \{(s_1^{(t-1)})\} \rightarrow s_1$$

$$s_x \& \{(s_1^{(t-1)}) \& (2s_1)\} \rightarrow s_0$$

of the lattice until it reaches the initial configuration. Note that it is necessary to specify the state of the lattice at the instant of time before the initial state. This would usually be set to be completely quiescent (every cell in state  $s_0$ ), but could be set to any of the possible global configurations of the lattice. The rule set for this CA simply alternate cells with the correct neighbourhood configuration but based on the state at the previous time step rather than the current state. This means the rule set belongs to the simplest reversible rule scheme proposed by Fredkin and Toffoli[50],

$$c_i^{(t+1)} = F [\{c_j^{(t)}: j \in R_i\}] - c_i^{(t-1)}$$

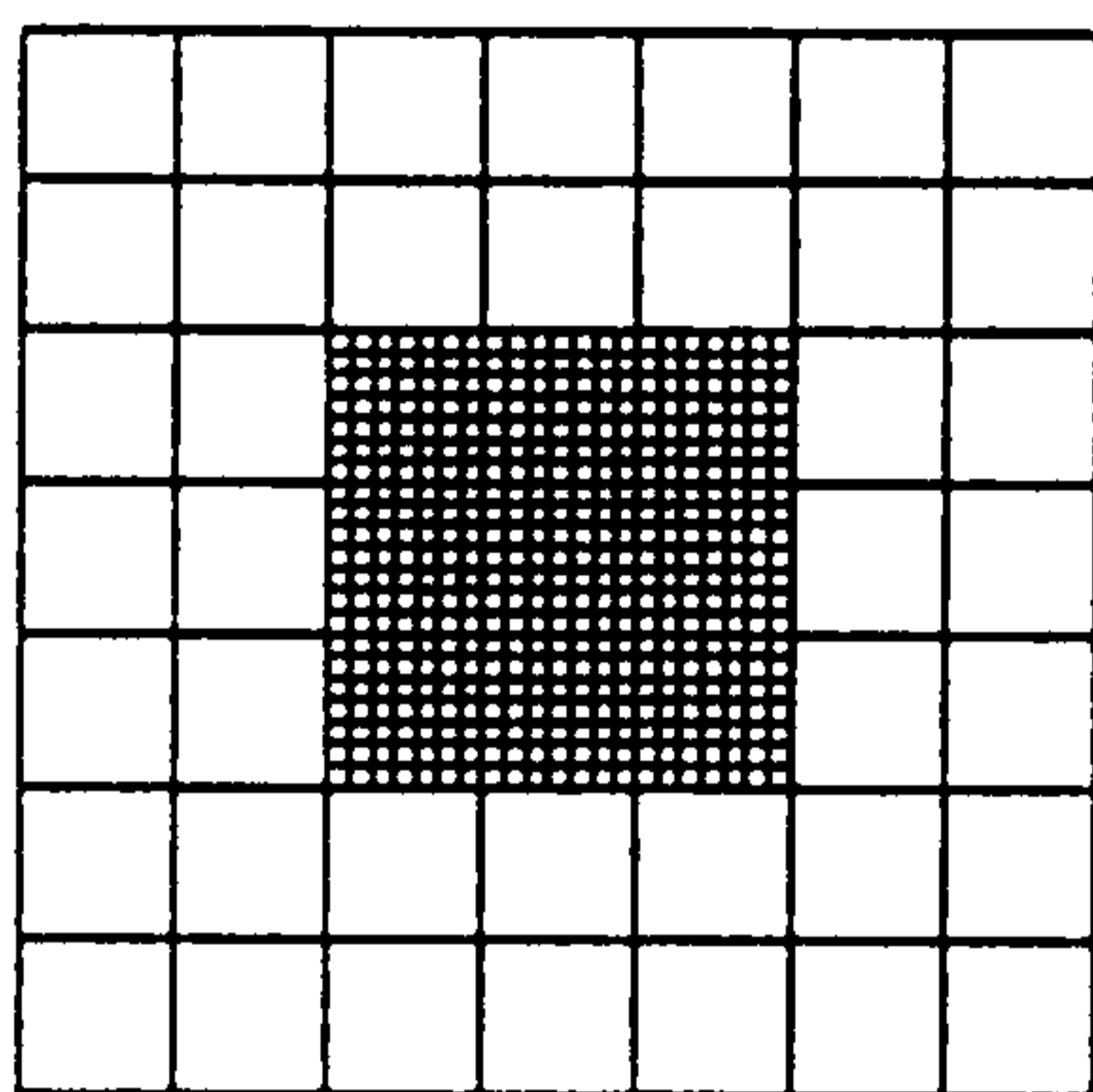
This CA behaves chaotically also and is thus class 3.

### **Twolife**

This CA is summarised in Figure 2.9. This CA is an extension of Conways ‘‘Game of Life’’[49]. We have two active states  $s_1$  and  $s_2$  which are both governed by the standard ‘‘Game of Life’’ rules. A cell is born, becomes non-quiescent, if it has three non-quiescent neighbours. A cell survives, remains non-quiescent, if it has two or three non-quiescent neighbours. Otherwise a cell dies or becomes quiescent. They are both defined around the same quiescent state  $s_0$  so that the two states will be competing with each other for the available ‘‘empty’’ space. This CA behaves in general as a standard ‘‘Game of Life’’ CA. Indeed if the lattice contains separate sparse regions of states  $s_1$  and  $s_2$  then we will obtain evolution exactly as for the standard ‘‘Game of Life’’. However when the two active states are mixed together there is a potential rule conflict since state  $s_0$  may change to either of the two active states and the two conditions for these transitions can be satisfied simultaneously. This conflict was resolved by taking the first rule in the set which was satisfied. This resulted in state  $s_1$  being favoured over state  $s_2$  and conse-

Figure 2.9: Summary of Twolife.

Three states ( $s_0, s_1, s_2$ ).



neighbourhood

$$s_x \rightarrow s_0$$

$$s_0 \& \{(3s_1)\} \rightarrow s_1$$

$$s_1 \& \{(2s_1)\} \rightarrow s_1$$

$$s_1 \& \{(3s_1)\} \rightarrow s_1$$

$$s_0 \& \{(3s_2)\} \rightarrow s_2$$

$$s_2 \& \{(2s_2)\} \rightarrow s_2$$

$$s_2 \& \{(3s_2)\} \rightarrow s_2$$



quently state  $s_2$  tended to reach a stable global state earlier than state  $s_1$ . The problem of rule conflicts will be considered further in Chapter 4. In the light of this analysis we may classify this CA as class 4.

## Summary

We have brought together a number of different mechanisms currently employed by workers to control the evolution of the CA lattice. These have been synthesised into a CA framework which gives us the flexibility to explore the complete range of CA behaviours. The framework is closely related other computational structures such as neural networks, threshold functions and logical operators. The rule space is however of a manageable size, so that it would be feasible to explore a large fraction of the entire rule space in order to discover, for example, the detailed distribution of behaviours in the rule space. Finally we described six CA whose behaviour is a result of utilising specific parts of our framework in the implementation of the rule set.

### 3. A Statistical Mechanics Approach

The second law of thermodynamics tells us that isolated microscopically reversible systems tend to states of maximal entropy or maximal disorder. Dissipative systems however which are microscopically irreversible evolve to more ordered states. This is due to the many-to-one mapping which is occurring, that is many states of the system will map or evolve to the same future state. This is stated formally in Liouville's theorem,

The volume occupied by any ensemble of states in phase space remains constant as the ensemble evolves in time for a Hamiltonian energy conserving system.

Furthermore for dissipative systems such ensembles must evolve to continually decreasing volumes of phase space and thus in the infinite time limit the volume of phase space occupied by the limit set of the system must be zero or alternatively must have a dimension at least one less than that of the phase space.

This is the physical basis of the self-organising behaviour of CA and indeed of natural and physical systems[26, 27, 29-32, 51, 52]. However, Vichniac[36] points out that reversible non-dissipative CA can exhibit self-organisation, although here it is due to the absence of information losses. Spatial correlations which form locally conserved structures act as organising centres inducing waves which move across the lattice at supraluminal speed (that is greater than the width of the neighbourhood per time step). In fact no information travels at greater than "the speed of light", the waves belong to the future light cones of the initial locally conserved shapes. This behaviour is a consequence of the fact that the locally reversible dynamics are also deterministic unlike the quantum mechanical nature of real physical processes.

Investigations of simple self-organisation phenomena in physical and chemical systems have typically used simple mathematical models. These have been based for example on

the Boltzmann transport equation obtained by averaging over an ensemble of microscopic states and assuming successive collisions between molecules are statistically uncorrelated. Another group of systems which have been used extensively are idealisations of the Navier-Stokes hydrodynamic equations which lead to dissipative non-linear differential equations. The typical behaviour of the limit sets for these types of systems as the non-linearity is increased is for them pass from limit points to limit cycles. The number of limit cycles then increase until chaotic behaviour occurs. This is analogous to the observation made by Langton[45] regarding the natural ordering of the qualitative CA classes and by Wolfram[43] noted in section 2 concerning *totalistic* CA.

However this type of numerical modelling of the physical world is far removed from the actual processes which are occurring. Firstly we have stylised the discrete physical system into continuous differential equations. Then in order to solve these by numerical integration we convert the differential equations to finite difference equations with discrete space and time. Finally we constrain real valued variables into finite computer words leading to round-off errors.

Now most physical quantities, for example density, temperature, pressure, are defined statistically. As we consider smaller and smaller volumes of a gas the mean density in that region will begin to fluctuate more and more in both space and time. This will be due to the number of particles (atoms or molecules) in the volume considered becoming very small and thus the statistical variations becoming noticeable. Conversely as larger and larger volumes are considered the mean density of the gas will approximate a continuous function closer and closer. Note that as the volume is made larger the resolution of the density increases but the spatial resolution decreases or as we reduce the error in the density the error in the spatial position of that density decreases. In other words we have an



uncertainty relation between the error in the density and the error in its spatial position. This applies to the temporal domain also, that is between the rate of change of density and time, as noted above.

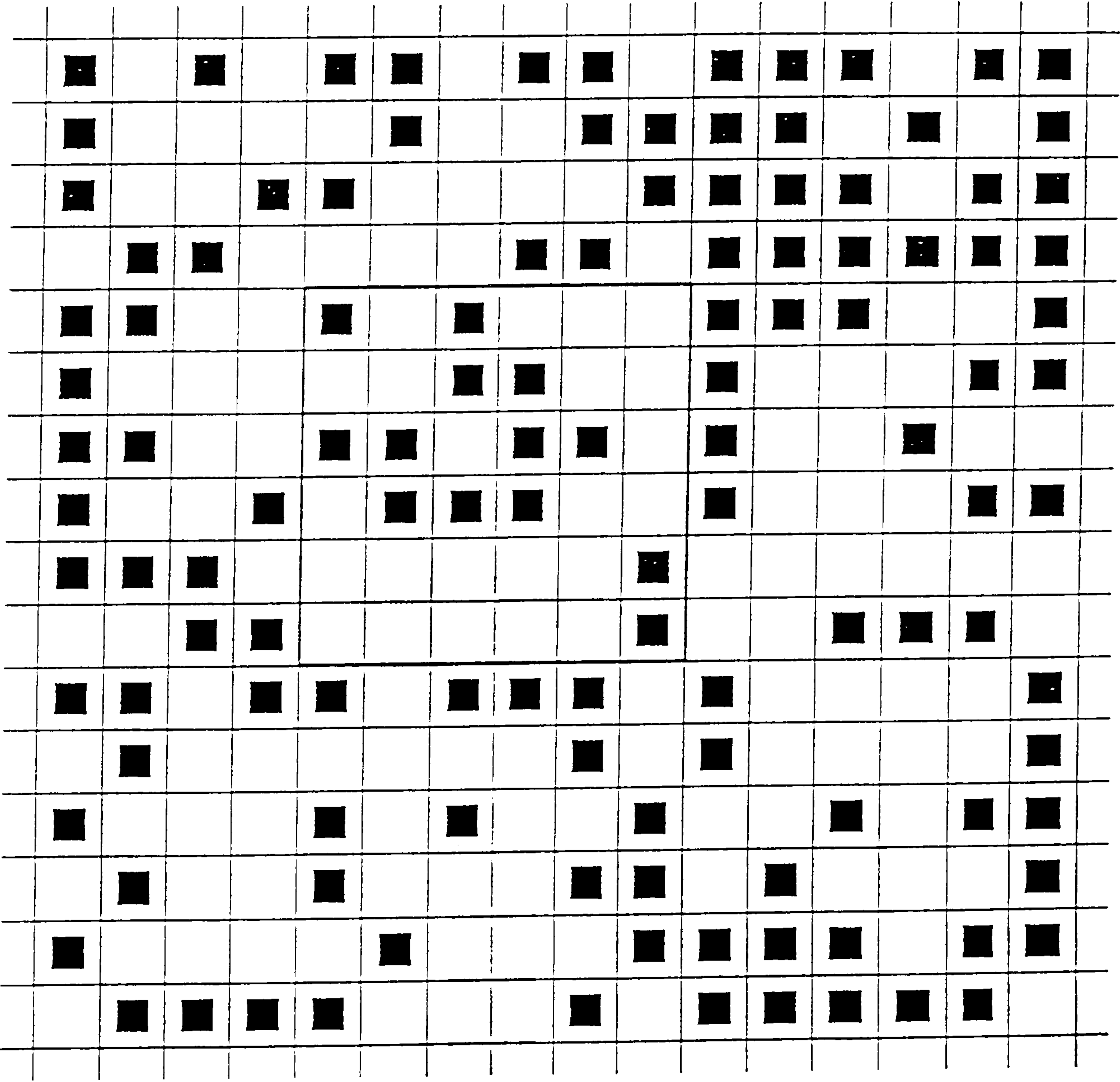
If we consider space and time to be divided into discrete uniformly sized cells such that each cell may hold only one particle then a volume will be equivalent to a number of cells. Thus for a volume  $V$  (i.e.  $V$  cells) the density may take values between 0 (no cells occupied) and 1 (all cells occupied) and we have an accuracy in terms of “bits” of  $\log_2(V)$ . Note however, that for additional cells the density is specified to this accuracy at the cost of that additional cell only. Thus for volumes of space much larger than the sampling volume  $V$  we may specify the density to any required accuracy by using a suitably large  $V$  with only 1 bit per cell as can be seen in Figure 3.1. We still lose spatial accuracy though as we increase the density accuracy. A further point is that with this representation each bit has equal significance, whereas with the conventional representation of real numbers the bits have exponentially ranked significance. Furthermore, in numerical simulations the lower order bits will contain no useful information at all due to roundoff errors.

The situation so far considered has been static, if we now consider the particles of the gas to be behaving dynamically then the same arguments will now apply in the temporal domain. Thus the statistical behaviour of the density will be just as if it were driven by a dynamical equation.

It can be seen that the discrete system introduced above is equivalent to a CA with two possible states per cell ( $k=2$ ). A CA with more than two states per cell allows us to specify the “density” of more than one variable in each cell. This is slightly different to the normal view of CA in which a cell may take one of  $k$  possible states, however this is



Figure 3.1: Statistical density function.



simply equivalent to  $\log_2(k)$  bits of storage per cell or  $\log_2(k)$  of our “density” variables per cell.

For  $V=1$  we have a Boolean valued density field the evolution of which is given exactly by the CA rule. If however  $V>1$  we have a scalar field whose evolution is non-deterministic. This is because for a given local density  $\rho$  the lattice could have many different configurations and each will in general evolve differently under the CA rule. We can calculate the probability of a change in density  $\Delta\rho$  by calculating the probability of the cell transitions which would result in  $\Delta\rho$ . In general there will many possible sets of transitions so we must perform a summation of all possible sets. The probabilities will be in terms of the local density  $\rho$ . We will thus obtain a probability distribution  $P(\Delta\rho)$  of the density changes  $\Delta\rho$ .

If the probability function is very peaked then the evolution will be very close to the deterministic evolution of the CA (for a delta function the evolution would be exactly that of the CA). This is equivalent to a finite difference algorithm for the evolution of the density as noted by Toffoli[35] and will occur if the rule and  $V$  are suitably chosen and the value of the density field is not too close to its extrema (i.e 0 and 1). Note that conventional differential equations will tend to fail at these extrema also which correspond to zero and infinite density respectively.

As we increase the size of the neighbourhood  $r$  the CA rule approaches a classical density function. When  $r=N^2$  the rule specifies the global transition function for the CA lattice. Our probability function will then be at its least accurate because the highest possible number of neighbourhood configurations will correspond to single density value.

If the density is initially uniform across the entire lattice so that each cell is assigned state

---

<sup>2</sup>  $N$  is the number of cells on the lattice.

one with probability  $p$  and a zero with probability  $(1-p)$  then the density will evolve similarly across the whole lattice and the density or the number of cells in state 1 ( $\rho$ ) can be modelled by a single dynamical equation,

$$\rho_{t+1} = \rho_t + P_{0,1}(1-\rho_t) - P_{1,0}\rho_t \quad 3.1$$

where  $P_{0,1}$  is the probability of a cell in state 0 changing to state 1 and similarly for  $P_{1,0}$ .

For the generalised CA developed in section 2 which was specified as follows,

$$s_i \& \{C_k(s_k)\} \rightarrow s_o \mid \text{Pr} = p \quad 3.2$$

We may generalise equation 3.1 to model the variation of the numbers of cells in each state. We thus obtain a system of dynamical equations,

$$n_i^{(t+1)} = n_i^{(t)} + \left[ \sum_j P_{ji} n_j^{(t)} \right] - \left[ \sum_j P_{ij} \right] n_i^{(t)} \quad 3.3$$

Where  $n_i$  is the number of cells in state  $s_i$  which we will call the *state populations*.  $P_{ij}$  is the probability of state  $s_i$  changing to state  $s_j$ . The  $P_{ij}$  are related to the probabilities of the conditions  $C_k(s_k)$  occurring.

The dynamical system (DS) 3.3 describes the evolution of the number of cells in the various states or the density of those states. The states are assumed to be uniformly randomly distributed over the lattice. This will be true on a large scale if the initial state was uniform. However locally the CA rule will in general impose a structure on the distribution of states. If we ignore this complication for the moment and assume that the CA maintains the uniform random distribution of states and only alters the relative concentrations of the various states then the dynamical system will model the behaviour of the CA very

---

<sup>3</sup> Note this is only strictly true for an ensemble average, assuming the lattice remains homogeneous. The probabilities  $P_{0,1}$  and  $P_{1,0}$  are not constant but depend, as discussed later, on the number of cells in each state[33,34,53].



well. We should also note that the exact evolution of the CA will differ for different initial random states with the same state populations. The dynamical system will give the average behaviour over all possible initial random states with the same state populations.

This suggests that we should ask what the probability of a given set of state populations will be at some time  $t$  in the evolution of the lattice or alternatively what is the probability distribution for the state populations vector as a function of time. This is called the master equation or probability generating function in stochastic dynamical systems theory. This approach has been used extensively in the field of population modelling, especially in relation to the dynamics of epidemics[53-57]. A stochastic approach is necessary since the numbers of infected individuals is initially very small and so statistical variation plays a very important role in determining the subsequent development of the epidemic. This approach has also been used to some extent in modelling CA[58-60] although the rules are typically simple threshold rules. The master equation for a CA can be written in general terms as follows,

$$P(\{s_i'\}, t+1) = \sum_{\{s_i\}} Q(\{s_i'\} | \{s_i\}) P(\{s_i\}, t) \quad 3.4$$

where

$$Q(\{s_i'\} | \{s_i\}) = \prod_i Q(s_i' | s_i, \{s_i'\}) \quad 3.5$$

Equation 3.4 states that the probability of obtaining the set of states  $\{s_i'\}$  at time  $t+1$  is given by the sum over all possible sets of states of the product of the probability of the set of states  $\{s_i\}$  going to  $\{s_i'\}$  and the probability of having the set of states  $\{s_i\}$  at time  $t$ . Equation 3.5 states that the probability of obtaining the new set of states given the old set is the product of the probabilities of each of the states  $s_i$  going to  $s_i'$  given their



neighbourhood  $\{s_{i'}\}$ . In certain simple rule schemes the probabilities  $Q(s_i' | s_i, \{s_{i'}\})$  have a simple form. For example for 1D lattice, totalistic, binary ( $k=2$ ) CA Grinstein[60] has shown that if we require detailed balance to hold for transitions between the two states then we have the following expression,

$$Q(s_i' | s_i, \{s_{i'}\}) = 1/2[1 + s_i s_i' \tanh(a + b s_i \sum_{i'} s_{i'})] \quad 3.6$$

However for a general totalistic rule set the probabilities  $Q(s_i' | s_i, \{s_{i'}\})$  will be hypergeometric functions and solution of the master equation will be very difficult if not impossible. However as noted above use of the master equation is only necessary when the population levels are likely to be very small and so our dynamical system approach will be sufficient for almost all the phase space of the CA. A second use of the master equation (3.4) is to indicate the typical spread of population values we would expect for a set of evolutions of the lattice. This is given by the width of the probability distribution. We can obtain this by simply performing a large enough set of evolutions to give us an adequate sample of the distribution. However we can calculate the standard deviation of the probability distribution of the changes in the state populations in the following way. Firstly we note the following result from statistical theory, let  $x_1$  be normally distributed as  $N[\mu_1, \sigma_1]$  and  $x_2$  be normally distributed as  $N[\mu_2, \sigma_2]$  where  $x_1$  and  $x_2$  are independent random variables, then  $a_1 x_1 \pm a_2 x_2$  is normally distributed as  $N[a_1 \mu_1 \pm a_2 \mu_2, \sqrt{a_1^2 \sigma_1^2 + a_2^2 \sigma_2^2}]$ . Now the changes in state population  $n_i$  can be expressed as follows,

$$\Delta n_i = \sum_{j=1}^N \Delta_{ji} - \sum_{j=1}^N \Delta_{ij} \quad 3.7$$

where  $\Delta_{ij}$  means the number of cells changing from state  $s_i$  to state  $s_j$ . Now the  $\Delta_{ij}$  are

not strictly independent but in general will be approximately independent. The probability distribution of the  $\Delta_{ij}$  is simply our transition probability function. The standard deviation of the state population changes or more correctly their variance is thus given by the following expression,

$$\sigma^2(\Delta n_i) = \sum_{j=1}^N \sigma^2(\Delta_{ji}) + \sum_{j=1}^N \sigma^2(\Delta_{ij}) \quad 3.8$$

We will see later that the transition probability function is very close to the normal distribution function and a simple expression exists for its standard deviation in terms of the state populations. Note that equation 3.8 implies that an approximation to the probability distribution of the  $\Delta n_i$  is given by,

$$N \left[ \sum_{j=1}^N \mu_{ji} - \sum_{j=1}^N \mu_{ij}, \sigma(\Delta n_i) \right] \quad 3.9$$

The transition probabilities depend on the probabilities of the conditions  $C_k(s_k)$ . The conditions  $C_k(s_k)$  will consist of numbers of cells in certain states required for the given state transition to occur. Thus the transition probabilities are related to the probability of finding a given number of cells in a given state in a region size  $r$  on a lattice with the distribution of states  $\{n_i : i = [0, \dots, k-1]\}$ . We need to be able to specify this probability in terms of  $n_i$ ,  $N$ ,  $r$  and  $k$  since this is the only information which the dynamical system can contain about the state of the lattice.

This probability is related to a probability system termed the *urn models*[48] in which we imagine drawing samples of coloured balls from an urn. If we identify the CA lattice with the urn, the sample with the neighbourhood and the coloured balls with the various states then the probability of finding a given number of cells in a given state will be

equivalent to the probability of finding a given number of balls of a given colour in our urn sample. There are three types of sampling which may be applied to this model,

### Unordered samples

The number of balls in the sample are drawn simultaneously.

### Ordered samples with replacement

The balls are drawn one at a time and each is replaced before the next ball is drawn.

### Ordered samples without replacement

The balls are drawn one at a time but are not replaced before the next ball is drawn.

The sampling relevant to our CA is the *unordered sampling* which in fact is equivalent to *ordered sampling without replacement* because the important factor is whether a given ball/cell can be sampled more than once.

The probability that an unordered sample of size  $r$  drawn from an urn containing  $n$  red balls and  $N-n$  white balls (where  $N$  is the total number of balls) contains exactly  $c$  red balls is,

$$P_c = \frac{\binom{n}{c} \binom{N-n}{r-c}}{\binom{N}{r}} \quad 3.10$$

These are known as the *hypergeometric probabilities*. For an ordered sample with replacement the probability is,

$$P_c = \frac{\binom{r}{c} n^c (N-n)^{r-c}}{N^r} \quad 3.11$$



or equivalently,

$$P_c = \binom{r}{c} p^c q^{r-c} \quad 3.12$$

where  $p=n/N$  and  $q=1-p$  and these are known as the *binomial probabilities*.

Now if we consider the CA neighbourhood to be a random sampling of  $r$  cells from the entire CA lattice then the binomial probabilities are appropriate since a cell may be sampled twice. Random sampling of the lattice will also remove the effect of non-uniformity in the distribution of states caused by the CA rule imposing characteristic patterns on the lattice. This is equivalent to “mixing” the CA lattice after each time step, that is swapping the states of two randomly chosen cells until every cell has been swapped. Note that mixing the CA lattice is not strictly identical to using a randomly sampled neighbourhood since it does not allow a cell to be sampled more than once. However in practice for all but very small lattices the hypergeometric and binomial probabilities are almost identical (the actual error will be studied later). The reason for mentioning the point about random sampling of the CA neighbourhood is that there exists a normal distribution approximation to the binomial probabilities which is identical for most situations (again the errors will be studied later).

$$P_c = \frac{1}{\sqrt{2\pi r p q}} \exp\left[-\frac{(c-rp)^2}{2r p q}\right] \quad 3.13$$

This approximation is more manageable than the factorial expressions in equations 3.10, 3.11 and 3.12.

An interesting point to note here about the transition probability function is that it is scale invariant. That is to say if we consider scaling all the population variables  $n_i$  by some



factor  $K$  the value of the transition probability remains the same. This is because  $p=Kn/KN=n/N$  and  $q=1-p$ . This means that the evolution of the dynamical system is scale invariant as we can see if we write down the general dynamical system with the scaled population variables,

$$\bar{n}_i^{(t+1)} = K n_i^{(t)} + \left[ \sum_j P_{ji} K n_j^{(t)} \right] - \left[ \sum_j P_{ij} \right] K n_i^{(t)} \quad 3.14$$

$\bar{n}_i^{(t+1)}$  is simply  $K$  times the original  $n_i^{(t+1)}$  so we can obtain the evolution of a dynamical system in which the initial conditions are  $K$  times greater by simply scaling every point in the evolutionary path by  $K$ . This suggests that we may obtain the global behaviour of an very large CA lattice by evolving a much smaller lattice.

Wolfram[34] and Packard and Wolfram[46] have introduced the concept of entropy of a CA configuration in order to quantify their randomness. The spatial set entropy for a set of two dimensional CA configurations is defined by considering a region of the lattice of  $X$  by  $Y$  cells. In general the number of configurations  $N(X,Y)$  which you would find by sampling a large number (in the limit an infinite number) of such regions will be less than the maximum possible number of configurations  $k^{XY}$ . The set entropy can then be defined thus,

$$s = \lim_{X,Y \rightarrow \infty} \frac{1}{XY} \log_k N(X,Y) \quad 3.15$$

The spatial set entropy characterises the set of configurations generated in the evolution of a CA without taking into account their probability of occurrence. A lattice may have all possible configurations present and still be highly non-random if some configurations have a much greater probability of occurrence than others. Wolfram therefore also defines a spatial measure entropy in terms of the probabilities  $p_i$  for the possible

configurations of the  $XY$  region,

$$s_{\mu} = \lim_{X,Y \rightarrow \infty} \frac{-1}{XY} \sum_{i=1}^{k^{XY}} p_i \log_k p_i \quad 3.16$$

Further tests of randomness are also necessary in order to guarantee the lattice is purely random. For example, one could imagine a lattice on which all possible configurations occur in some definite order. The above tests would lead to the conclusion that the lattice was purely random. The permutation frequency distribution - all possible orderings of all  $XY$  region configurations occur with equal frequency - deals with this problem. Wolfram[61] discusses tests of the randomness of CA configurations in detail (see Knuth[62] for statistical tests of randomness.)

In an analogous way we may define the entropy of a two dimensional lattice configuration. If we consider the number of different configurations in all possible  $XY$  regions on the lattice,  $N(X,Y)$  then we may define the entropy of the lattice as follows,

$$S = \lim_{X,Y \rightarrow \infty} \frac{1}{XY} \log_k N(X,Y) \quad 3.17$$

This can be generalised to the entropy of a  $d$ -dimensional lattice by considering the number of configurations in all possible volumes defined by the generator vectors  $v_i$  as follows,

$$S = \lim_{\alpha_1 \rightarrow \infty} \cdots \lim_{\alpha_d \rightarrow \infty} \frac{1}{\alpha_1 \cdots \alpha_d} \log_k N(\alpha_1 v_1, \cdots, \alpha_d v_d) \quad 3.18$$

where the  $\alpha_i$  are scalar parameters and  $N(\alpha_1 v_1, \cdots, \alpha_d v_d)$  is the number of different configurations in the volume defined by the  $\alpha_i v_i$ .

Now starting from an initial random state a given CA lattice will have a characteristic

entropy for each time step in its evolution. This entropy will in general decrease as the CA evolves due to the order imposed on the lattice by the CA. However mixing the CA lattice at each time step will destroy any patterns on the lattice and give a weighted random distribution of states whose entropy will in general be greater than that of the unmixed lattice. However if the lattice is not completely mixed then only those cells which have been mixed will have this random distribution and thus this is equivalent to applying a state weighted noise to the those cells that were mixed. This will increase the entropy of the lattice by some amount  $\Delta S$  and thus we can quantify the effect of mixing the lattice, which as we have noted is equivalent to adding noise, in terms of the increase in entropy  $\Delta S$ .

In order to simulate mixing of the CA lattice we can randomly choose two cells on the lattice and swap their states. However if we repeat this operation  $N_s$  times the number of swapped cells will not be  $2N_s$  since as we swap more cells the chances of choosing a cell which has already been swapped increases. Once a cell has been swapped it is effectively randomised and any further swaps will have no effect statistically. We therefore need an expression for the number of swapped cells as a function of the number of random swaps performed.

Let  $n_s$  be the number of swapped cells and  $N$  be the total number of cells in the lattice. The probability of choosing a swapped cell will be  $n_s/N$ . Consider a region of  $r$  cells, the probability of finding  $c$  unswapped cells will be given by equation 3.13 (strictly equation 3.10 however has already mentioned these are identical for most situations). So if we now swap all  $r$  cells then the increase in the number of swapped cells will be given by the mean number of unswapped cells in the region of  $r$  cells.

Now for our normal distribution approximation to  $P_c$  the mean number of cells is  $rp$  or



$r(N-n_s)/N$ . We can now write down the rate of change of the number of swapped of cells with respect to the number of swaps as follows,

$$\frac{dn_s}{dN_s} = \frac{r(N-n_s)}{rN} \quad 3.19$$

Integrating this gives,

$$\int_0^{n_s} \frac{1}{N-n_s} dn_s = \int_0^{N_s} dN_s \quad 3.20$$

the solution of which is,

$$n_s = N(1-\exp(-N_s/N)) \quad 3.21$$

This expression tells us that as we randomise cells we have to perform exponentially more swaps each time to randomise the same number of cells. The theoretical expression for the entropy of a lattice requires the limit of an infinitely large lattice to be taken. This is obviously not possible in practice and so we need to consider the effect of finite lattices and especially finite sampling regions. Wolfram[61] suggests that one could calculate the probability that a truly random (and infinitely large) sample would have the properties of the observed finite sample by evaluating  $\chi^2$ ,

$$\chi^2 = \sum_1^v (p_o - p_e)^2 / p_e \quad 3.22$$

Here  $v$  is the total number of possible different observations (in our case  $k^r$ ) and  $p_o$  and  $p_e$  are the observed and expected frequencies of occurrence of the different observations.

However it would be useful to know how many configurations we would expect to find on a truly random lattice if we sample with a finite sampling region a finite number of



times. Now for a given lattice and a very small sampling region we will obtain all possible configurations even with a very ordered lattice. As we increase the size of the sampling region this must fall until when the sampling region is the same size as the lattice we will obtain only one configuration even for a completely random lattice. We therefore require an expression for theoretical number of configurations we can expect for a given size lattice and sampling region. Now in sampling a random lattice of  $N$  cells for configurations of size  $r$  we are actually randomly sampling the space of all possible size  $r$  configurations which is exactly equivalent to sampling a lattice of partially swapped cells for swapped cells and so equation 3.21 with the appropriate variables is exactly the expression we require. We must replace the size of the lattice  $N$  with the number of possible configurations  $k^r$  and the number of samples  $N_s$  by the size of the lattice in units of the sampling region size  $N/r$ . This give us the following expression for the number of configurations  $n_c$  we will find on a random lattice,

$$n_c = k^r (1 - \exp(-N/rk^r)) \quad 3.23$$

Equation 3.21 tells us the number of swapped cells as a function of the number of swaps or equivalently the number of randomised cells  $n_r$  on the lattice as a function of the number of randomisations  $N_r$  we perform. We can differentiate this to give us the rate of change of the number of randomised cells with respect to the number of randomisations,

$$\frac{dn_r}{dN_r} = \exp(-N_r/N) \quad 3.24$$

We can specify  $N_r$  in terms of  $n_r$  by rearranging equation 3.21 as follows

$$N_r = N \ln(N/(N-n_r)) \quad 3.25$$

Substituting this into equation 3.24 we get an expression for the change in the number of

randomised cells for a small change in the number of randomisations in terms of the number of randomised cells

$$\Delta n_r = \exp(-\ln(N/(N-n_r)))\Delta N_r \quad 3.26$$

However for a given CA lattice we can only calculate the characteristic entropy. We can consider this to be equivalent to partial randomisation of a uniform lattice. Consider a lattice on which every cell has the same state, the entropy of this lattice will be zero. Now if we begin to randomise the states of randomly chosen cells the entropy of the lattice will increase. We require an expression for the number of configurations on the lattice as a function of the number of randomised cells. Consider a lattice on which all cells are in the quiescent state (0). If we now randomise  $n_r$  randomly chosen cells such that each state has an equal probability ( $1/k$ ) of being assigned to a cell then  $(k-1)n_r/k$  cells will have changed from the quiescent state to one of the other possible states on average. If we now take  $N_s = N/r$  samples of  $r$  cells from the lattice the probability of finding  $c$  cells which are non-quiescent will be given by our probability function  $P_c(c)$ . We would therefore expect to find  $N_s P_c(c)$  sample regions with  $c$  non-quiescent cells. Now the number of possible configurations of  $c$  objects in  $r$  bins is,

$$\binom{r}{c}$$

Now each of these objects can be in  $k-1$  different states (or colours), therefore the total number of possible configurations for our sample will be,

$$s_c = (k-1)^c \binom{r}{c}$$

The number of configurations we would expect to see when taking  $N_s P_c(c)$  samples will be given by equation 3.21. The total number of configurations on the lattice will

therefore be the sum of the number for  $c$  non-quiescent cells over all values of  $c$ ,

$$n_c = \sum_0^r s_c (1 - \exp(-N_s P_c(c)/s_c))$$

However this expression cannot be rearranged to obtain the number of randomised cells in terms of the number of configurations. The variation of the number of configurations as a function of the number of randomised cells is shown in Figure 3.2.

As a first approximation we can assume that the number of configurations increases linearly with the number of randomised cells. Thus the number of configurations is given by the ratio of the number of randomised cells to the total number of cells times the theoretical number of configurations we would expect on a completely random lattice with a sampling region of size  $r$  (equation 3.23),

$$n_c = \frac{n_r}{N} k^r (1 - \exp(-N/rk^r)) + 1 \quad 3.27$$

The addition of a one is to account for the limiting condition of no randomised cells ( $n_r = 0$ ) in which case we will have only one configuration present on the lattice, the null configuration.

Substituting this into equation 3.17 and rearranging for  $n_r$  we obtain,

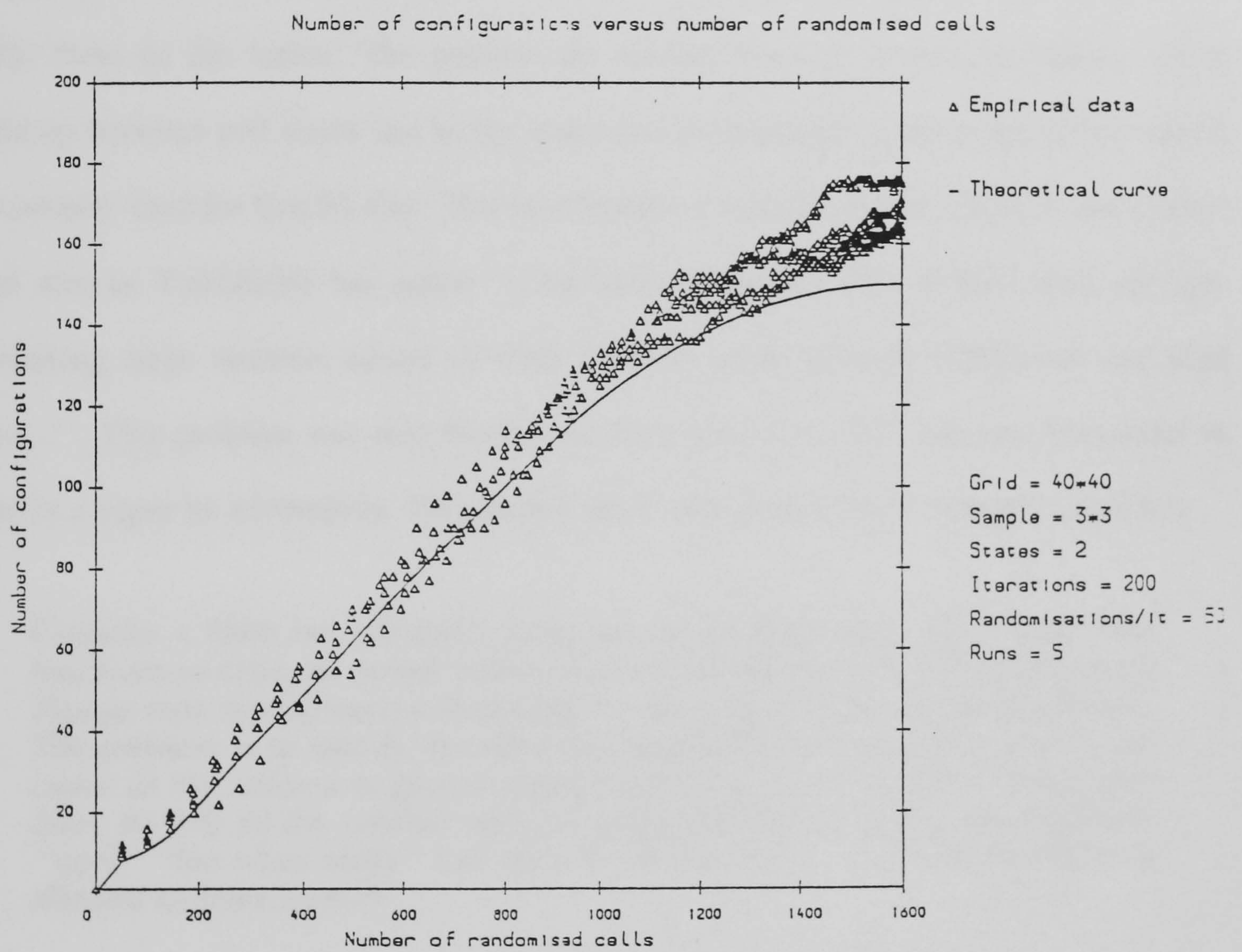
$$n_r = \frac{N(k^{S_c r} - 1)}{k^r (1 - \exp(-N/rk^r))} \quad 3.28$$

Where  $S_c$  is the characteristic entropy of the CA lattice.

If we now randomise  $\Delta n_r$  more cells on the lattice (equation 3.26 tells us how many randomisations we have to do to achieve this) the new noise added entropy  $S_n$  is given by,



Figure 3.2: Number of configurations as a function of the number of randomised cells.





$$S_n = 1/r \log_k \left[ \frac{n_r + \Delta n_r}{N} k^r (1 - \exp(-N/rk^r)) + 1 \right] \quad 3.29$$

Thus the noise which we have added by randomising  $\Delta n_r$  cells can be quantified by the increase in entropy,

$$\Delta S = S_n - S_c \quad 3.30$$

By mixing the CA lattice, using randomly chosen neighbourhood cells or adding weighted noise to the CA lattice we will tend to destroy the patterns which would normally form on the lattice. The patterns are formed because of the correlations which build up between cell states due to the synchronous evolution of the entire lattice which is normally used for CA[60, 63]. This synchronous evolution is very artificial and unphysical and as Toffoli[64] has noted “...the implementation of a global clock for synchronising large iterative arrays of cells involves great physical difficulties and high costs...”. This problem was first devised by John Myhill in 1957 and was first stated in print in a paper by Moore[65]. He called it the *Firing Squad Synchronisation problem*,

Consider a finite but arbitrarily long one-dimensional array of  $N$  finite state machines or cells, all called *soldiers* and one at the end called the *general*, all change state synchronously depending on the state of their nearest neighbours. The problem is to specify the states and transitions such that the *general* can cause all the *soldiers* to go into a particular state, the *firing state*, at the same time. At  $t=0$  all the *soldiers* are in a quiescent state and the *general* simply “says” “fire when ready” and takes no further action. Note that no *soldier* is allowed to count up to  $N$ .

The problem was first solved by John McCarthy and Marvin Minsky. Since then Waksman[66], and Balzer[67] have obtained optimal solutions. Moore and Langdon[68] considered the problem where the *general* is allowed to be any machine in the linear array. Varshausky[69] and Varshausky *et al*[70] consider cells having random pair-wise interactions and non-uniform speed of response and communications delays respectively.

Herman[71, 72] has generalised the problem to cells which cannot distinguish right from left (i.e totalistic) and arrays which grow during the process of reaching the *firing state*. Further generalisations by Rosensthiel[73] and Rosensthiel *et al*[74] have allowed the cells to be arbitrarily connected and they obtain synchronisation in  $2N$ . Grasselli[75] has considered the problem in two-dimensions and shows that for a special class of arrays which he calls *information lossless* the problem essentially reduces to the one-dimensional case. More recently Romani[76] has shown that much faster solutions can be obtained by utilising properties of the network structure. Vichniac[36] notes that synchronous updating of the lattice can lead to a “feedback catastrophe” when modelling Ising spin systems. Essentially if any region of the lattice takes on a checkerboard pattern it acts as a *forcing structure*[77, 78] which spreads across the entire lattice. The action of the forcing structure relies on cells in both states switching simultaneously and is thus an artificial property of the synchronisation of the lattice.

In the standard description of a CA each cell may store only one state and cannot remember any of its past states (although we have included this characteristic in our general model). Also when a given cell evolves its state must change instantaneously since intermediate states are not defined. Thus in order for the CA to evolve correctly the instantaneous change of state must occur at the same time for every cell. If this were not the case some cells would see the future state of cells in their neighbourhood rather than their current state. The synchronisation of the state changes can be achieved in several ways. Each cell may have its own internal clock which all run at the same rate or there could be a global clock with signal lines to every cell (in this case the individual lines must all have the same effective length). Which ever method is used any noise or error in the individual clocks or the individual lines from the global clock will cause the evolution of the cells to gradually become out of synchronisation. So this type of



synchronisation is inherently unstable, but this problem can be removed by a slight alteration in the CA and the way it evolves. Firstly the cells have a memory of their past state and secondly the CA has a two phase global clock. The CA now evolves as follows, in the first phase of the clock each cell transfers its current state into its past state memory, then in the second phase each cell changes its state using the past state memories of the cells in its neighbourhood. Providing the phases of the clock are much greater than the characteristic noise or error in the time when each cell receives the clock pulse then the CA will evolve synchronously.

If we now consider a CA which evolves asynchronously (ACA), this may be achieved in two ways, by random iteration or independent clocks[79]. With random iteration cells have a certain probability of evolving which they do one at a time. The other method is for each cell to have its own internal clock each having a different period with the periods of the clocks having a normal distribution about some mean for example. In terms of our general model the random iteration method is more appropriate as it fits naturally with the rule probability. We may now require that the ACA evolve in an equivalent manner to the synchronous CA (SCA) such that we can obtain a global state of the ACA which is identical to the corresponding global state of the SCA. Now as the ACA and SCA evolve from the same initial state the evolutions will steadily diverge because some cells on the ACA will have been evolved more than others and in terms of the SCA the ACA cells will be using cell states from different time steps in determining their next state. However if we allow the cells of the ACA to have an infinite amount of memory so that they can store all their past states. A given cell can then interrogate the memories of its neighbourhood cells corresponding to its own "age" and we would be able to recover a global state from the ACA equivalent to that of the SCA at a given point in its evolution. In practice for a finite lattice there will be a finite range of ages and so

the ACA will only need a memory large enough to store states younger than the eldest possible cell on the lattice. We should also note that under the assumption that the CA maintains a uniform distribution of cell states each cell will “see” the correct distribution regardless of its age and we would therefore expect the ACA to evolve similarly to the SCA in terms of the state populations.

We may now ask what will be the distribution of ages on the lattice for an ACA.

Let  $n_\zeta$  be the number of cells which have evolved  $\zeta$  times. Then in an analogous way to equation 3.24 we can define the rate of change of the number of  $\zeta$ -evolved cells with respect to the number of evolutions which have occurred  $N_e$  as follows,

$$\frac{\Delta n_\zeta}{\Delta N_e} = \frac{\bar{c}_{\zeta-1} - \bar{c}_\zeta}{r} \quad 3.31$$

where  $\bar{c}_\zeta$  is the mean number of  $\zeta$ -evolved cells which would be found in a sample of size  $r$ . This simply expresses the gain and loss of  $\zeta$ -evolved cells when the  $r$  cells are evolved. Any  $\zeta$ -evolved cells in the region will become  $\zeta+1$  evolved and no longer contribute to the count but  $\zeta-1$  evolved cells will become  $\zeta$ -evolved and thus now contribute to the count. Now for a normal distribution the mean number of  $\zeta$ -evolved cells will be  $rn_\zeta/N$ , substituting this into equation 3.31 and taking the limit of small sample sizes we obtain

$$\frac{dn_\zeta}{dN_e} = \frac{n_{\zeta-1} - n_\zeta}{N} \quad 3.32$$

We may solve this by first considering  $\zeta=0$  and since there are no  $\zeta=-1$  cells then we have a simple exponential decay expression,



$$n_0 = N \exp(-N_e/N) \quad 3.33$$

We can now substitute this into equation 3.32 with  $\zeta=1$  and solve this and so on for increasing values of  $\zeta$ . The general solution for equation 3.32 is as follows,

$$n_\zeta = \frac{N_e^\zeta}{\zeta! N^{\zeta-1}} \exp(-N_e/N) \quad 3.34$$

This expression tells us the variation in the number of  $\zeta$ -evolved cells as a function of the number of random evolutions which have occurred. The distribution of  $n_\zeta$  with  $N_e$  and  $\zeta$  is shown in Figure 3.3.

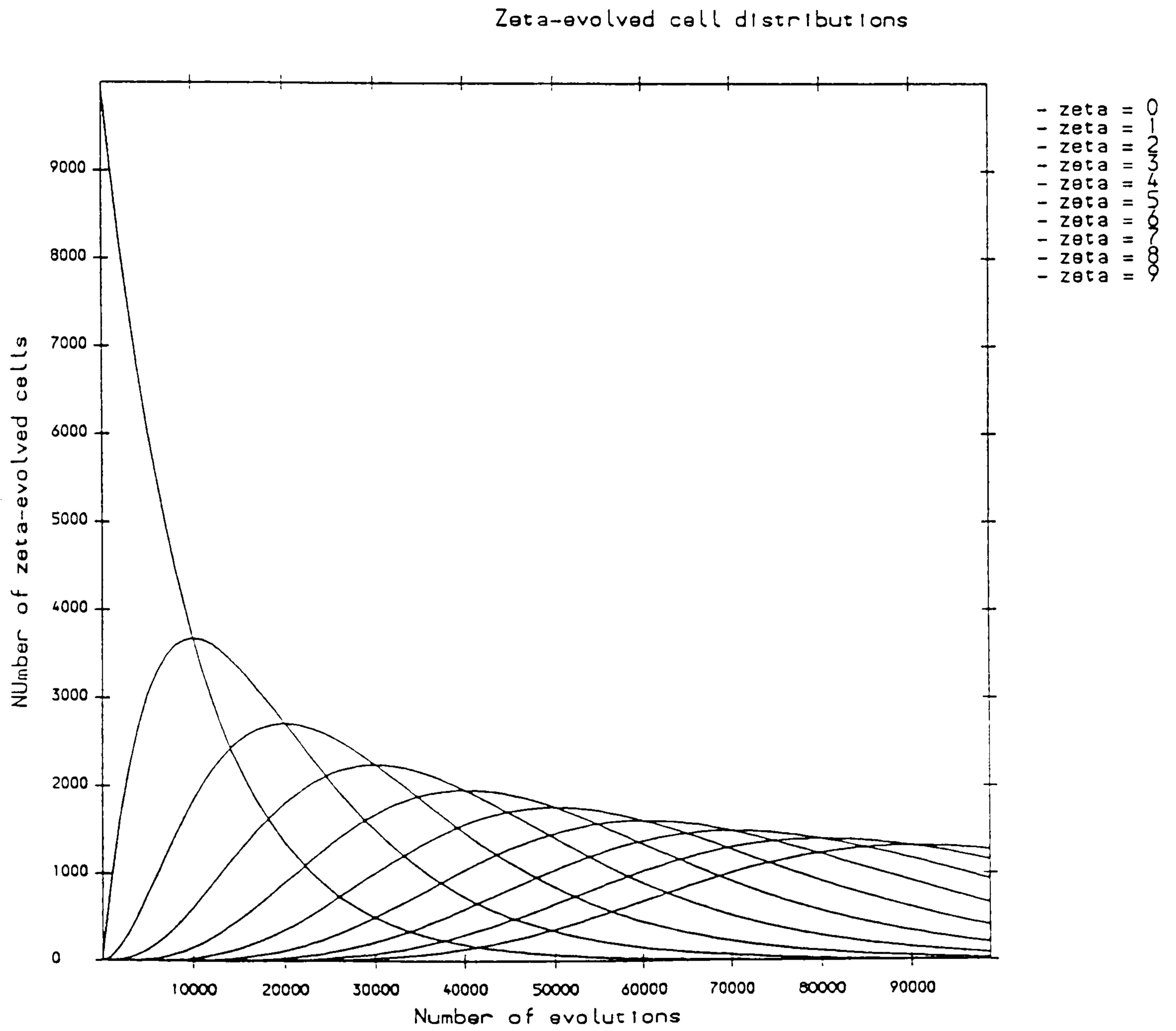
This distribution is in fact the Poisson distribution function and therefore the process of randomly evolving cells on the lattice is a Poisson process[53, 80]. A Poisson process can be defined as follows,

- (i) Events occurring in non-overlapping intervals of time are independent of each other.
- (ii) There is a constant  $\lambda$  such that the probabilities of occurrence of events in a small interval of time  $\Delta t$  are given as follows,
  - (a)  $P \{ \text{number of events in } (t, t+\Delta t] = 0 \} = 1 - \lambda \Delta t + O(\Delta t)$
  - (b)  $P \{ \text{number of events in } (t, t+\Delta t] = 1 \} = \lambda \Delta t + O(\Delta t)$
  - (c)  $P \{ \text{number of events in } (t, t+\Delta t] > 1 \} = O(\Delta t)$

where  $O(\Delta t)/\Delta t \rightarrow 0$  as  $\Delta t \rightarrow 0$ .

Under this definition the Poisson distribution may be stated thus,

Figure 3.3: Distribution of  $\zeta$ -evolved cells.



$$P [X(t)=k] = e^{-\lambda t} (\lambda t)^k / k! \quad 3.35$$

$\lambda$  is the mean number of events per unit time. An event in terms of our CA lattice is the evolution of a given cell. The mean number of evolutions a cell will undergo for every evolution of a random cell on the lattice (that is identifying  $N_e$  with  $t$ ) is  $1/N$ . Alternatively a cell will evolve once on average for every  $N$  evolutions of the lattice. Thus after  $N$  evolutions of the lattice a  $\zeta-1$  evolved cell will be  $\zeta$  evolved and the new probability of finding a  $\zeta$  evolved cell will be equal to the previous probability of finding a  $\zeta-1$  evolved cell. We therefore obtain,

$$\frac{dP_\zeta}{dN_e} = \frac{P_{\zeta-1} - P_\zeta}{N} \quad 3.36$$

Solving this in a similar fashion to equation 3.34 leads to the Poisson distribution (equation 3.35).

In order to determine the amount of memory required by the ACA in order to simulate an SCA we need the variation in the number of cells which have evolved  $\zeta$  or more times. The derivation follows the same path as before, we first obtain the rate of change of  $n_{\geq\zeta}$ ,

$$\frac{dn_{\geq\zeta}}{dN_e} = \frac{n_{\zeta-1}}{N} \quad 3.37$$

Solving these for progressively larger values of  $\zeta$  we obtain the general solution,

$$n_{>\zeta} = \frac{1}{\zeta! N^\zeta} \int_0^{N_e} N_e^\zeta \exp(-N_e/N) dN_e \quad 3.38$$

The integral cannot be solved exactly in general but approximate solutions exist[81, 82] and we may write the general solution as follows,

$$n_{>\zeta} = N \left[ 1 - \exp(-N_e/N) \sum_{i=0}^{\zeta} \frac{N_e^i}{i! N^i} \right] \quad 3.39$$

In order to see how the range of ages of the cells on the lattice changes as we continue to evolve the lattice we need the rate of change of  $n_{>\zeta}$  with respect to  $N_e$ . This is simply given by Equation 3.37,

$$\frac{\delta n_{>\zeta}}{\delta N_e} = \frac{N_e^\zeta}{\zeta! N^\zeta} \exp(-N_e/N) \quad 3.40$$

This is the Poisson distribution which has a maximum at,

$$\zeta = \log_e(N_e/N)$$

We can see from this that the youngest cells are not the fastest evolving cells and so will tend to be left behind. The range of ages on the lattice will continually increase and so an ACA would need an infinite amount of memory to simulate a SCA exactly for an infinite amount of time. However the range of ages and thus the amount of memory required decreases as the evolutionary time (or number of evolutions of the lattice) is decreased and so for finite times simulation of a SCA with an ACA is possible.

The system of dynamical equations we obtained which model the evolution of the numbers of cells in the various states,

$$n_i^{(t+1)} = n_i^{(t)} + \left[ \sum_j P_{ji} n_j^{(t)} \right] - \left[ \sum_j P_{ij} \right] n_i^{(t)} \quad 3.41$$

may be converted to differential equations by specifying the change in the numbers of cells in each state in a time period  $\Delta t$ ,

$$n_i^{(t+\Delta t)} = n_i^{(t)} + \left[ \left[ \sum_j P_{ji} n_j^{(t)} \right] - \left[ \sum_j P_{ij} \right] n_i^{(t)} \right] \Delta t \quad 3.42$$



We then take the limit as  $\Delta t$  becomes very small giving the differential equation,

$$\frac{dn_i}{dt} = \left[ \sum_j P_{ji} n_j^{(t)} \right] - \left[ \sum_j P_{ij} \right] n_i^{(t)} \quad 3.43$$

Note that our original dynamical system 3.3 expresses the behaviour of the SCA more exactly since the SCA rules give the transition probabilities  $P_{ij}$  and if there are  $n_i$  cells in state  $i$  then  $P_{ij} n_i$  will change to state  $j$  on average on the SCA lattice in a single evolutionary step. However we can reduce the number of cells which change from state  $i$  to state  $j$  by giving each rule a probability of acting  $p$  as in our general model 3.2. Now the number of cells changing state  $\Delta n_i$  for equation 3.42 is,

$$\Delta n_i = \left[ \left[ \sum_j P_{ji} n_j^{(t)} \right] - \left[ \sum_j P_{ij} \right] n_i^{(t)} \right] \Delta t \quad 3.44$$

and the number of cells changing state for a SCA with rules which have a probability  $p$  of acting is,

$$\Delta n_i = \left[ \sum_j p P_{ji} n_j^{(t)} \right] - \left[ \sum_j p P_{ij} \right] n_i^{(t)} \quad 3.45$$

So  $\Delta t$  and  $p$  are exactly equivalent. By reducing the size of the time step  $\Delta t$  in our finite difference equation 3.42 we can make it model the behaviour of the equivalent differential equation arbitrarily closely. Similarly by reducing the rule action probability  $p$  we can make the behaviour of the CA approach that of the equivalent dynamical system arbitrarily closely. An SCA with probabilistic rules will behave in an identical manner to an ACA in which the cells have a probability  $p$  of evolving. For both of these cases because the cells are not evolving synchronously correlations between cell states will not occur as readily and the lattice will tend to retain its random distribution of states.

## Summary

We have re-introduced a statistical mechanical approach to CA which has appeared in various forms in the literature previously but has not been extensively developed. We have shown that this leads us to a system of dynamical equations which model the changes in the state populations as the CA evolves, and that the accuracy of this simulation can be quantified. We have looked in detail at entropy of the CA lattice and various factors which effect it, and the relationship between synchronous and asynchronous evolution and deterministic and probabilistic evolution. This leads us to several variations on the standard SCA. Mixing, random neighbourhoods, weighted noise and probabilistic evolution which all have the effect of maintaining a random distribution of states and preventing the characteristic patterns of the SCA from developing. In conclusion we have shown that a discrete, probabilistic, totalistic CA can be viewed as equivalent to a continuous, deterministic dynamical system.

#### 4. Experimental Results

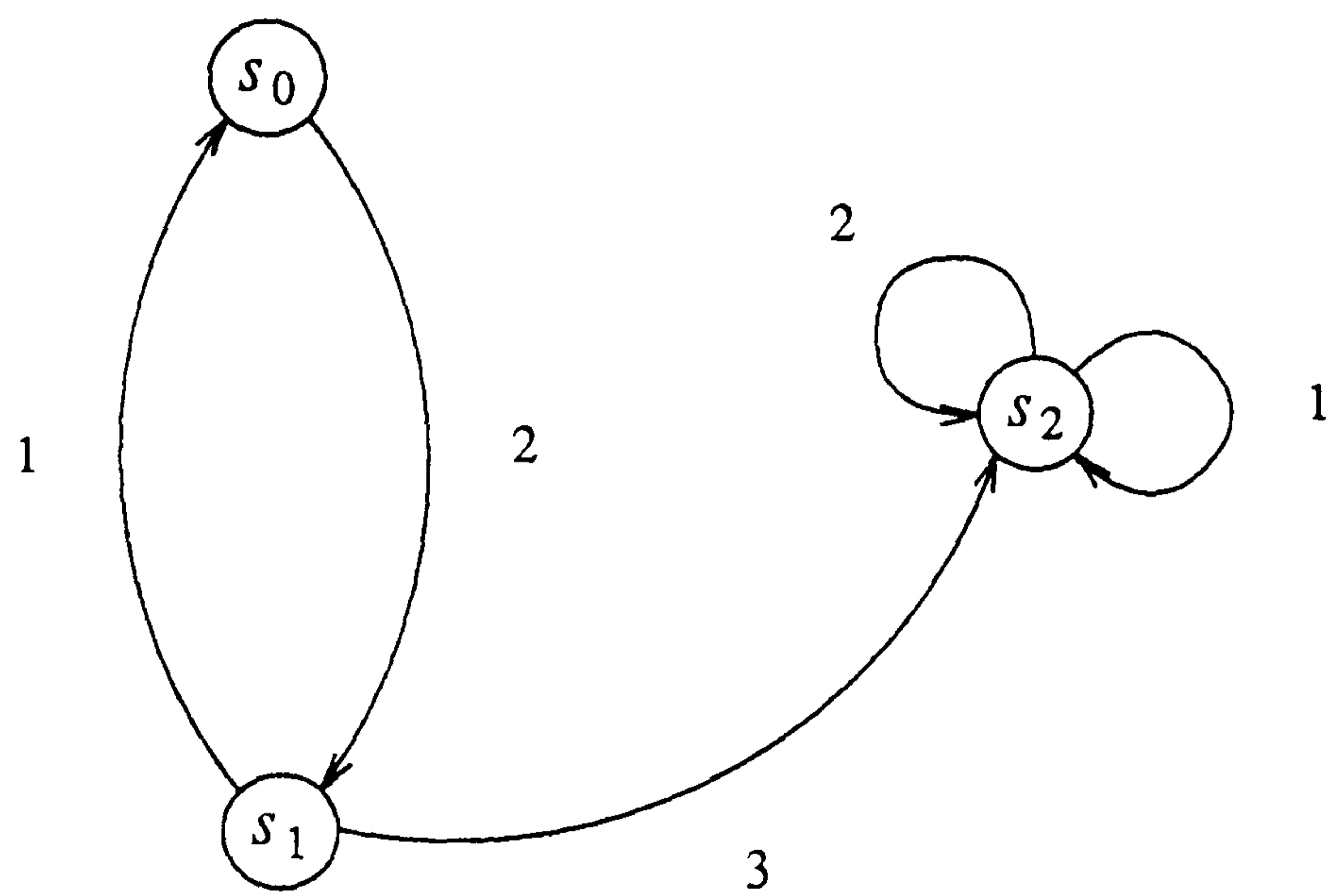
In order to study the relationship between a CA and its equivalent dynamical system (DS) we will use the phase space representation for the evolution of the systems[83, 84]. With this method of representation the state of a  $d$ -variable system is represented by a point in a  $d$ -dimensional space such that each variable corresponds to a dimension of the space. Its value is then represented by the position of the point in that dimension of the space. For a finite system (or indeed for an unbounded system with an appropriate transformation of the phase space[85]) a finite phase space diagram can represent any possible state of the system. The evolution of a system will be a series of points or a curve through the space depending on whether the system is discrete or continuous. These curves are known as trajectories, phase paths or orbits.

The phase space is especially useful with regard to identification of the limiting behaviour of a system. If we calculate the evolutionary trajectories of many initial points spread across the entire phase space for times long enough for the limiting behaviour to be approached. Then for multiple limit set systems, not only will the limit sets be clear but also the domains of attraction of the various limit sets.

Many authors have noted that bounded two variable systems always approach a limit point[83, 86]. We can see this if we represent the system by a directed graph in an analogous way to the representation of finite-state machines in automata theory.

Figure 4.1 is an example of a directed graph, here the *nodes* correspond to the states of the machine and the *directed arcs* correspond to the possible transitions between states. The *directed arcs* are labelled with the *input symbol* which generates that transition. In terms of our CA the nodes will represent states of the CA, the directed arcs will represent

Figure 4.1: Example of directed graph.





the state transitions defined by the rules for the CA and each will be labelled with the conditions which give rise to the transition.

Thus we can represent all the possible different 2D systems as shown in Figure 4.2. These systems contain all possible variations of the four basic transitions possible. Any more complex transition structure can be simplified to one of these with an appropriate adjustment of the transition conditions. All these systems will reach a limit point, the reason for this is that for a CA the total number of cells is a constant and so  $n_1 = N - n_0$ . We therefore effectively have only a single independent variable. The rate of change of  $n_0$  will therefore be a function of  $n_0$  only. When  $\dot{n}_0$  is positive  $n_0$  will increase until either  $\dot{n}_0 = 0$  or  $n_0$  reaches its maximum value of  $N$ , similarly if  $\dot{n}_0$  is negative then  $n_0$  will decrease until  $\dot{n}_0 = 0$  or  $n_0$  reaches its minimum of 0. Figure 4.3 depicts this graphically. Note the possibility of the unstable equilibrium points (B) and (D), such that if  $n_0$  has exactly the right value for  $\dot{n}_0$  to be zero then the system will remain in that state but if  $n_0$  is perturbed by an infinitesimal amount then the system will evolve away from the equilibrium point. Also the system can never evolve to these points so they may only exist as initial states of the system. Moore[4] has termed these ‘‘Garden-of-Eden’’ states in the context of CA.

Thus the simplest bounded system which can exhibit non-trivial limit sets is a three variable DS which corresponds to a three state CA ( $k=3$ ). It is this type of system which we will mainly study as it is sufficiently complex while remaining computationally tractable.

Now the total number of possible transitions is simply the product of the number of different pairs of states, the number of possible transition directions (i.e. two) and the number of states, since each transition can depend on any state. The number of pairs of states is simply the number of combinations of two states chosen from the  $k$  possible

Figure 4.2: Two variable systems.

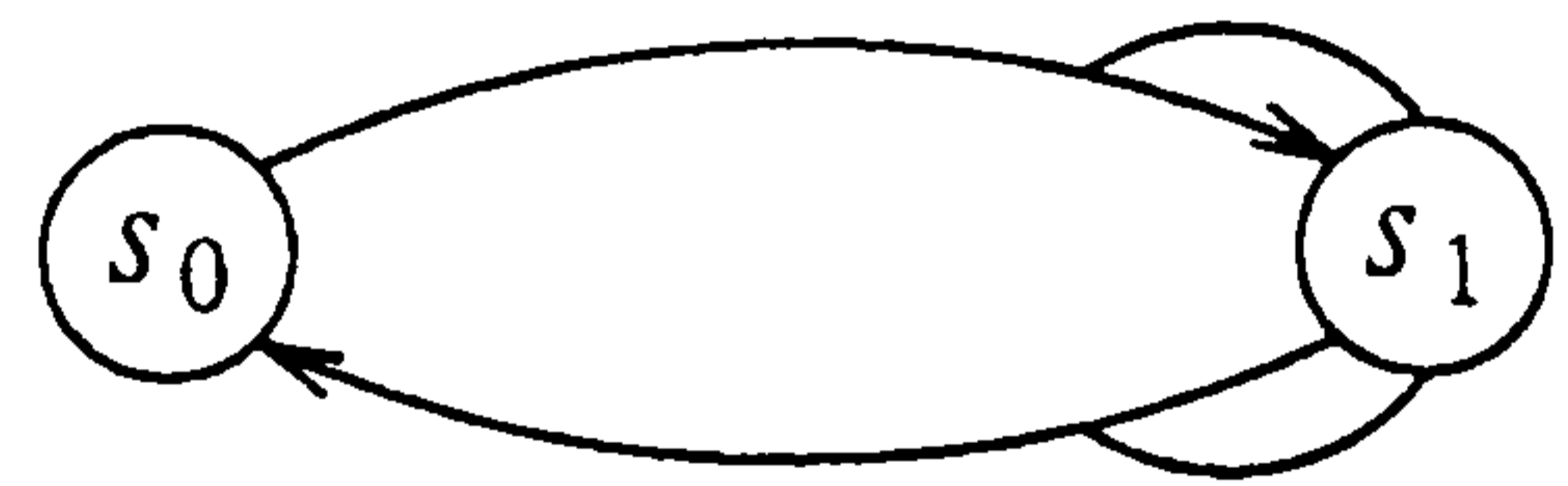
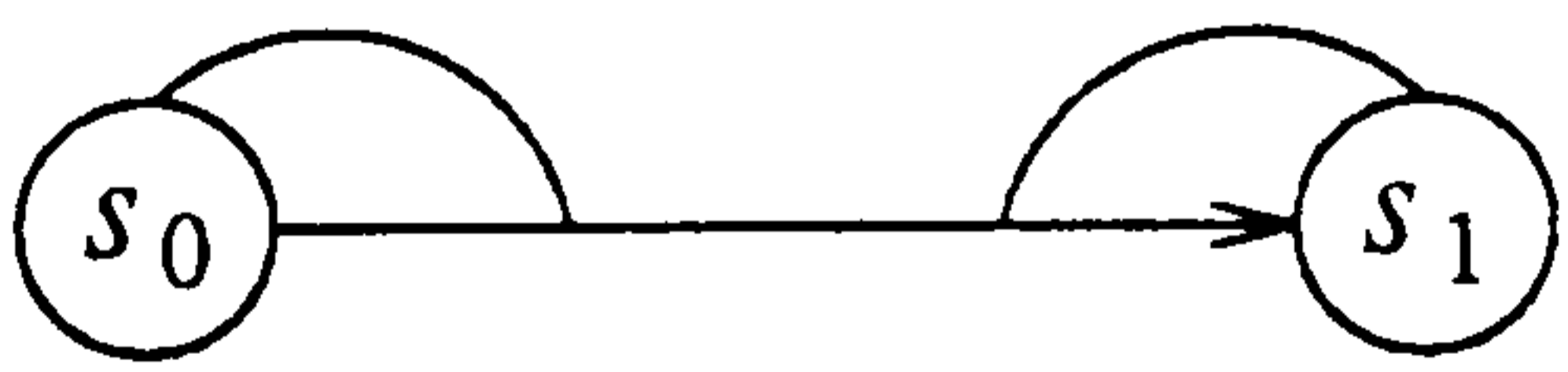
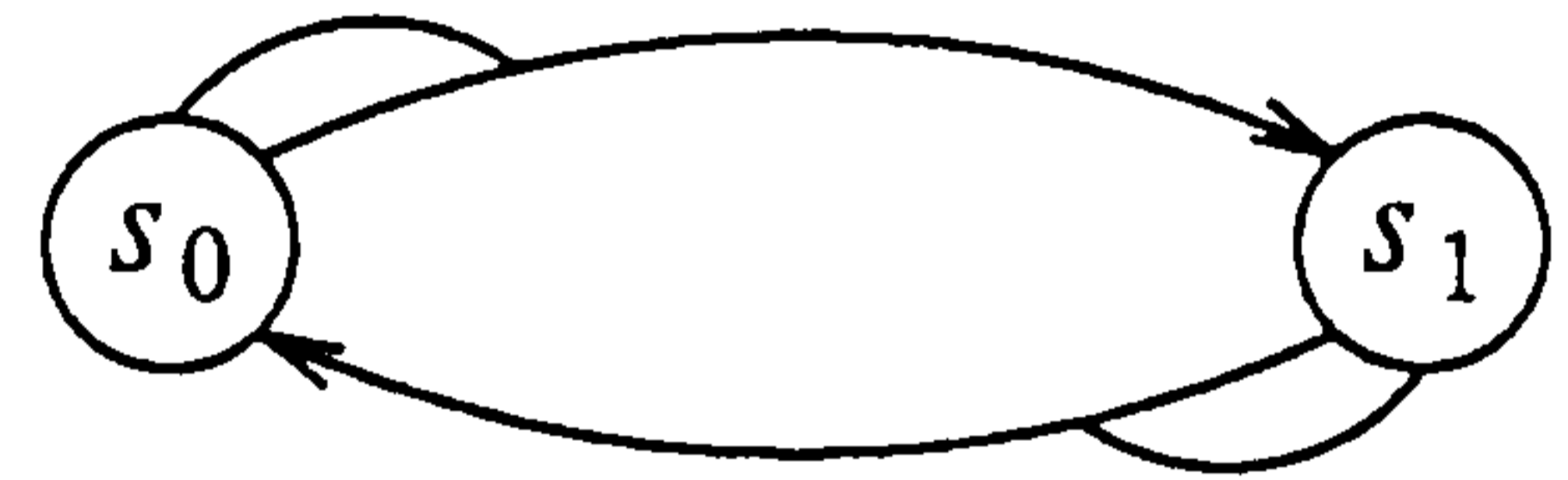
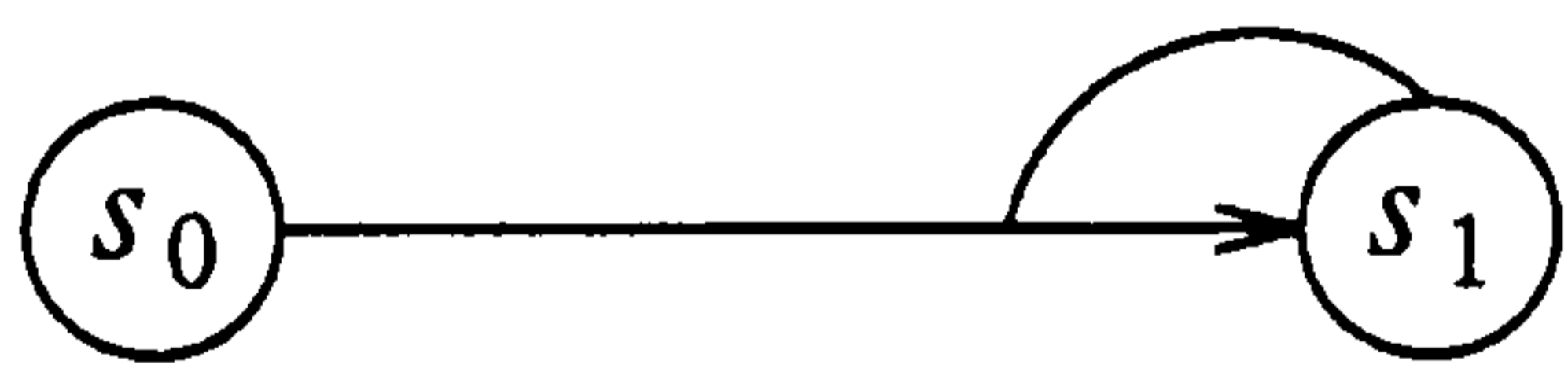
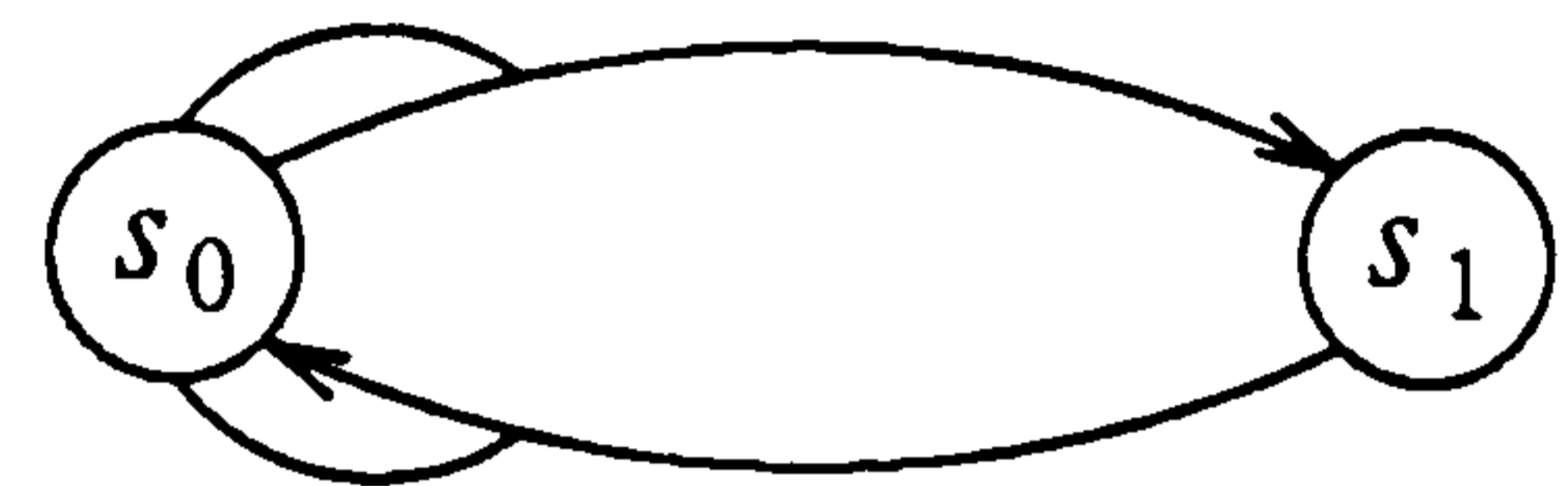
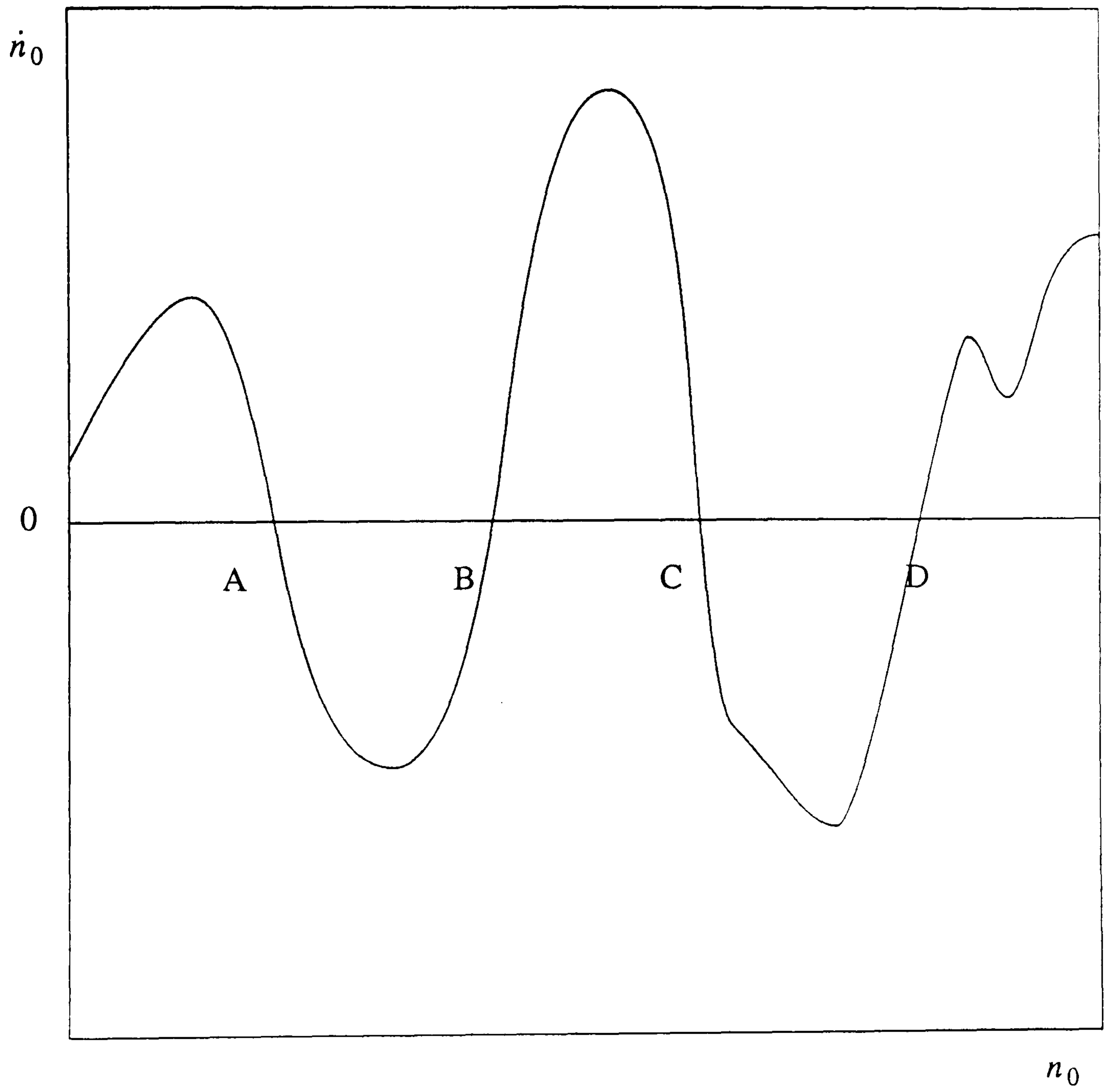


Figure 4.3: Example behaviour of  $\dot{n}_0$ .



states which is given by,

$$\frac{k!}{2!(k-2)!} = 1/2k(k-1) \quad 4.1$$

Thus the maximum number of different transitions is,

$$k^2(k-1) \quad 4.2$$

So for  $k=3$  there are 18 possible transitions, however we will in general restrict our attention to systems with a maximum of two transitions in opposite directions between any two states.

The coverage of the phase space was chosen so as to demonstrate the global behaviour of the system without making the phase space too crowded with trajectories. Each initial point on the phase space represents an initial number of each of the three state populations. This is equivalent to a state density on the lattice of  $n_i/N$ . This state density was used as a probability of assigning that state to each cell on the CA lattice.

Probabilities were simulated throughout using a non-linear additive feedback random number generator employing a table of 31 long integers to return pseudo-random numbers in the range 0 to  $(2^{31})-1$  available as a standard library routine on the SUN UNIX system used. The period of the random number generator was approximately  $16(2^{31}-1)$ [87].

In this way we obtained a random initial state for the lattice whose position on the phase diagram would be approximately that specified by the coverage. The lattice was then evolved for a time long enough for the limiting behaviour to be approached, typically this was less than a hundred iterations. At each step the state populations were saved so that the trajectories from each initial point could be plotted later.



An identical coverage was used for the DS, the state population variables being initialised to the appropriate values. The DS was then iterated for the same length of time as the CA and the state population variables stored at each time step.

Mixing of the CA lattice was simulated by choosing the neighbourhood cells at random from the whole lattice. As already noted in Chapter 3 this is equivalent to mixing the lattice by swapping the states of randomly chosen cells until all cells on average have been swapped.

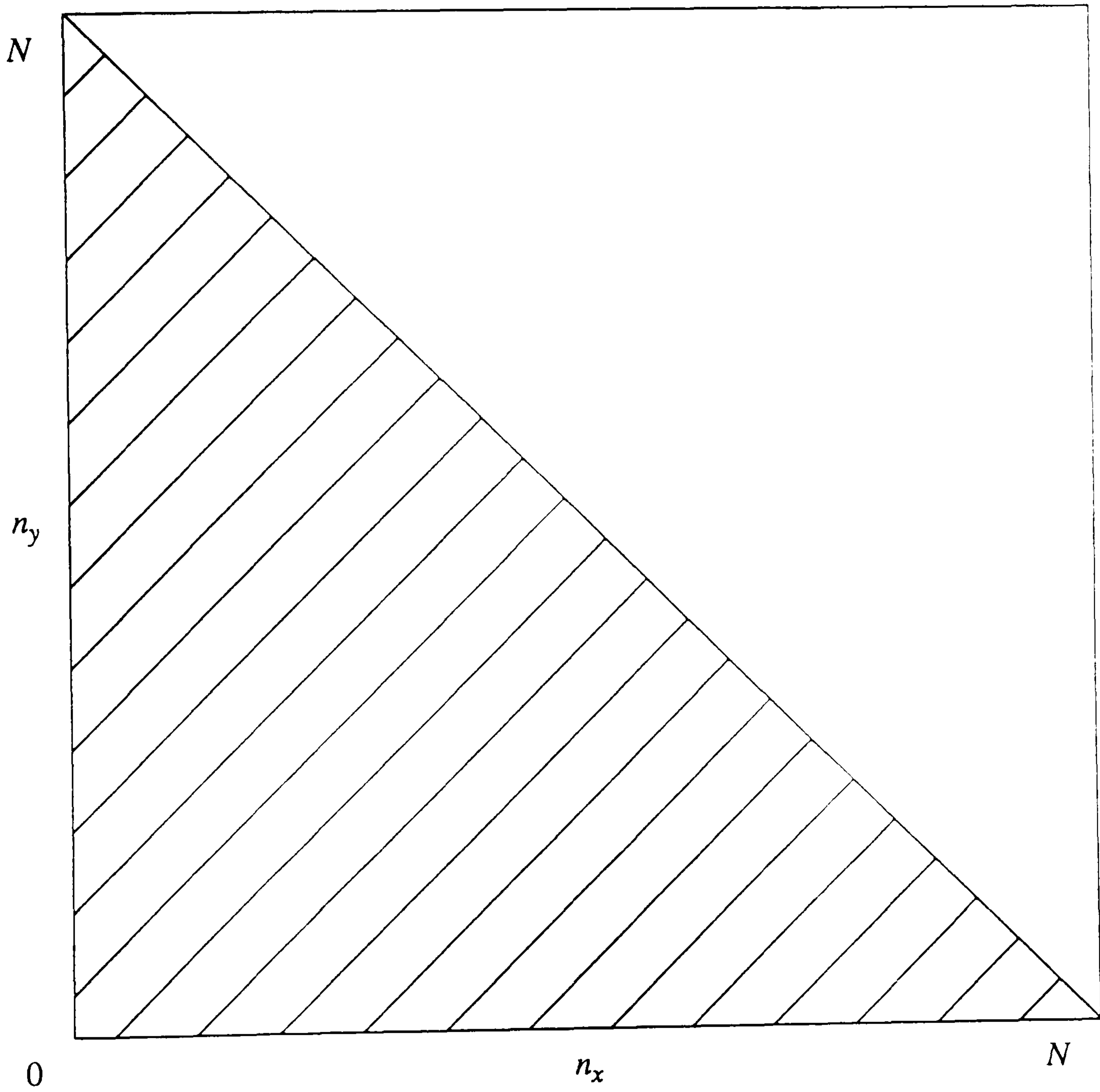
Now for a three variable system the phase space will be three-dimensional. However a two-dimensional projection of a three-dimensional phase space would not be sufficiently clear to be useful and so one must plot two-dimensional slices of the full phase space. However any given trajectory will not in general remain on a planar slice of the three-dimensional phase space and so any two-dimensional slice cannot be a complete description of the state of the system. This problem does not occur for our systems however because the systems are bounded and so have only two independent variables. Thus at any point in a 2D slice of the phase space the other variable is defined by  $N - n_x - n_y$  and thus this phase space slice is a complete representation of the state of the system. The bounded nature of the systems also has another effect in that all points on the 2D slice are not possible states of the system. This is because  $n_x + n_y \leq N$  which means the shaded region in Figure 4.4 is the only valid region of the phase space.

Firstly we will consider *threshold* systems which form a subset of the set of the CA defined by our general totalistic scheme. A threshold rule will be written as follows,

$$s_i \& \geq T s_T \rightarrow s_o \quad 4.3$$

such that the transition  $s_i$  to  $s_o$  occurs if  $T$  or more cells in the neighbourhood are in state

Figure 4.4: Valid region of phase plane for three variable CA.



$s_T$ . Threshold rules are interesting because the behaviour of the transition probability is simpler than that for a general totalistic rule. This is because the transition probability increases monotonically with the density of the threshold state  $s_T$ . Thus the point at which a given transition rate exceeds some other transition rate will occur at a single threshold state density whereas for a general totalistic rule there may be many such totalistic state densities depending on the complexity of the rule set. In order to understand this we need to study the transition probability function in more detail.

The hypergeometric transition probability function, the related binomial function and the normal distribution approximation were introduced in Chapter 3. These functions are plotted for typical values of  $N$  and  $c$  in Figure 4.5. The hypergeometric function is plotted again together with the difference between it and the binomial and normal functions in Figure 4.6. The binomial function is almost identical to the hypergeometric function for all parameter values the error always being less than 0.1%. The normal distribution approximation is not such a good approximation but the error is mostly less than 5% only exceeding this for extremal values of population size and neighbourhood cell count.

However there are two parameter regions where the errors become very large due to the normal approximation failing completely. This occurs when the cell count is either 0 or  $r$  and the population size is either very small or very close to the maximum  $N$ . The transition probabilities given by the normal approximation are greater than one and in fact tend to infinity as the population size tends to 0 or  $N$ . The conditions which lead to these errors are rarely encountered in CA as rules which specify cell counts of either 0 or  $r$  are not normally useful or interesting. Thus by avoiding these rules this problem can be eliminated. Furthermore the advantages of the normal approximation far outway these slight disadvantages and so we will work entirely with the normal approximation.

Figure 4.5: Hypergeometric, Binomial and Normal functions.

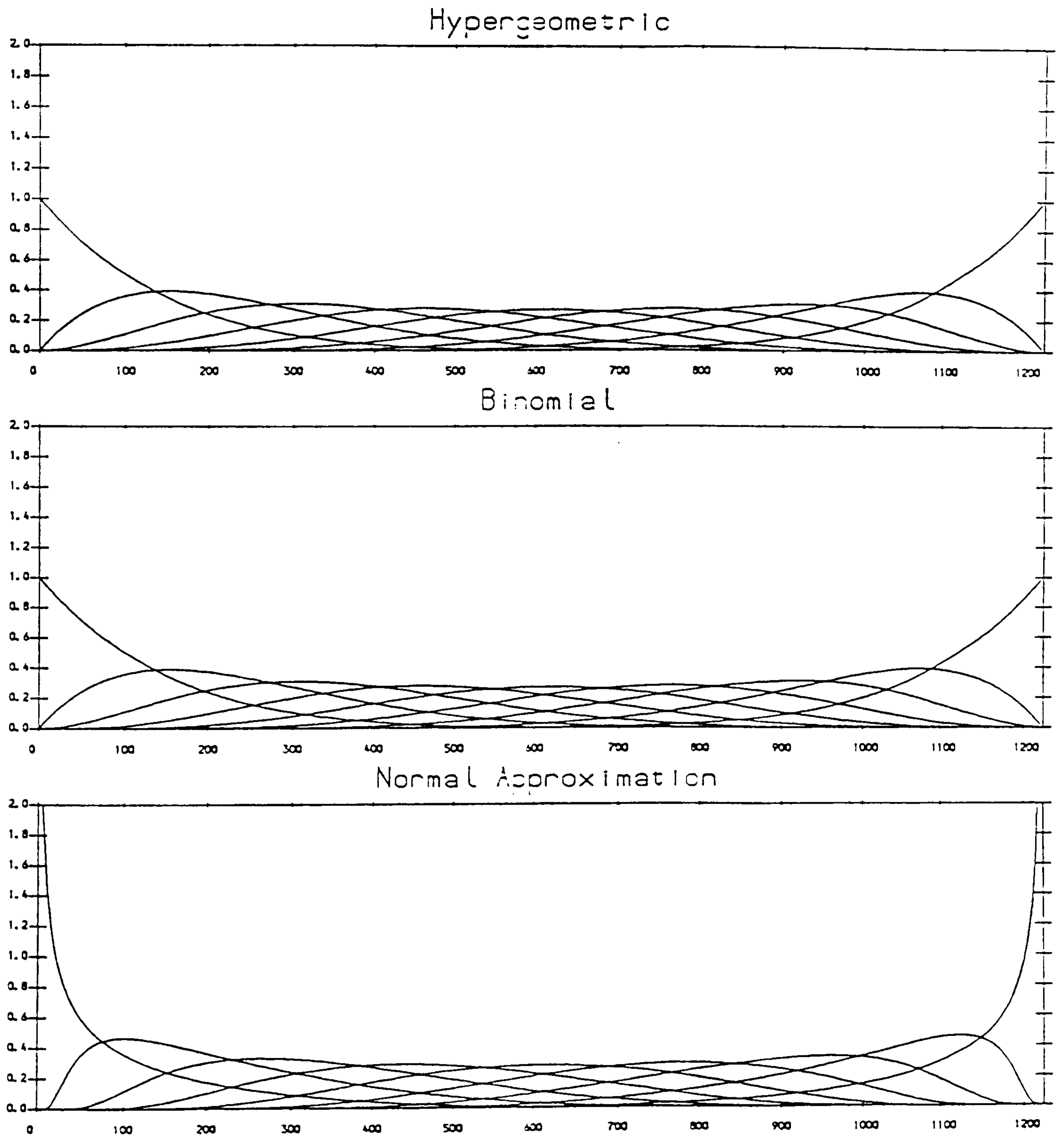
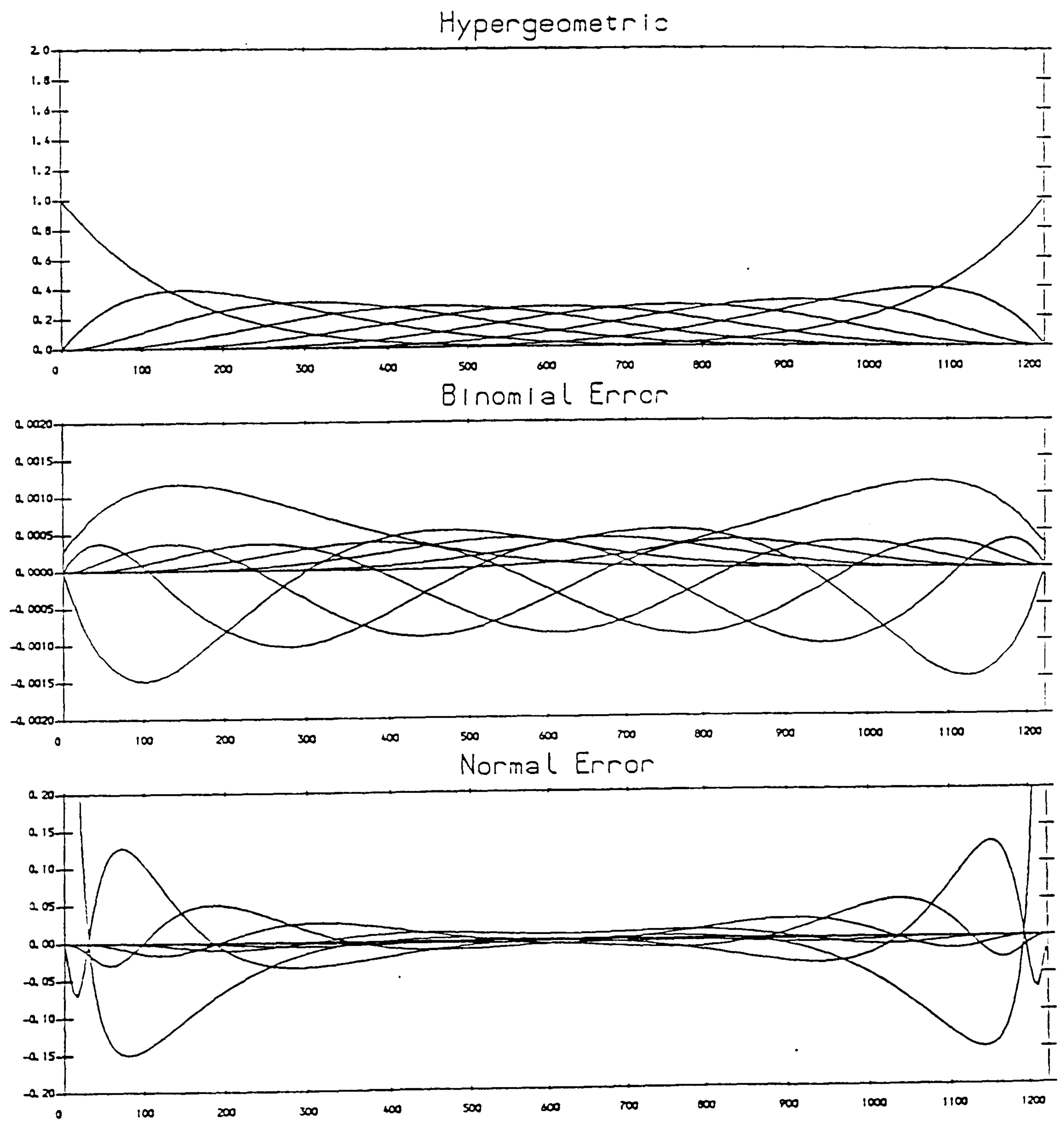




Figure 4.6: Hypergeometric function and Binomial and Normal errors.



If we now consider the variation of the transition probability with the dependent population and the count of cells in the neighbourhood we obtain the curves shown in Figure 4.7.

Thus the transition probability for a totalistic CA, where the transition depends on several cell counts in the neighbourhood, may have many maxima. However if we now consider the variation of the threshold transition probability which is the summation of the transition probabilities for cell counts greater than or equal to the threshold (up to the size of the neighbourhood). We obtain the curves shown in Figure 4.8.

Here we see that the transition probability monotonically increases with the dependent population. Therefore the transition probability for a transition which depends on many thresholds will still be a monotonically increasing function of the dependent population.

The threshold systems which we will study are summarised in Figure 4.9, their state transition diagrams in Figure 4.10, vector field diagrams in Figure 4.11 and phase diagrams in Figure 4.12.

System 1 and 2 are trivial in that states change to themselves and so any initial CA lattice will remain fixed for all time. These are used to show that the equivalent DS will also be immediately stable which is confirmed by consideration of the DS equations.

System 3 is effectively a two state system as the population of state  $s_2$  remains fixed. We thus have a bounded two variable system in terms of the DS and the system should reach a limit point for all initial states. Furthermore, since the probability of a cell changing to a given state depends on the population of that state, then the state which initially increases will continue to increase until the other state has disappeared. Both these points are confirmed by the phase diagrams of the system shown in Figure 4.12 and the

Figure 4.7: Variation of transition probability with dependent population.

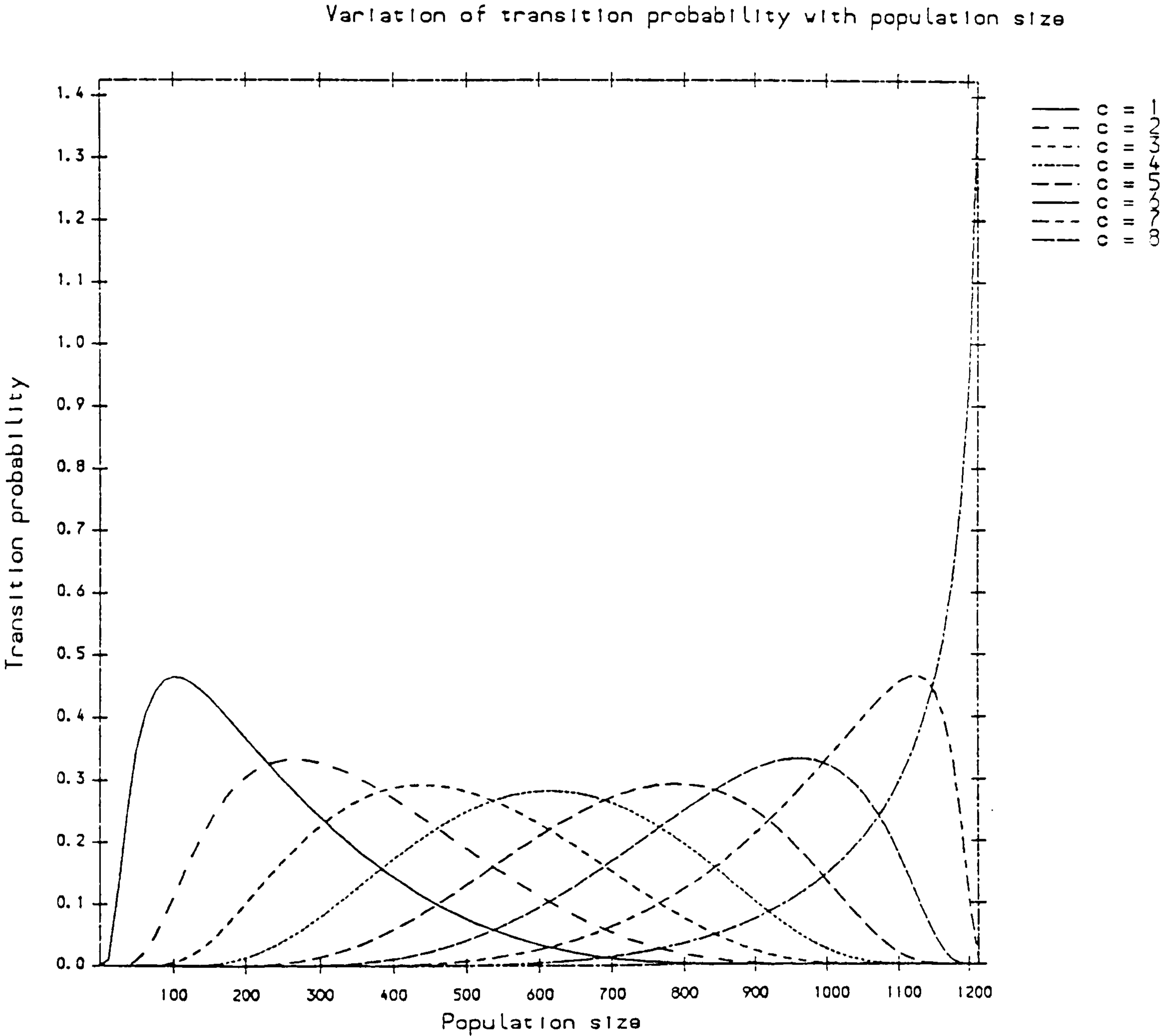






Figure 4.9: Threshold systems summary (a).

**System 1**

$$s_0 \ \& \ \geq 2s_0 \rightarrow s_0$$

$$s_1 \ \& \ \geq 4s_1 \rightarrow s_1$$

$$s_2 \ \& \ \geq 6s_2 \rightarrow s_2$$

$$\dot{n}_0 = P_{0,0}(2,n_0)n_0 - P_{0,0}(2,n_0)n_0$$

$$\dot{n}_1 = P_{1,1}(4,n_1)n_1 - P_{1,1}(4,n_1)n_1$$

$$\dot{n}_2 = P_{2,2}(6,n_2)n_2 - P_{2,2}(6,n_2)n_2$$

**System 2**

$$s_0 \ \& \ \geq 2s_0 \rightarrow s_0$$

$$s_1 \ \& \ \geq 4s_2 \rightarrow s_1$$

$$s_2 \ \& \ \geq 6s_1 \rightarrow s_2$$

$$\dot{n}_0 = P_{0,0}(2,n_0)n_0 - P_{0,0}(2,n_0)n_0$$

$$\dot{n}_1 = P_{1,1}(4,n_2)n_1 - P_{1,1}(4,n_2)n_1$$

$$\dot{n}_2 = P_{2,2}(6,n_1)n_2 - P_{2,2}(6,n_1)n_2$$

**System 3**

$$s_0 \ \& \ \geq 4s_1 \rightarrow s_1$$

$$s_1 \ \& \ \geq 6s_0 \rightarrow s_0$$

$$s_2 \ \& \ \geq 2s_2 \rightarrow s_2$$

$$\dot{n}_0 = P_{1,0}(6,n_0)n_1 - P_{0,1}(4,n_1)n_0$$

$$\dot{n}_1 = P_{0,1}(4,n_1)n_0 - P_{1,0}(6,n_0)n_1$$

$$\dot{n}_2 = P_{2,2}(2,n_2)n_2 - P_{2,2}(2,n_2)n_2$$

Figure 4.9: Threshold systems summary (b).

**System 4**

$$s_0 \ \& \ \geq 3s_2 \rightarrow s_1$$

$$s_1 \ \& \ \geq 5s_0 \rightarrow s_2$$

$$s_2 \ \& \ \geq 1s_1 \rightarrow s_0$$

$$\dot{n}_0 = P_{2,0}(1, n_1)n_2 - P_{0,1}(3, n_2)n_0$$

$$\dot{n}_1 = P_{0,1}(3, n_2)n_0 - P_{1,2}(5, n_0)n_1$$

$$\dot{n}_2 = P_{1,2}(5, n_0)n_1 - P_{2,0}(1, n_1)n_2$$

**System 5**

$$s_0 \ \& \ \geq 5s_0 \rightarrow s_1$$

$$s_1 \ \& \ \geq 6s_1 \rightarrow s_2$$

$$s_2 \ \& \ \geq 7s_2 \rightarrow s_0$$

$$\dot{n}_0 = P_{2,0}(7, n_2)n_2 - P_{0,1}(5, n_0)n_0$$

$$\dot{n}_1 = P_{0,1}(5, n_0)n_0 - P_{1,2}(6, n_1)n_1$$

$$\dot{n}_2 = P_{1,2}(6, n_1)n_1 - P_{2,0}(7, n_2)n_2$$

**System 6**

$$s_0 \ \& \ \geq 3s_0 \rightarrow s_1$$

$$s_1 \ \& \ \geq 4s_1 \rightarrow s_2$$

$$s_2 \ \& \ \geq 5s_2 \rightarrow s_0$$

$$\dot{n}_1 = P_{2,0}(5, n_2)n_2 - P_{0,1}(3, n_0)n_0$$

$$\dot{n}_1 = P_{0,1}(3, n_0)n_0 - P_{1,2}(4, n_1)n_1$$

$$\dot{n}_2 = P_{1,2}(4, n_1)n_1 - P_{2,0}(5, n_1)n_2$$

Figure 4.9: Threshold systems summary (c).

**System 7**

$$s_0 \ \& \ \geq 2s_0 \rightarrow s_1$$

$$s_1 \ \& \ \geq 3s_1 \rightarrow s_2$$

$$s_2 \ \& \ \geq 4s_2 \rightarrow s_0$$

$$\dot{n}_0 = P_{2,0}(4, n_2)n_2 - P_{0,1}(2, n_0)n_0$$

$$\dot{n}_1 = P_{0,1}(2, n_0)n_0 - P_{1,2}(3, n_1)n_1$$

$$\dot{n}_2 = P_{1,2}(3, n_1)n_1 - P_{2,0}(4, n_2)n_2$$

**System 8**

$$s_0 \ \& \ \geq 4s_0 \rightarrow s_2$$

$$s_0 \ \& \ \geq 6s_0 \rightarrow s_1$$

$$s_1 \ \& \ \geq 6s_1 \rightarrow s_0$$

$$s_2 \ \& \ \geq 4s_2 \rightarrow s_1$$

$$\dot{n}_0 = P_{1,0}(6, n_1)n_1 - P_{0,2}(4, n_0)n_0$$

$$\dot{n}_1 = P_{0,1}(6, n_0)n_0 + P_{2,1}(4, n_2)n_2 - P_{1,0}(6, n_1)n_1$$

$$\dot{n}_2 = P_{0,2}(4, n_0)n_0 - P_{0,2}(6, n_0)n_0 - P_{2,1}(4, n_2)n_2$$

**System 9**

$$s_0 \ \& \ \geq 3s_0 \rightarrow s_2$$

$$s_0 \ \& \ \geq 4s_0 \rightarrow s_1$$

$$s_1 \ \& \ \geq 4s_1 \rightarrow s_0$$

$$s_2 \ \& \ \geq 3s_2 \rightarrow s_1$$

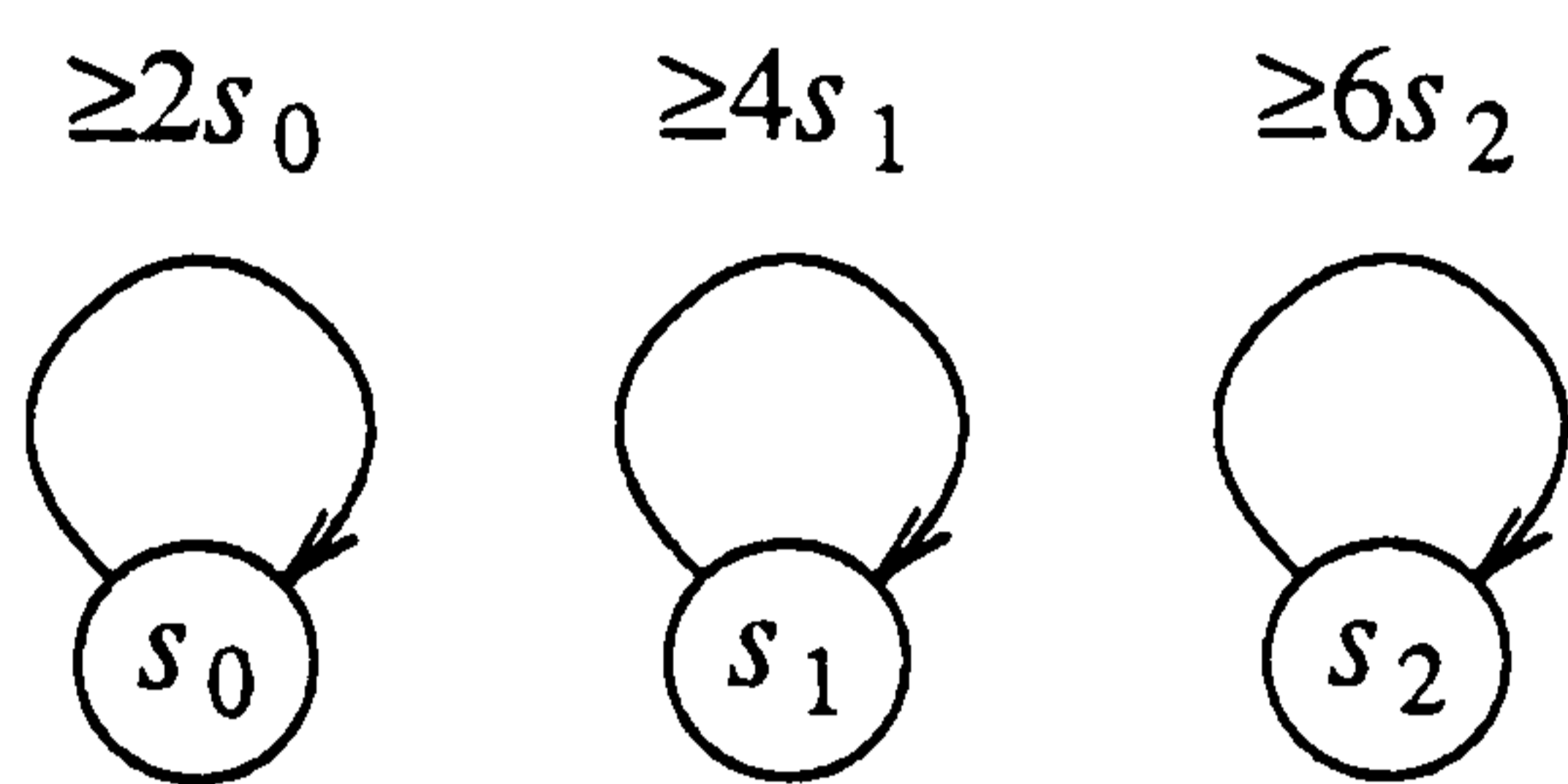
$$\dot{n}_0 = P_{1,0}(4, n_1)n_1 - P_{0,2}(3, n_0)n_0$$

$$\dot{n}_1 = P_{0,1}(4, n_0)n_0 + P_{2,1}(3, n_2)n_2 - P_{1,0}(4, n_1)n_1$$

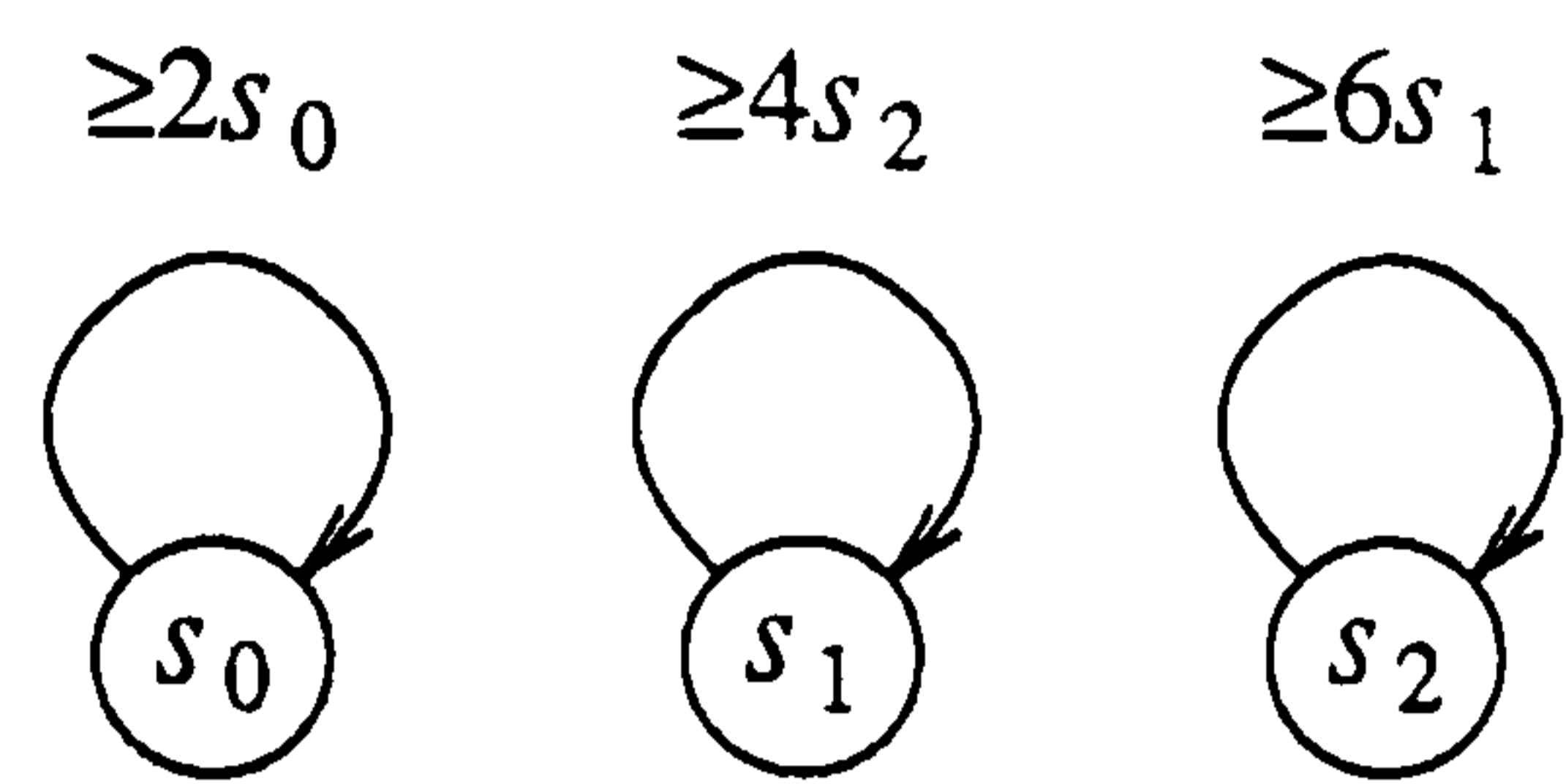
$$\dot{n}_2 = P_{0,2}(3, n_0)n_0 - P_{0,2}(4, n_0)n_0 - P_{2,1}(3, n_2)n_2$$

Figure 4.10: Threshold systems state transition diagrams (a).

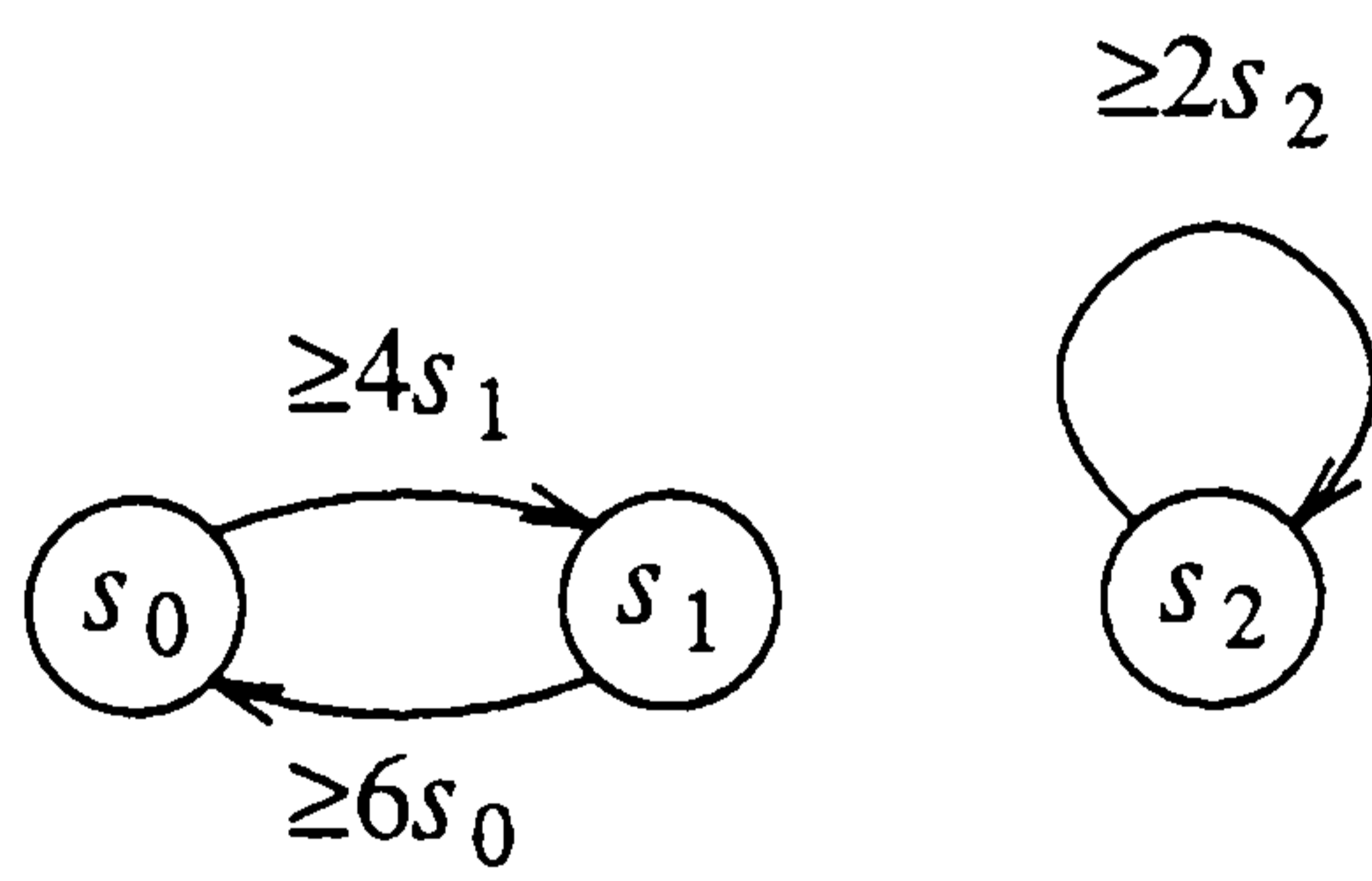
System 1



System 2



System 3



System 4

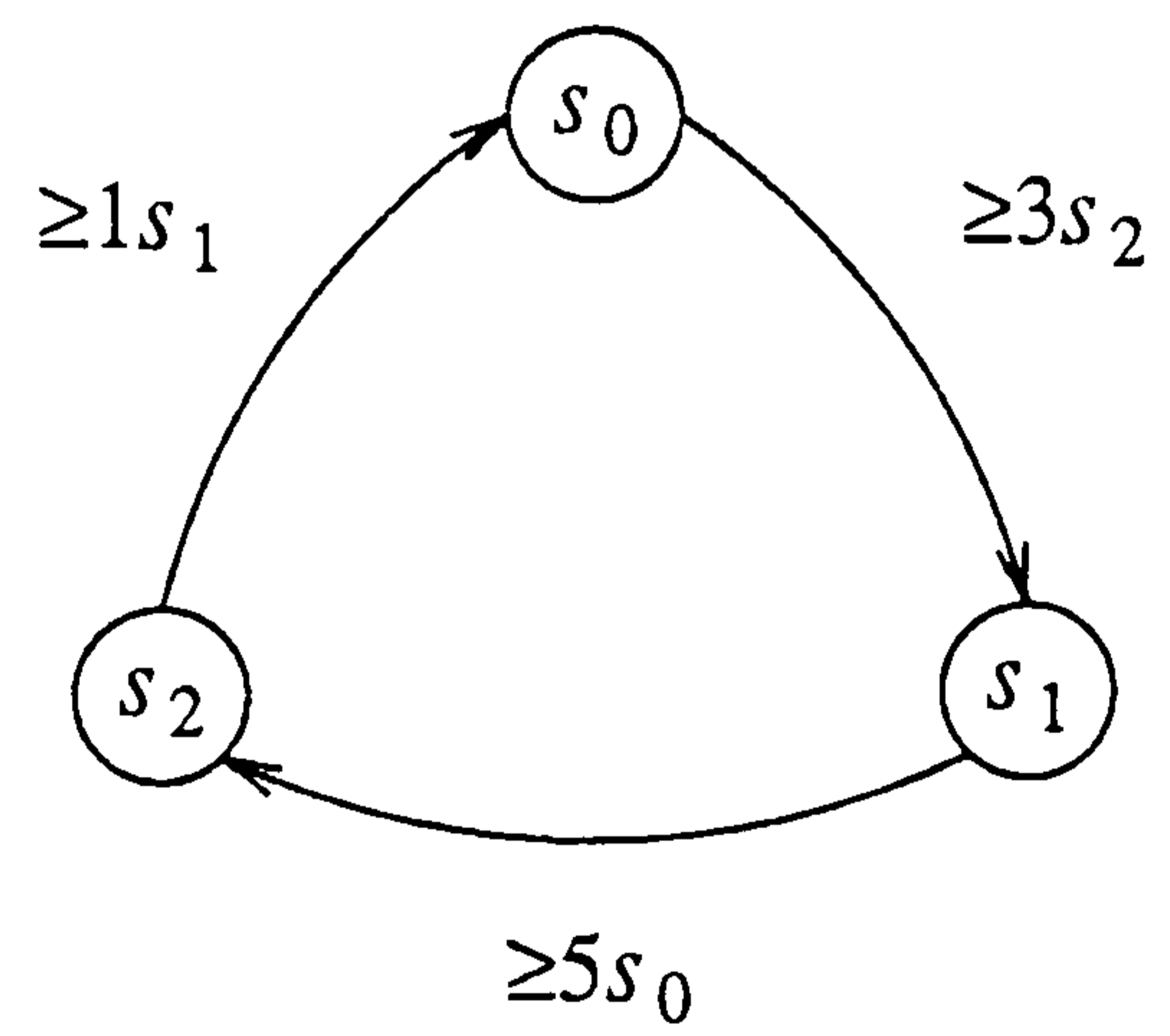
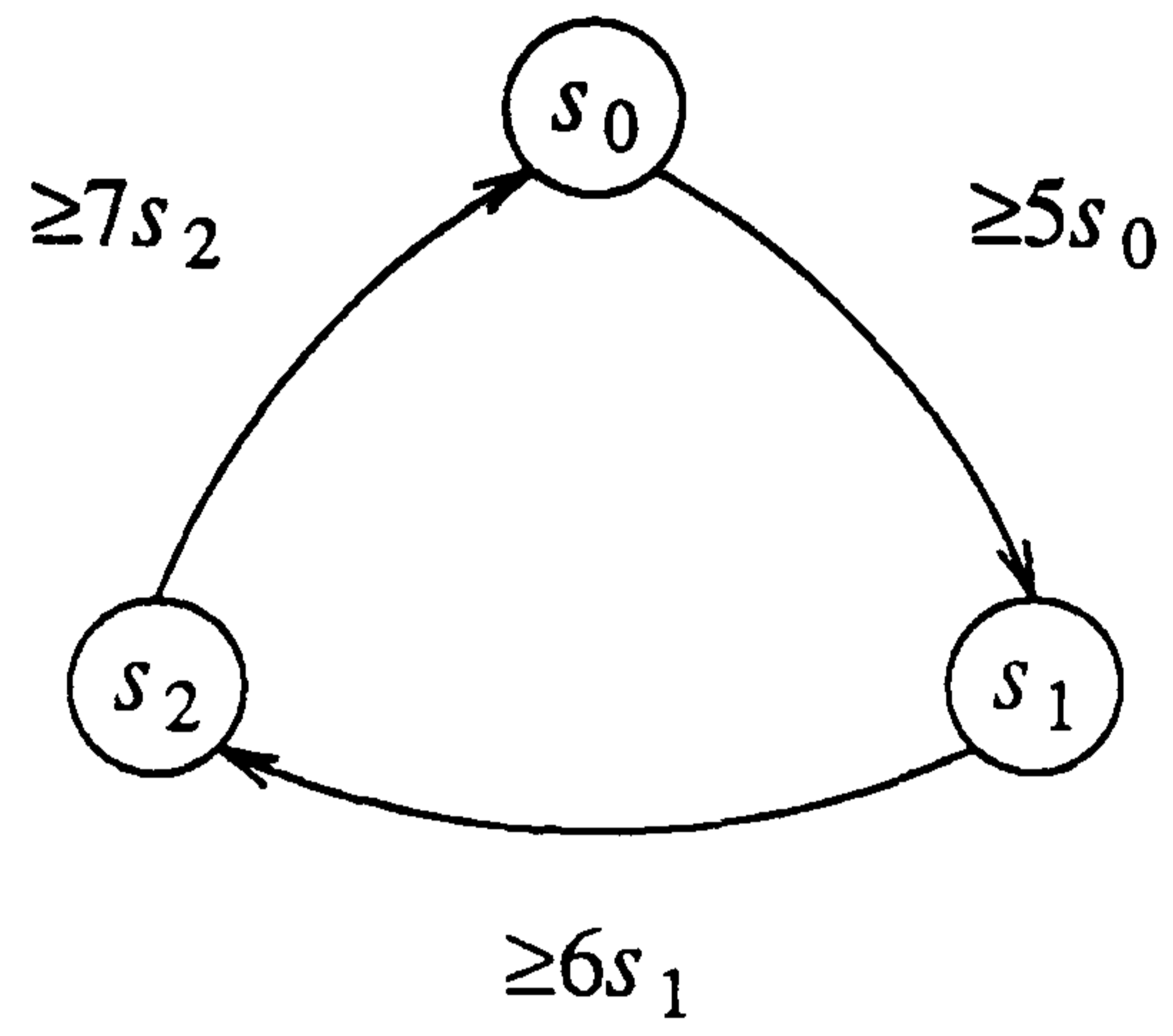


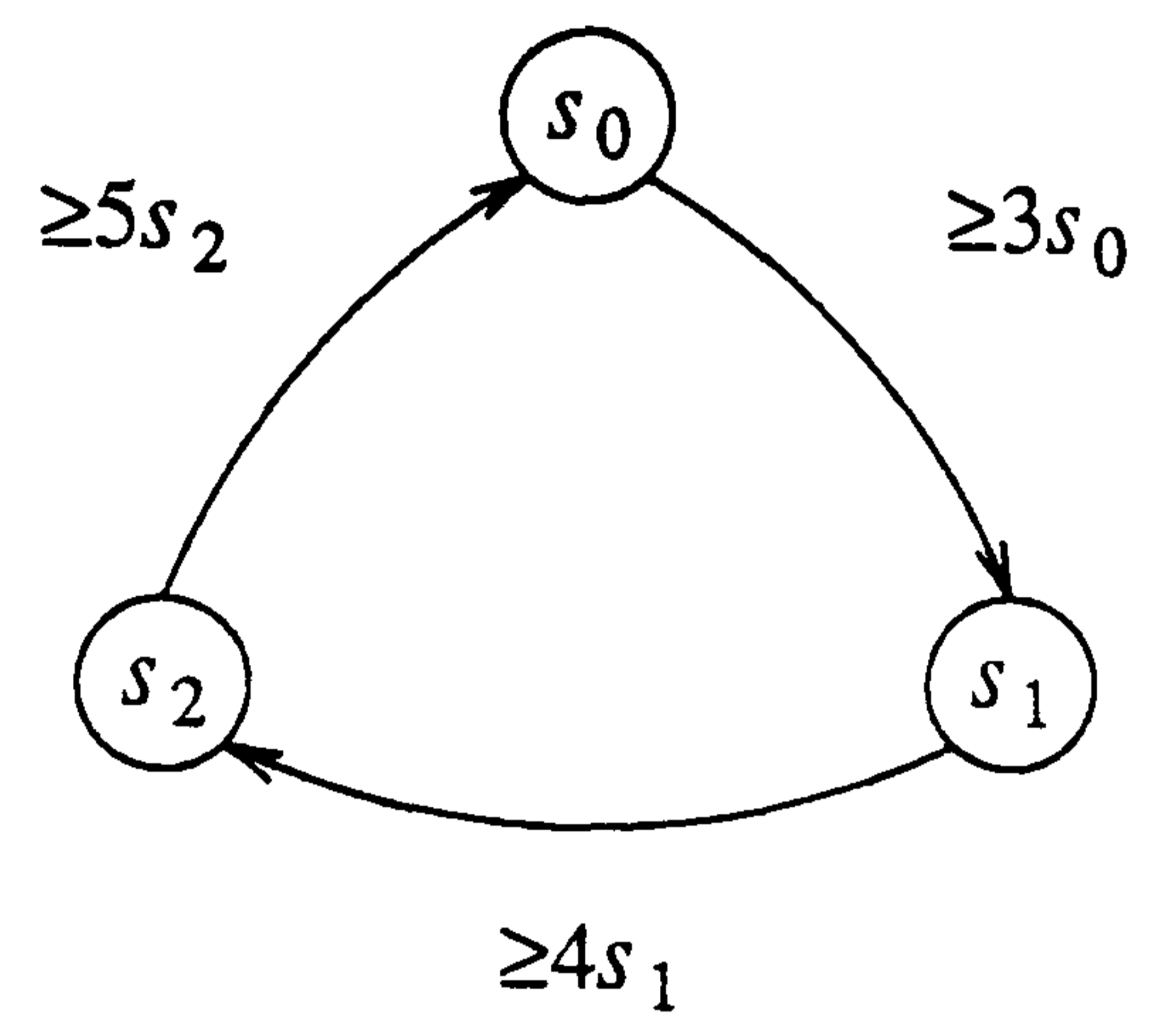


Figure 4.10: Threshold systems state transition diagrams (b).

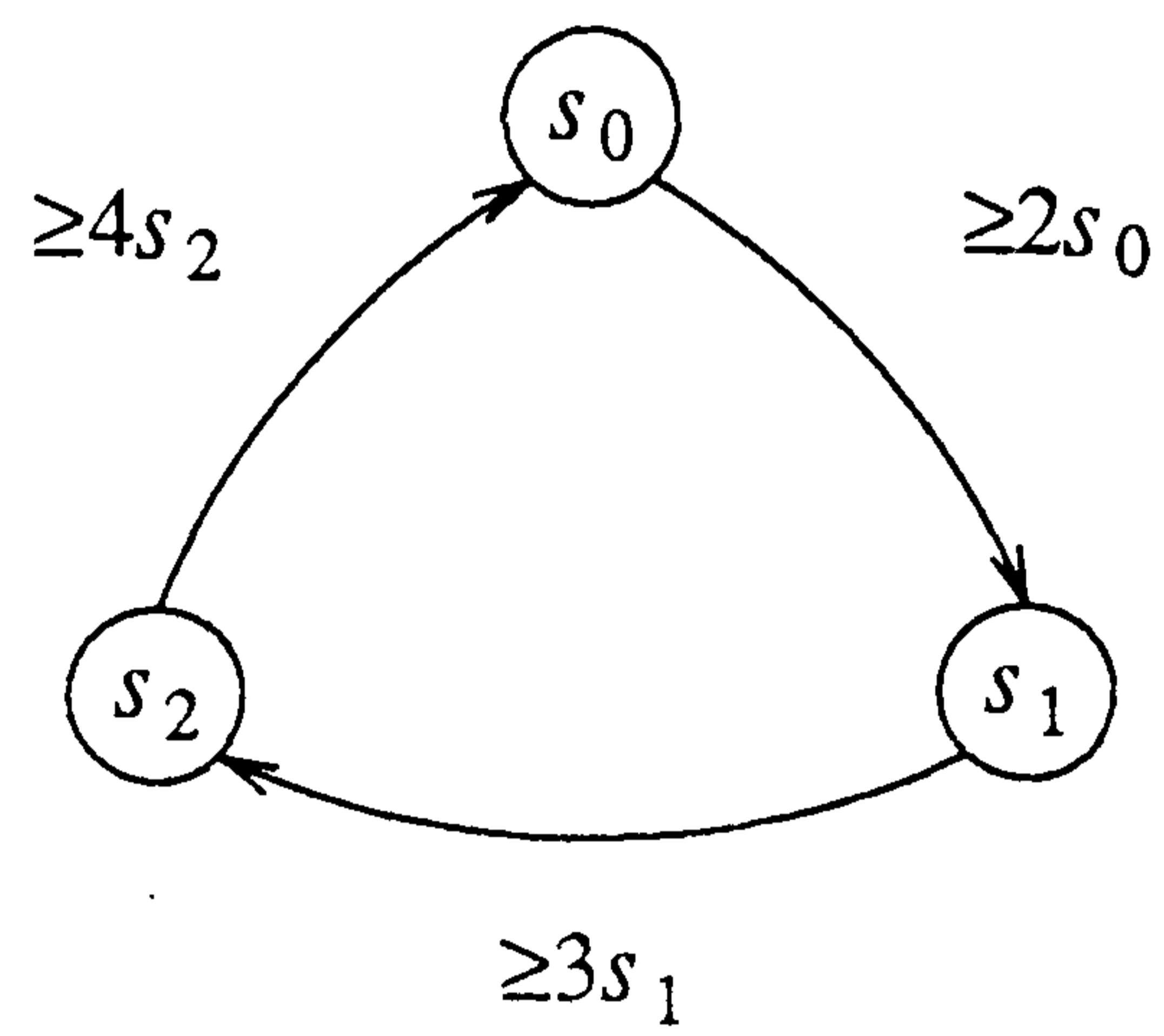
System 5



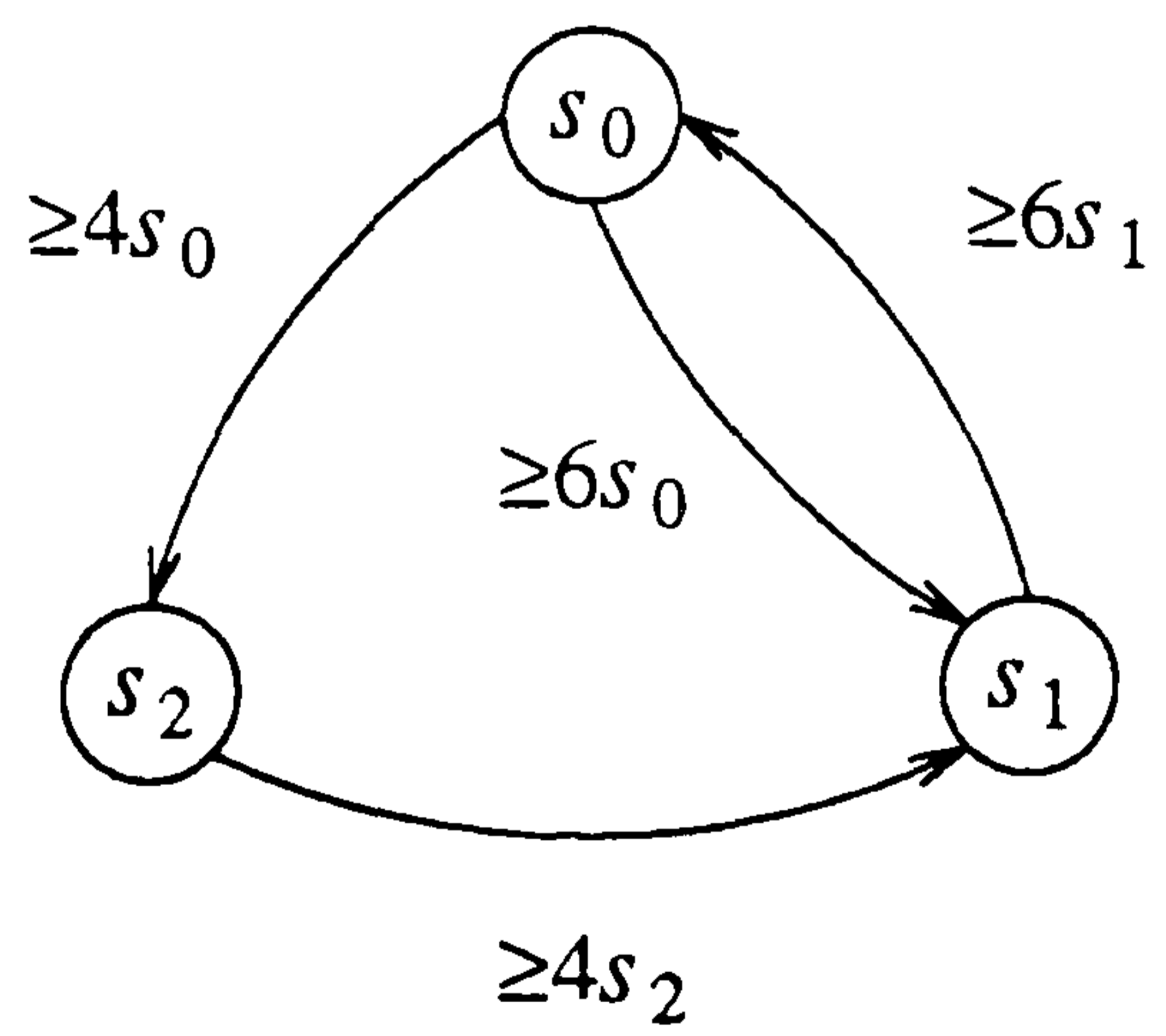
System 6



System 7



System 8



System 9

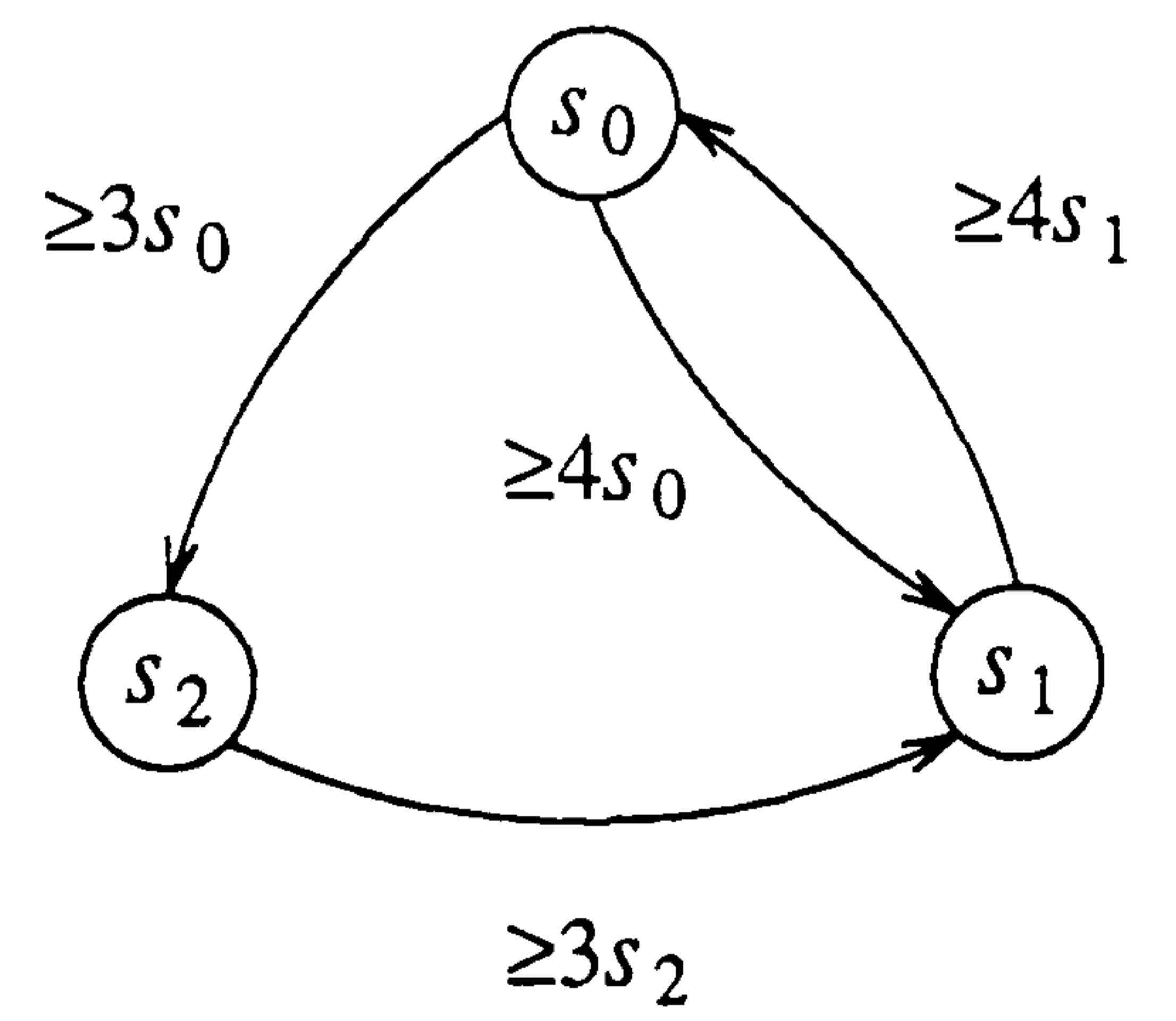


Figure 4.11: Threshold systems vector field diagrams.  
(a) System 3.

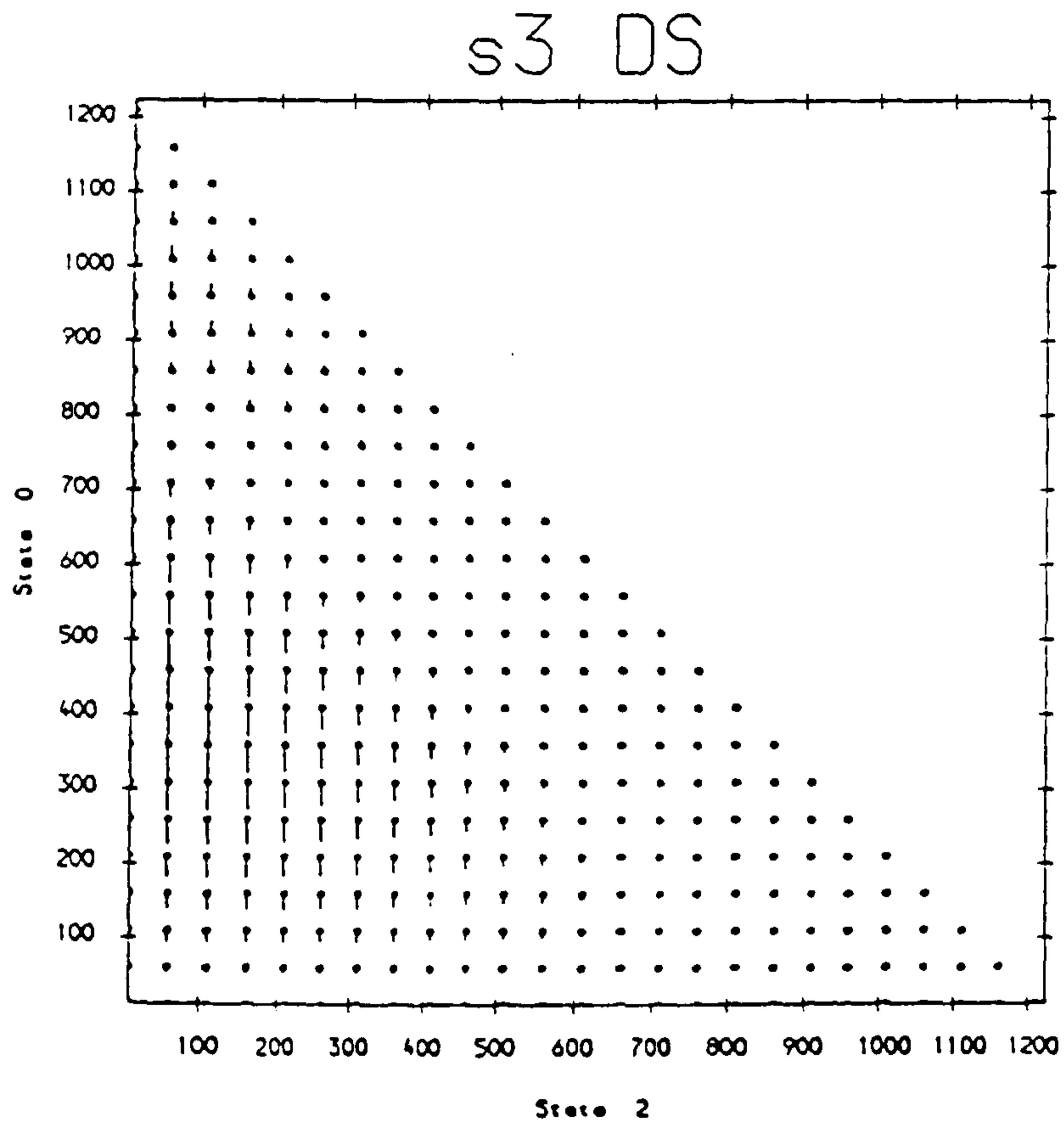
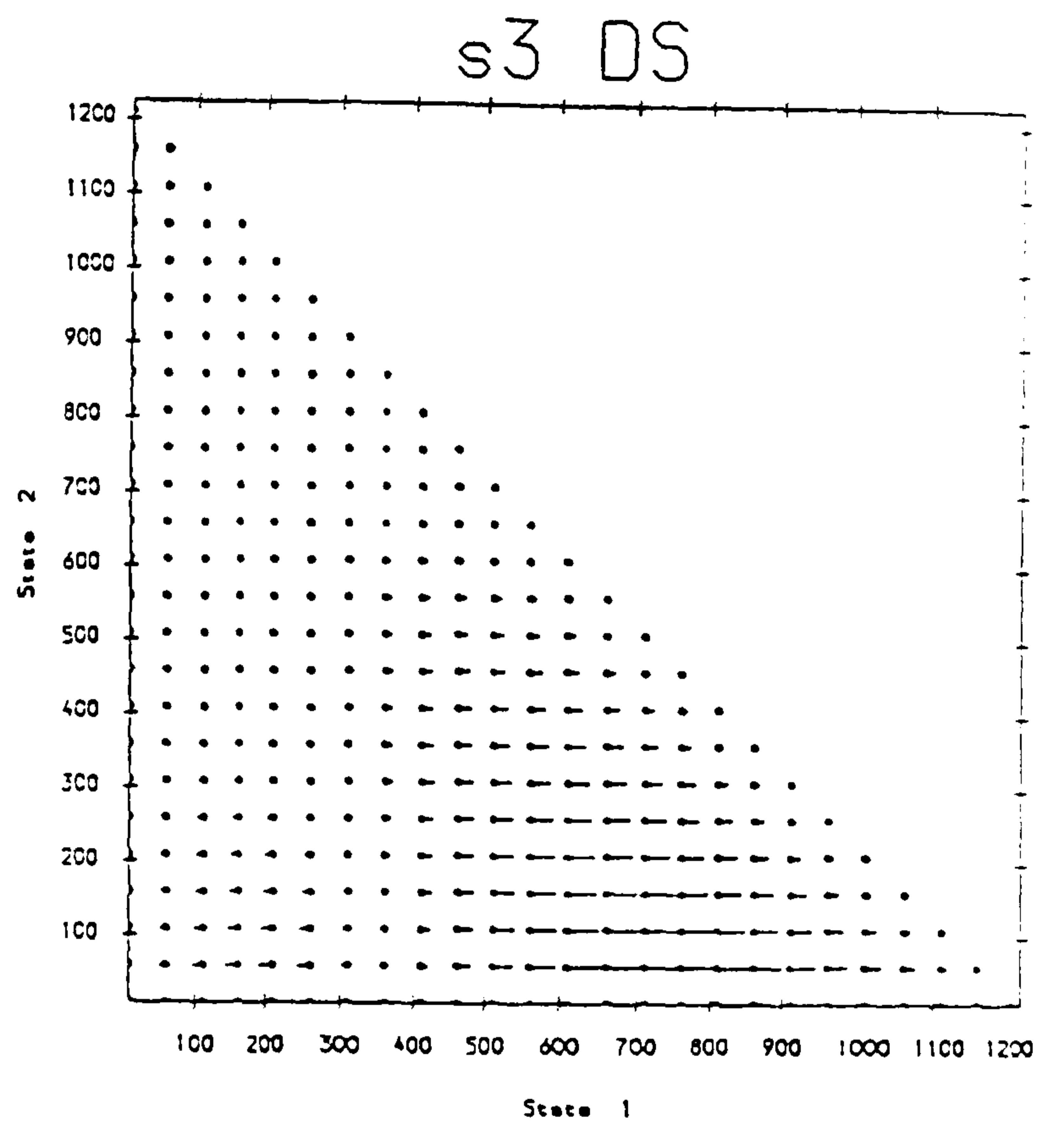
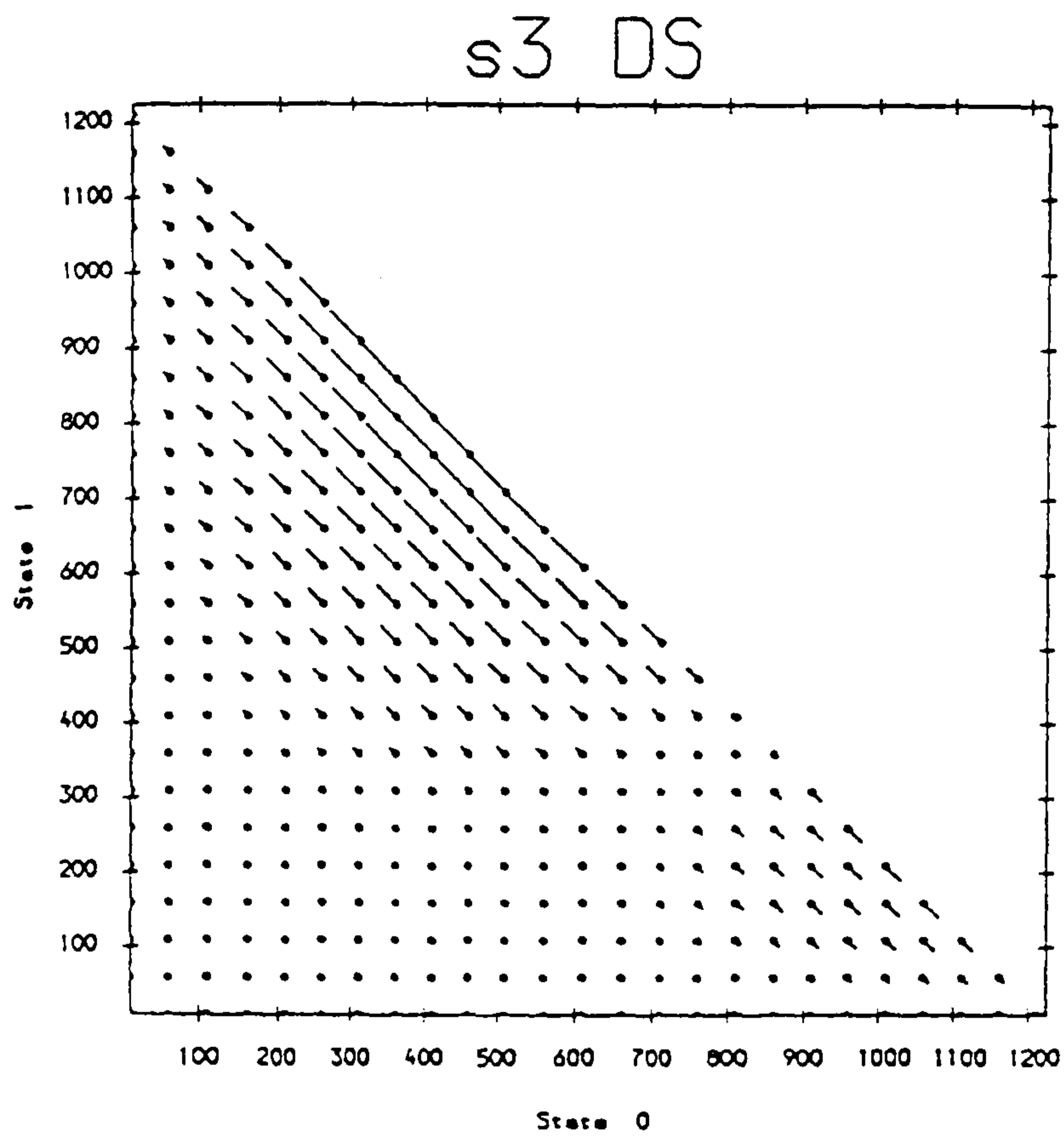


Figure 4.11: Threshold systems vector field diagrams.  
(b) System 4.

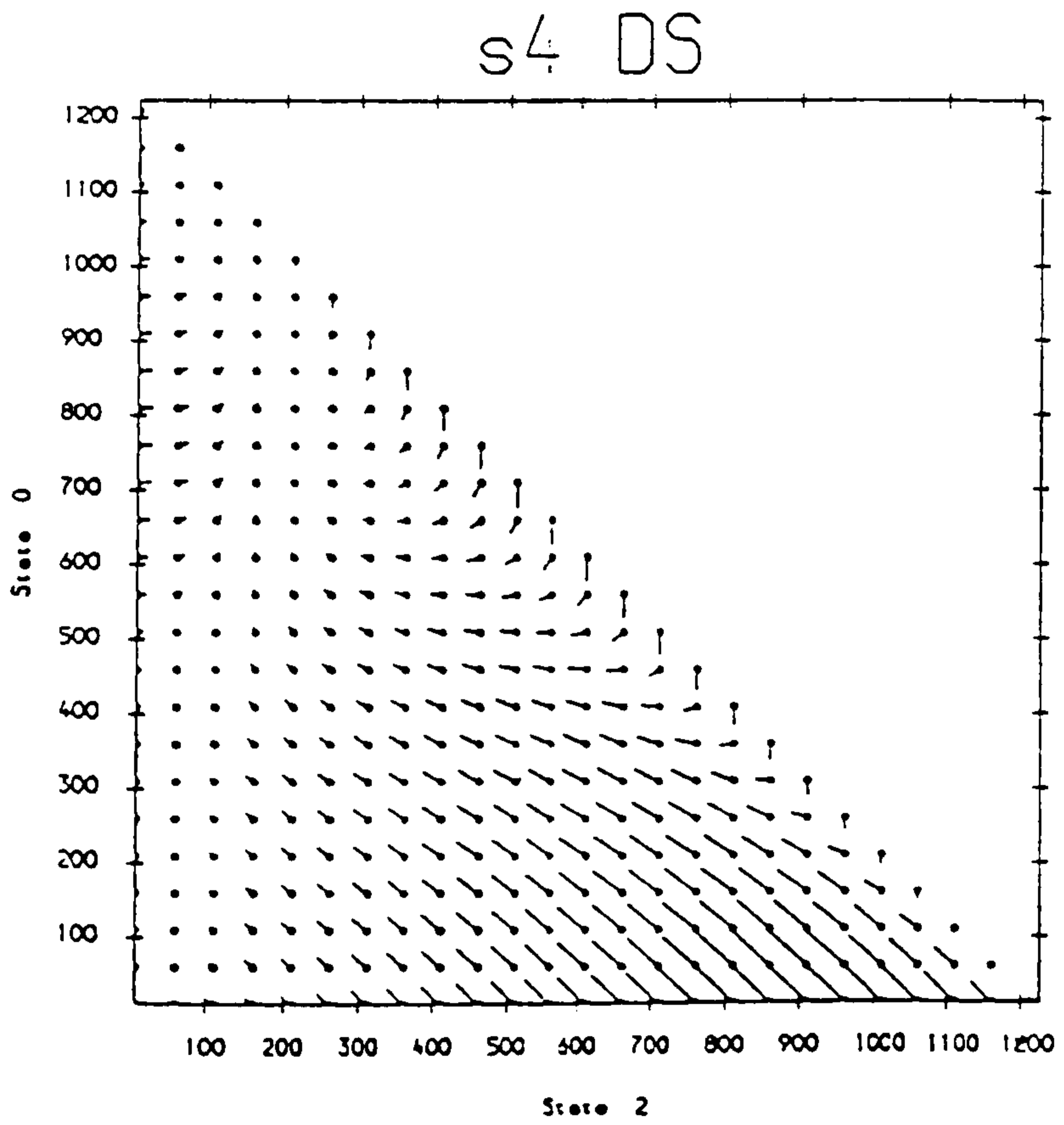
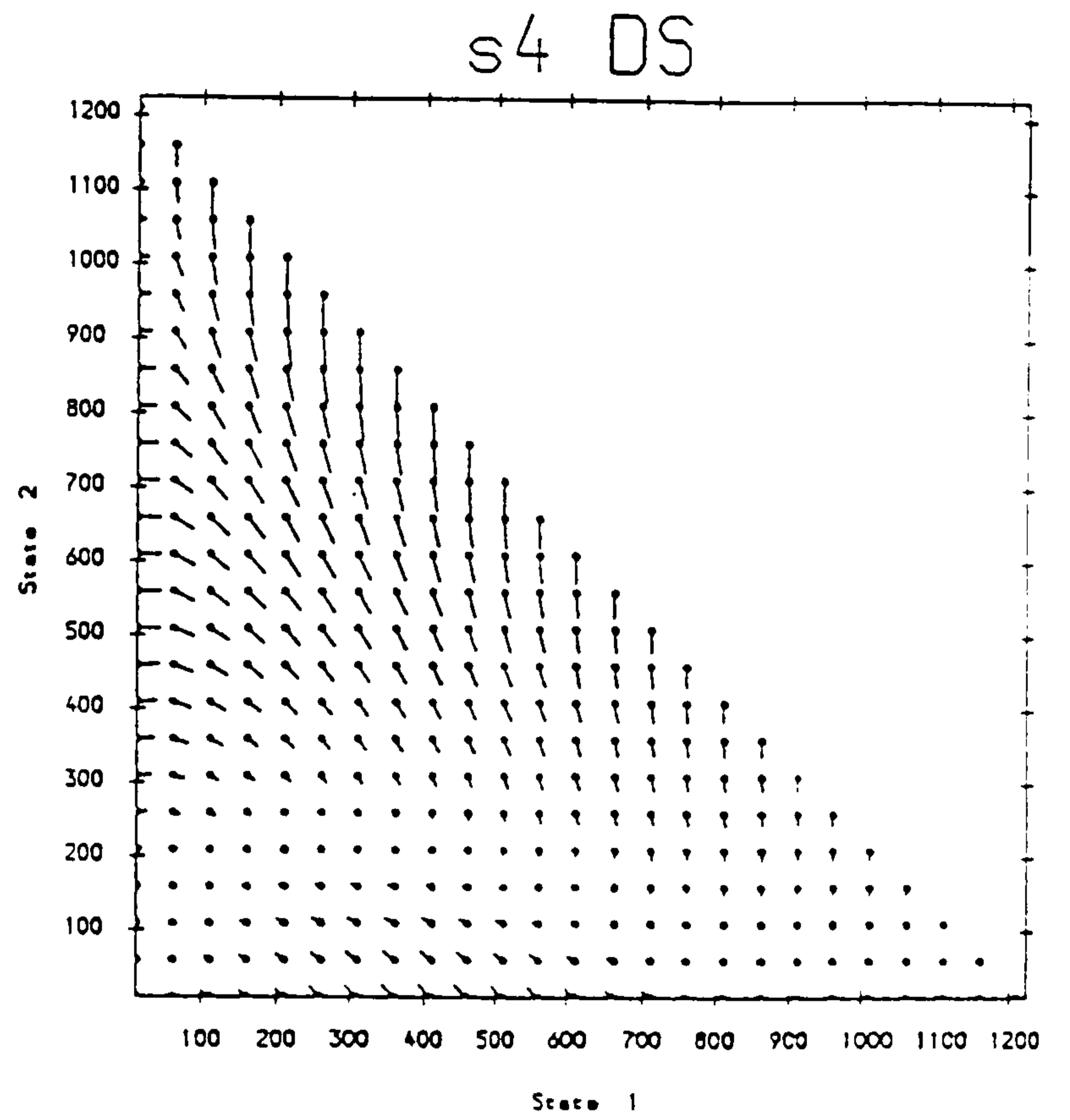
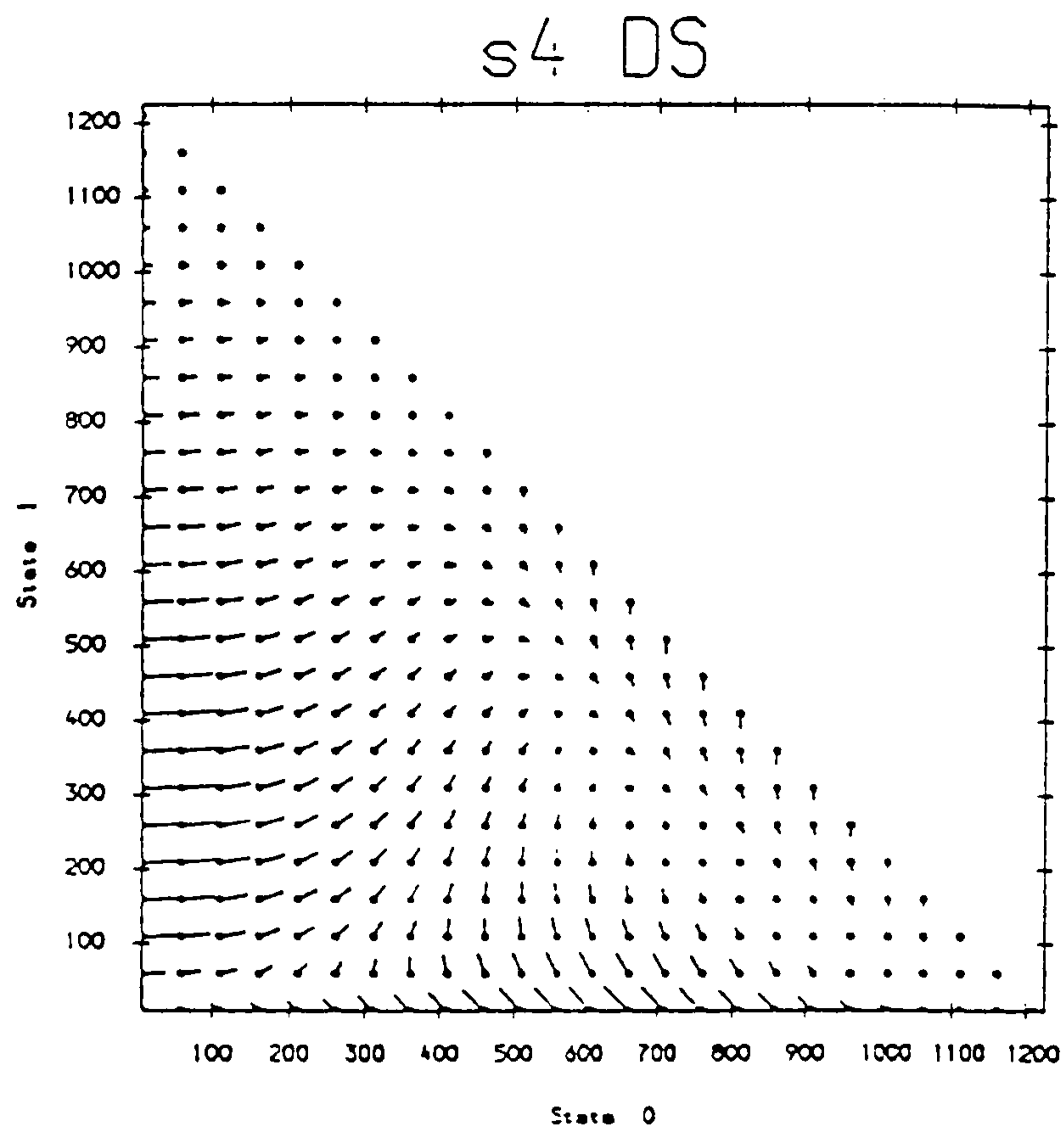


Figure 4.11: Threshold systems vector field diagrams.  
(c) System 5.

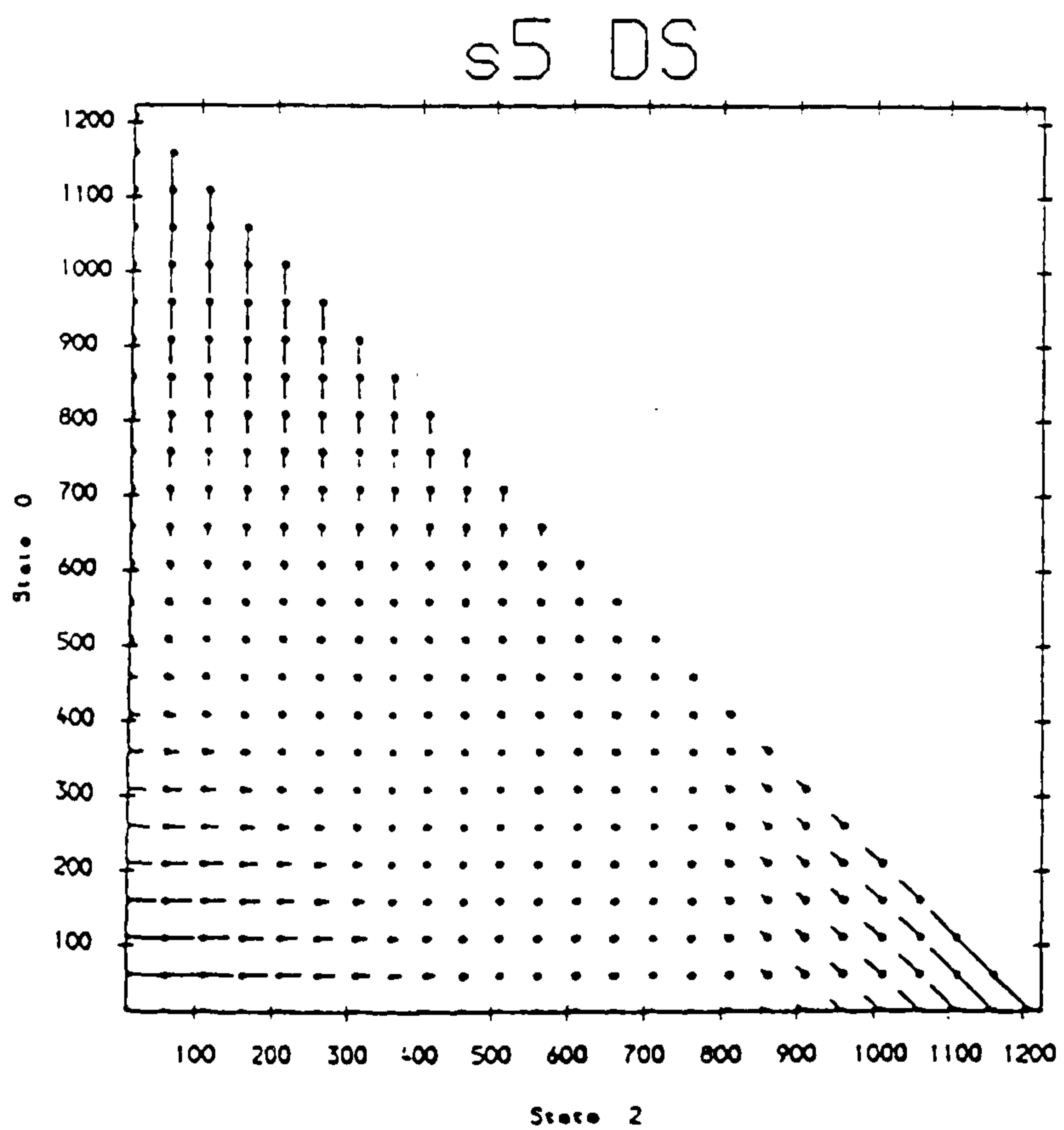
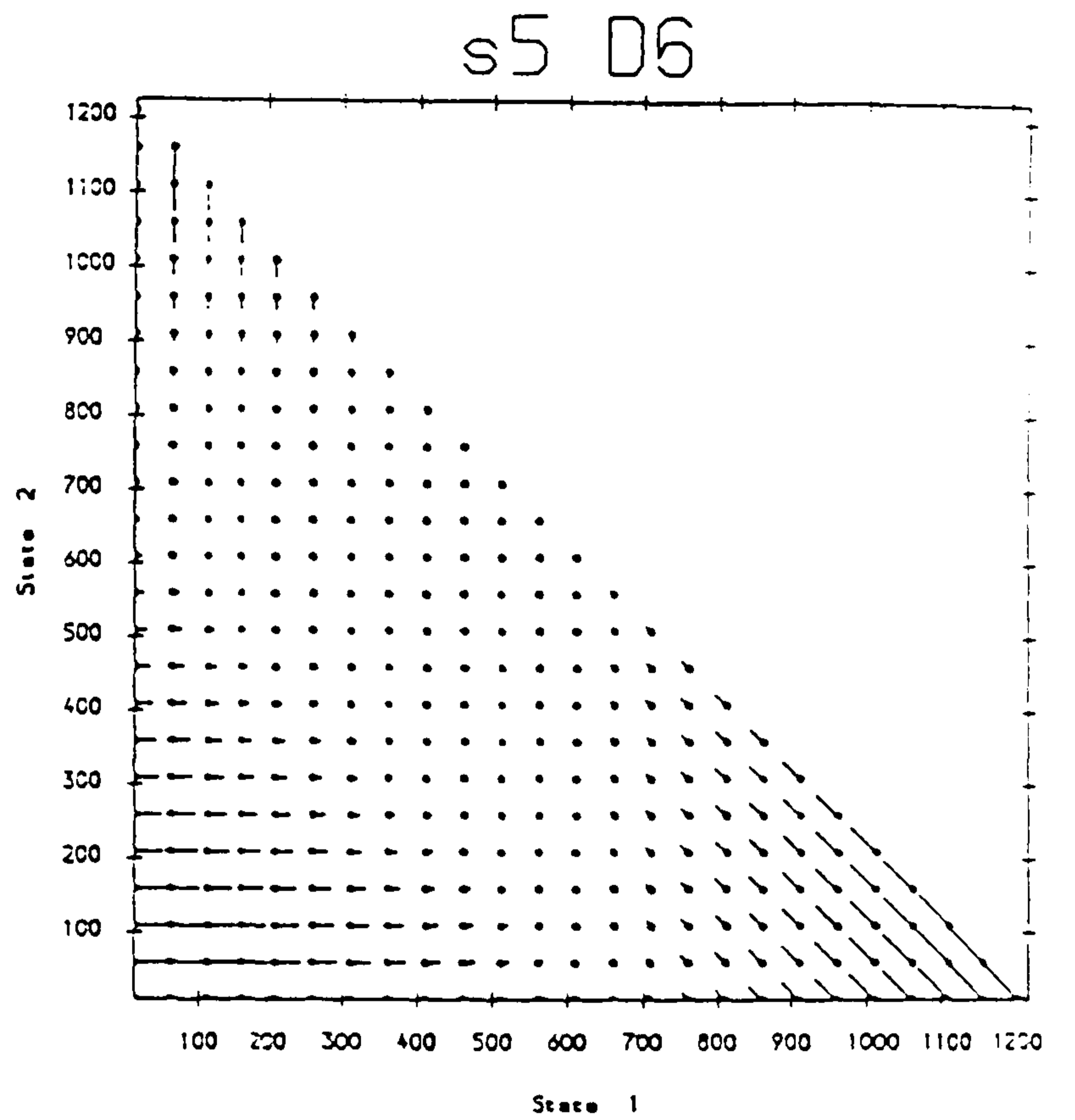
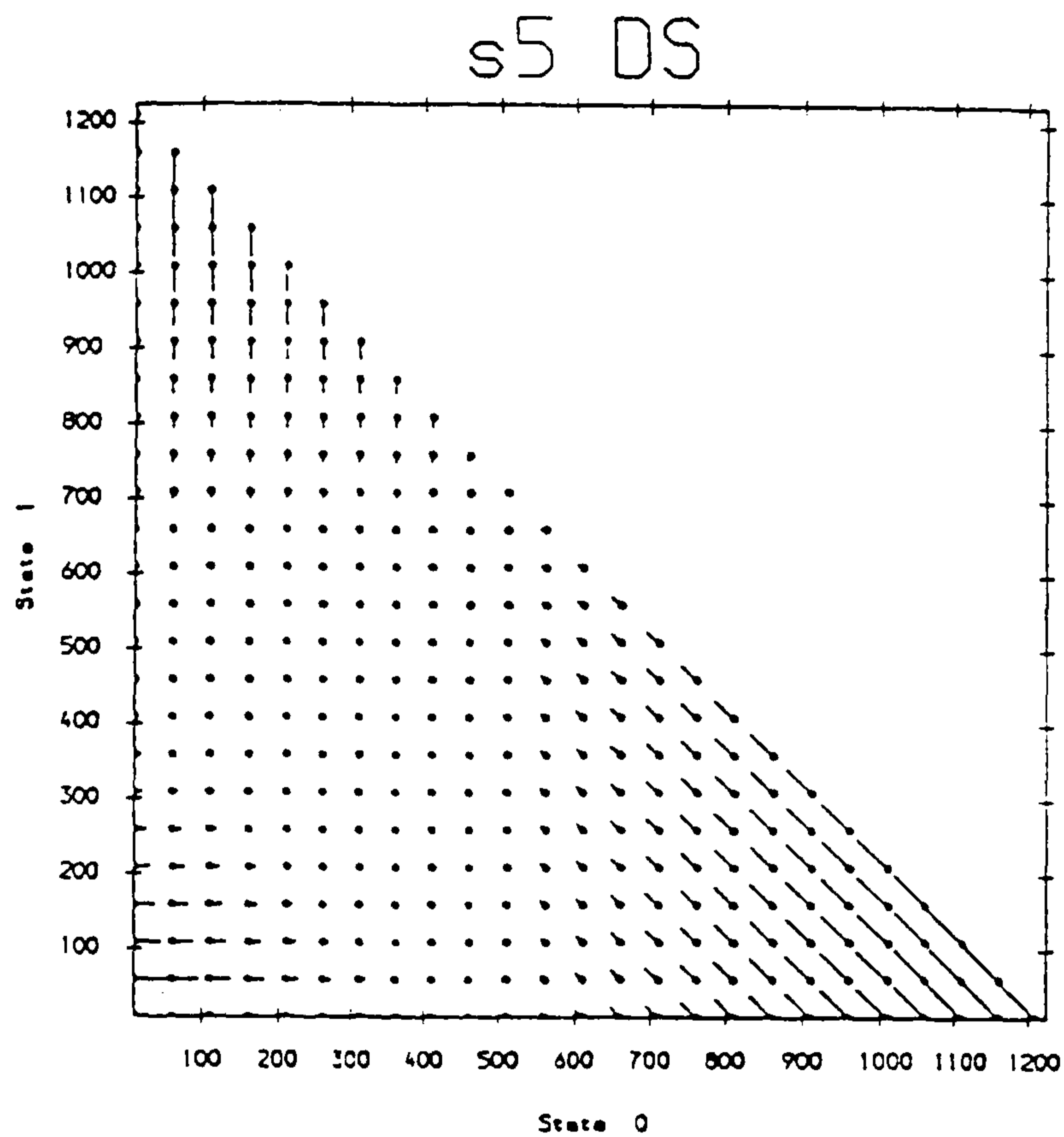




Figure 4.11: Threshold systems vector field diagrams.  
(d) System 6.

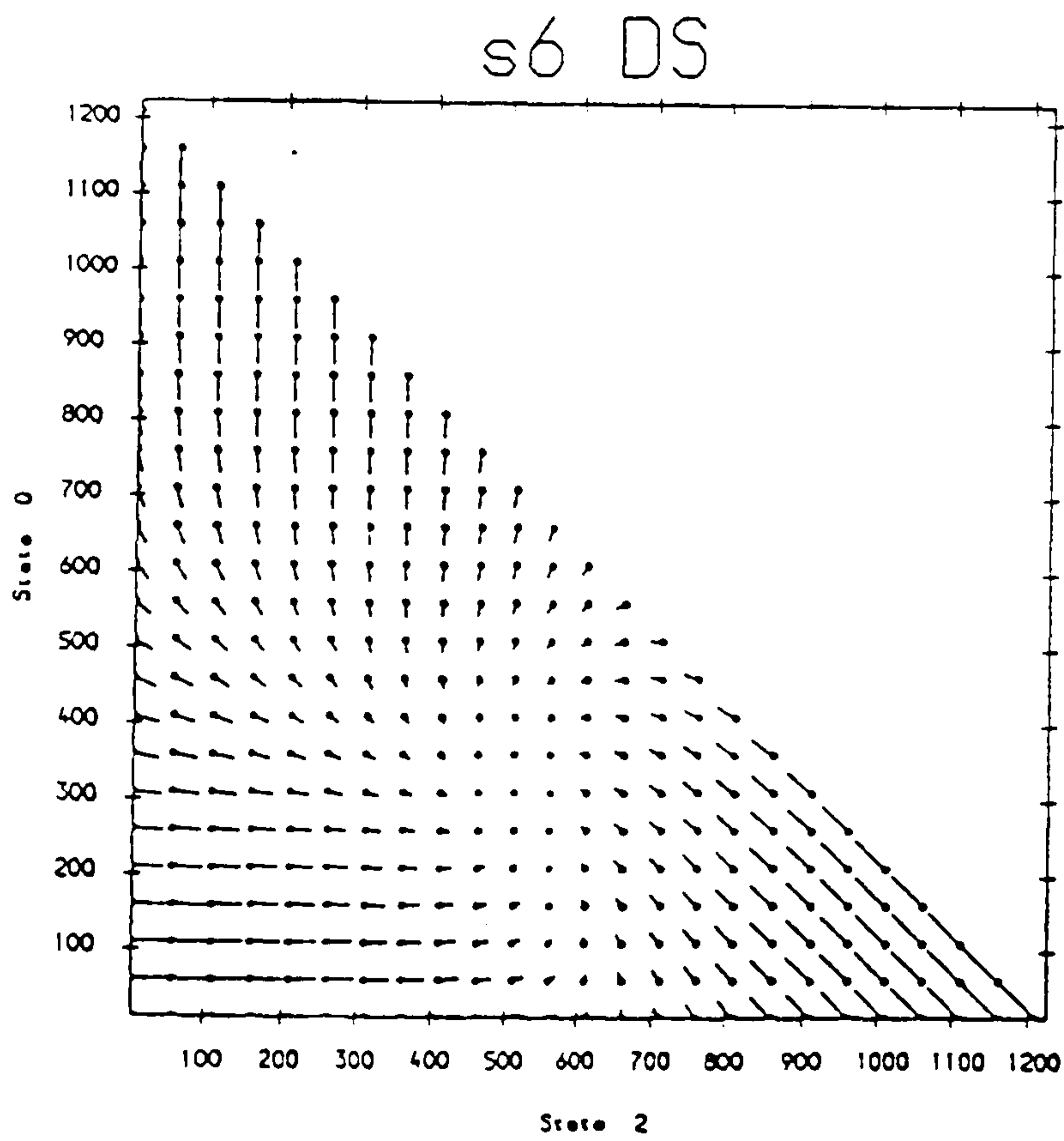
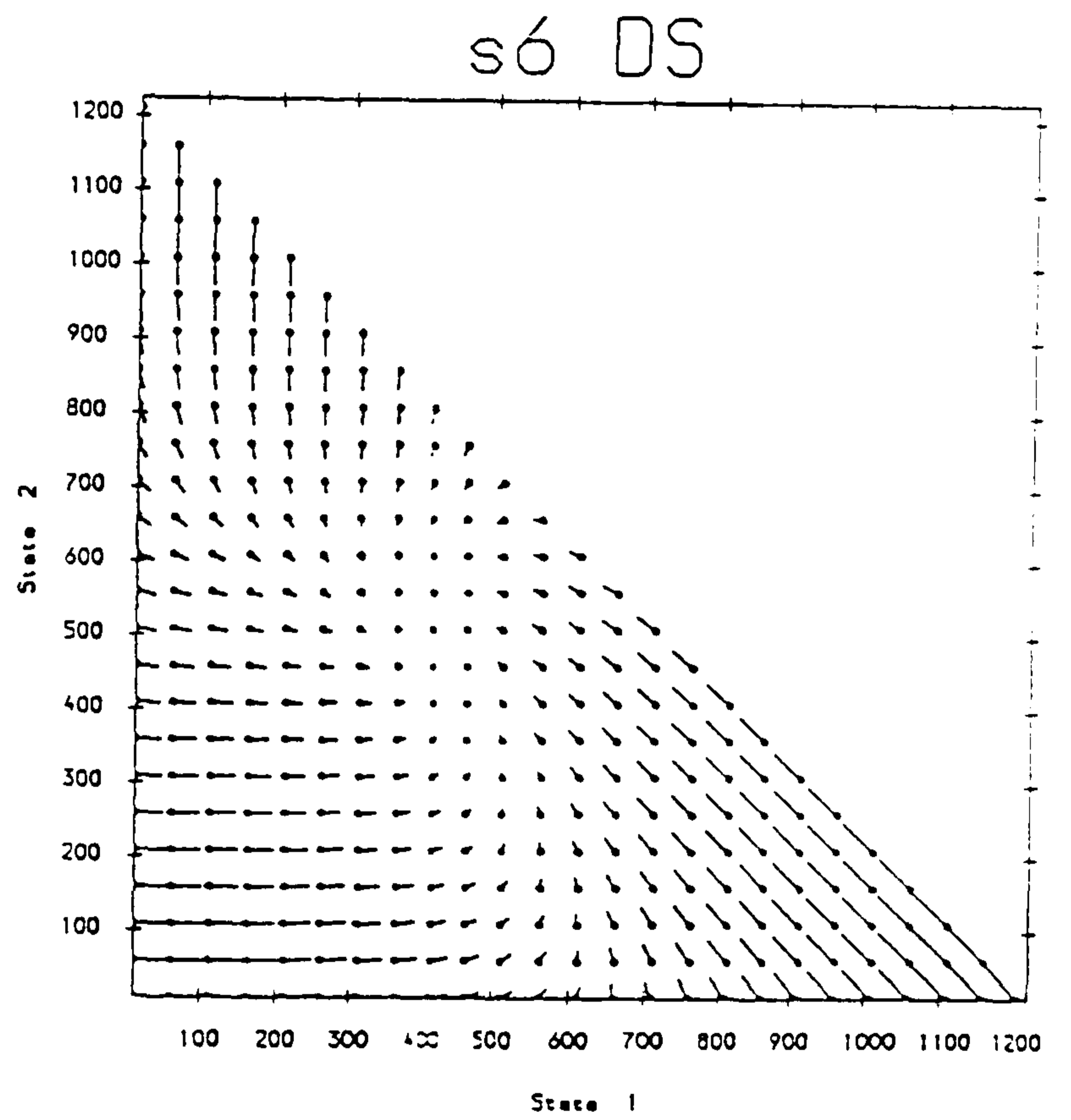
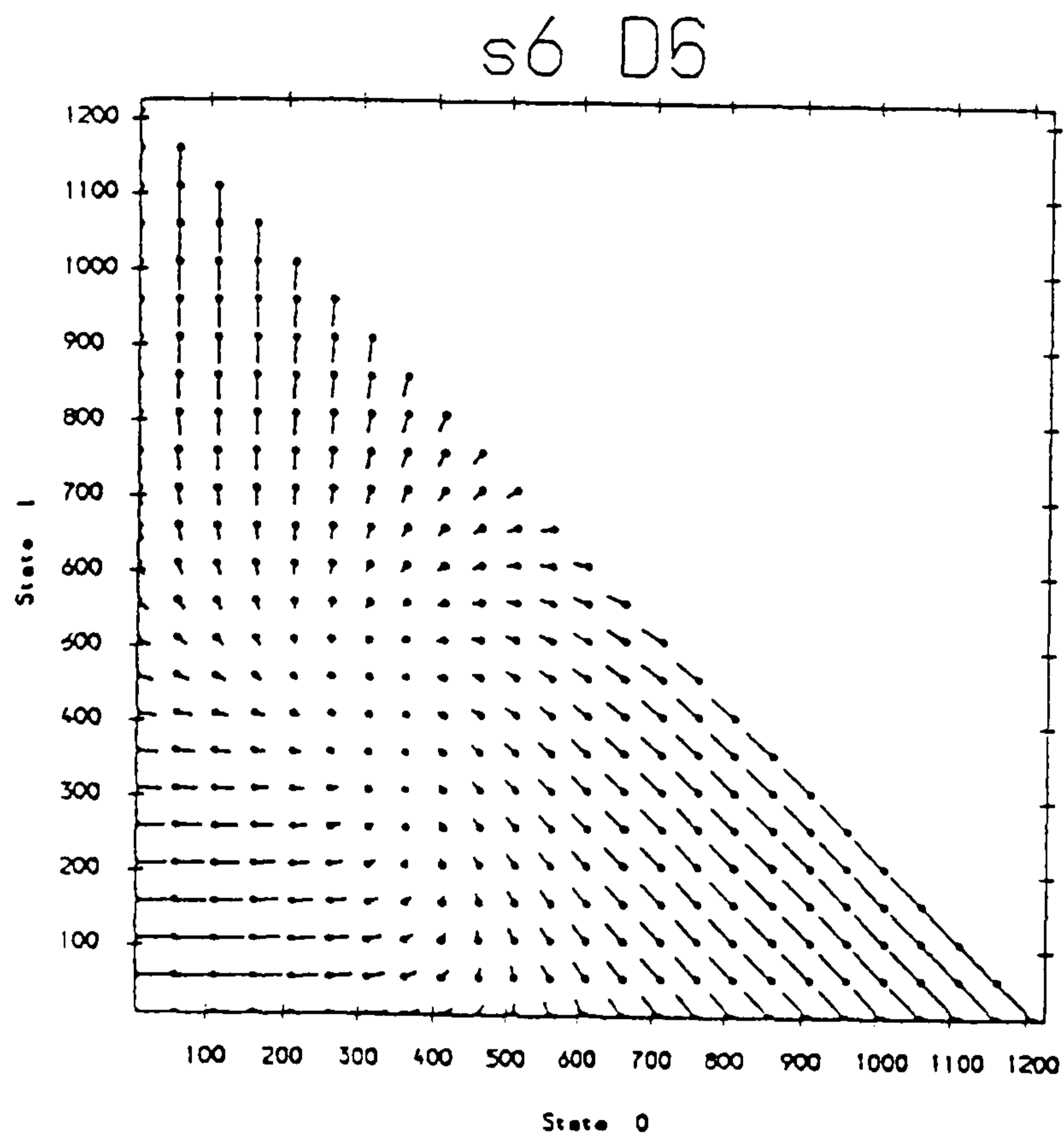


Figure 4.11: Threshold systems vector field diagrams.  
(e) System 7.

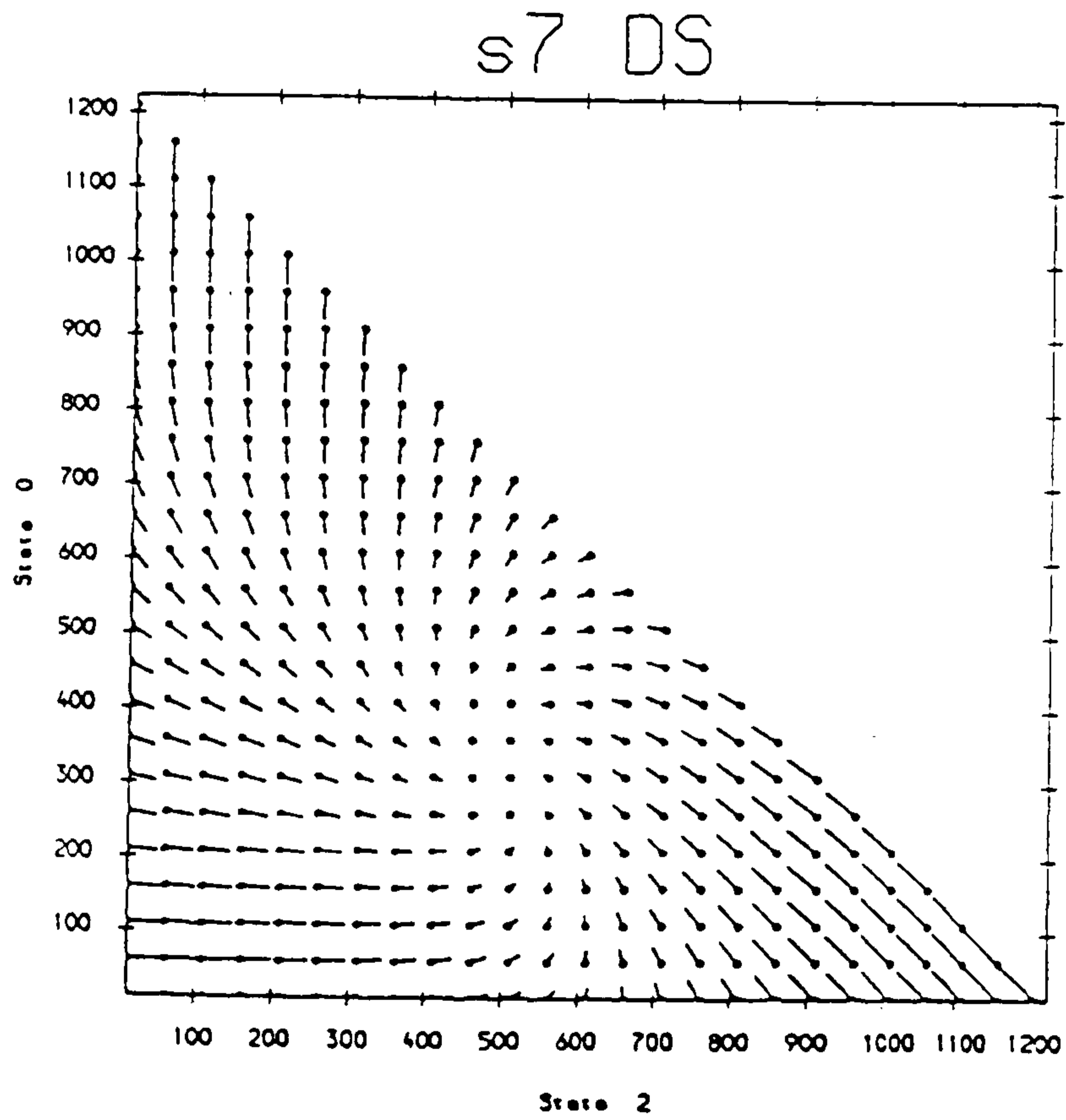
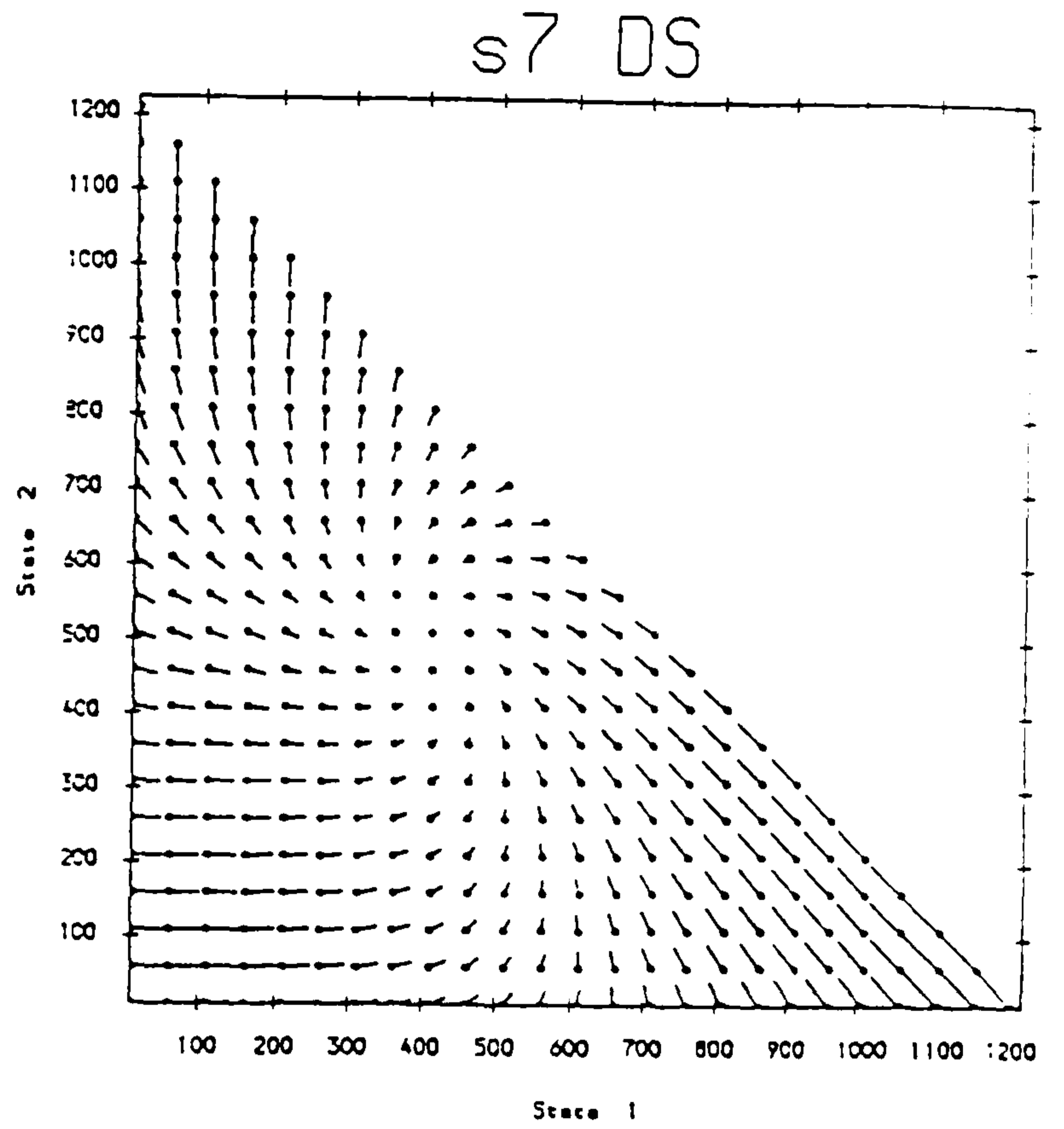
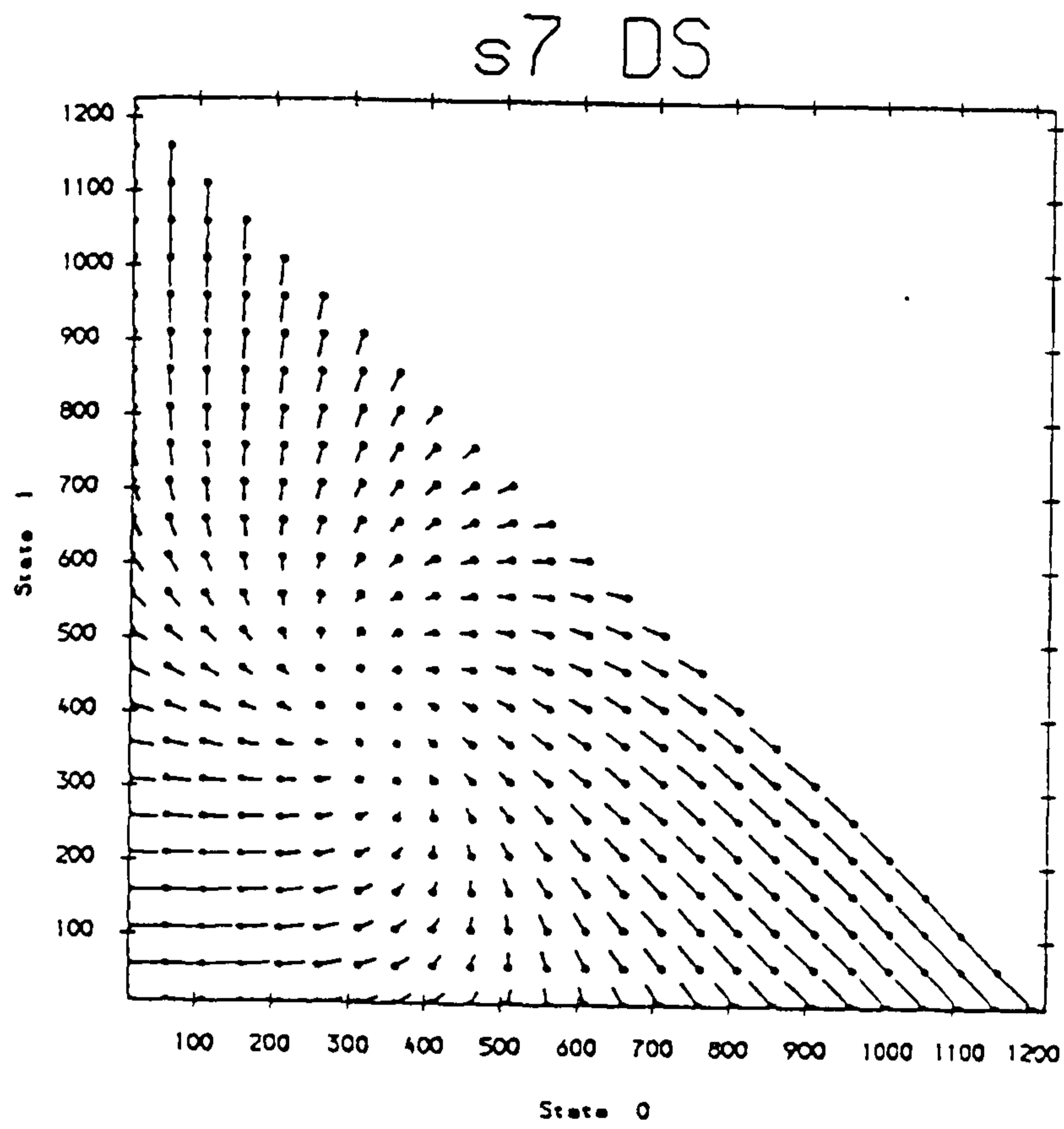


Figure 4.11: Threshold systems vector field diagrams.  
(f) System 8.

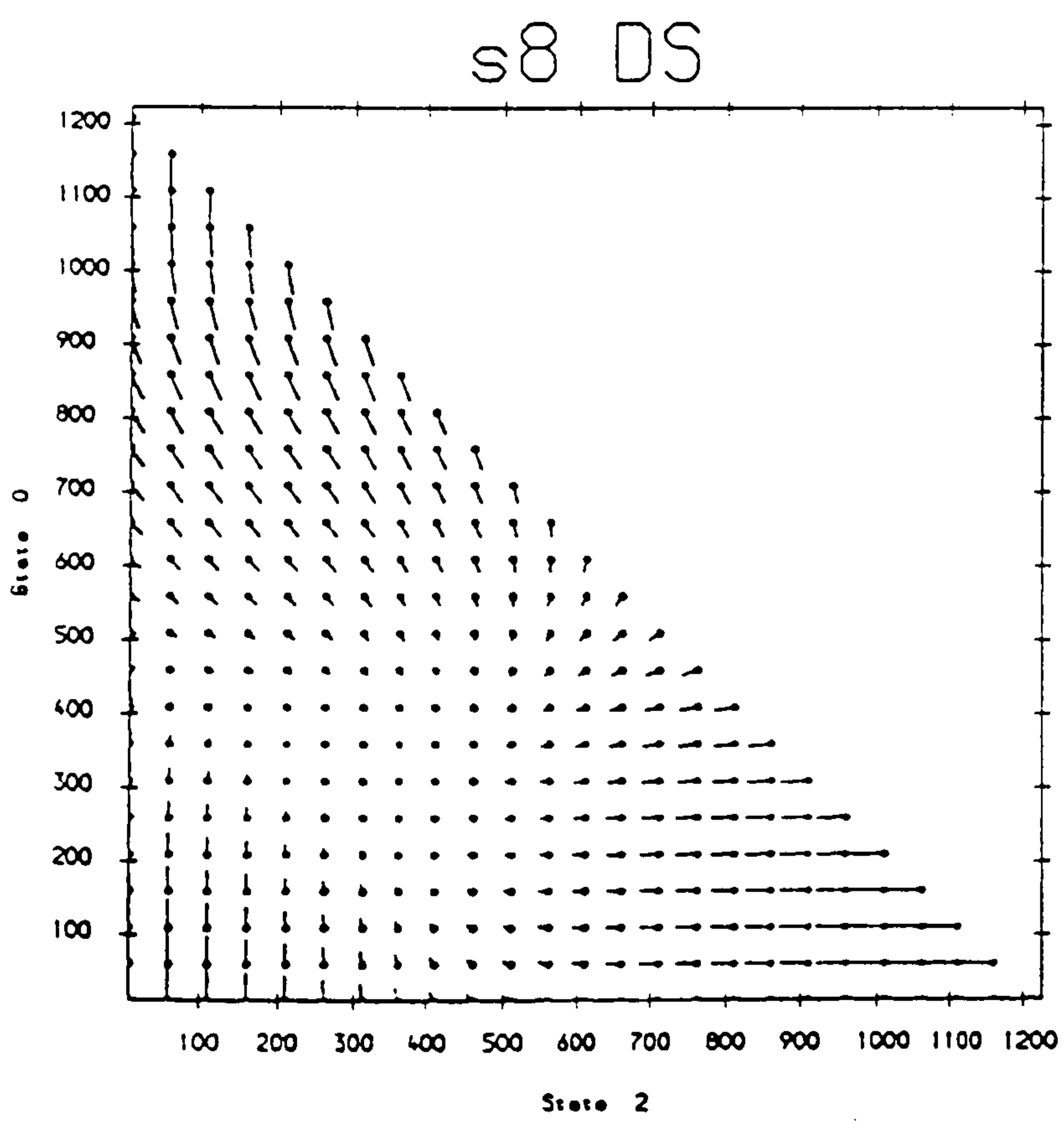
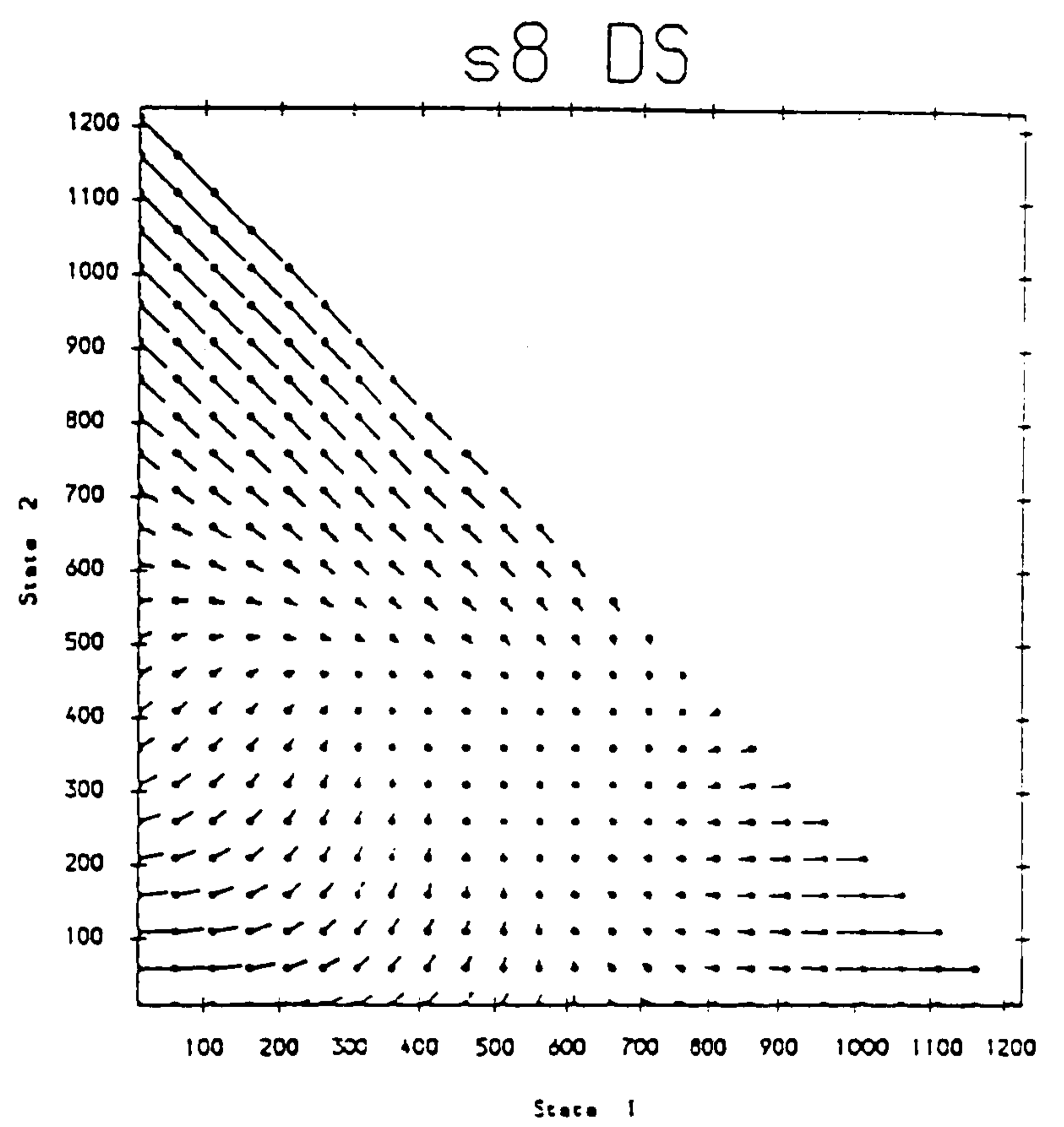
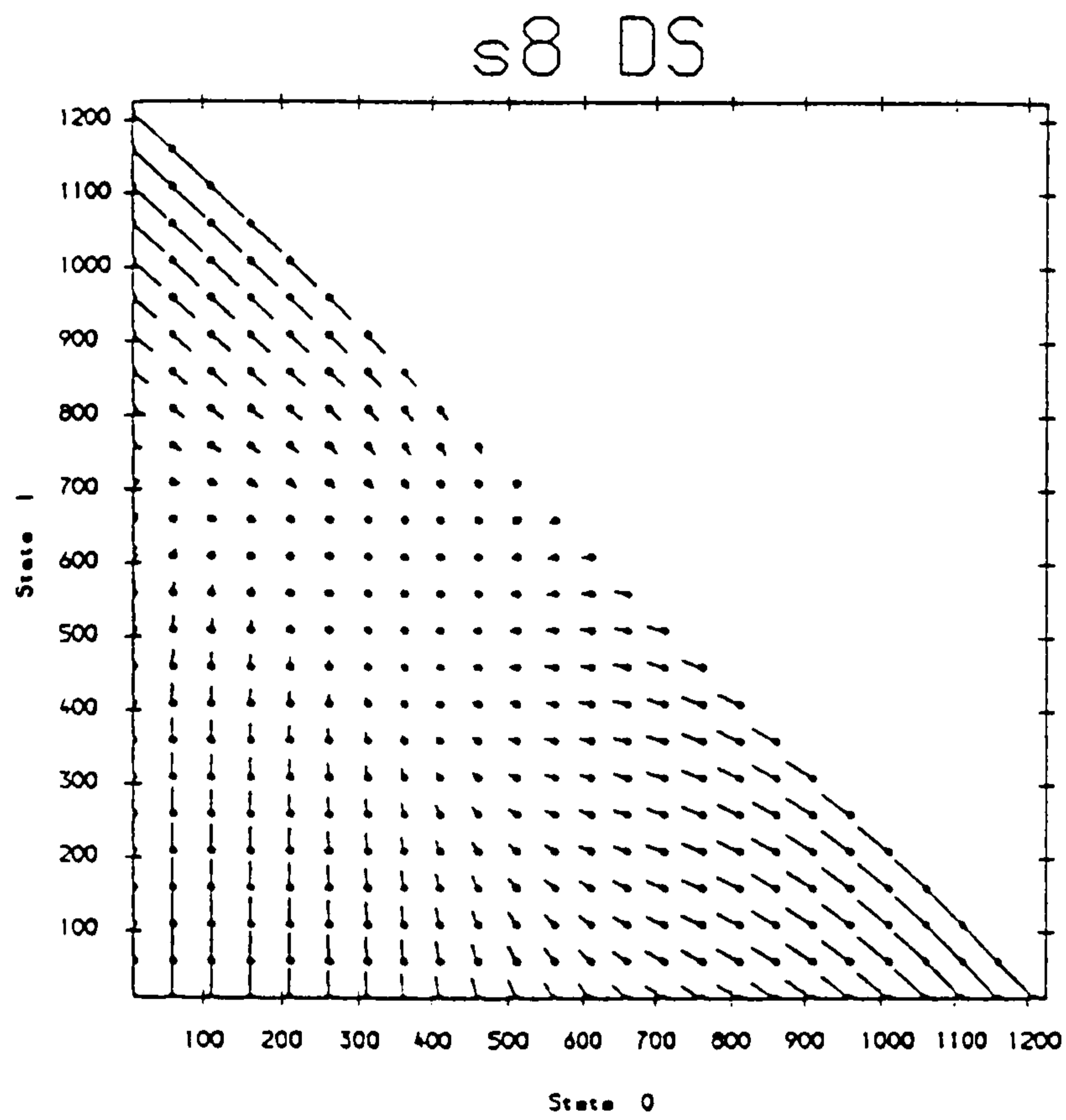


Figure 4.11: Threshold systems vector field diagrams.  
(g) System 9.

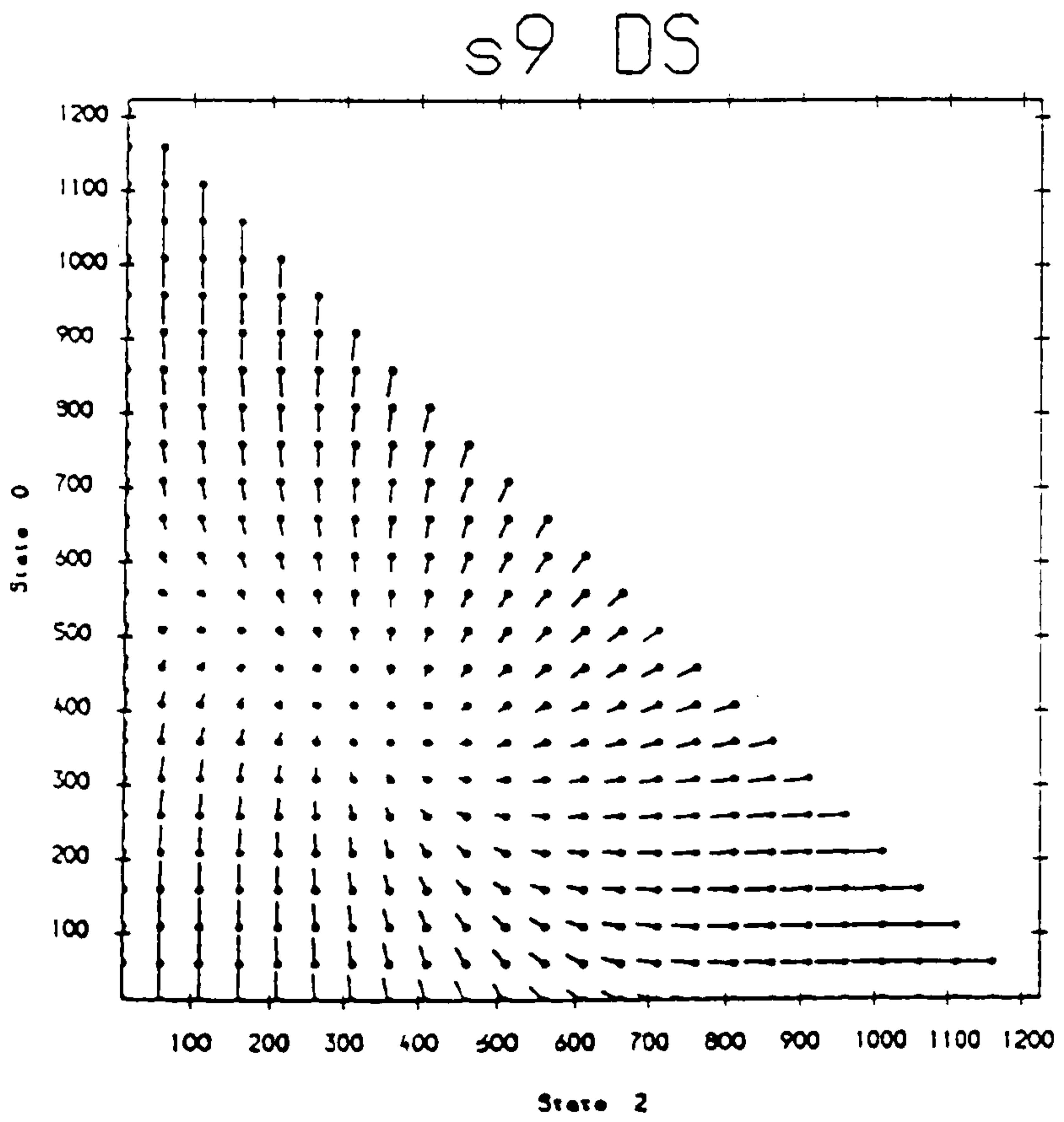
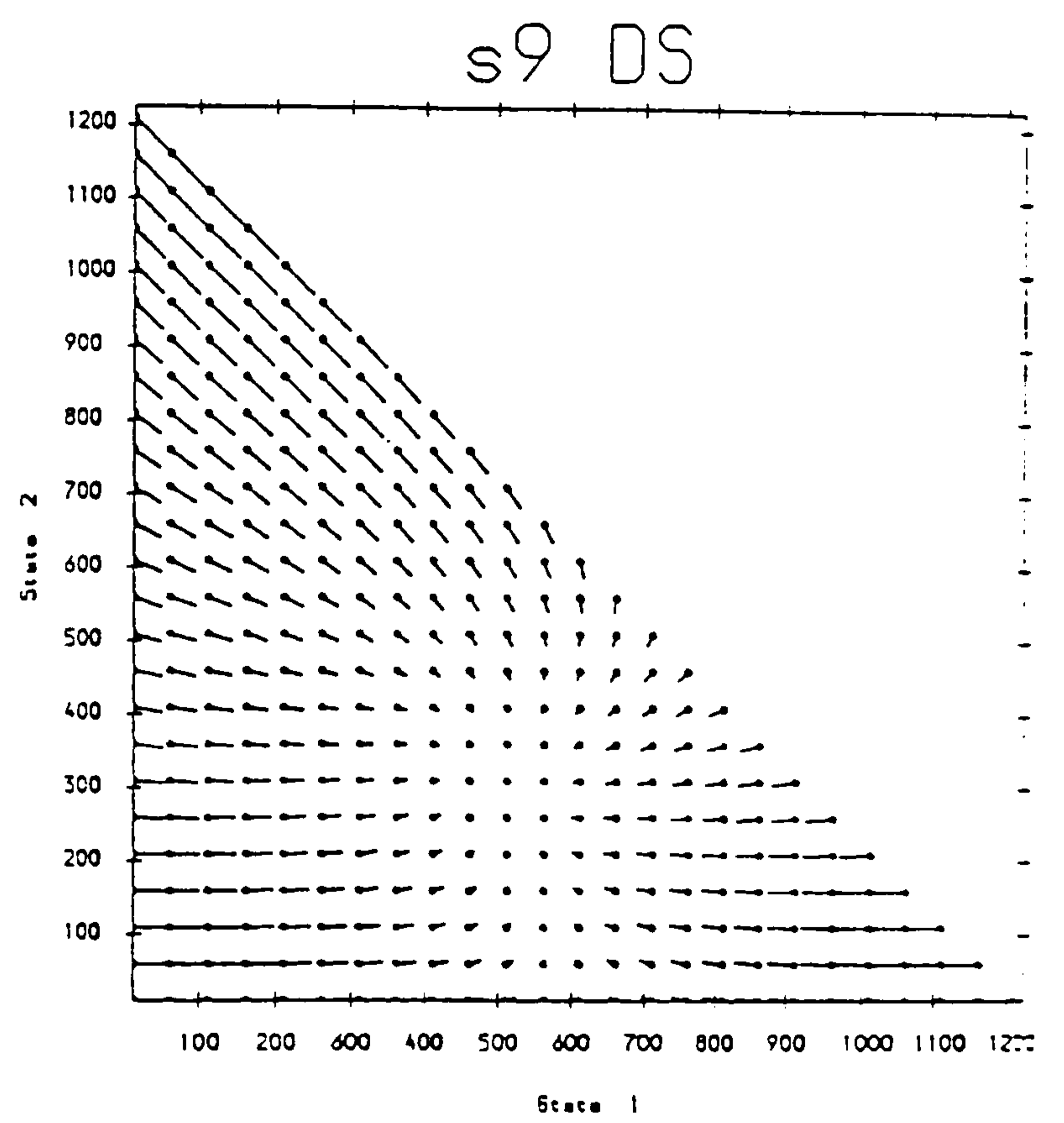
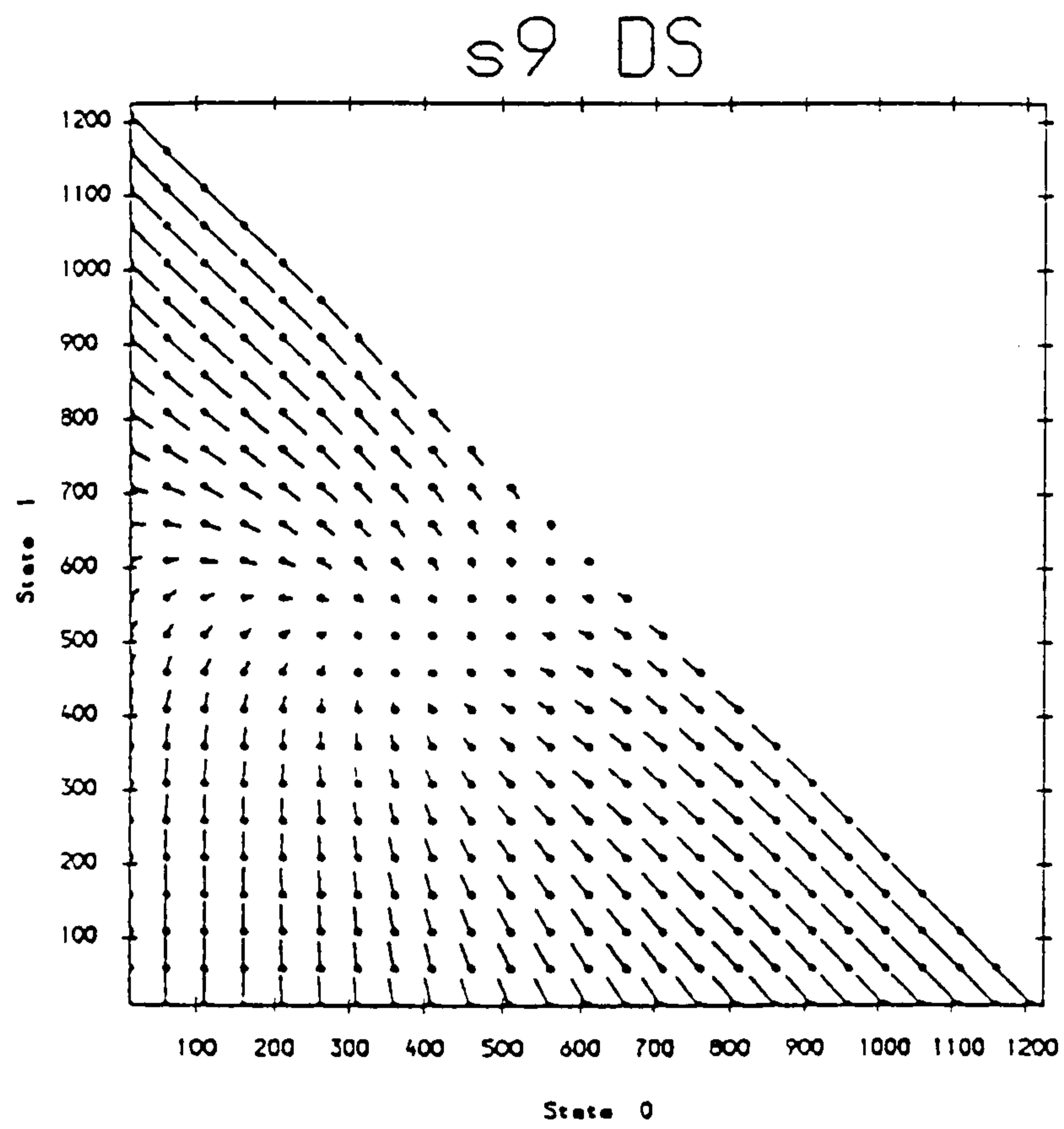




Figure 4.12: Threshold systems phase diagrams.  
 (a) System 3: Deterministic.

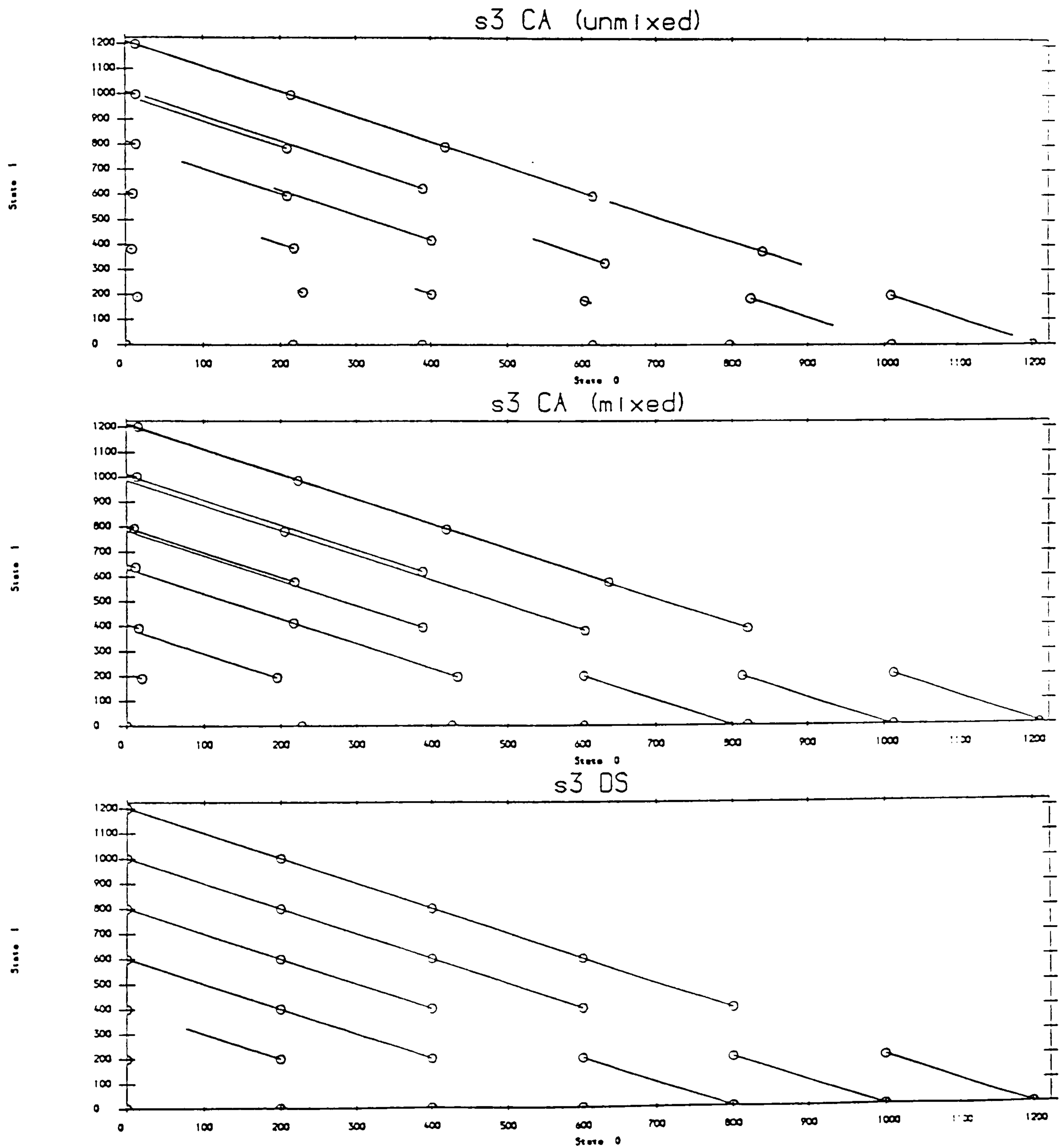




Figure 4.12: Threshold systems phase diagrams.  
(b) System 4: Deterministic.

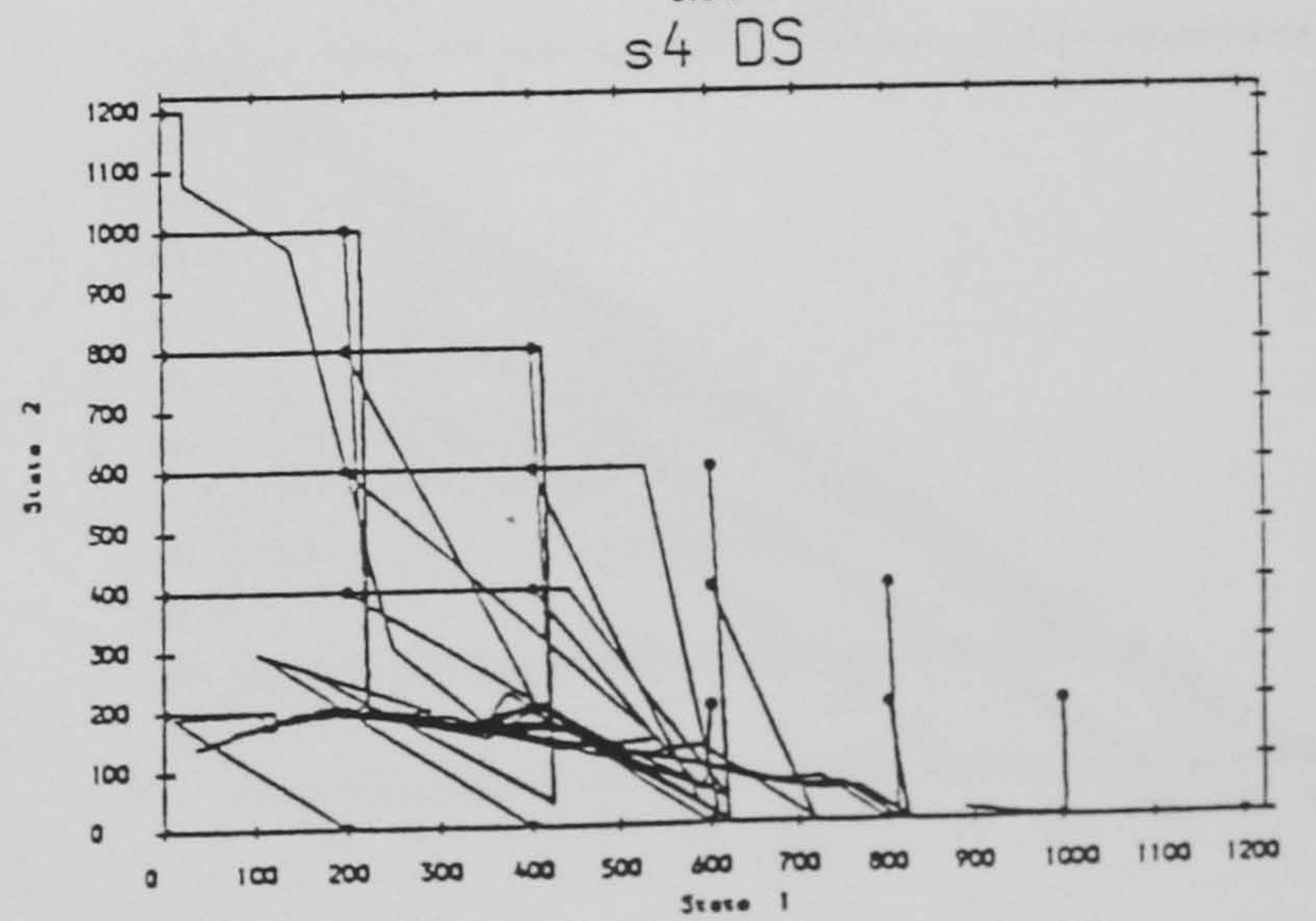
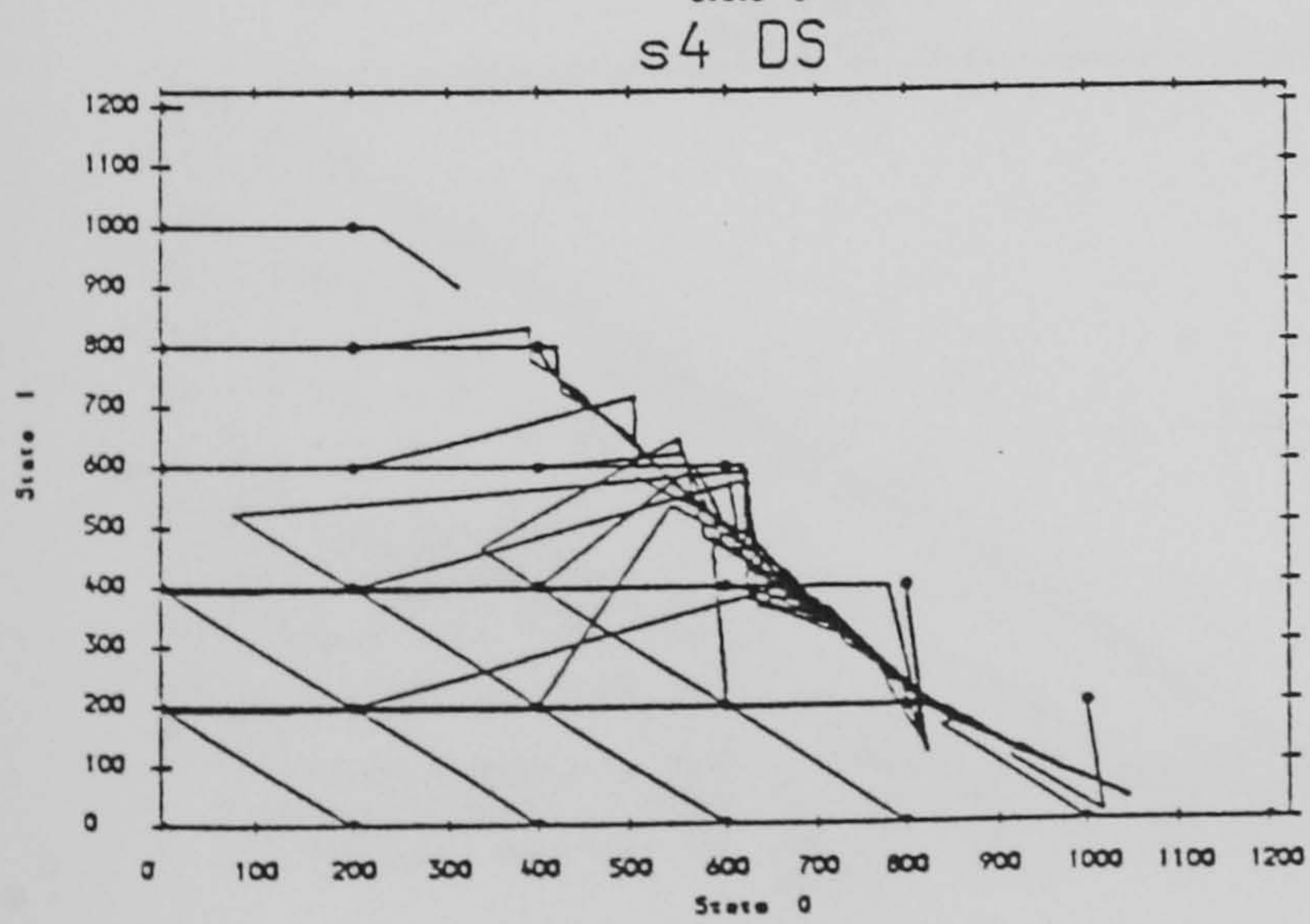
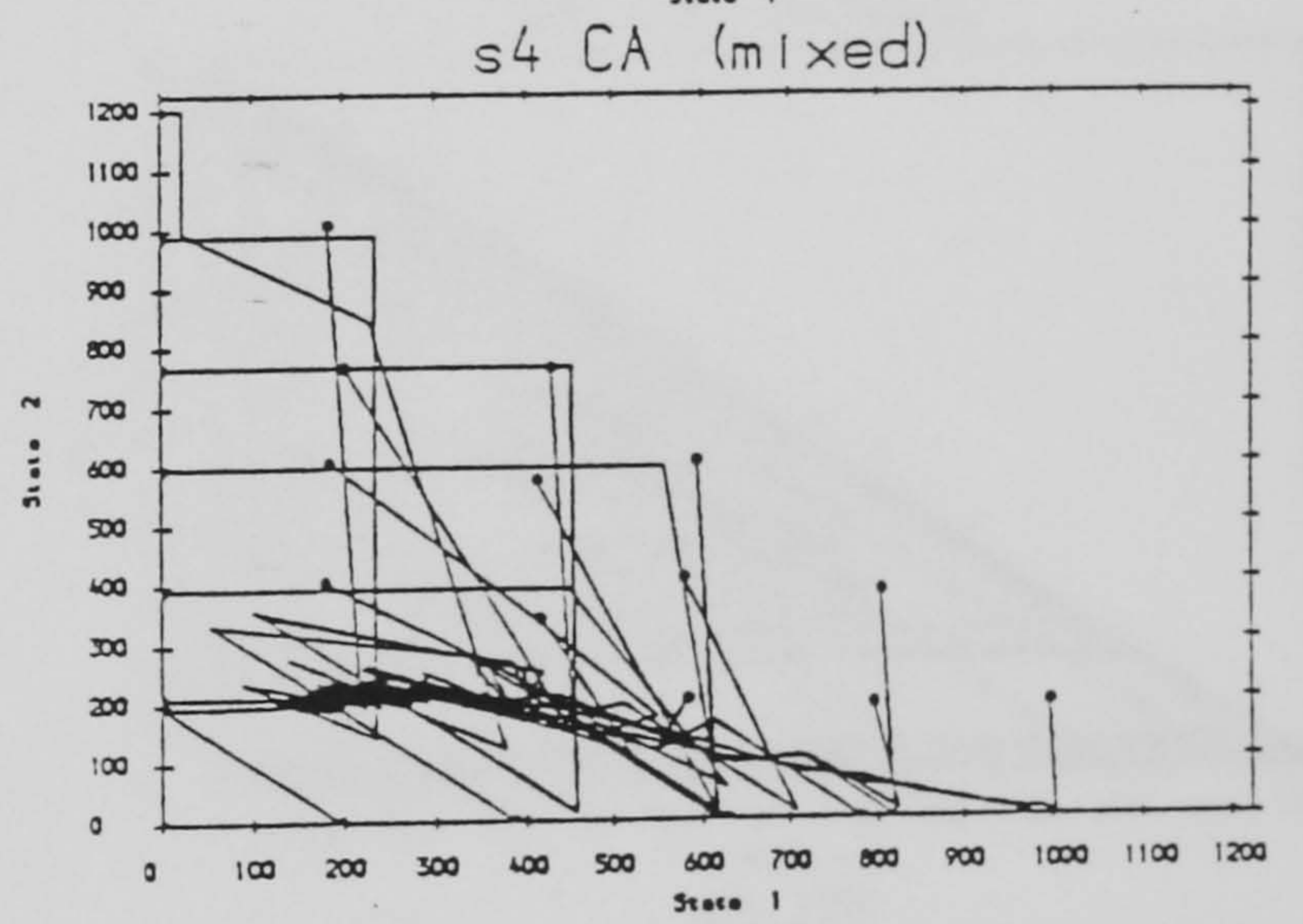
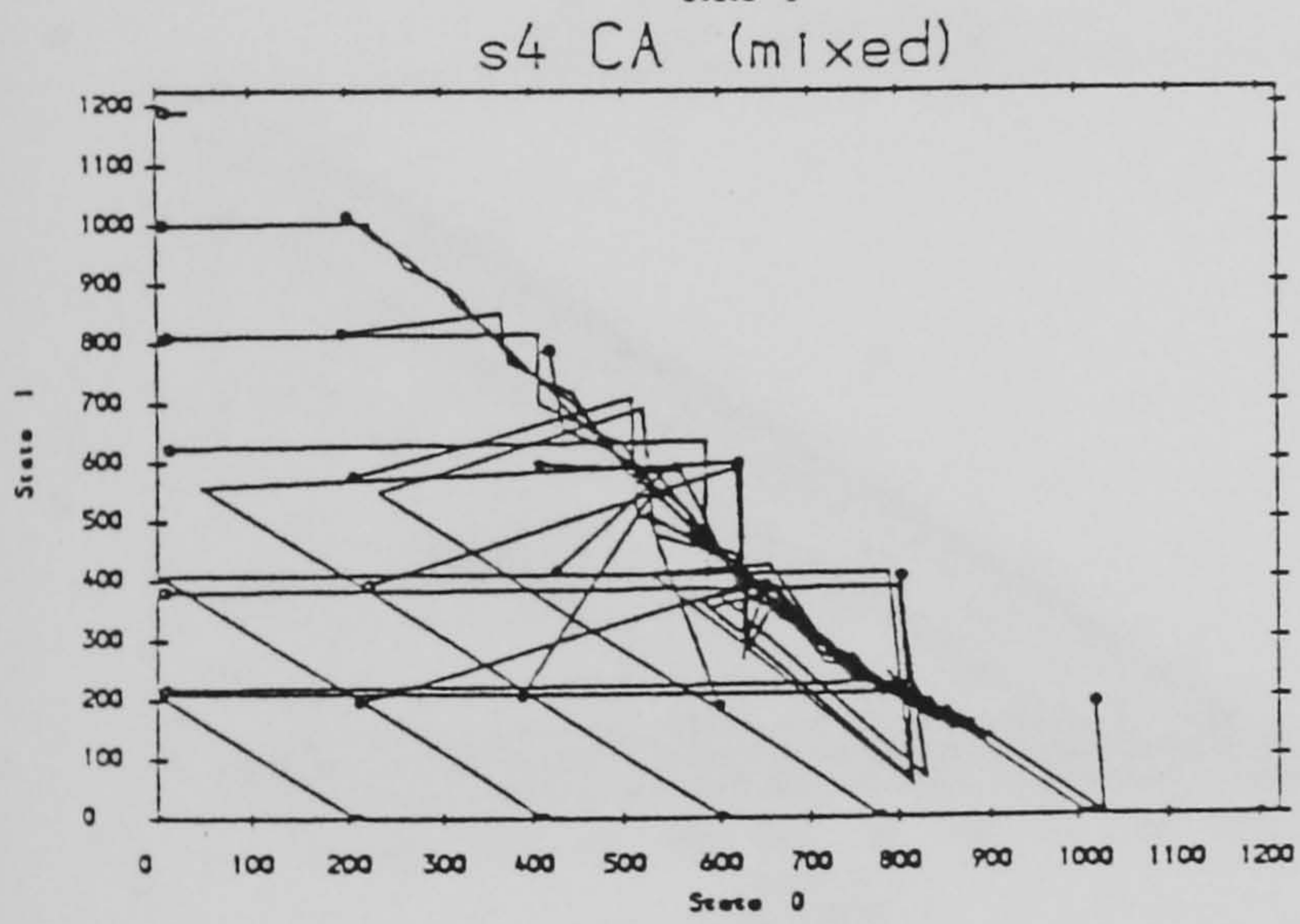
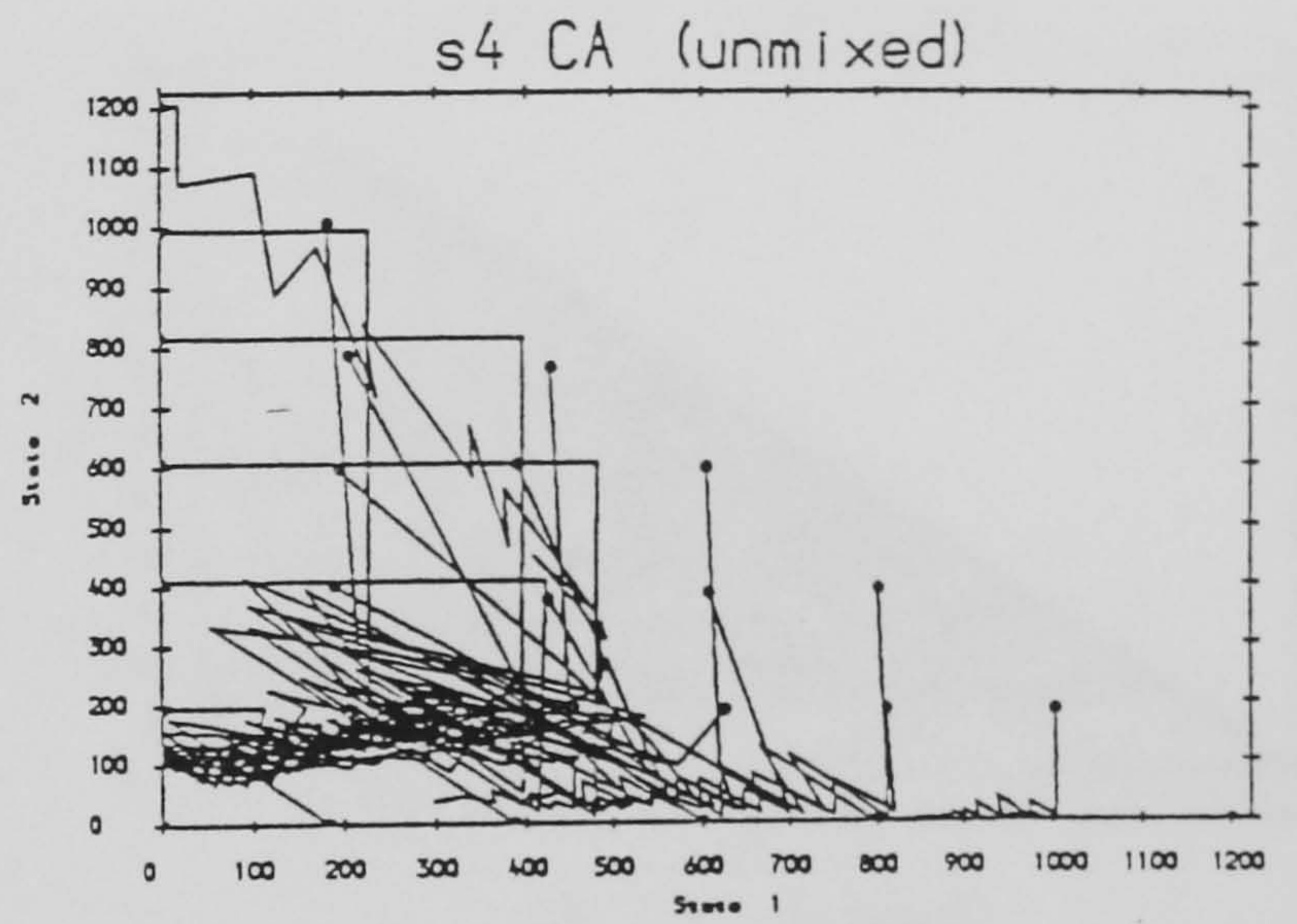
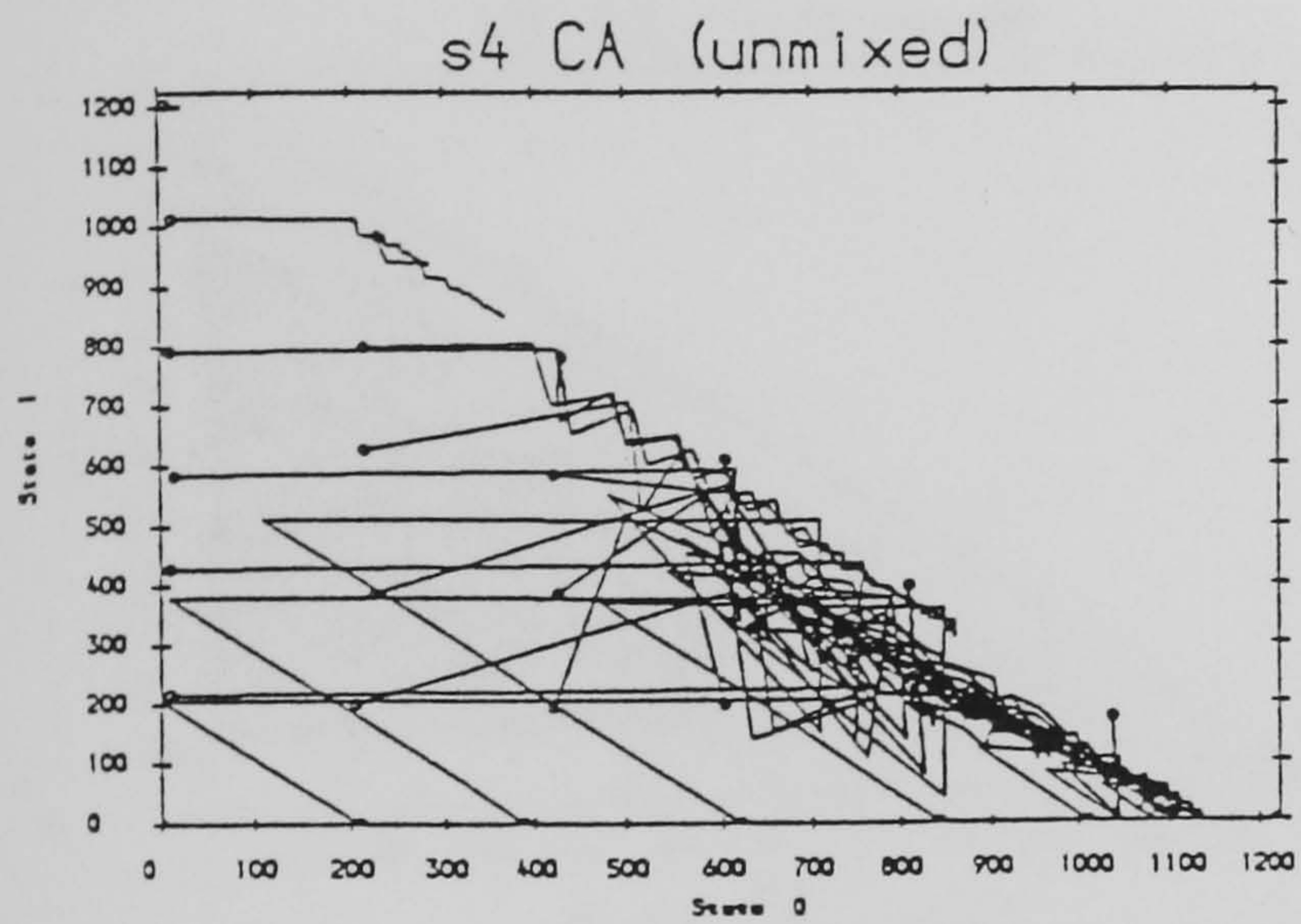




Figure 4.12: Threshold systems phase diagrams.  
(c) System 5: Deterministic.

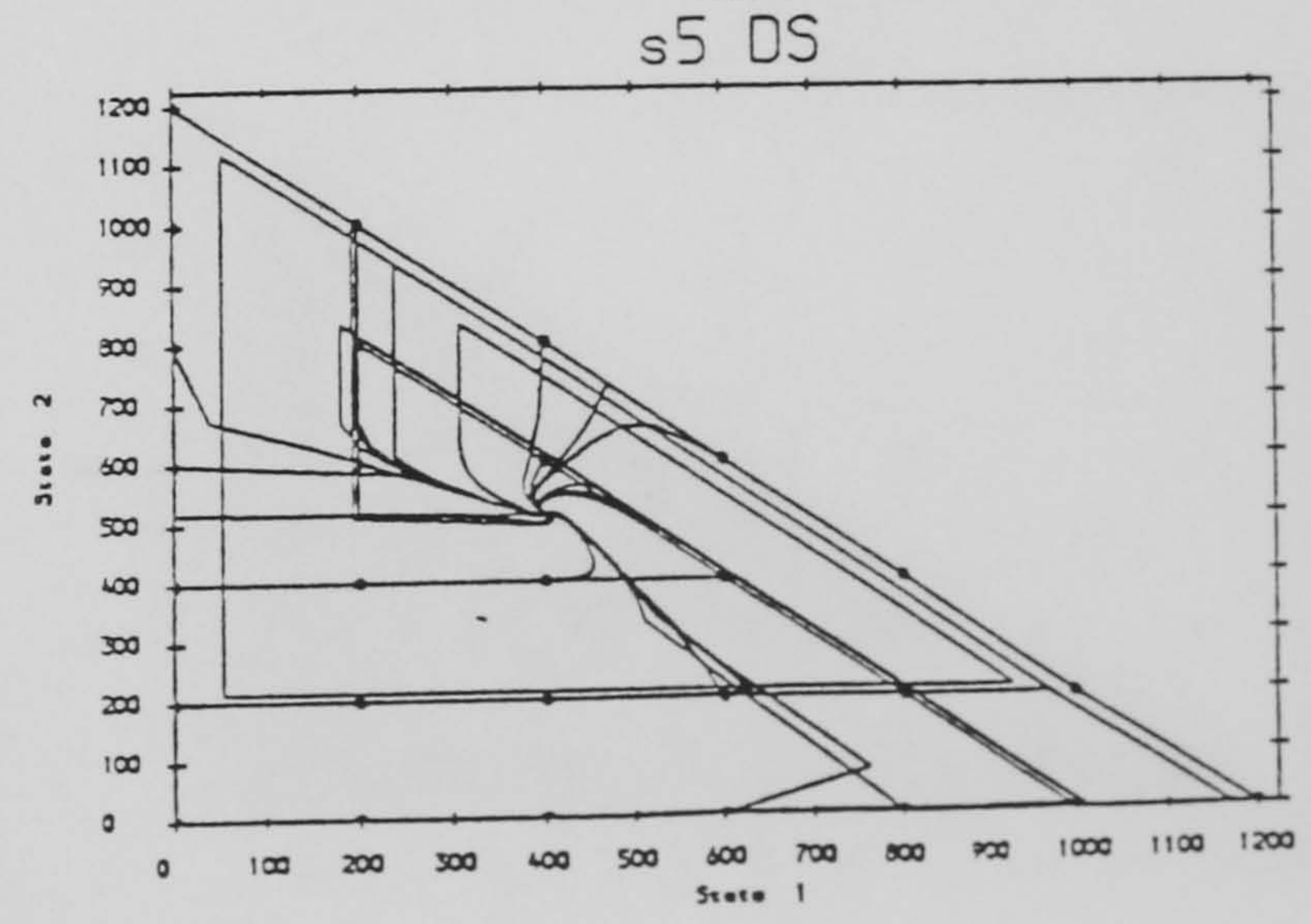
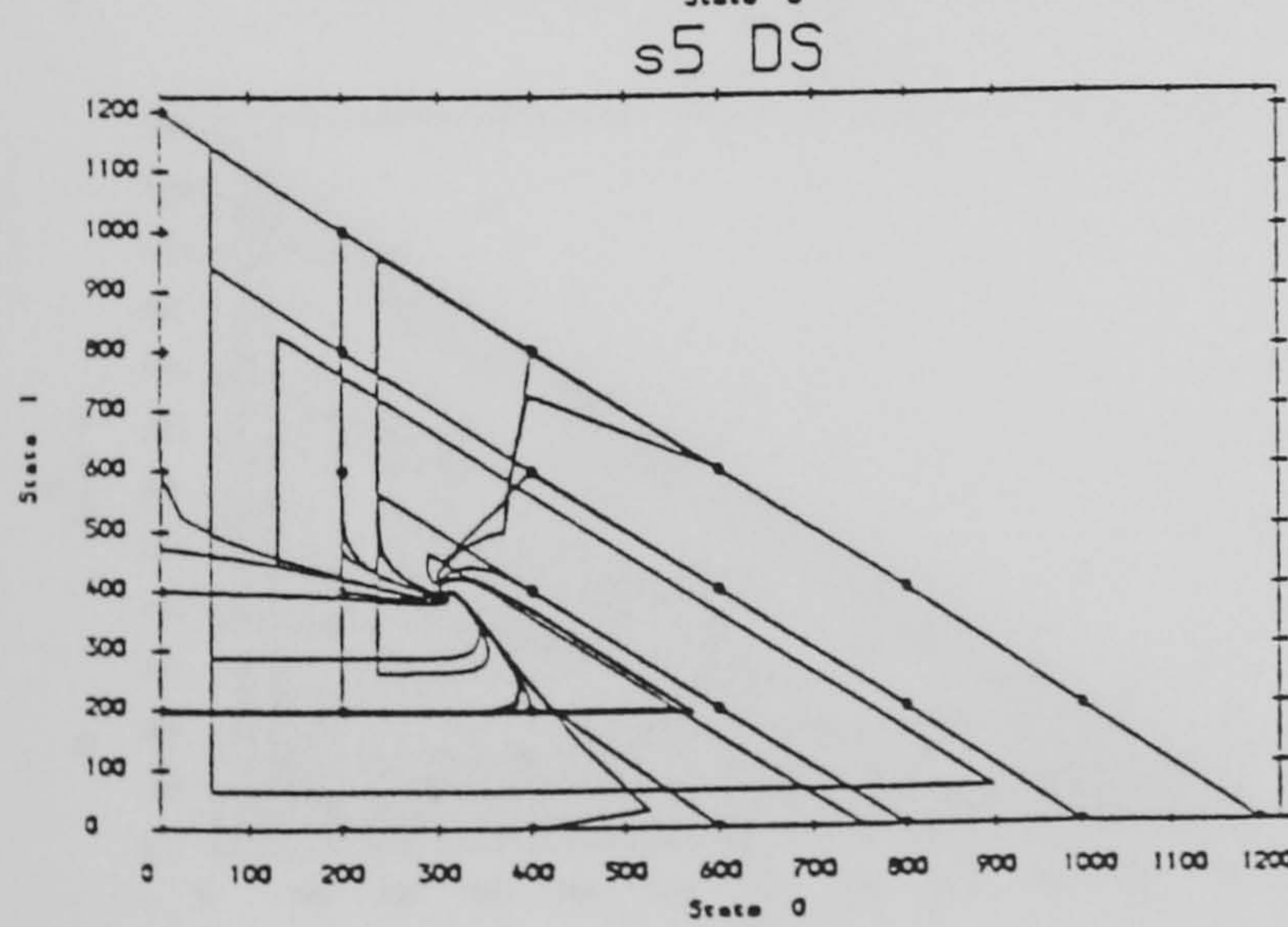
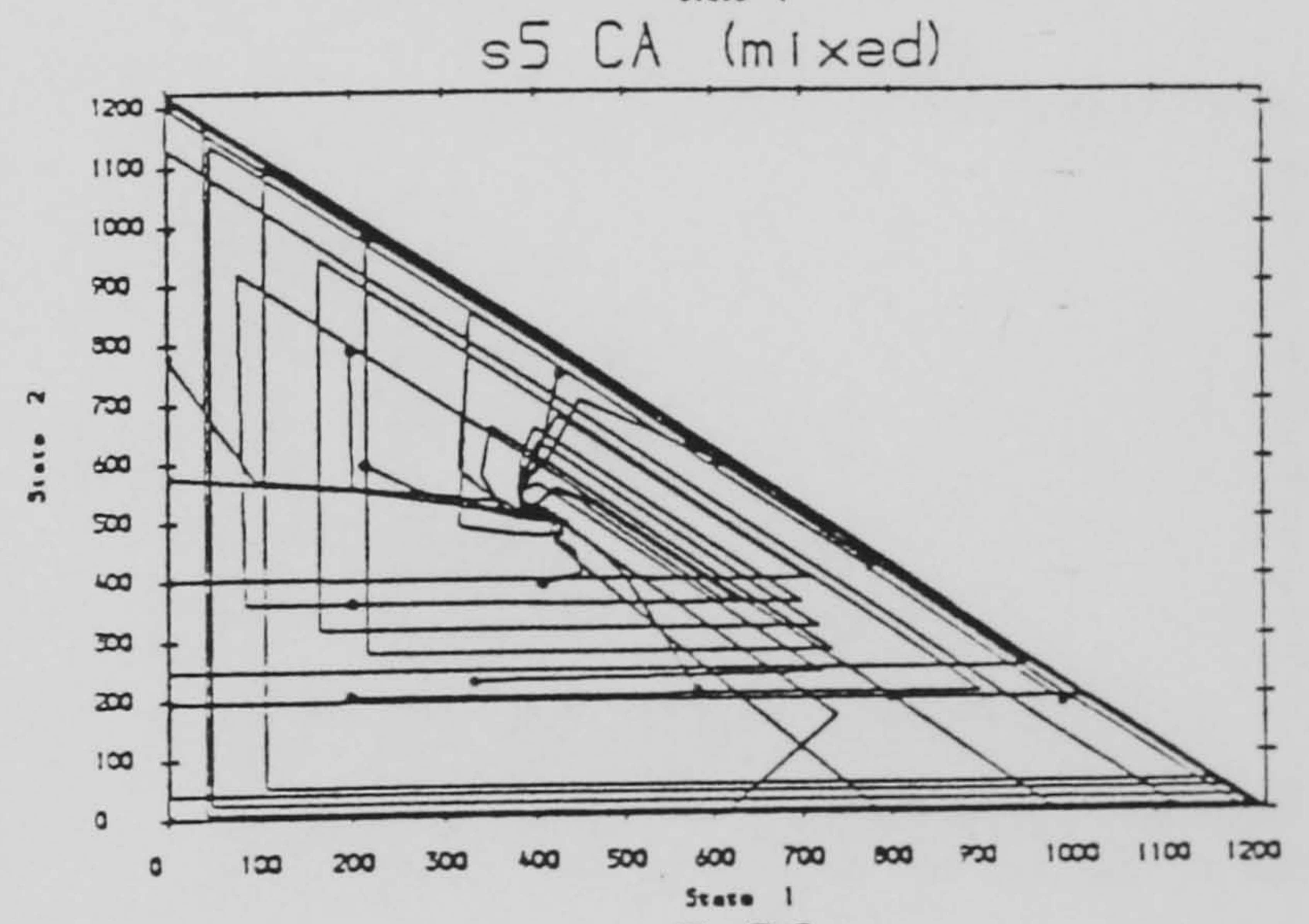
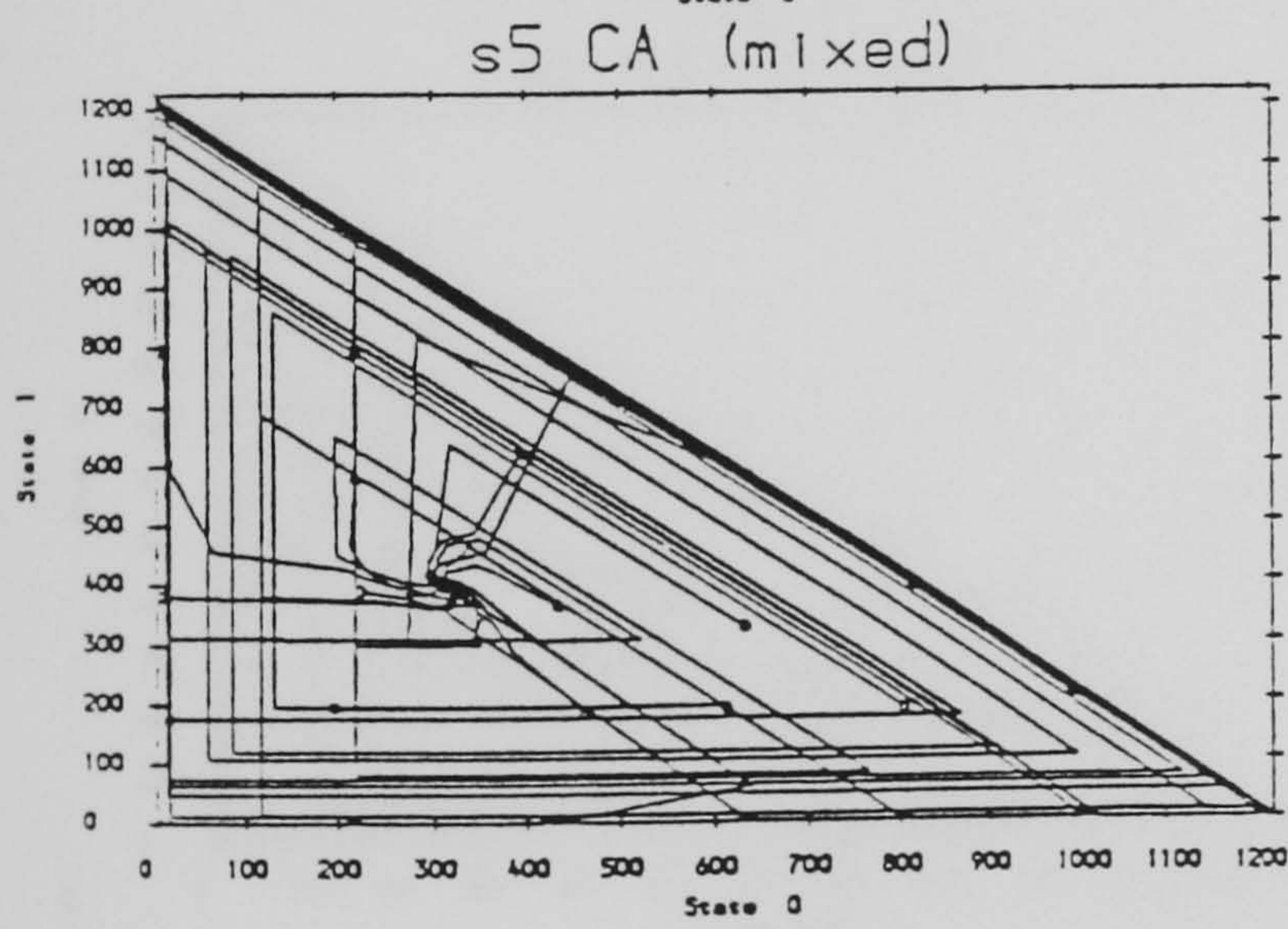
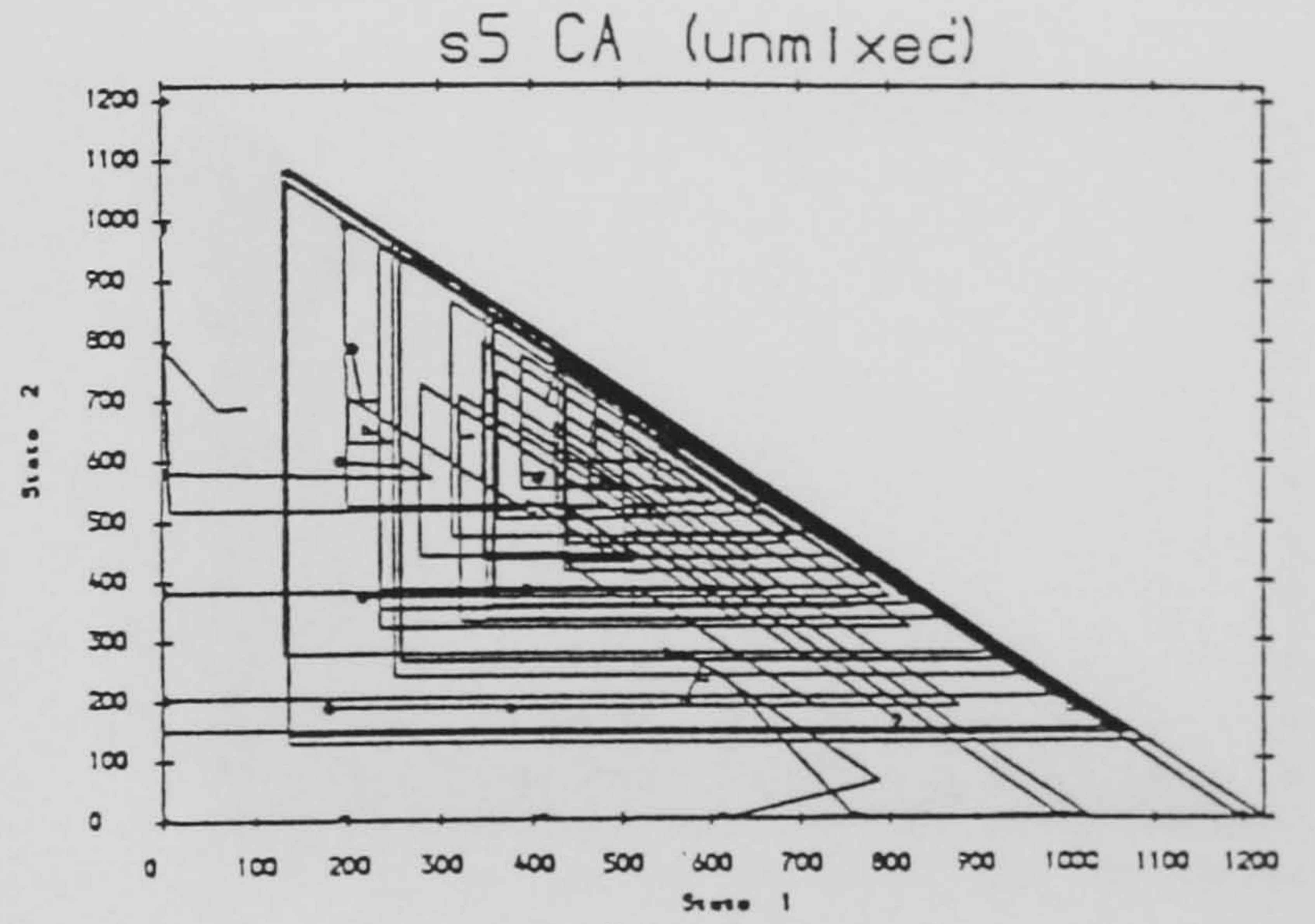
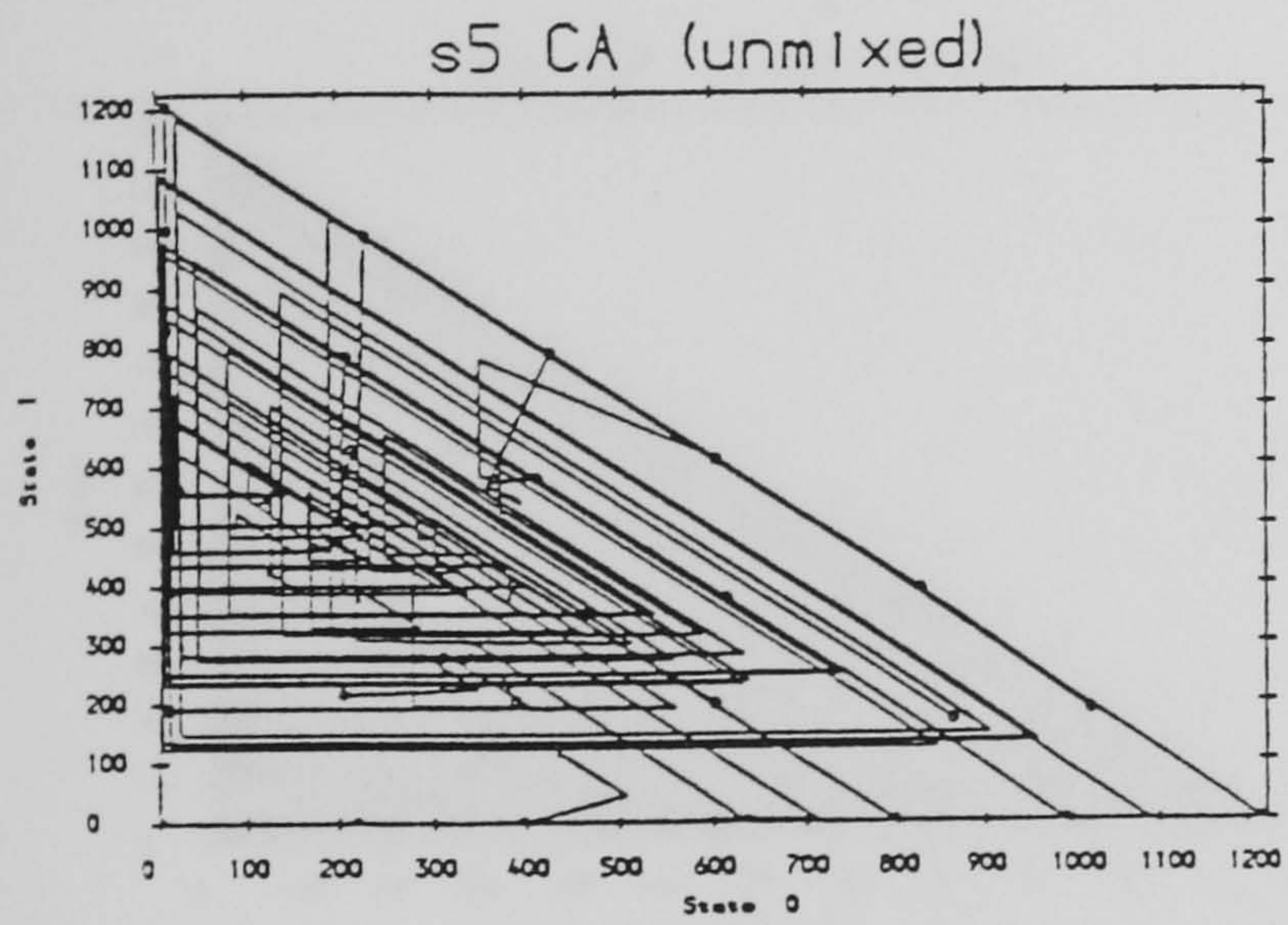




Figure 4.12: Threshold systems phase diagrams.  
(d) System 6: Deterministic.

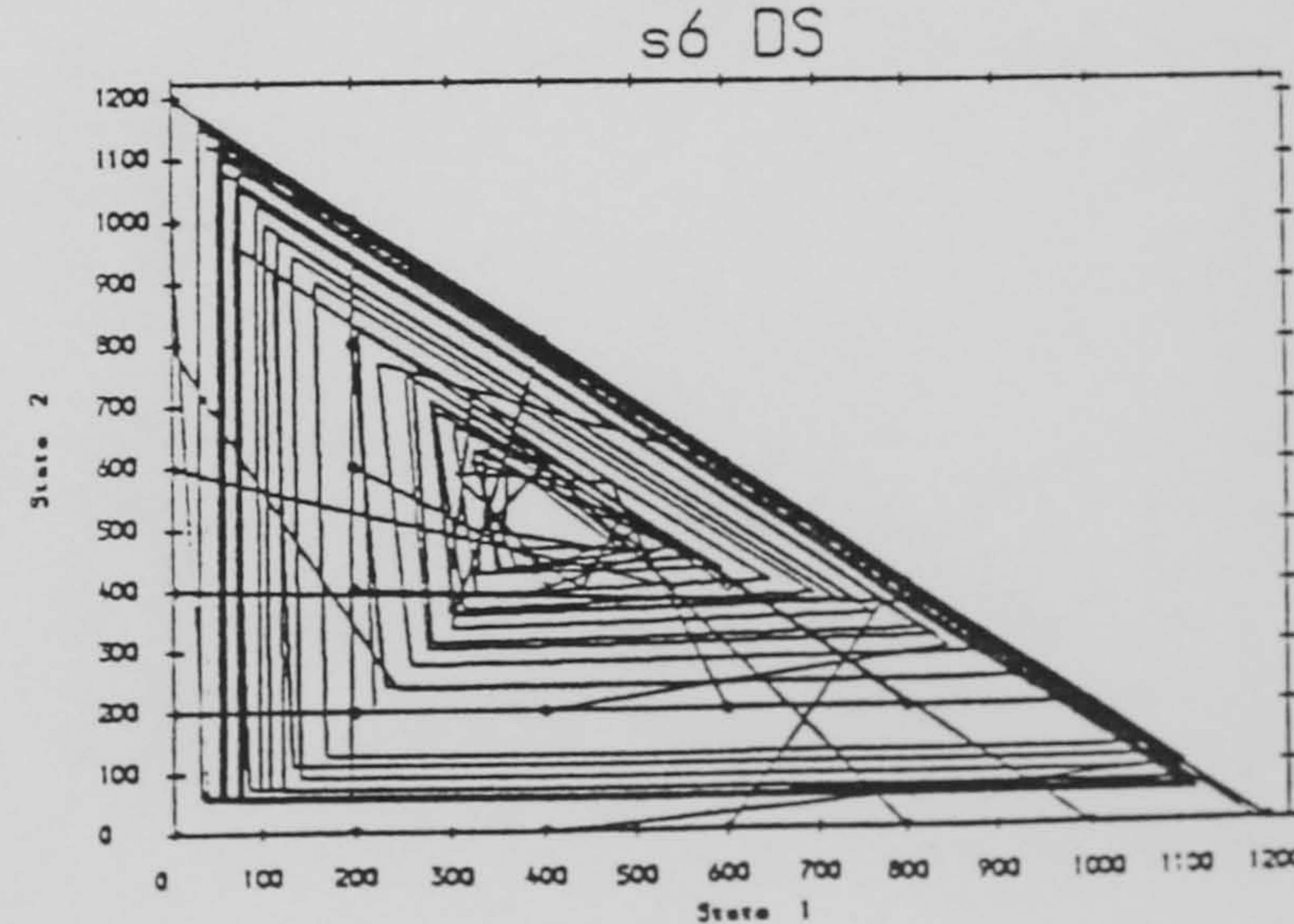
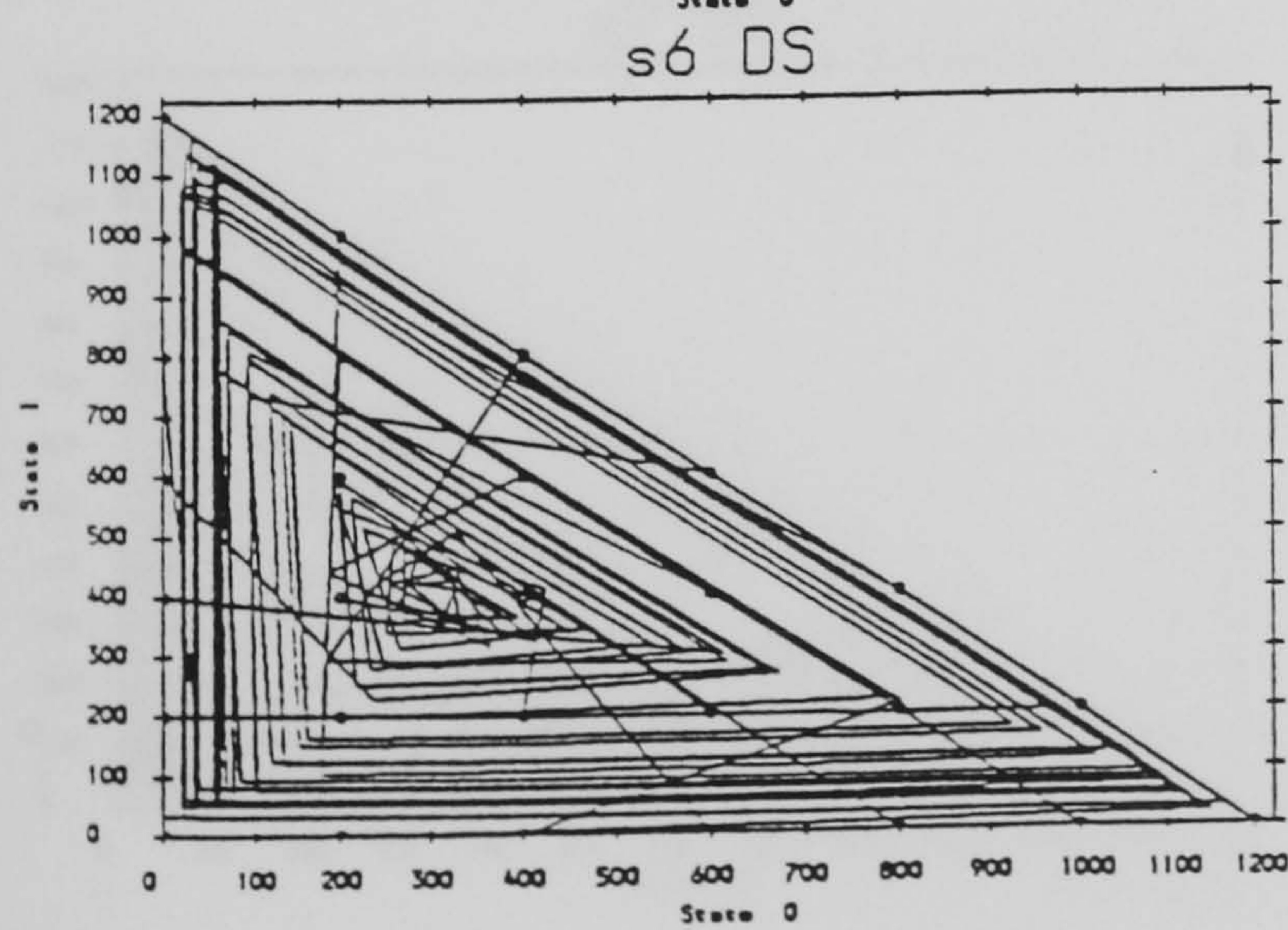
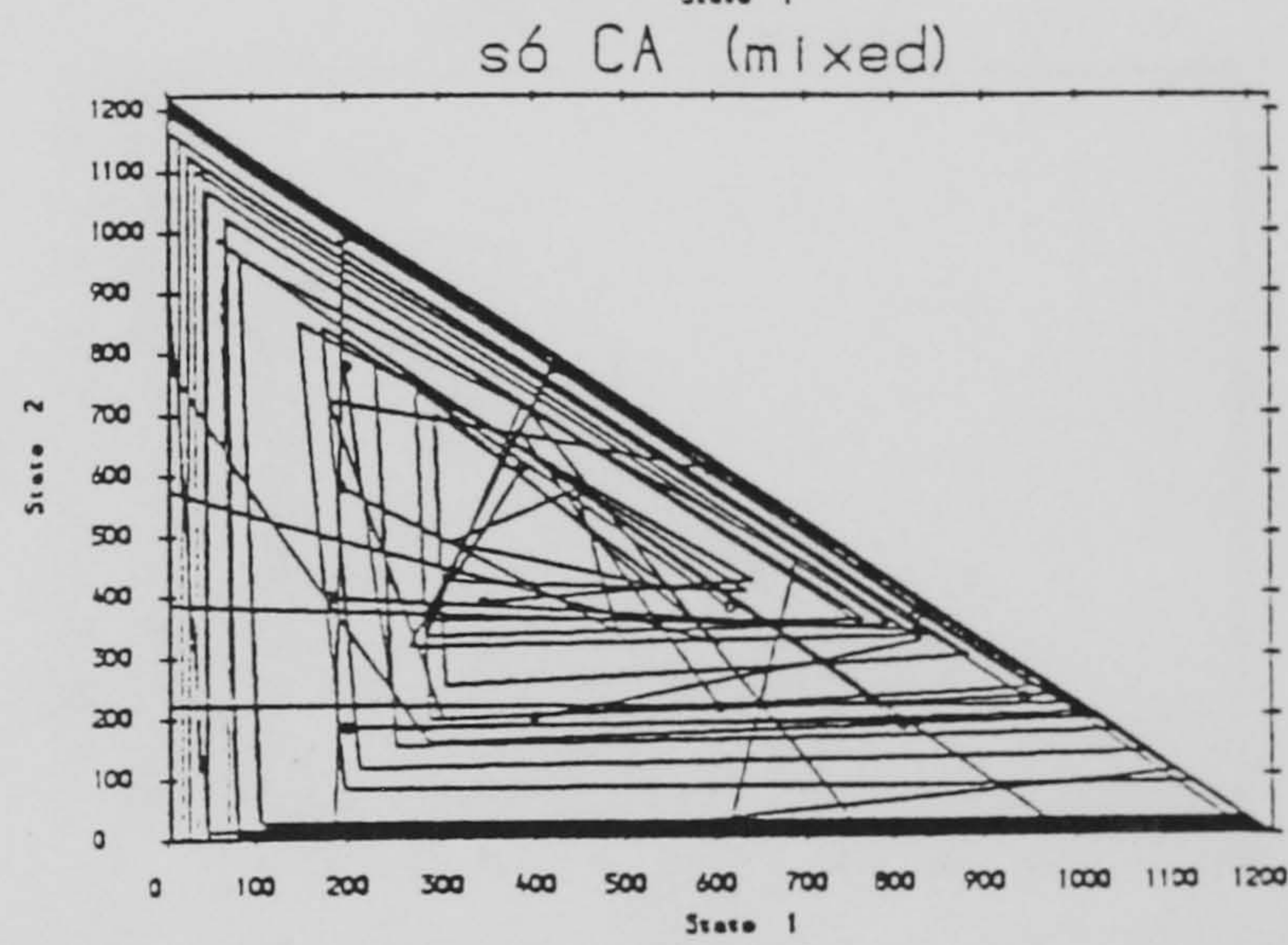
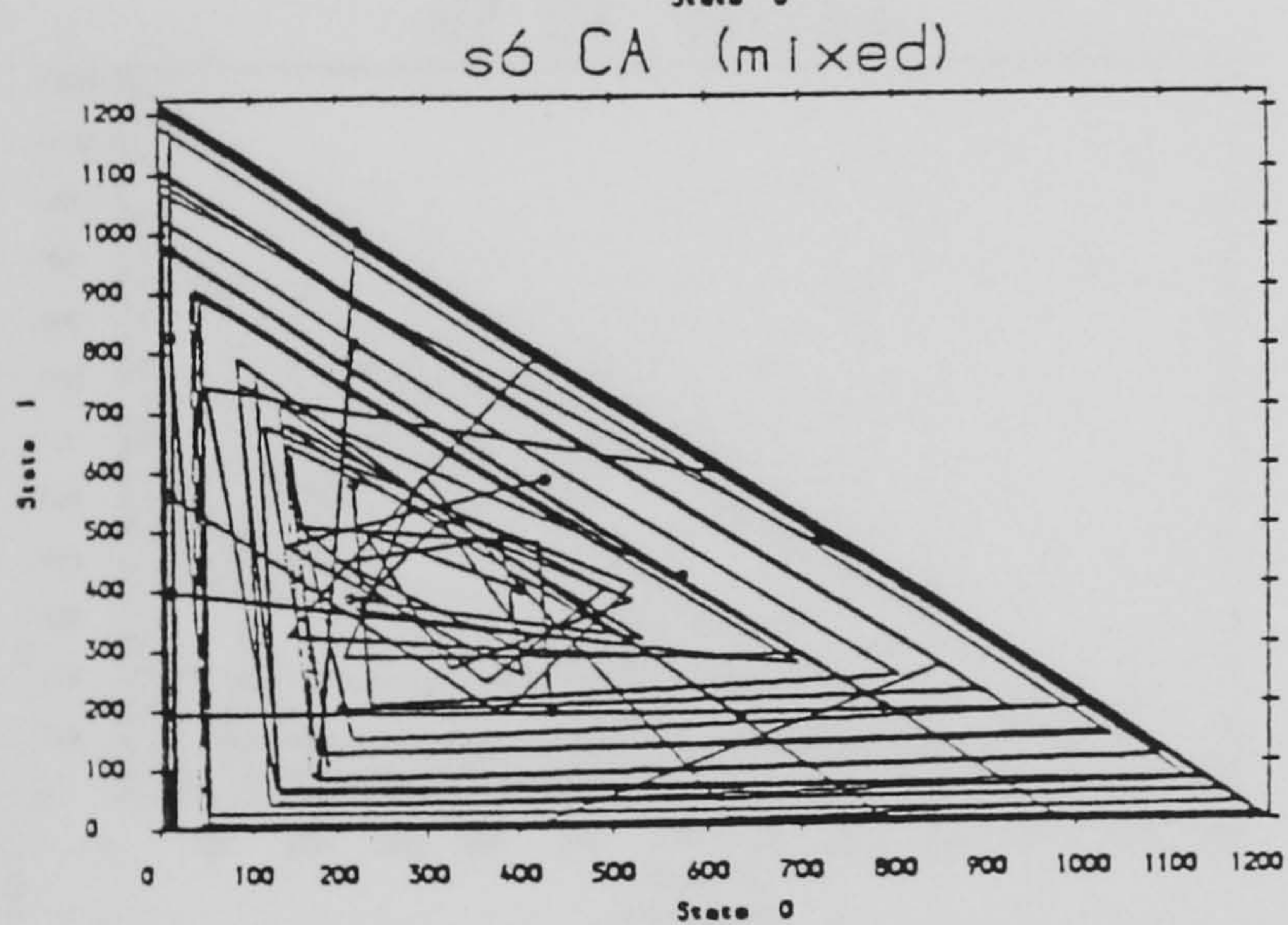
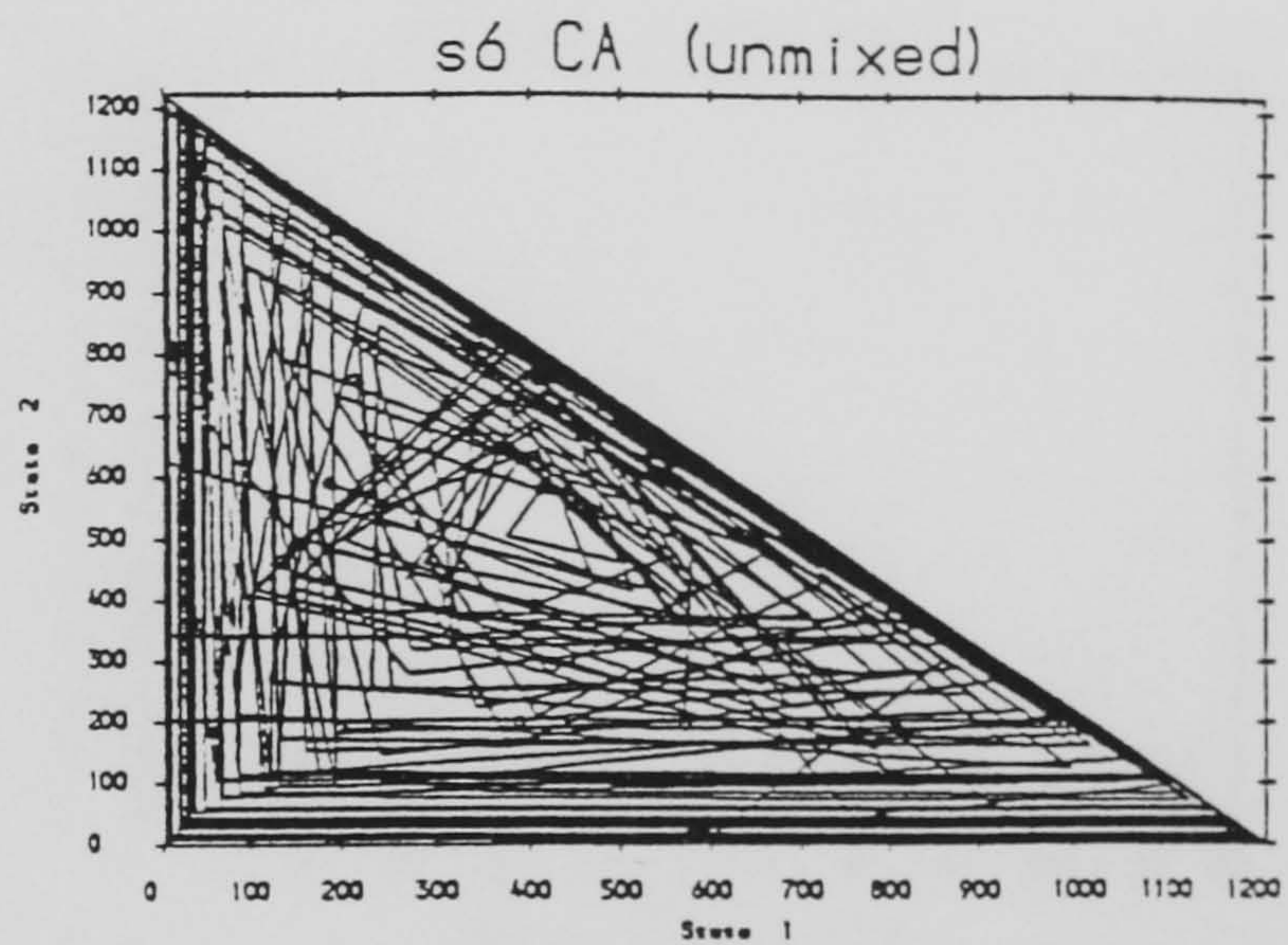
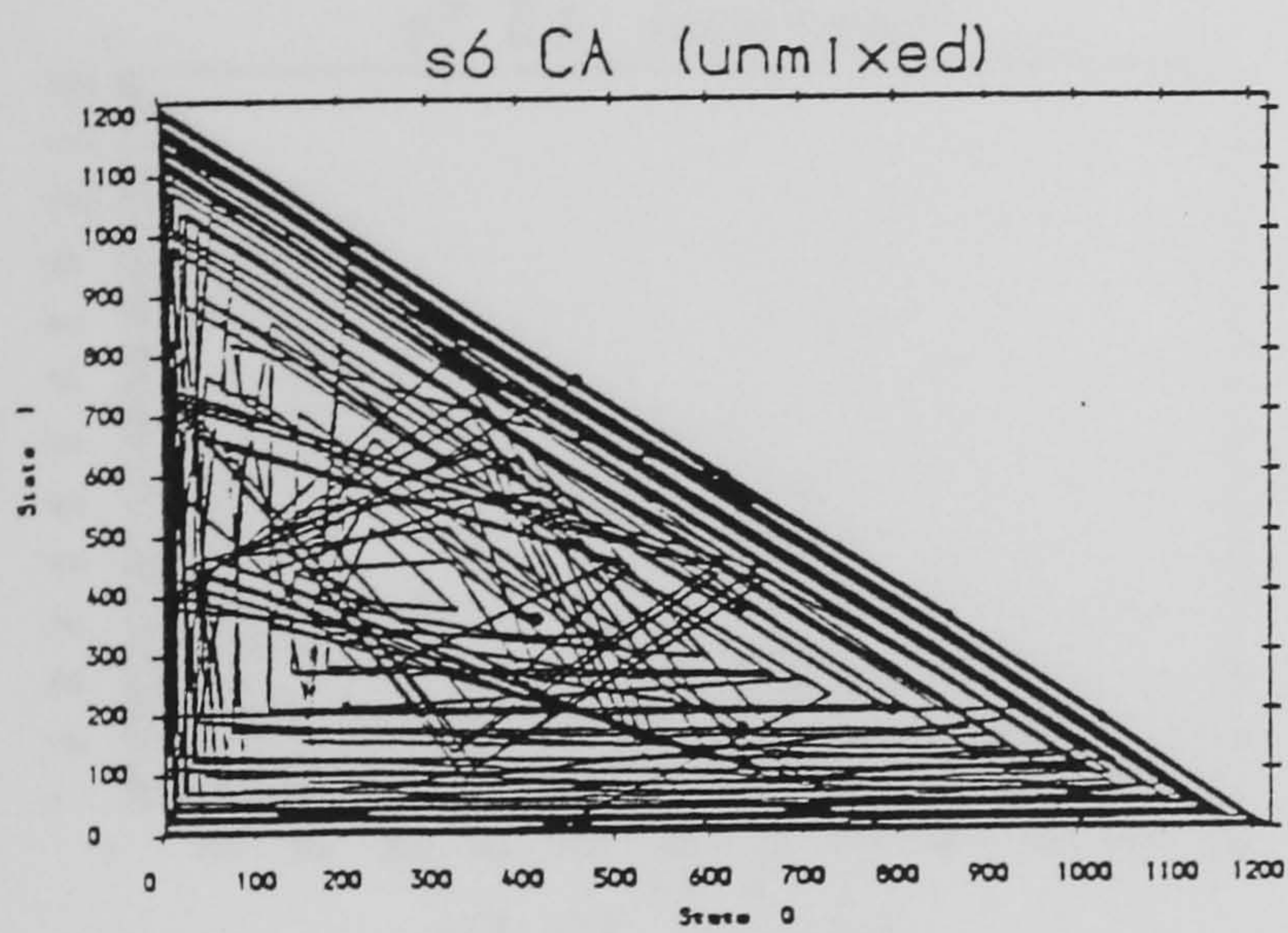




Figure 4.12: Threshold systems phase diagrams.  
(e) System 7: Deterministic.

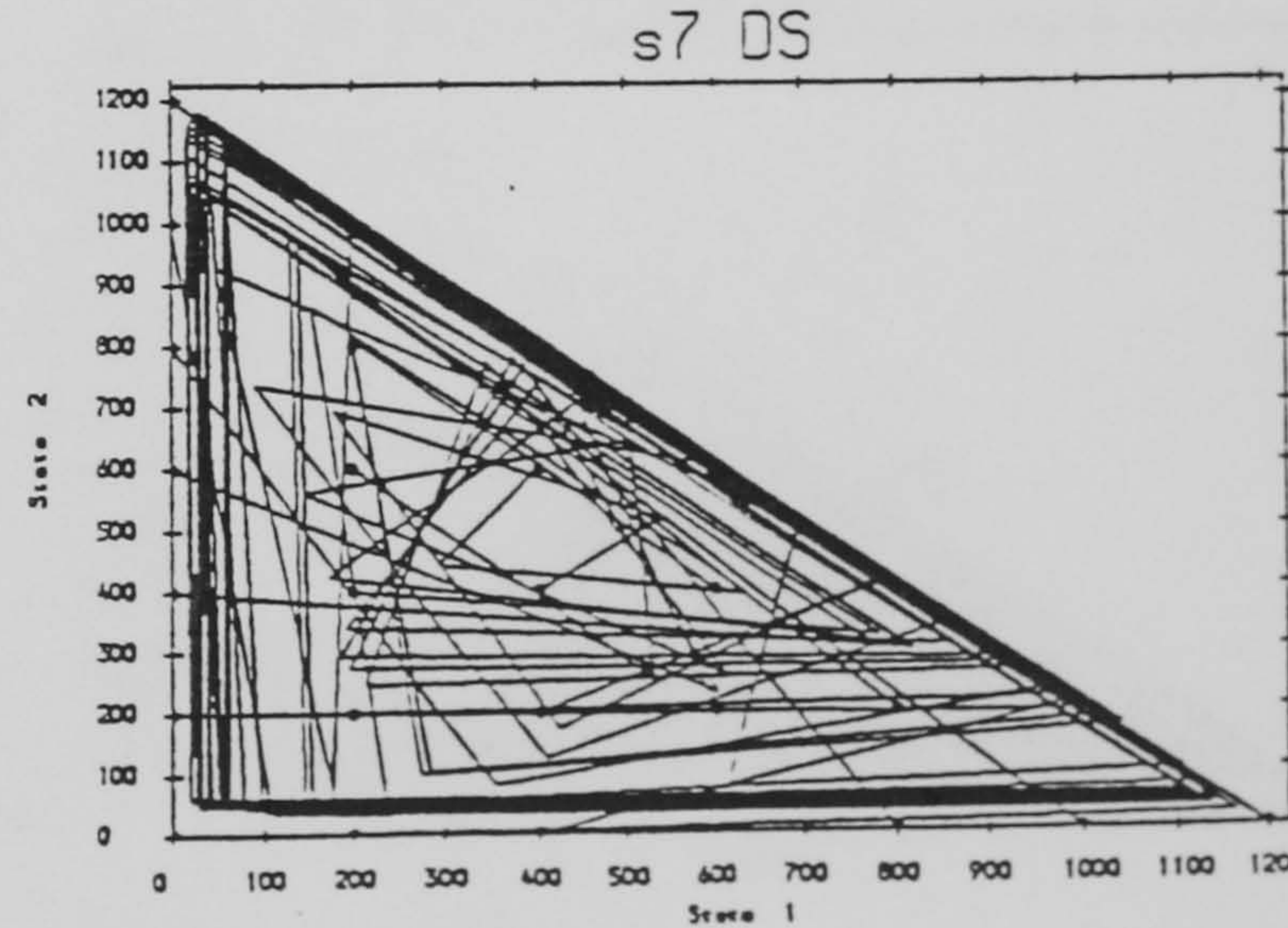
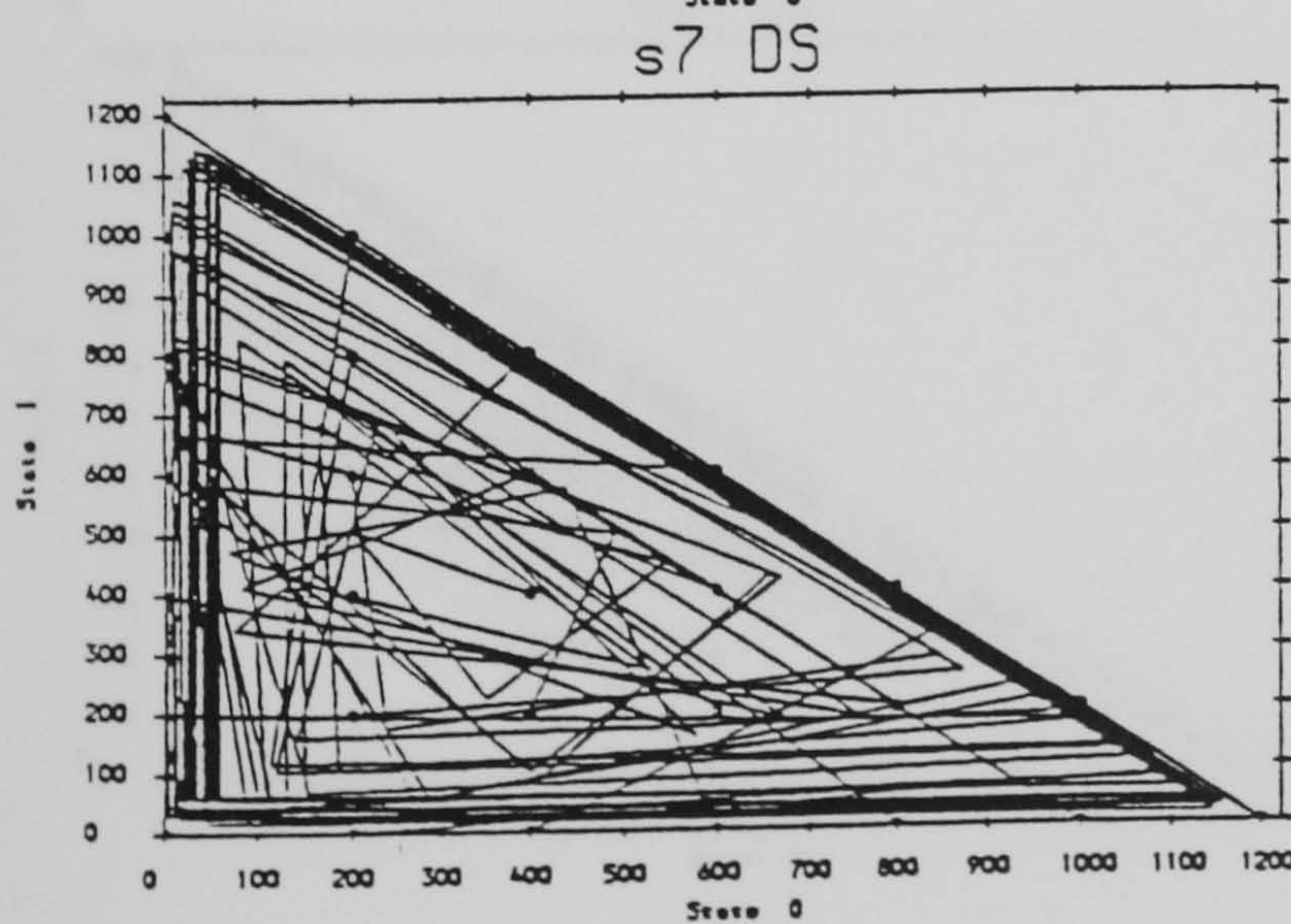
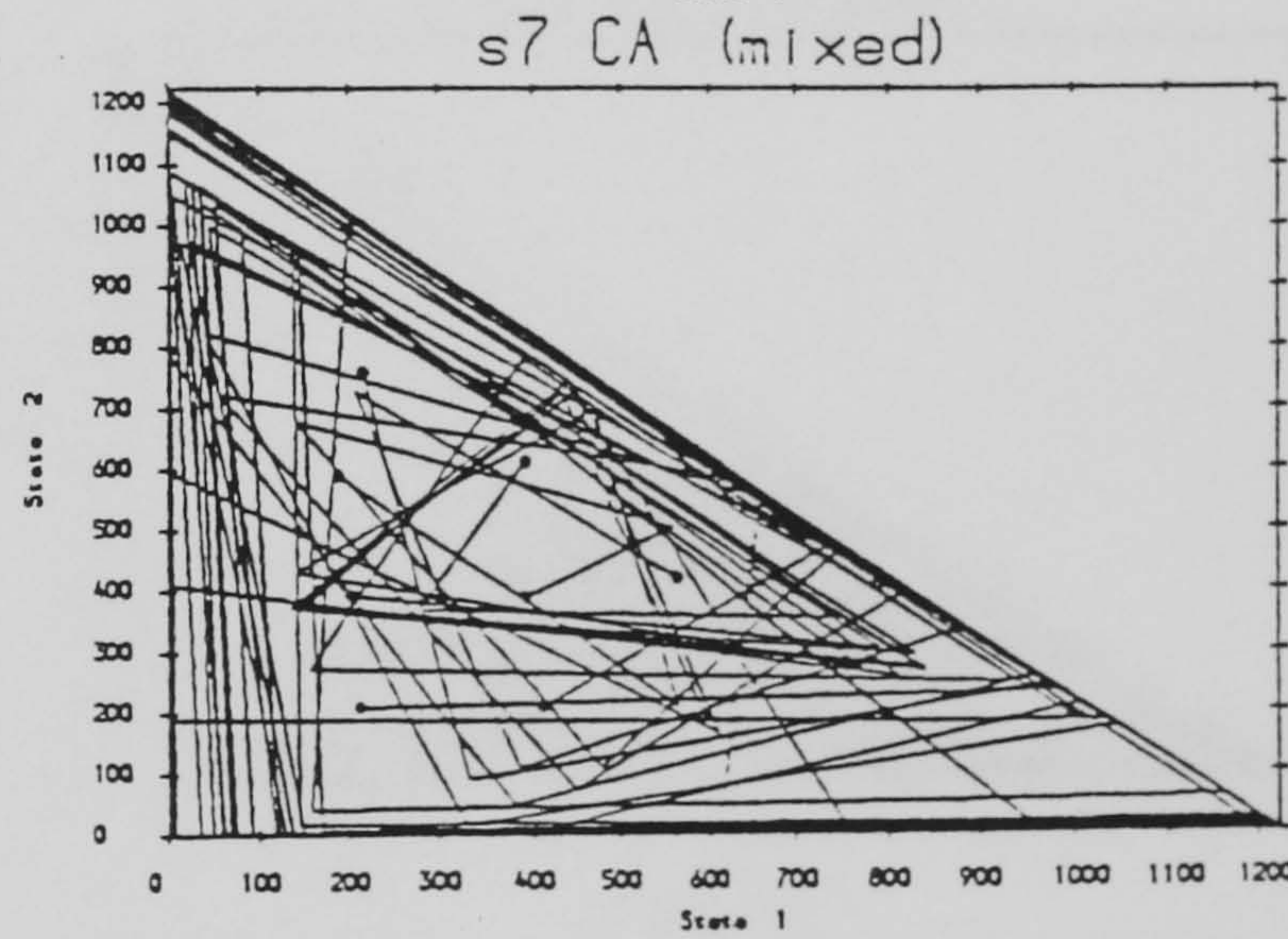
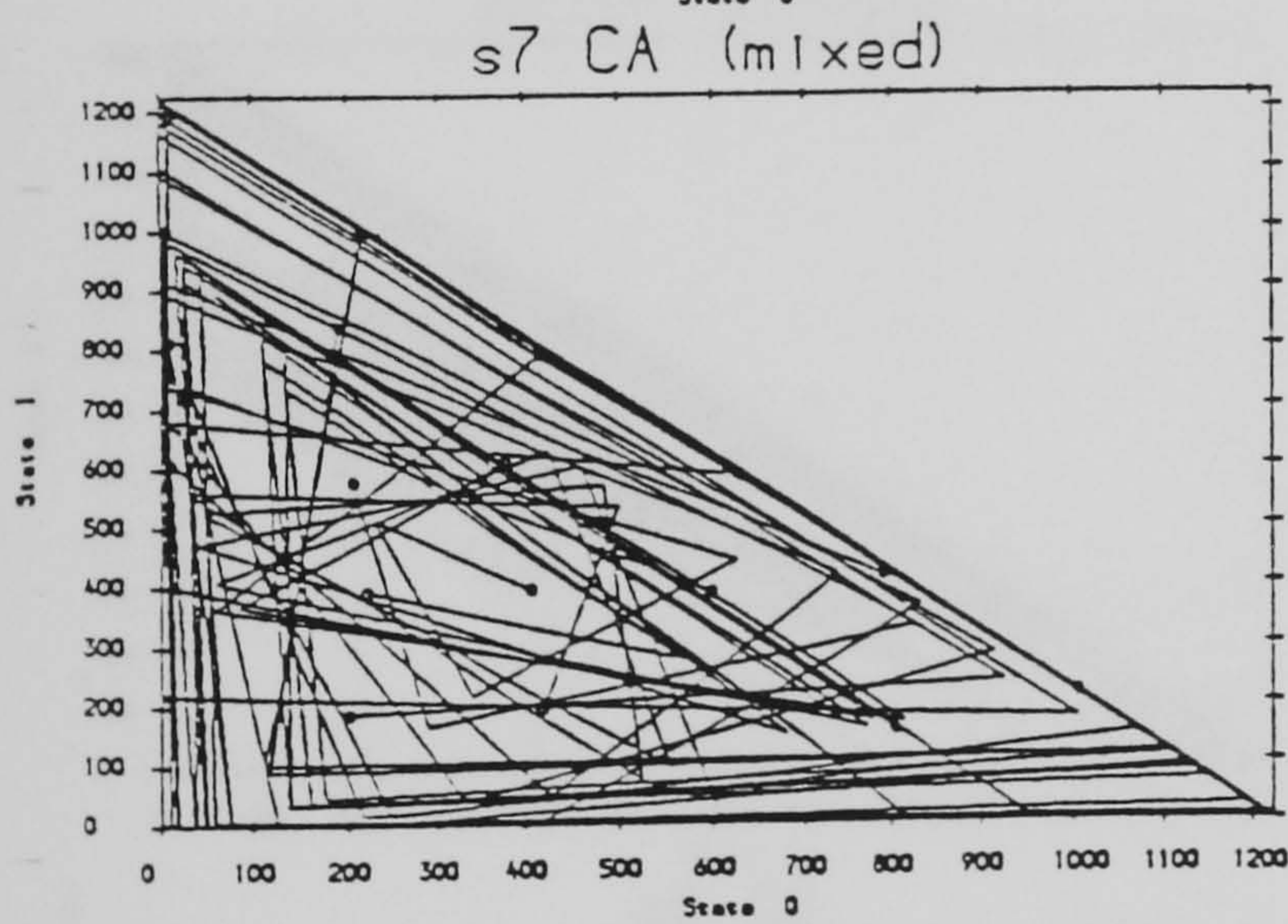
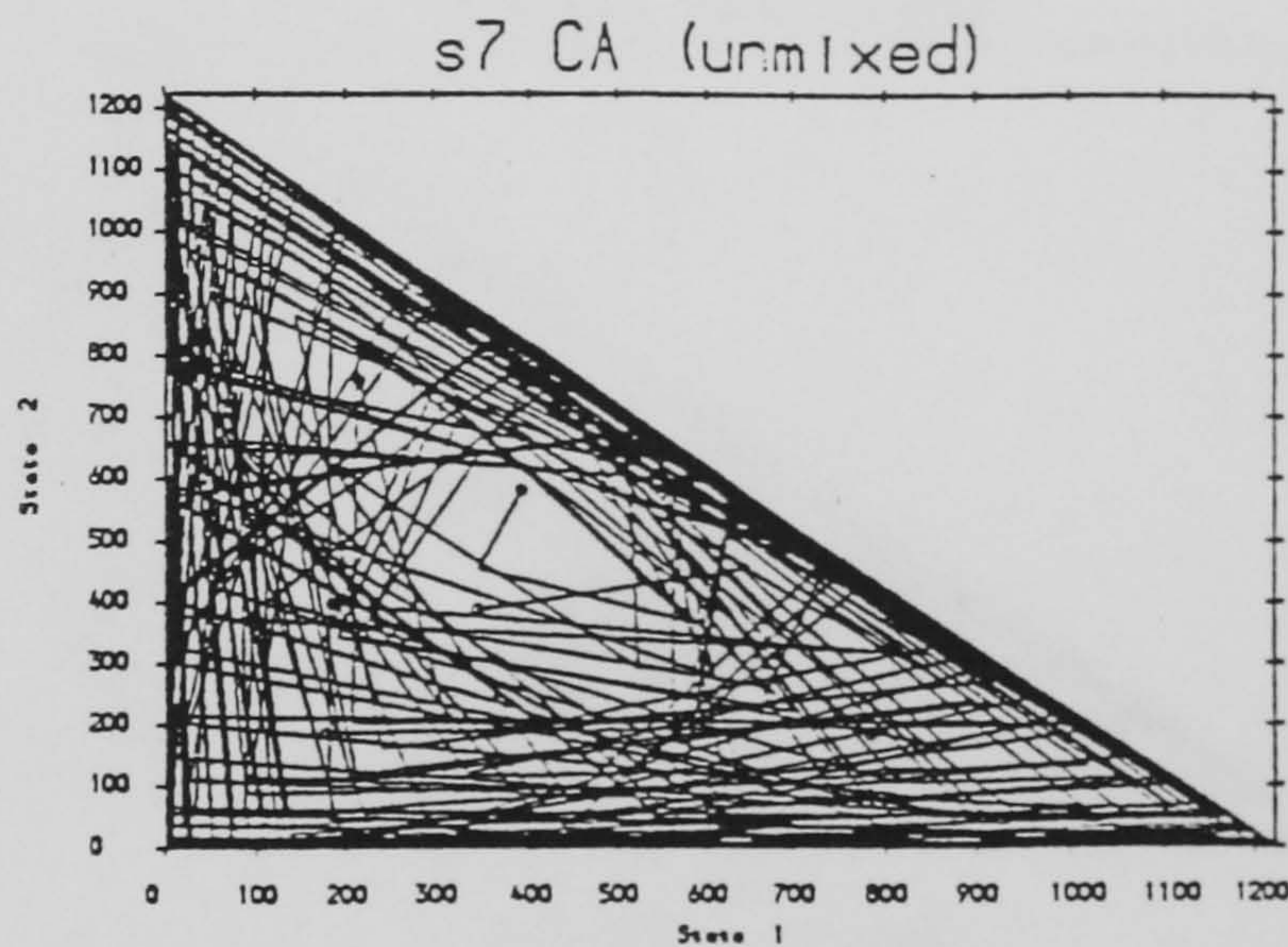
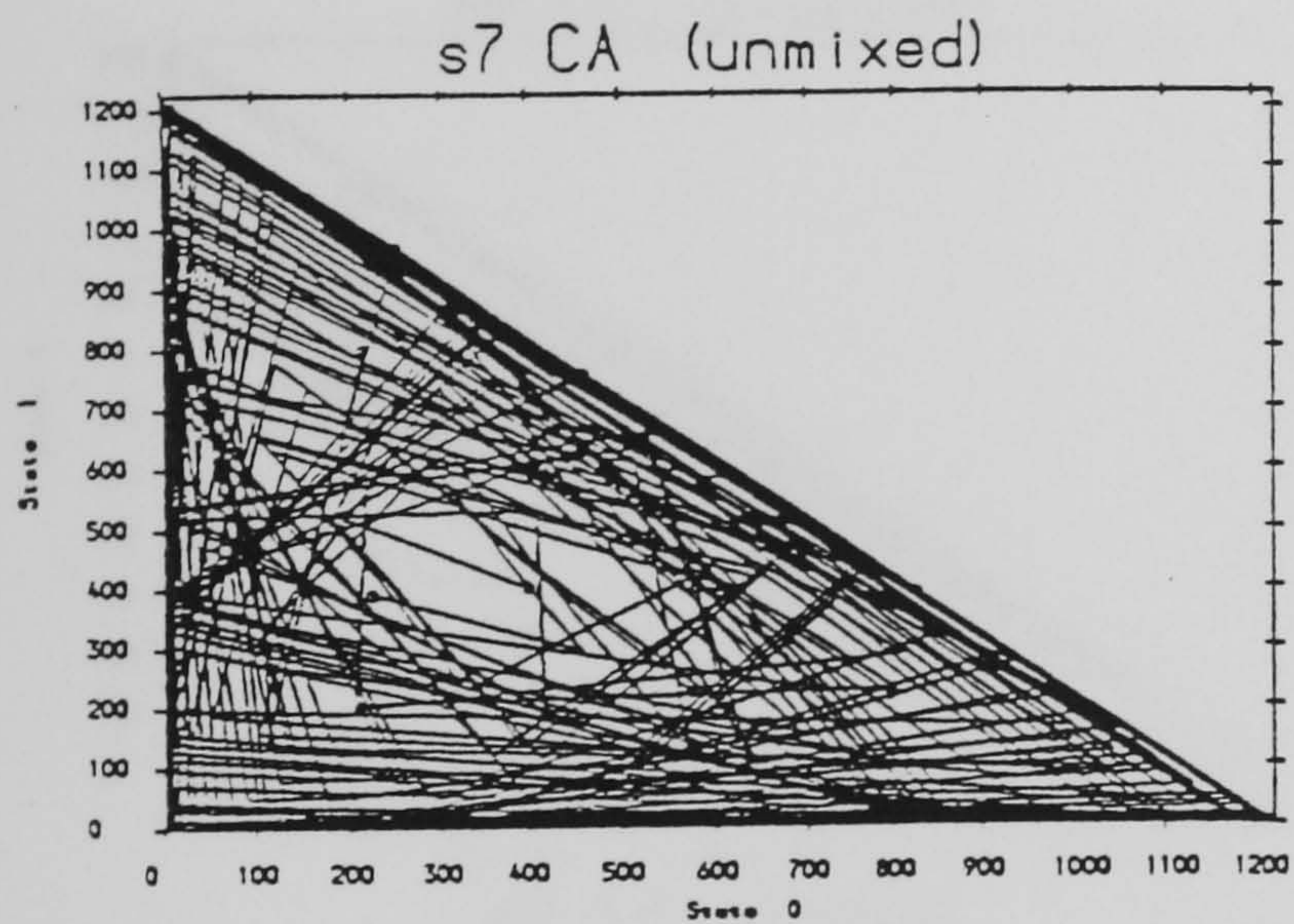




Figure 4.12: Threshold systems phase diagrams.  
(f) System 8: Deterministic.

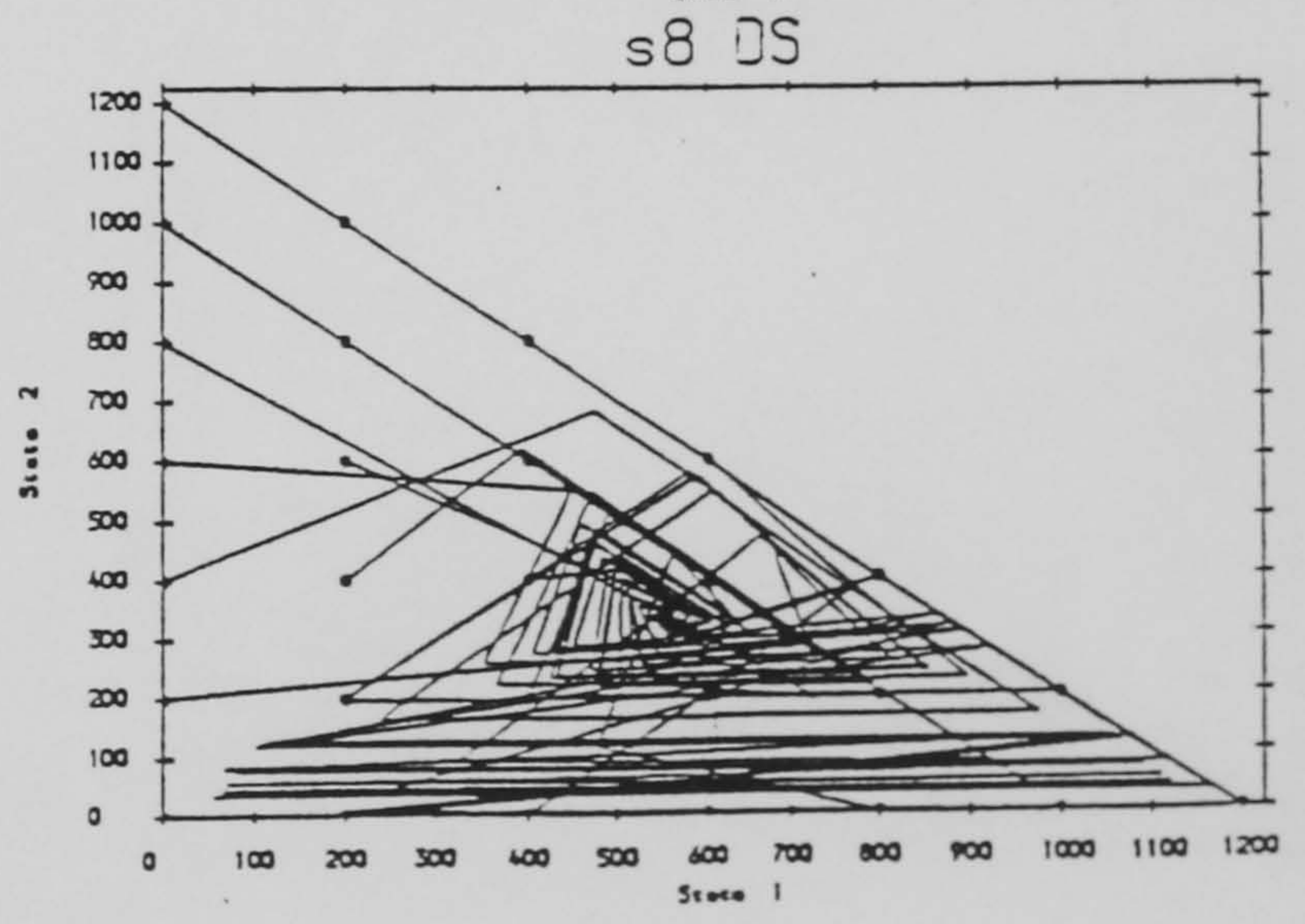
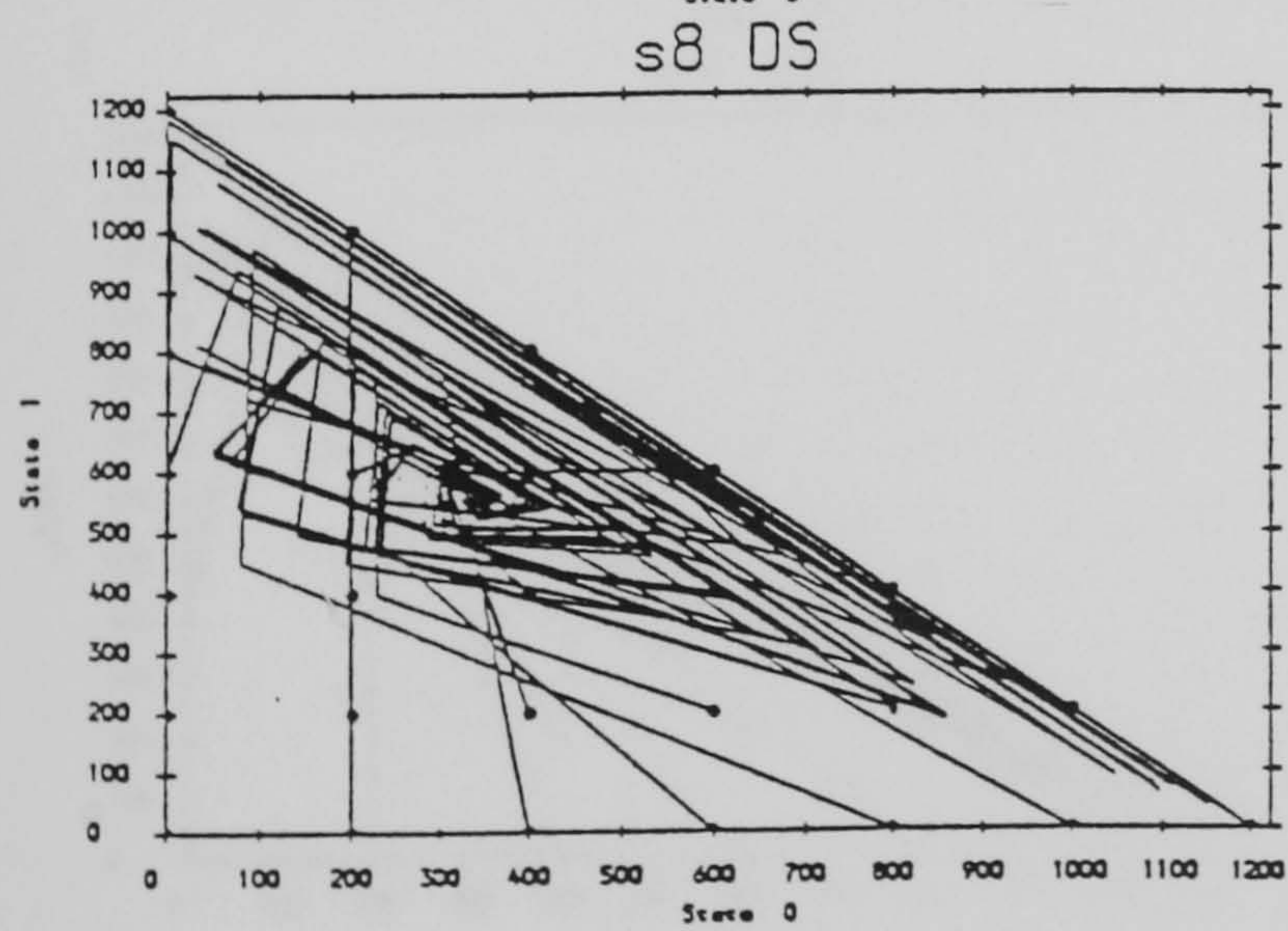
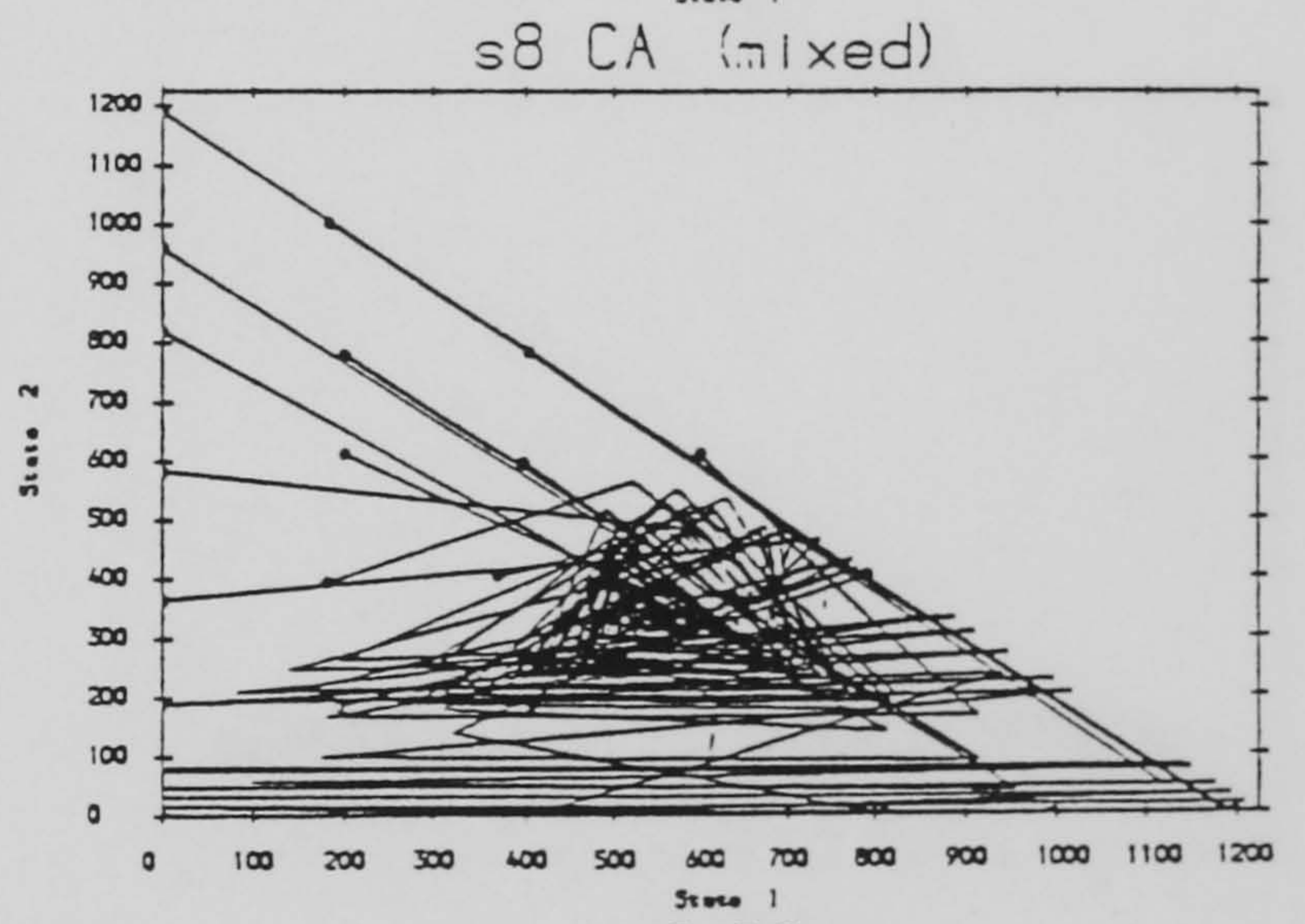
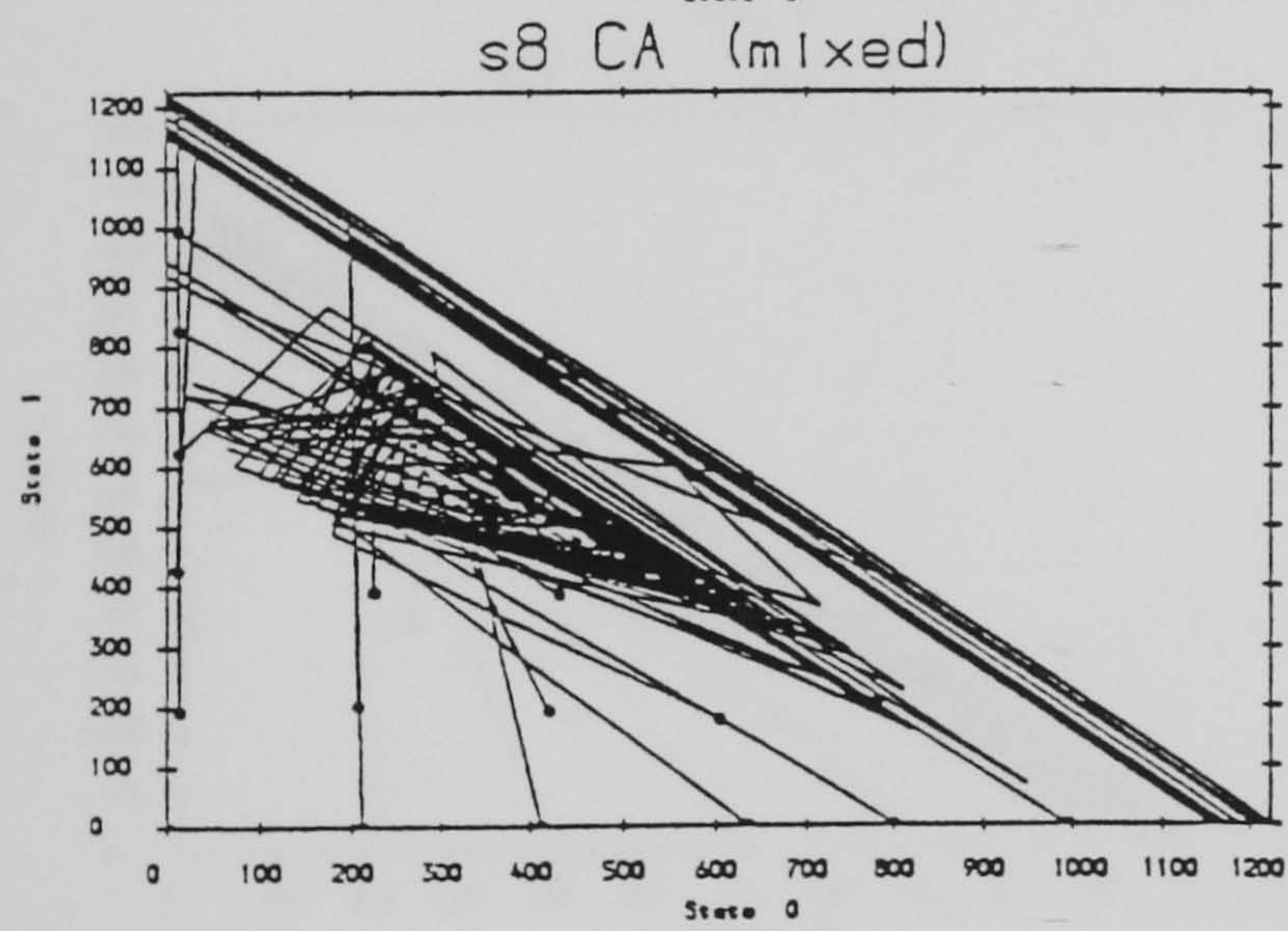
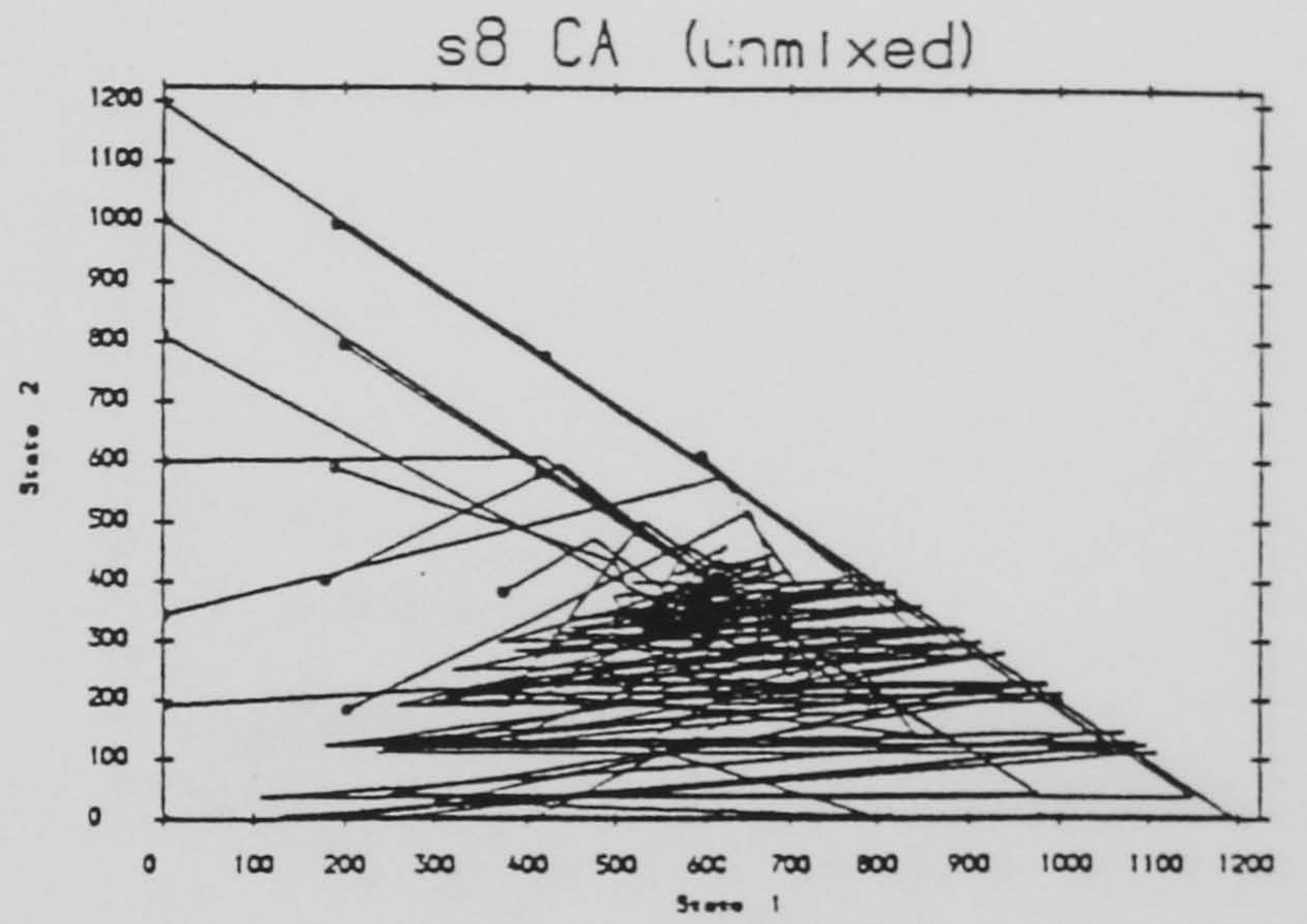
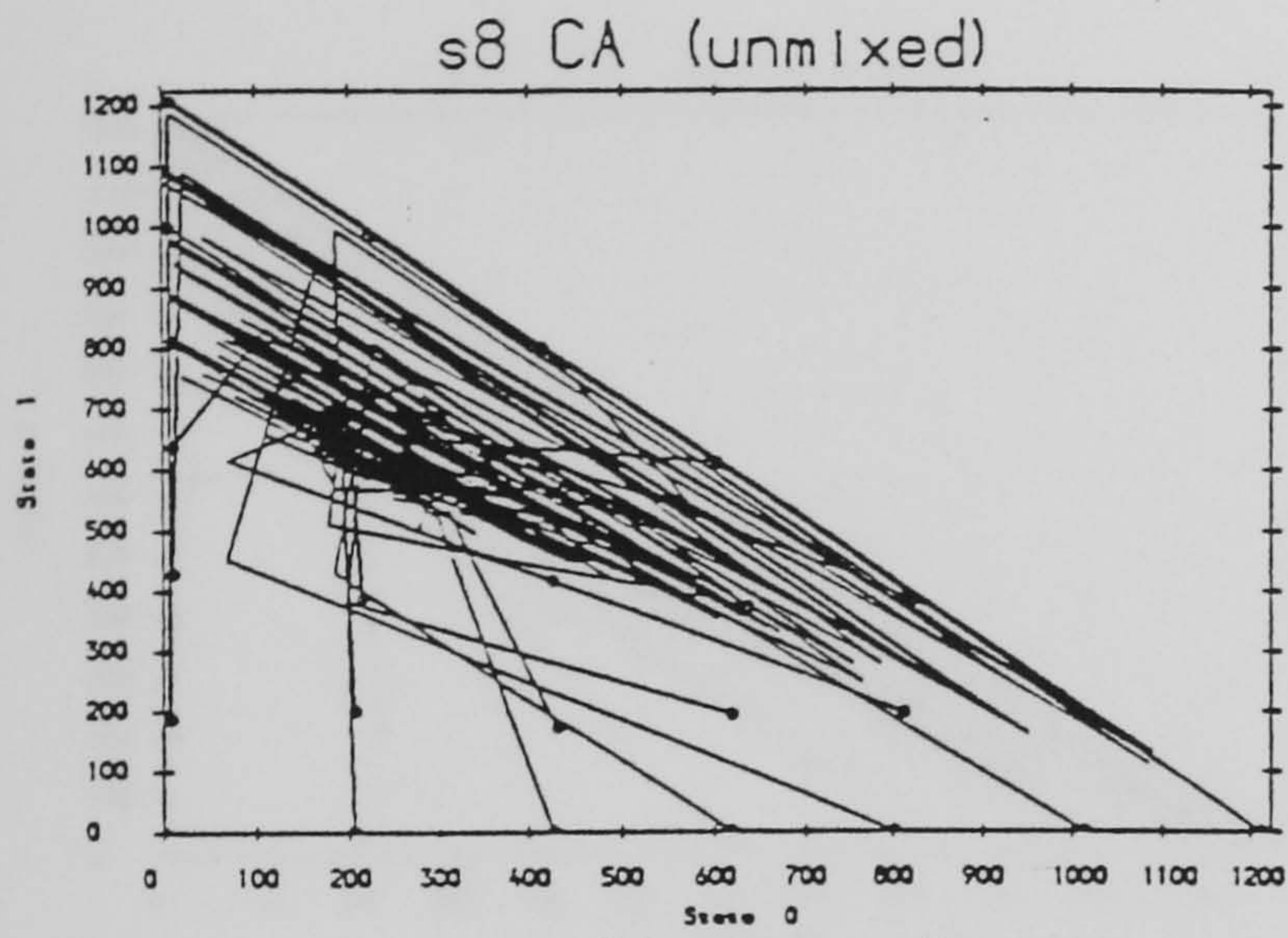




Figure 4.12: Threshold systems phase diagrams.  
(g) System 9: Deterministic.

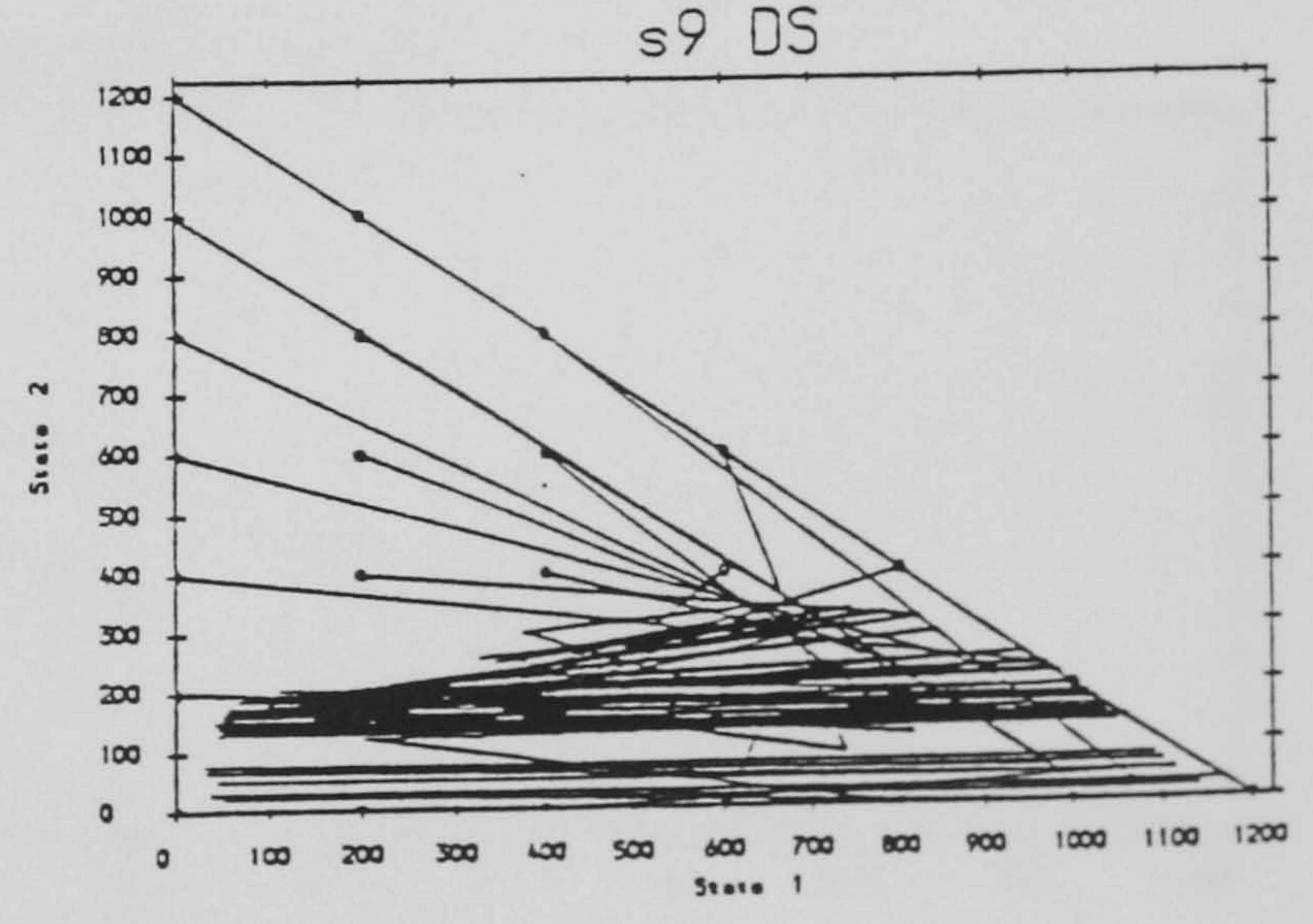
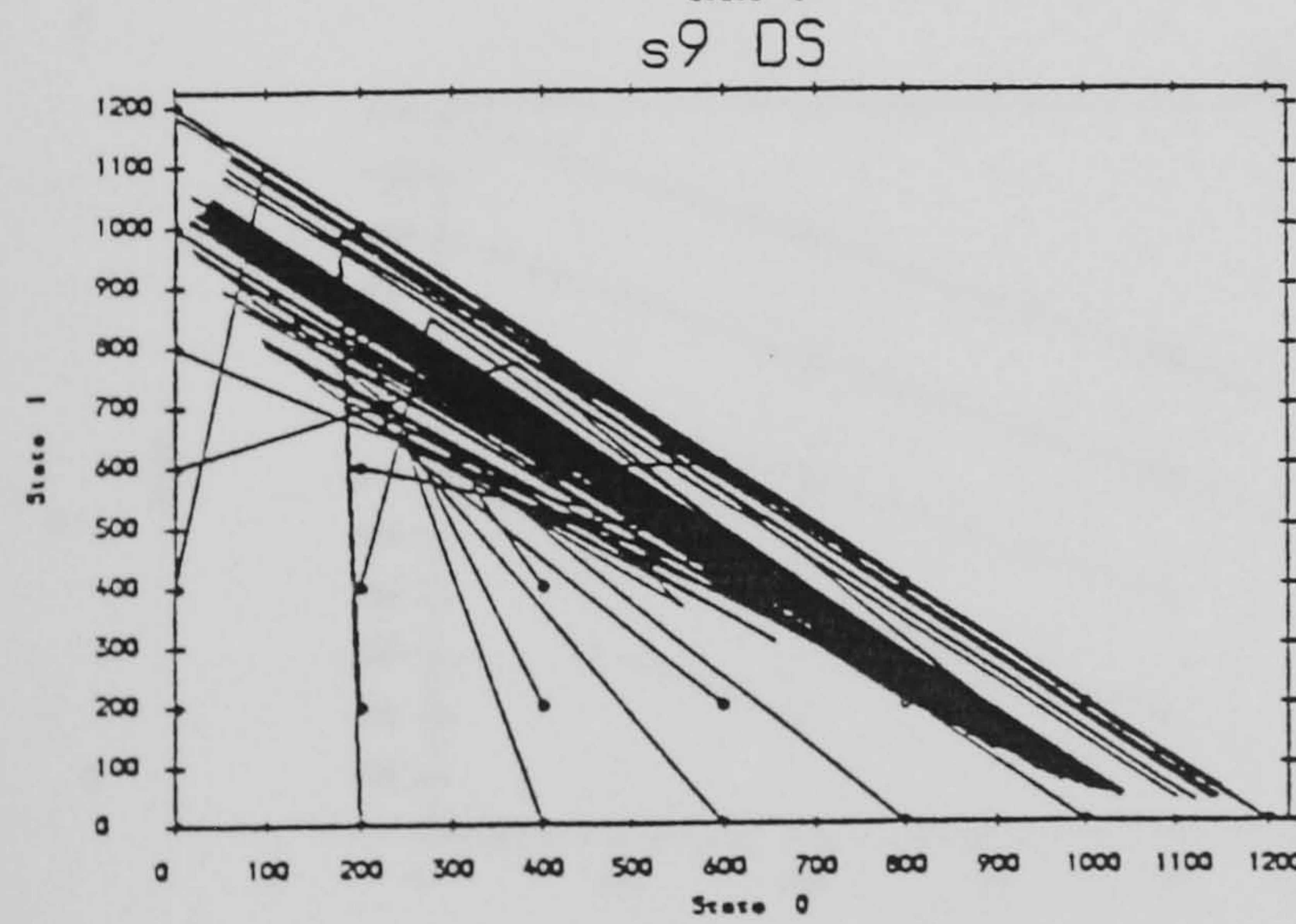
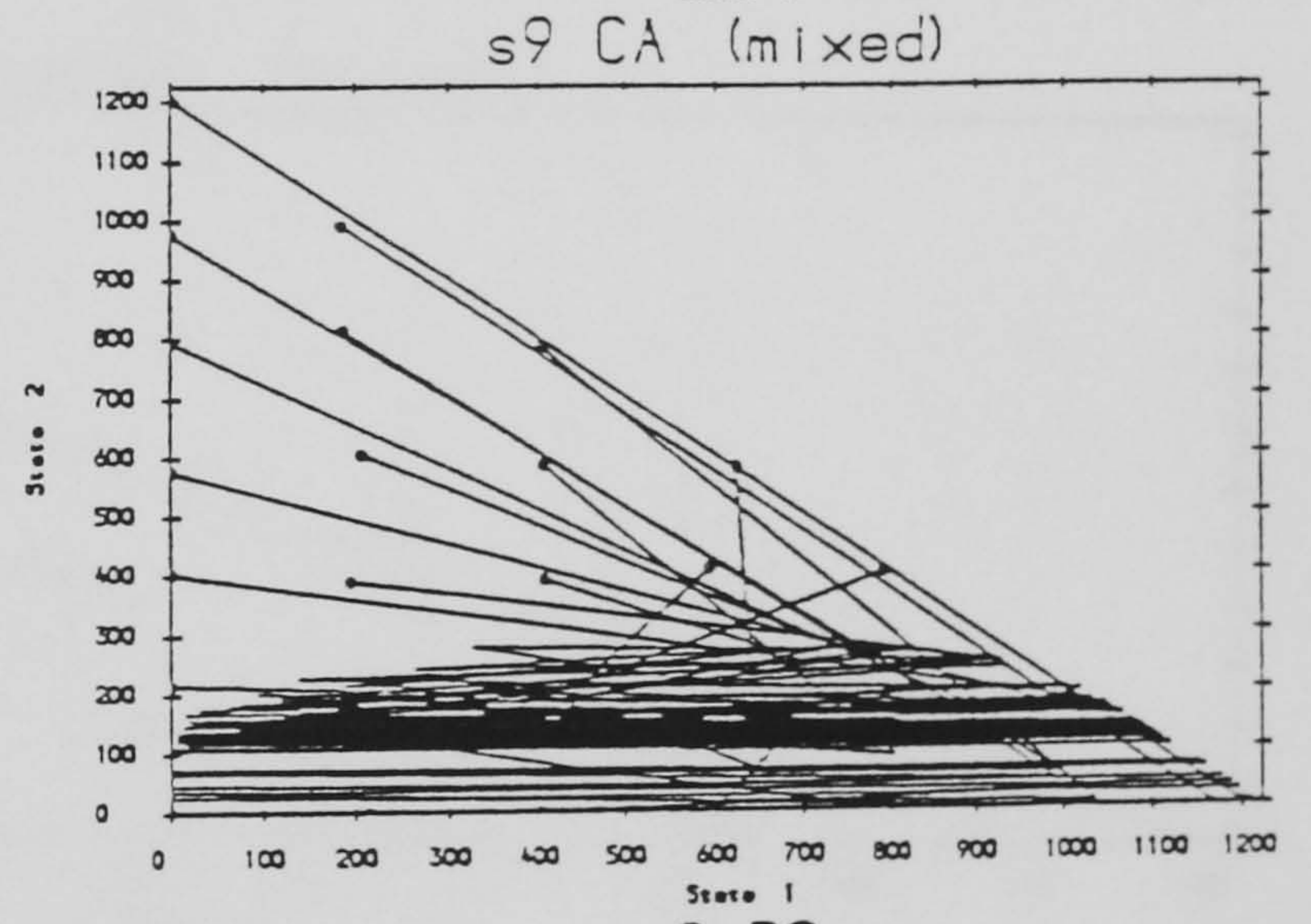
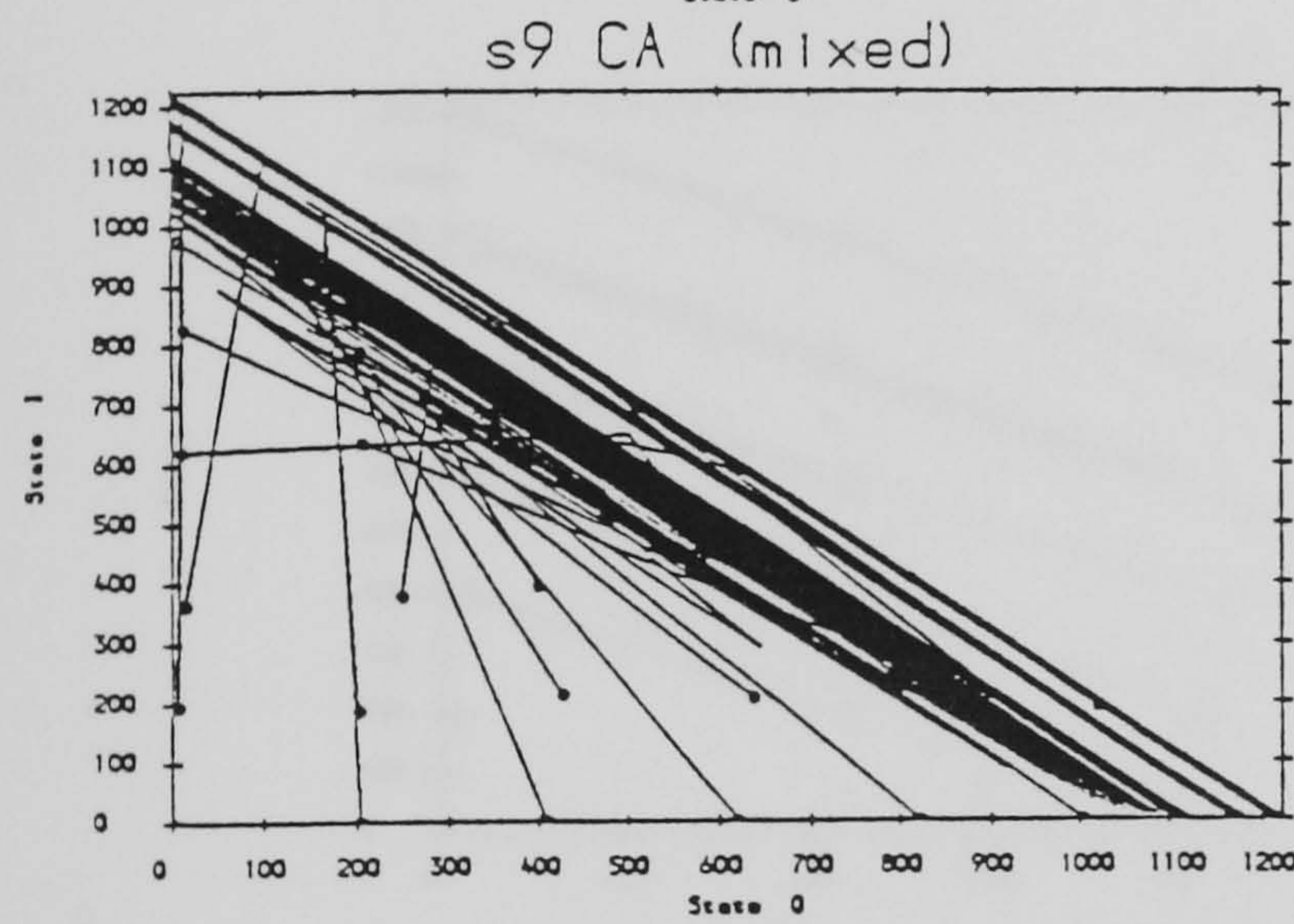
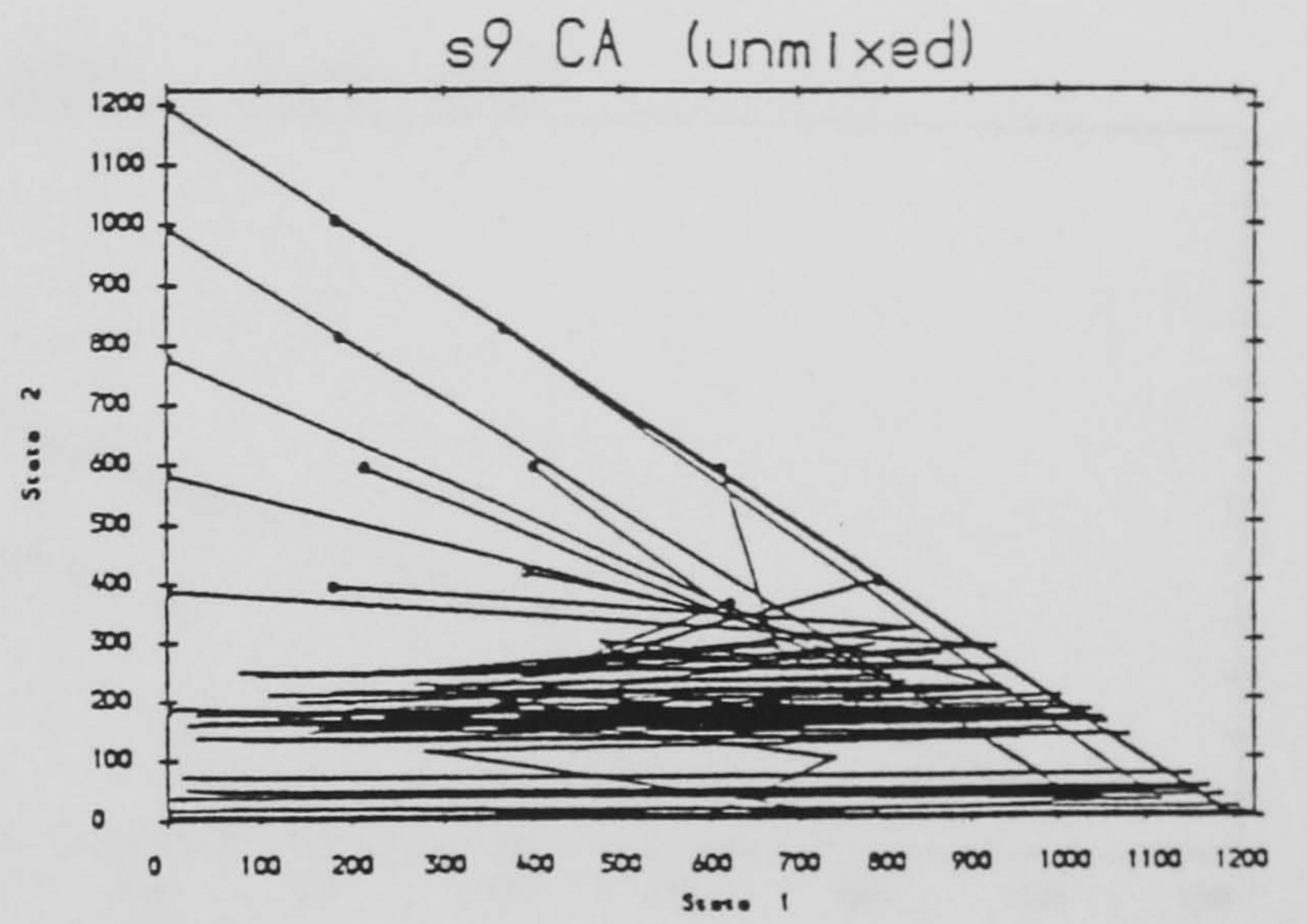
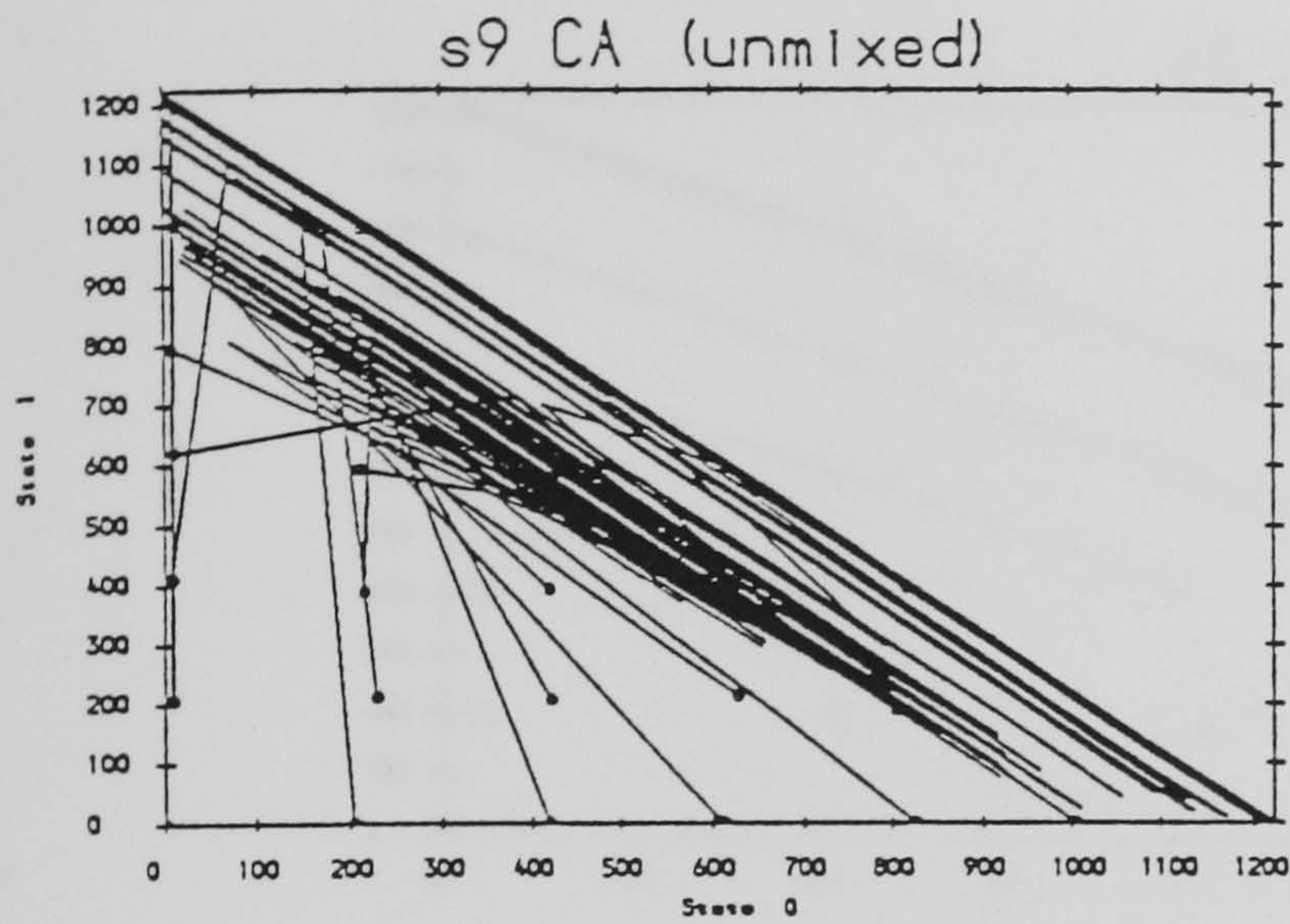




Figure 4.12: Threshold systems phase diagrams.  
(a) System 3: Probabilistic.

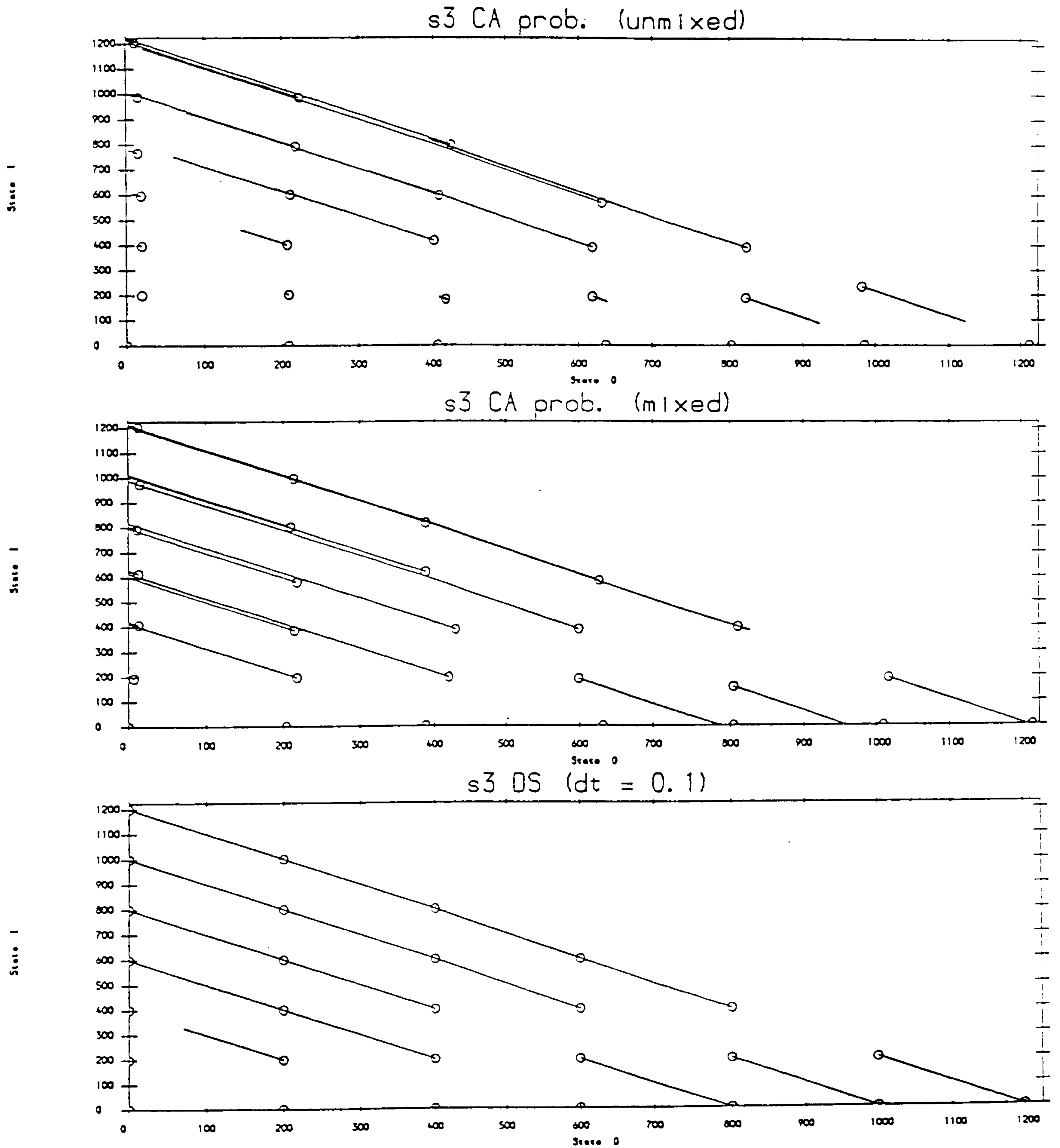




Figure 4.12: Threshold systems phase diagrams.  
 (b) System 4: Probabilistic.

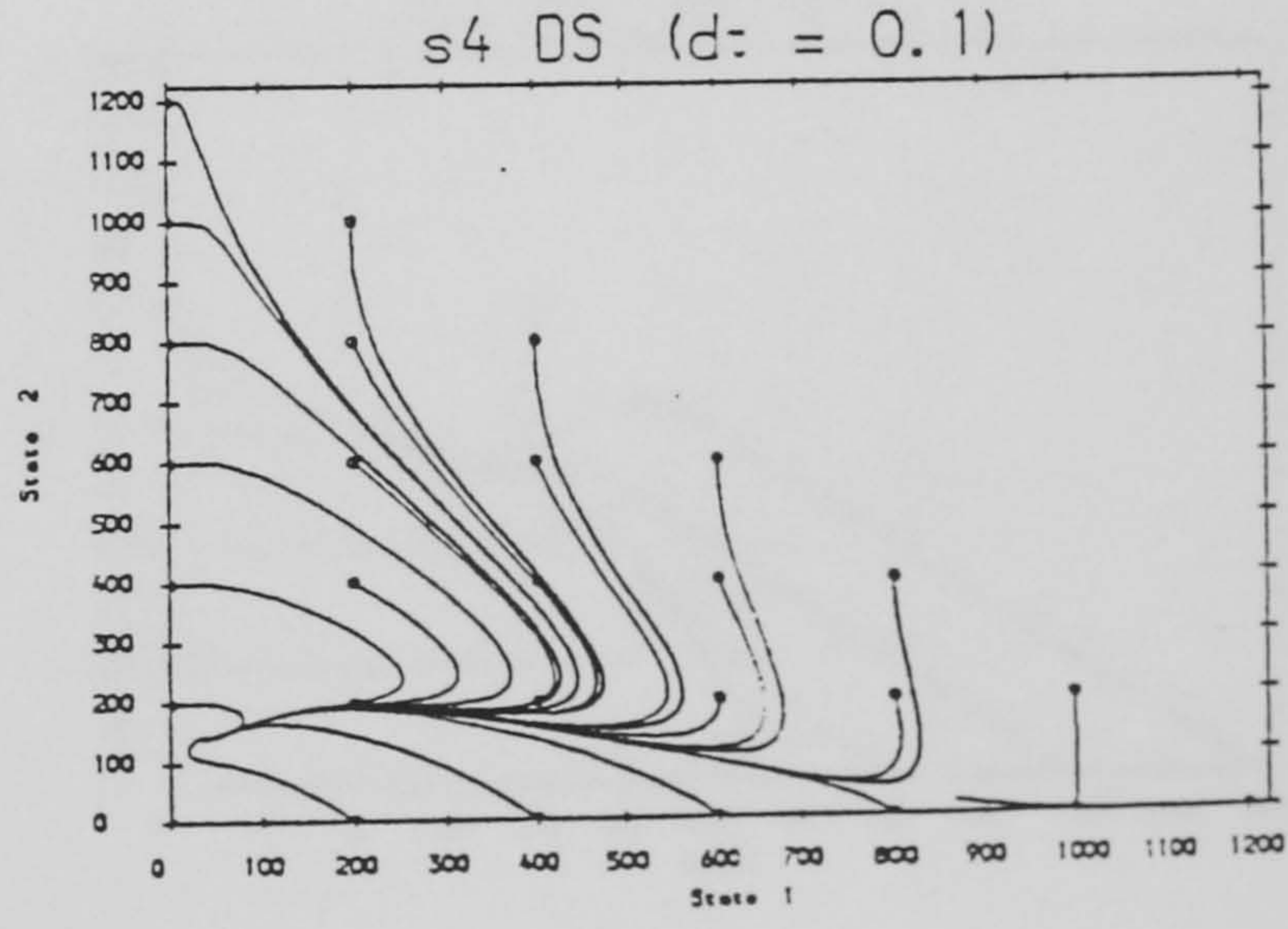
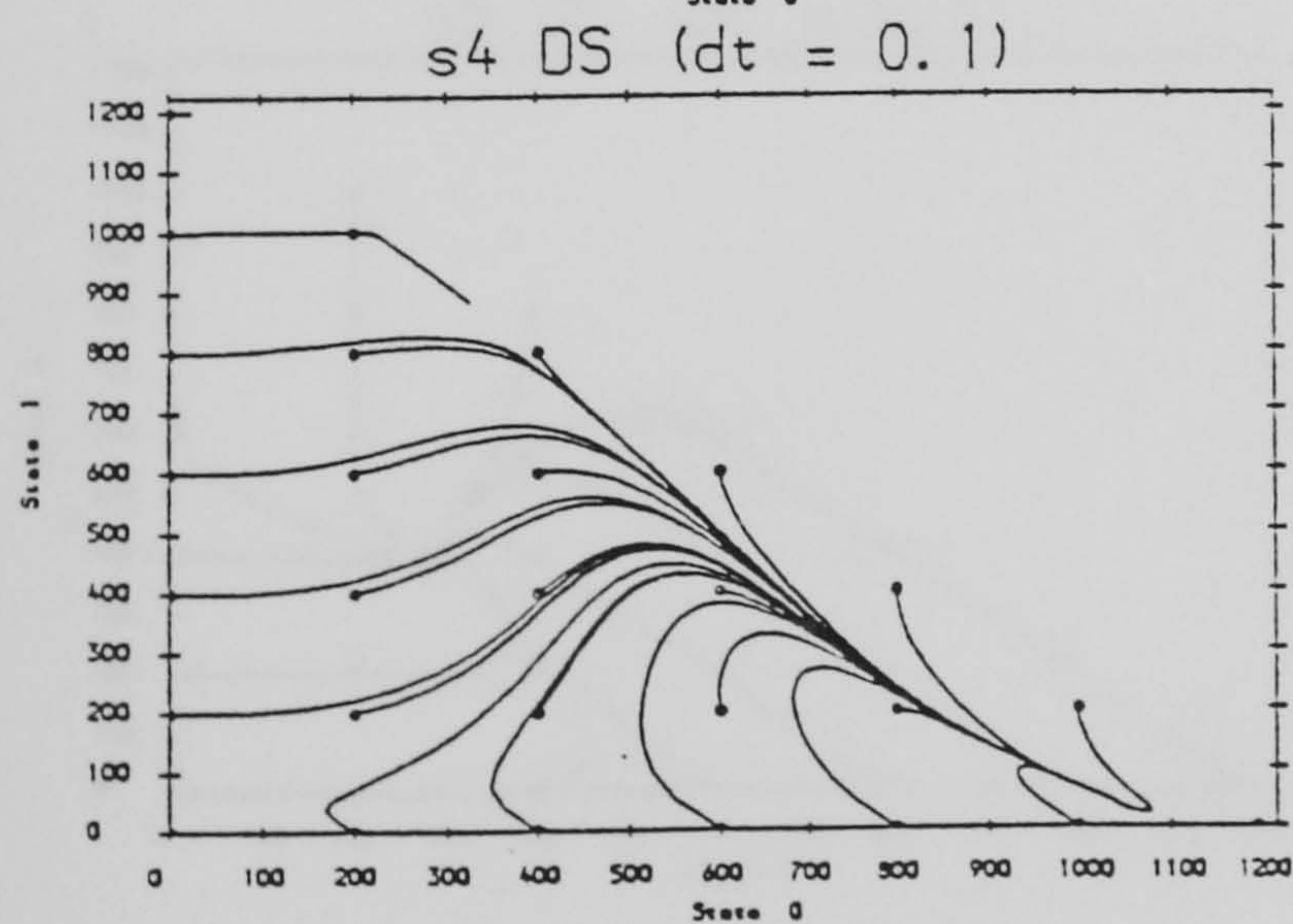
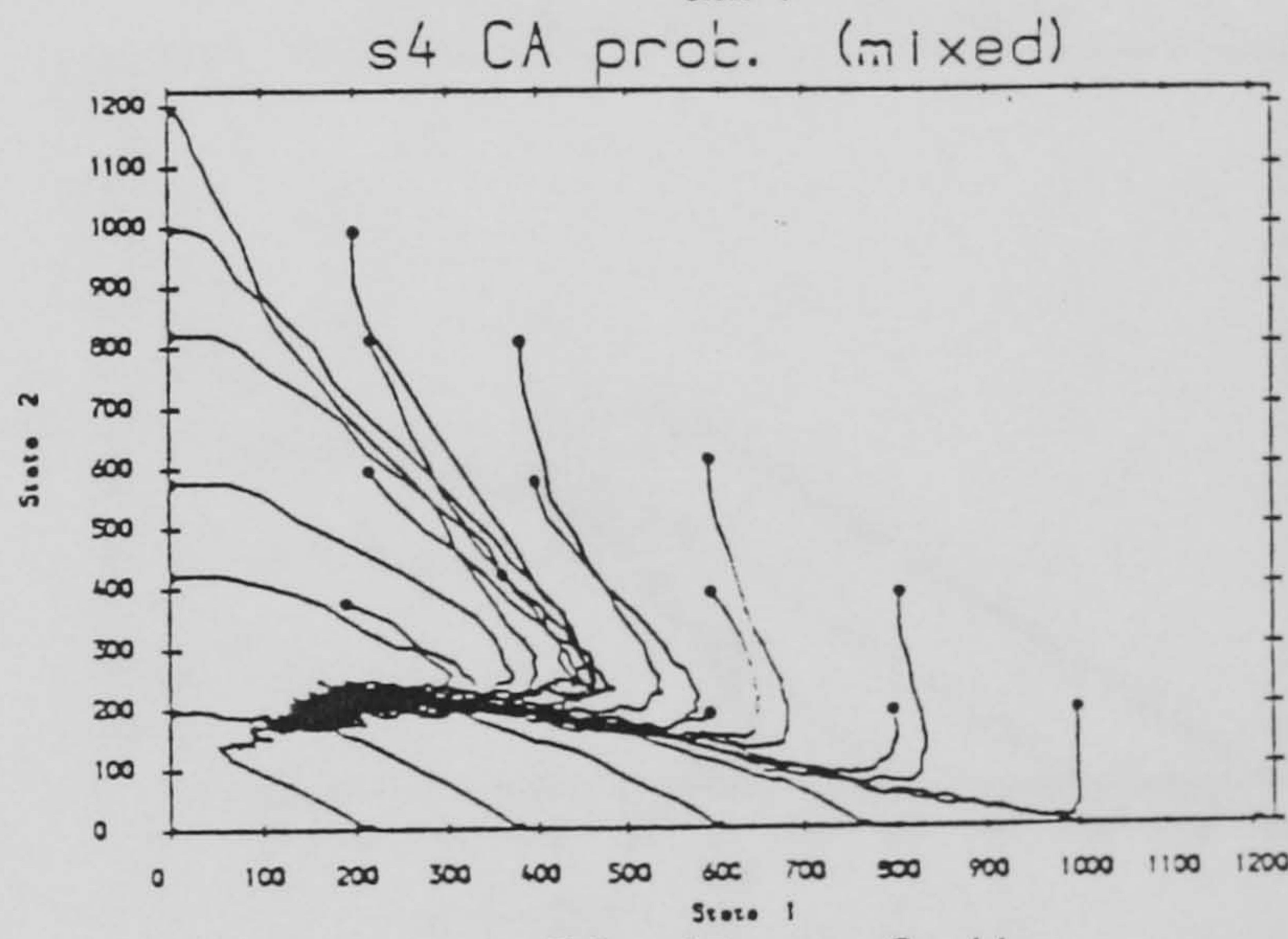
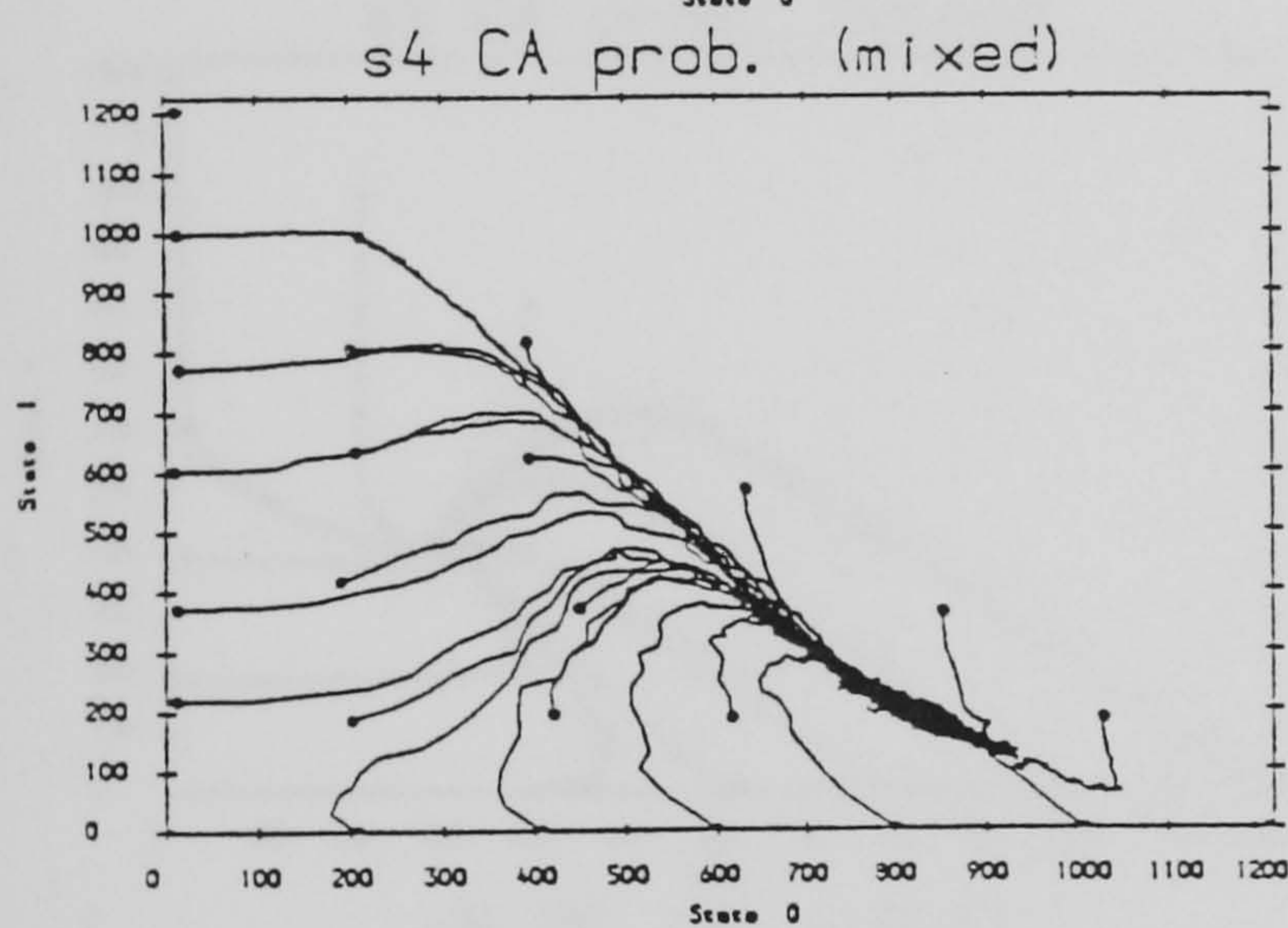
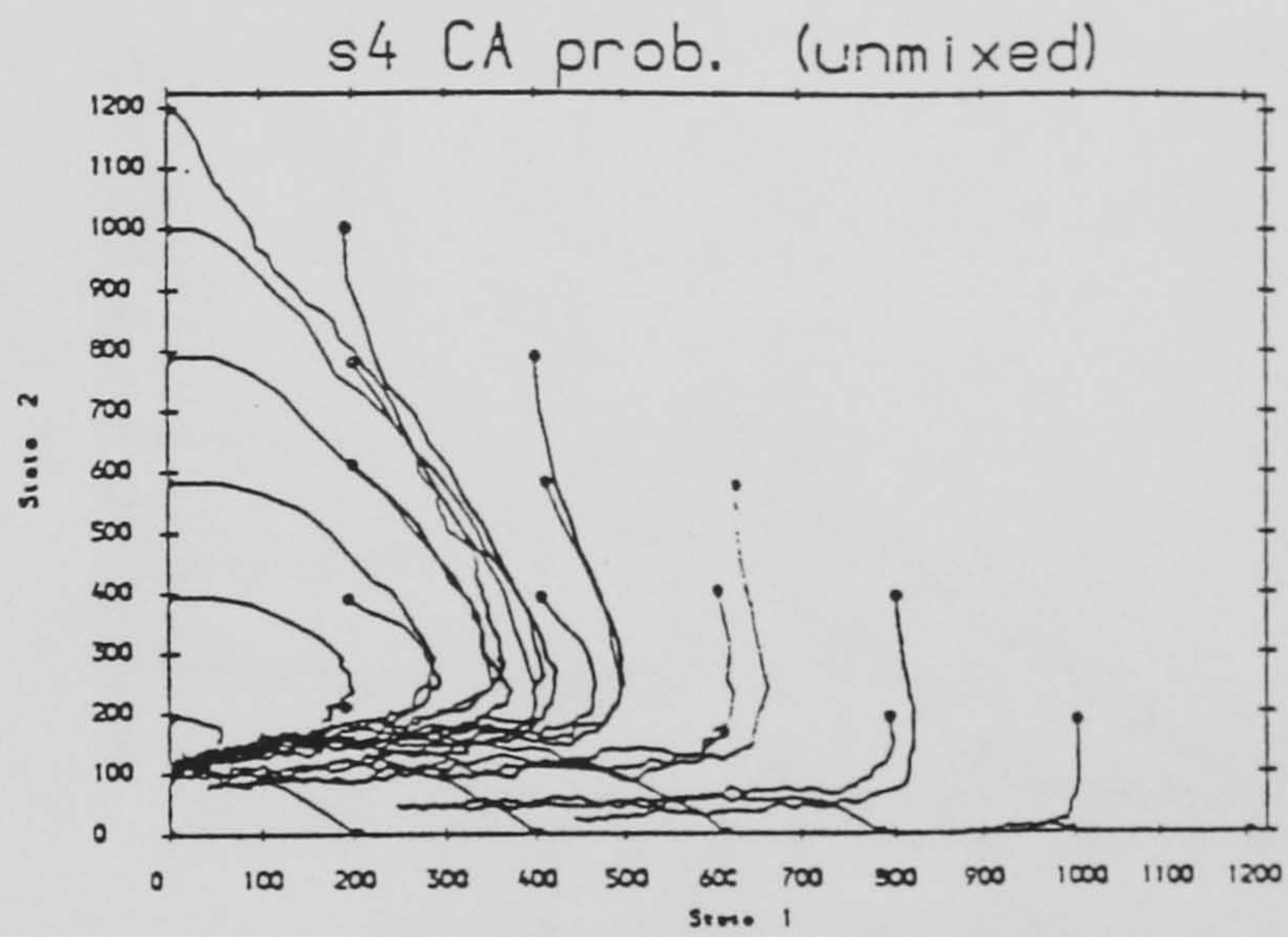
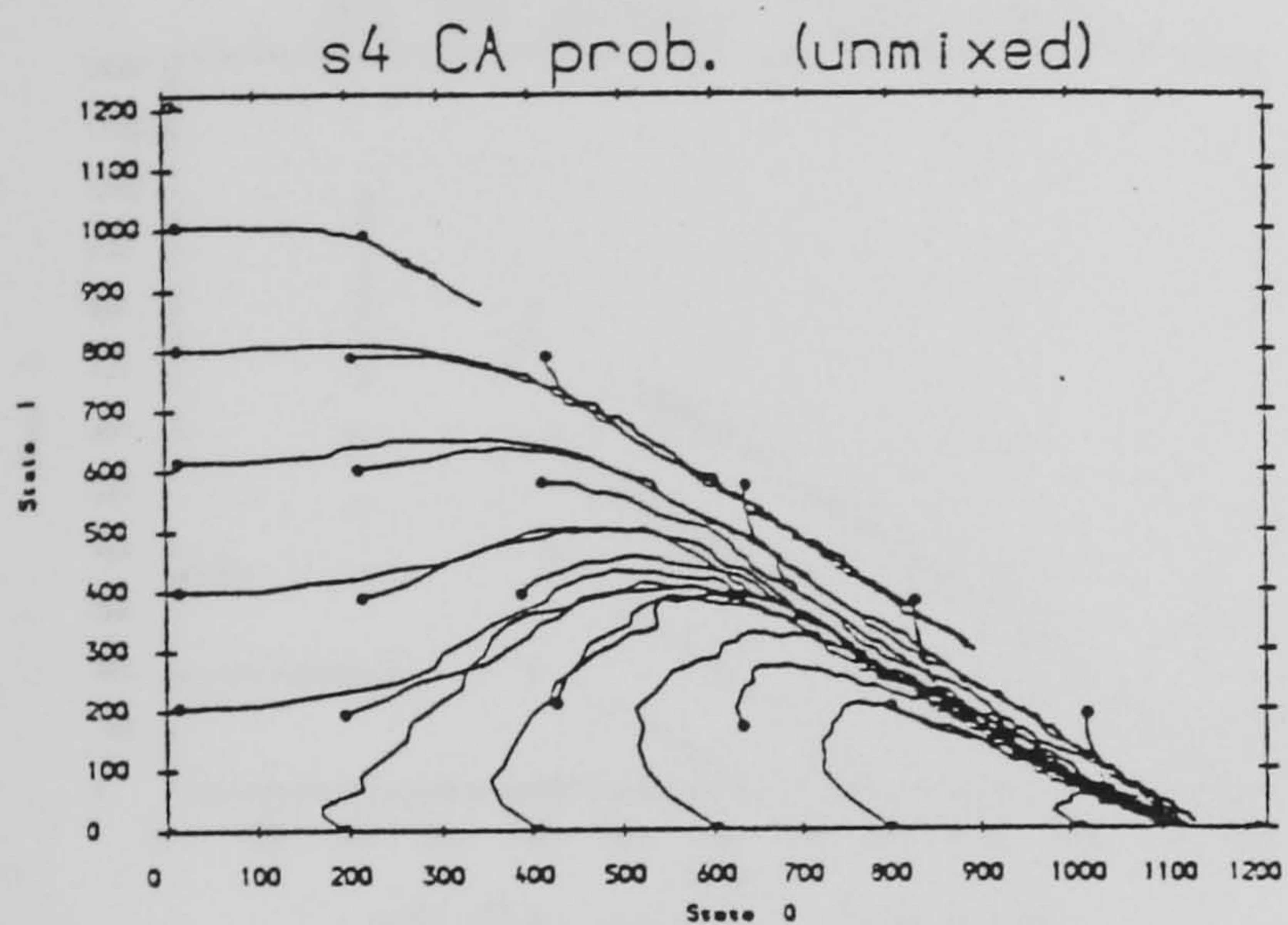




Figure 4.12: Threshold systems phase diagrams.  
(c) System 5: Probabilistic.

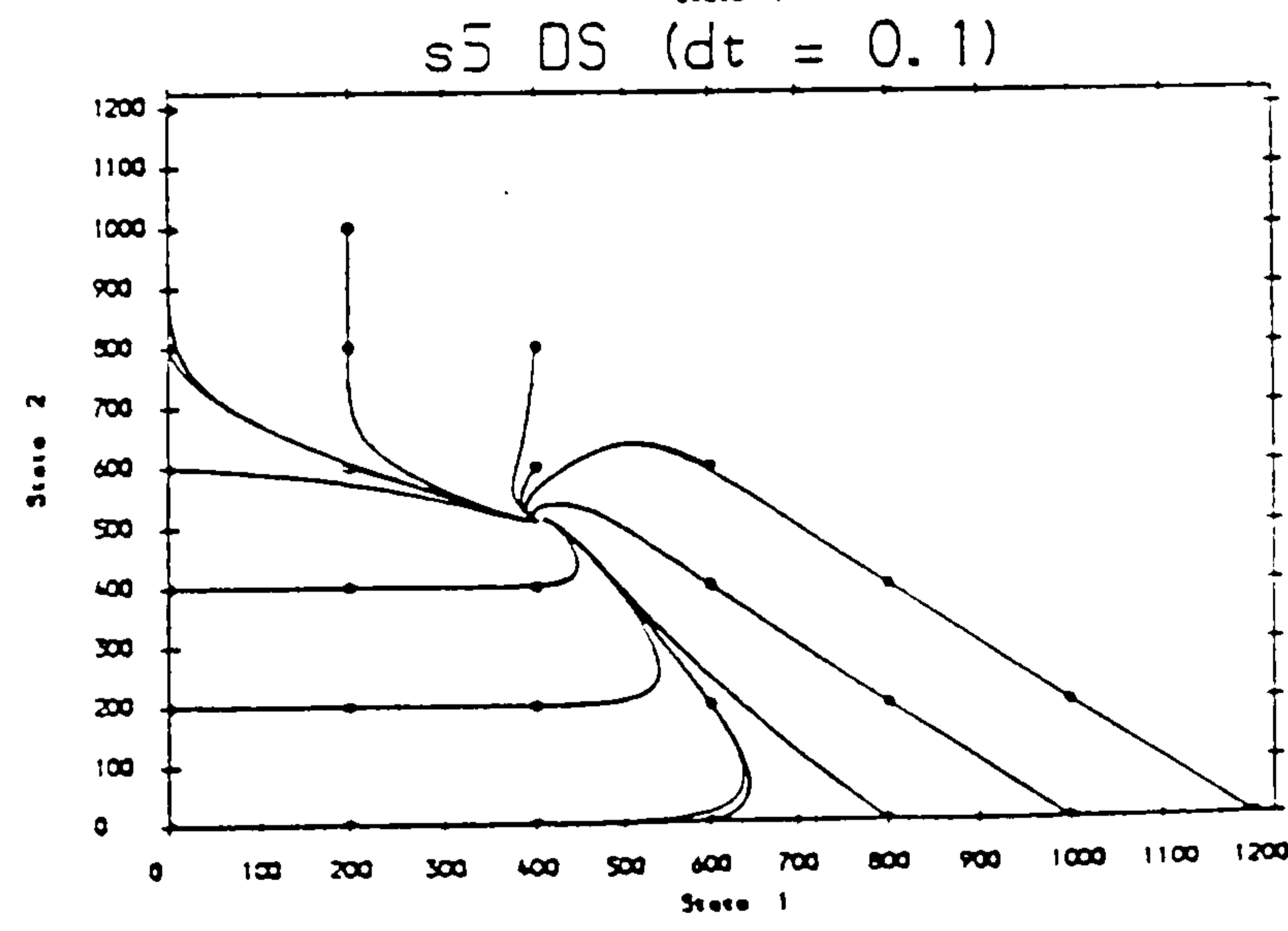
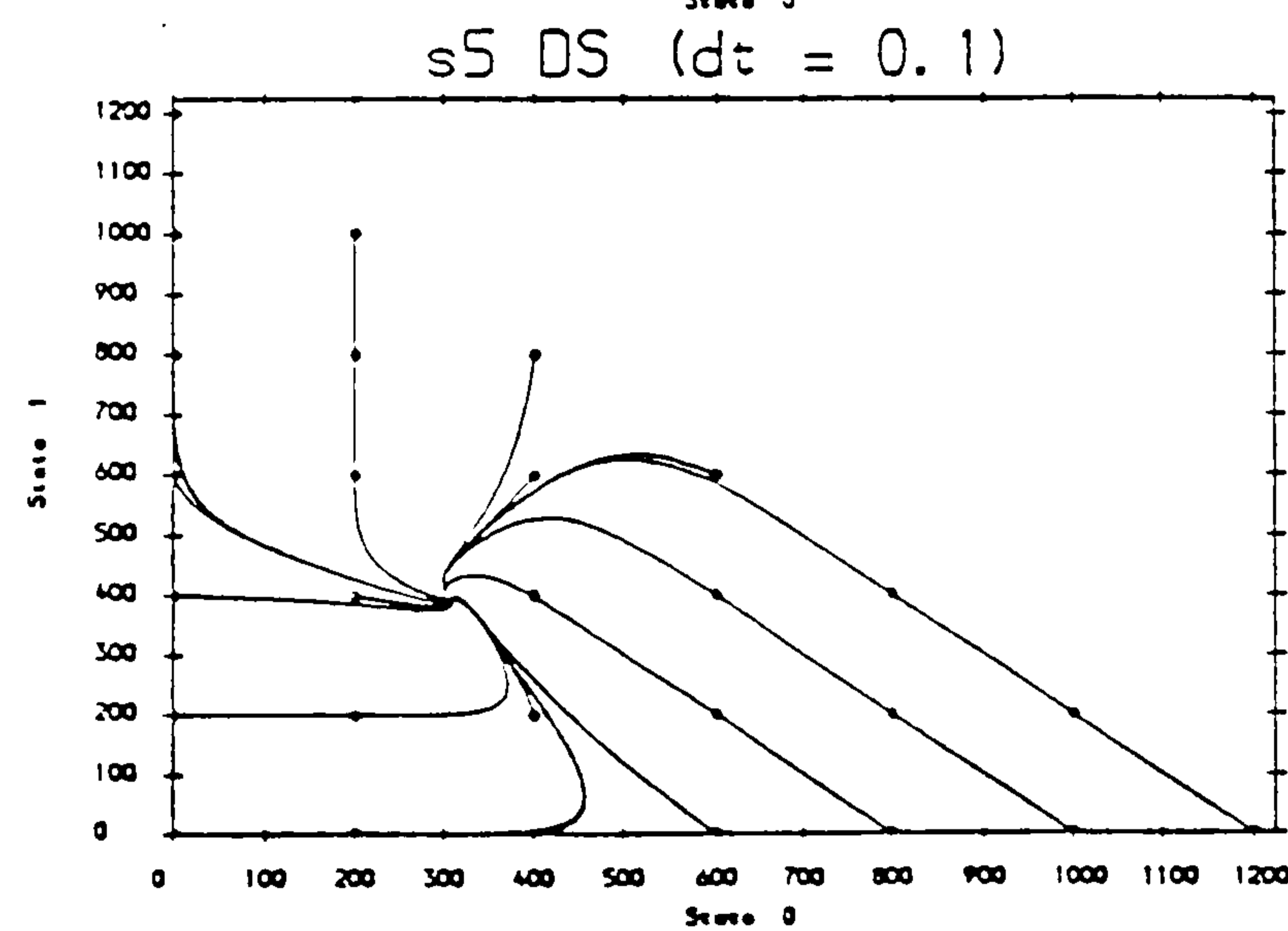
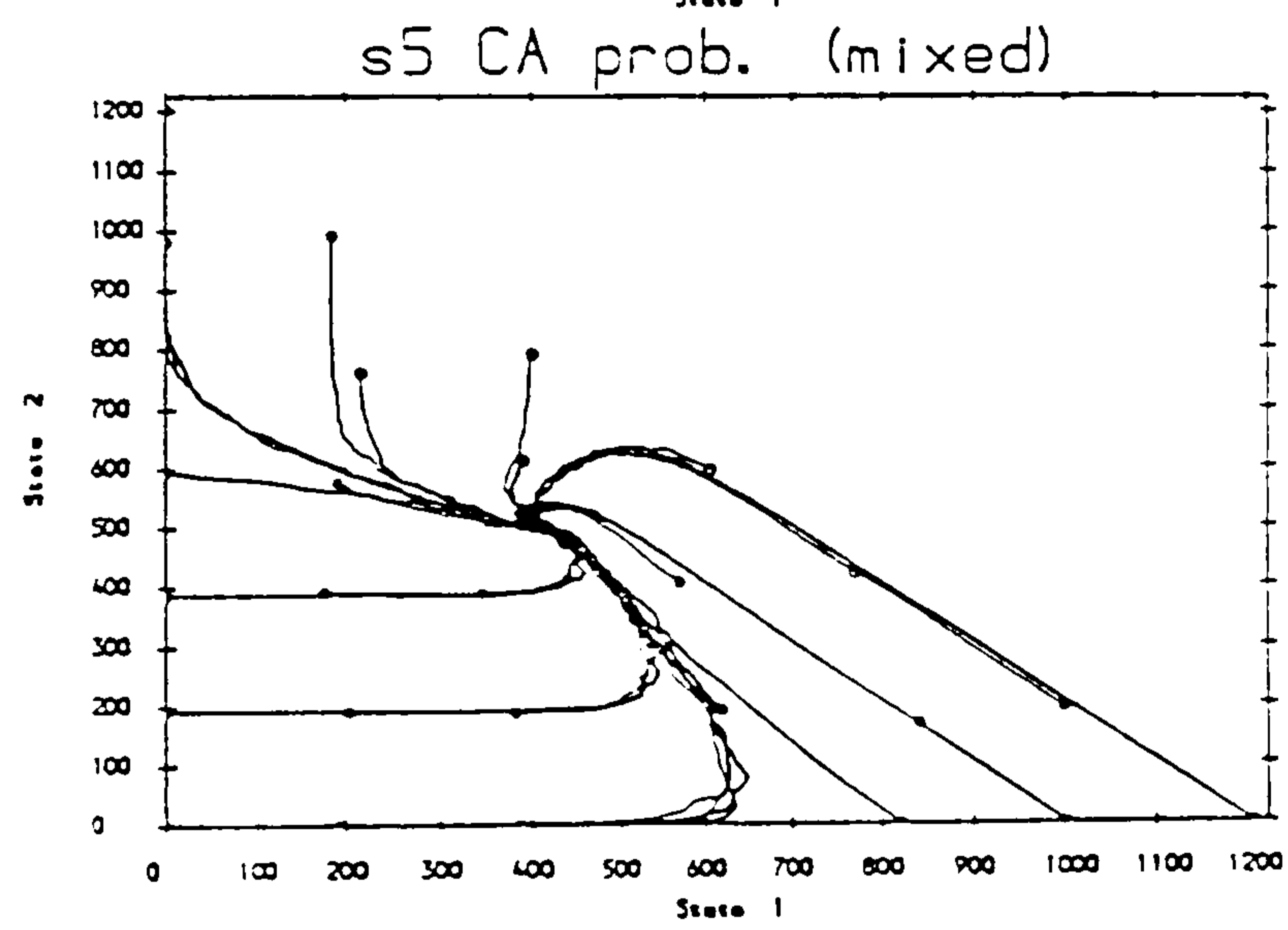
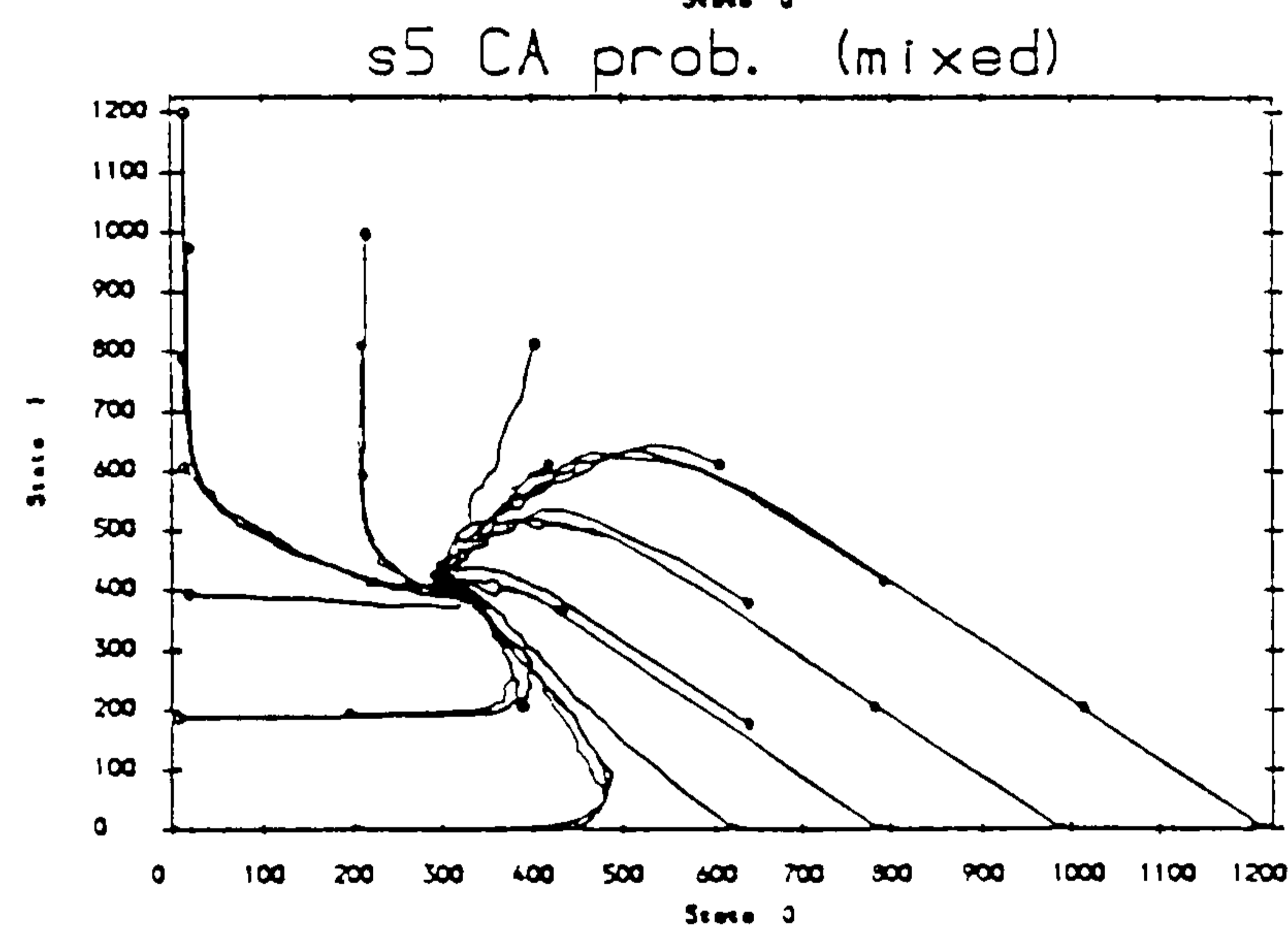
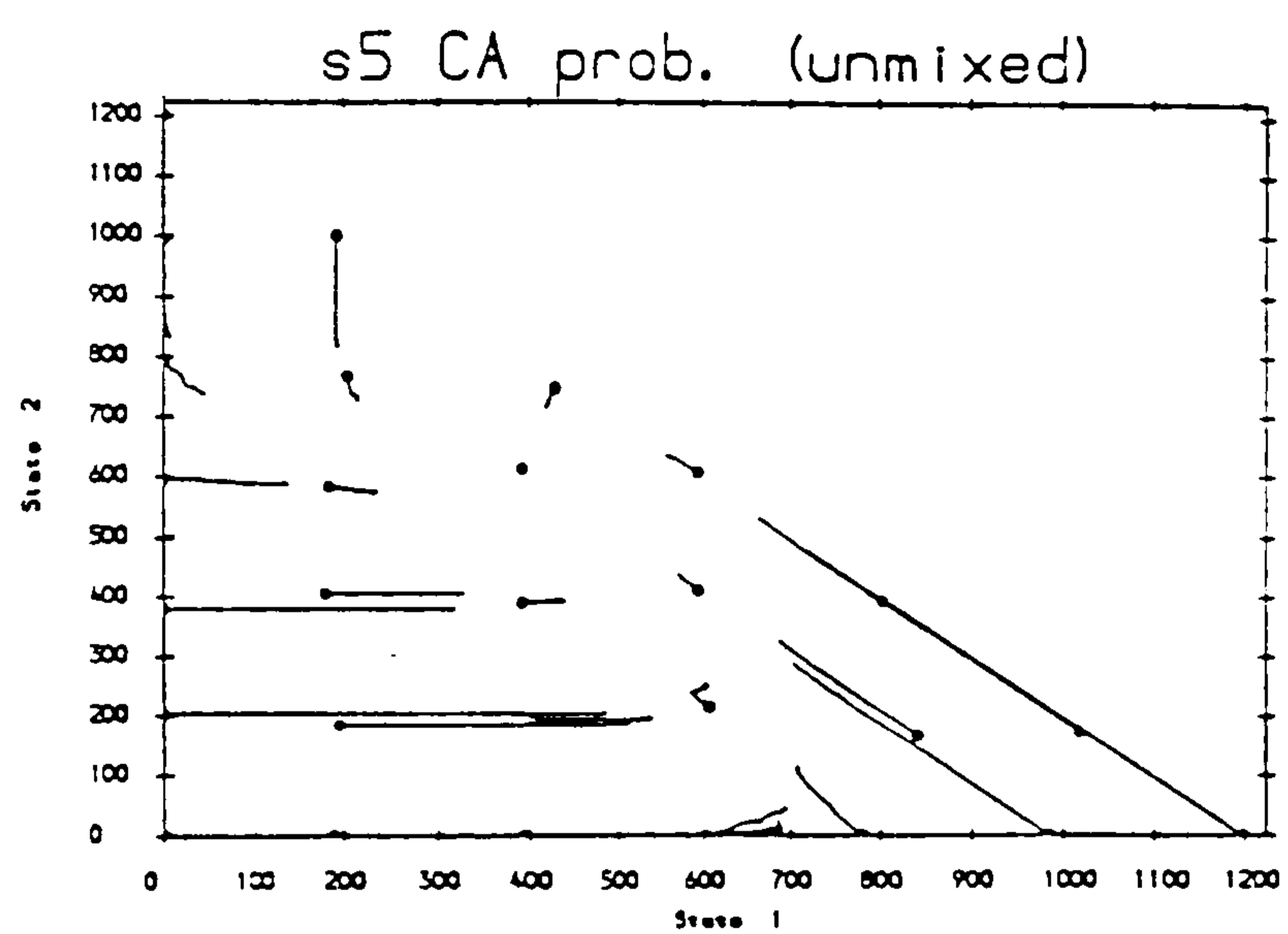
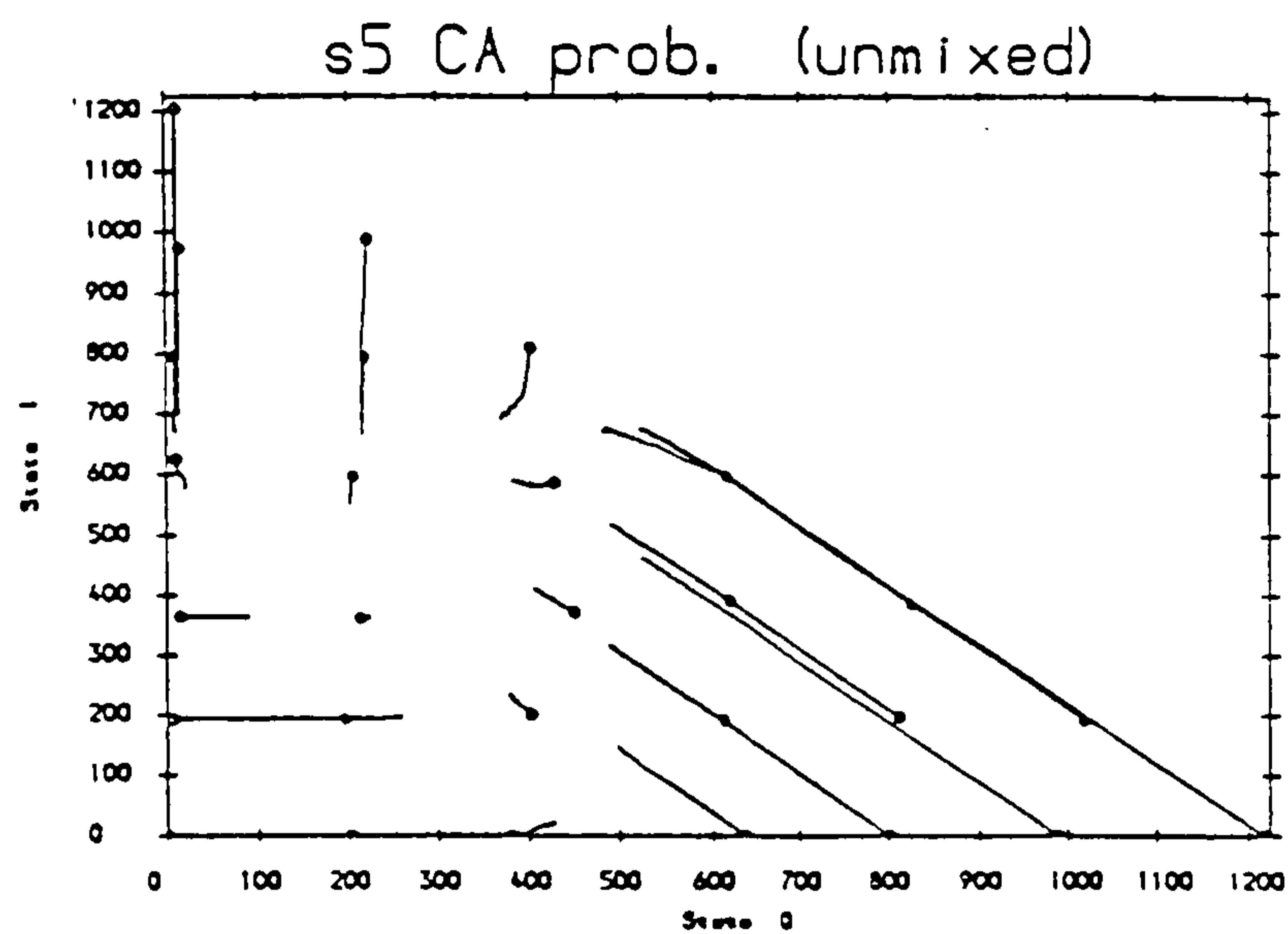




Figure 4.12: Threshold systems phase diagrams.  
(d) System 6: Probabilistic.

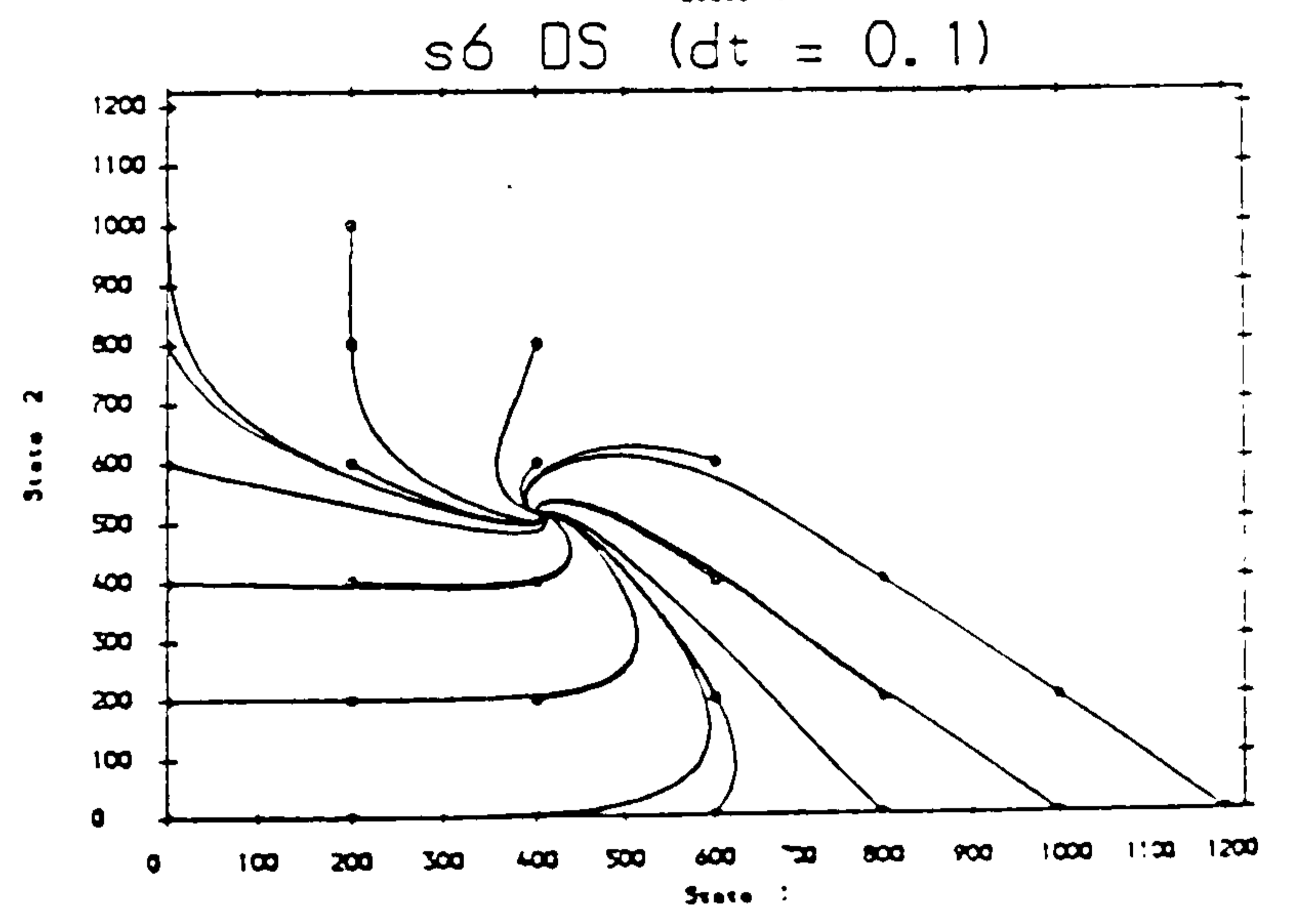
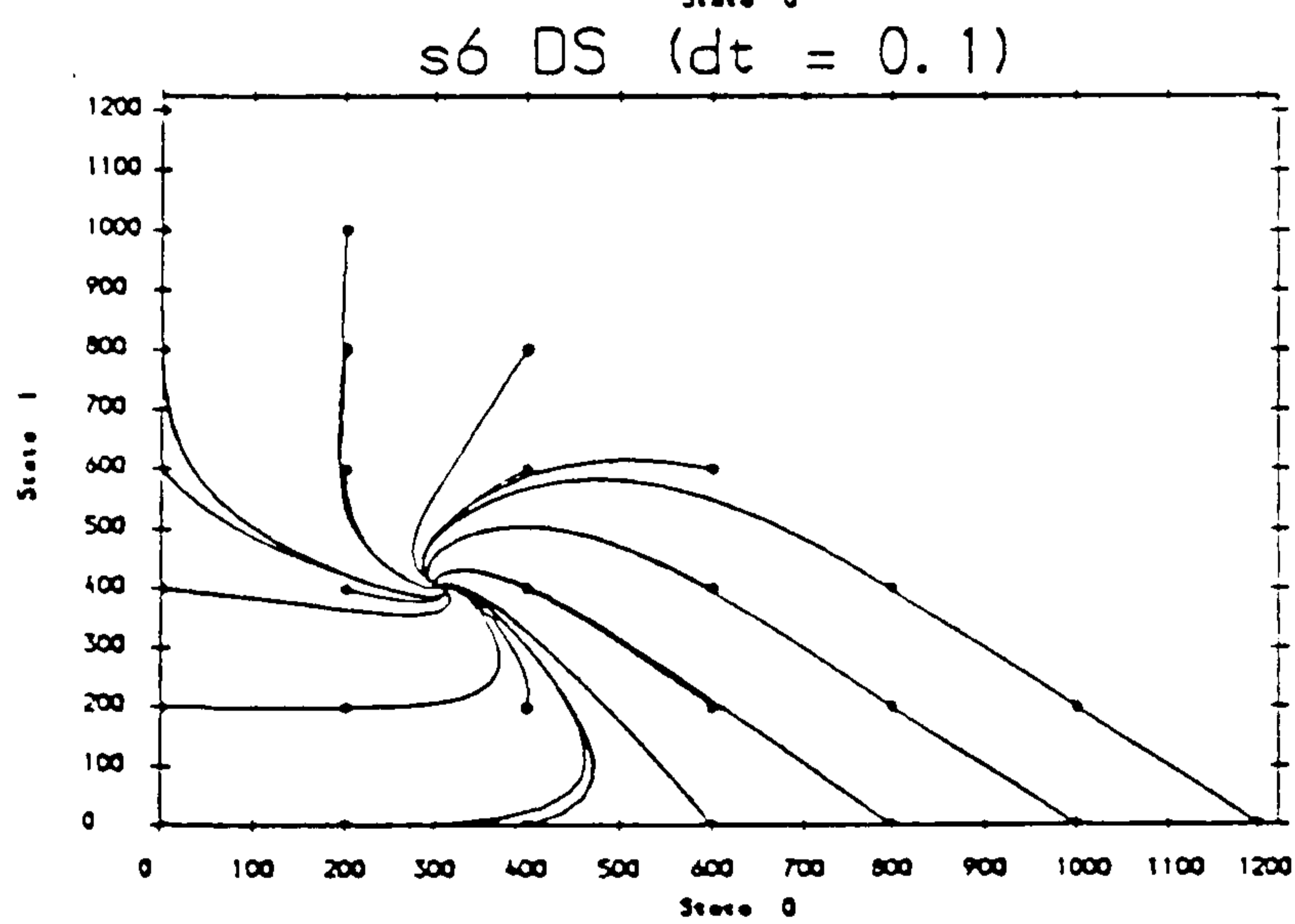
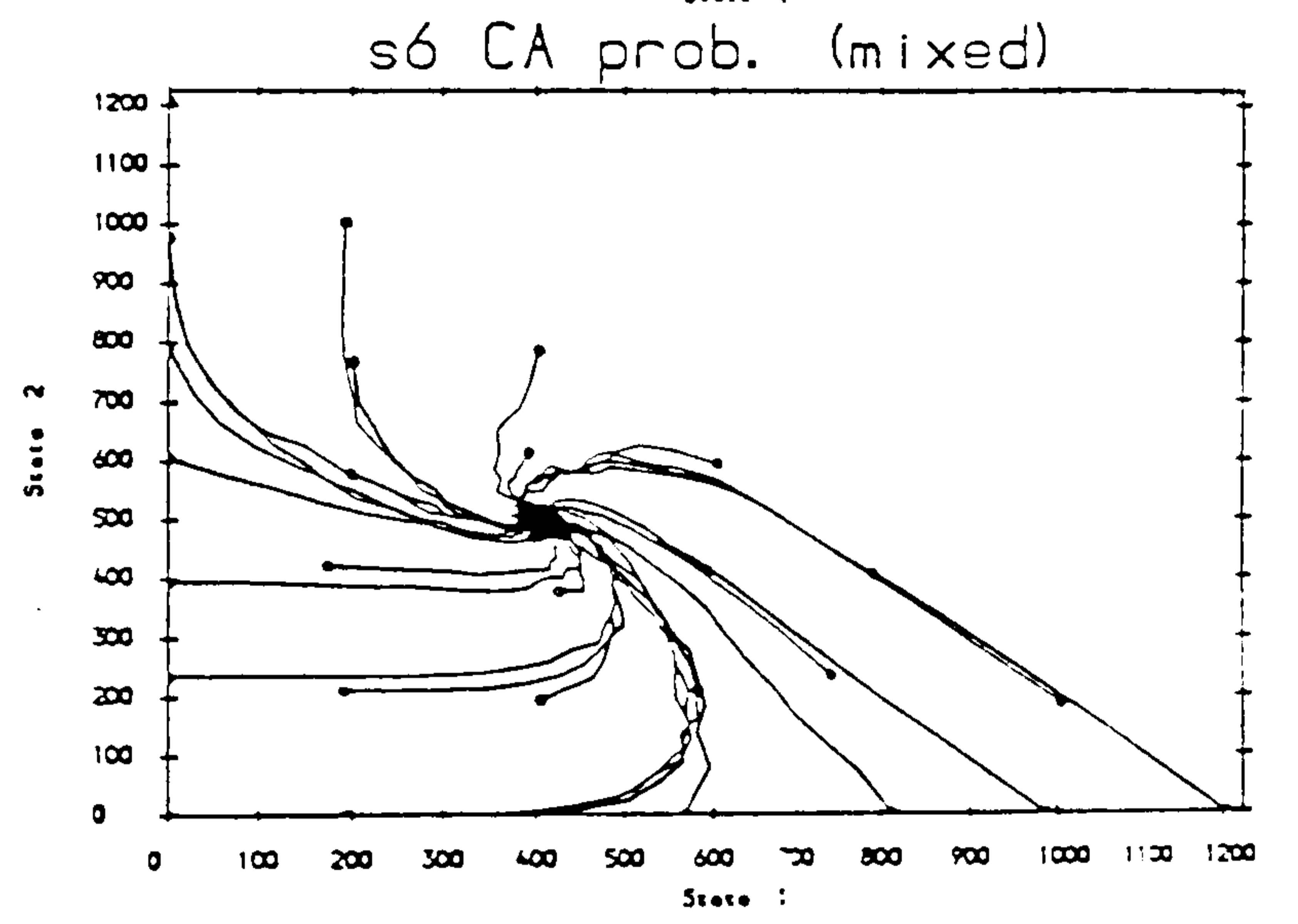
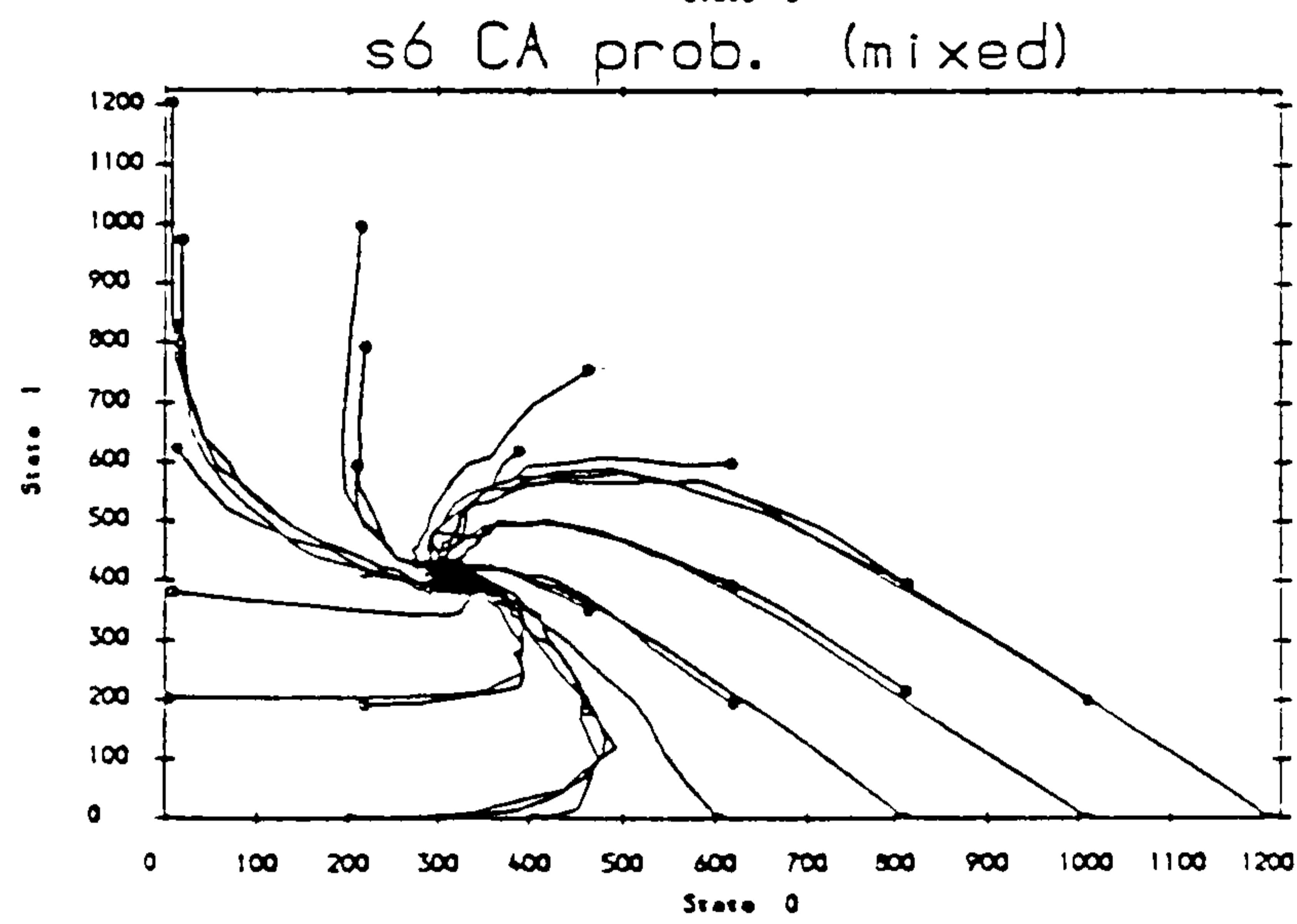
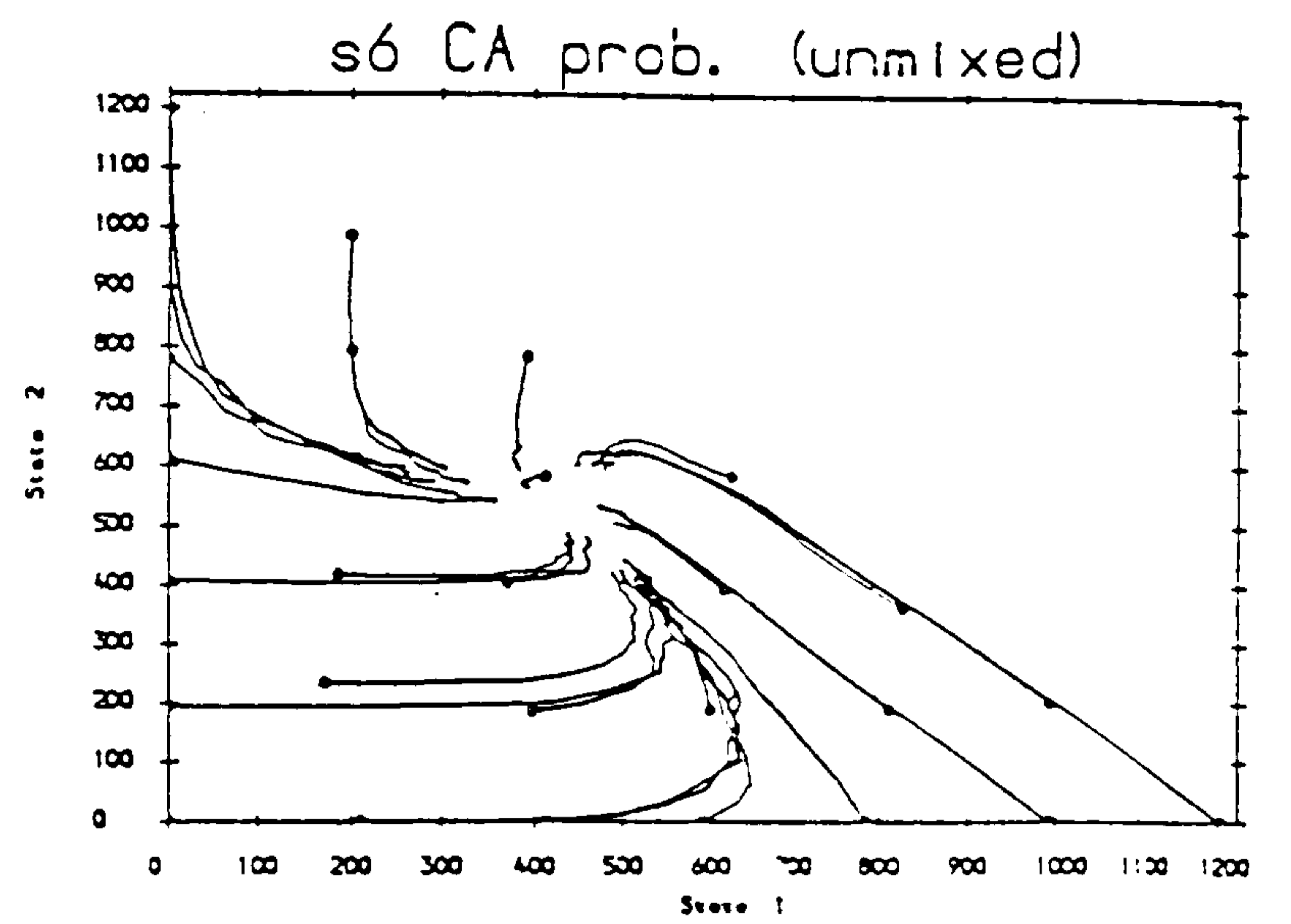
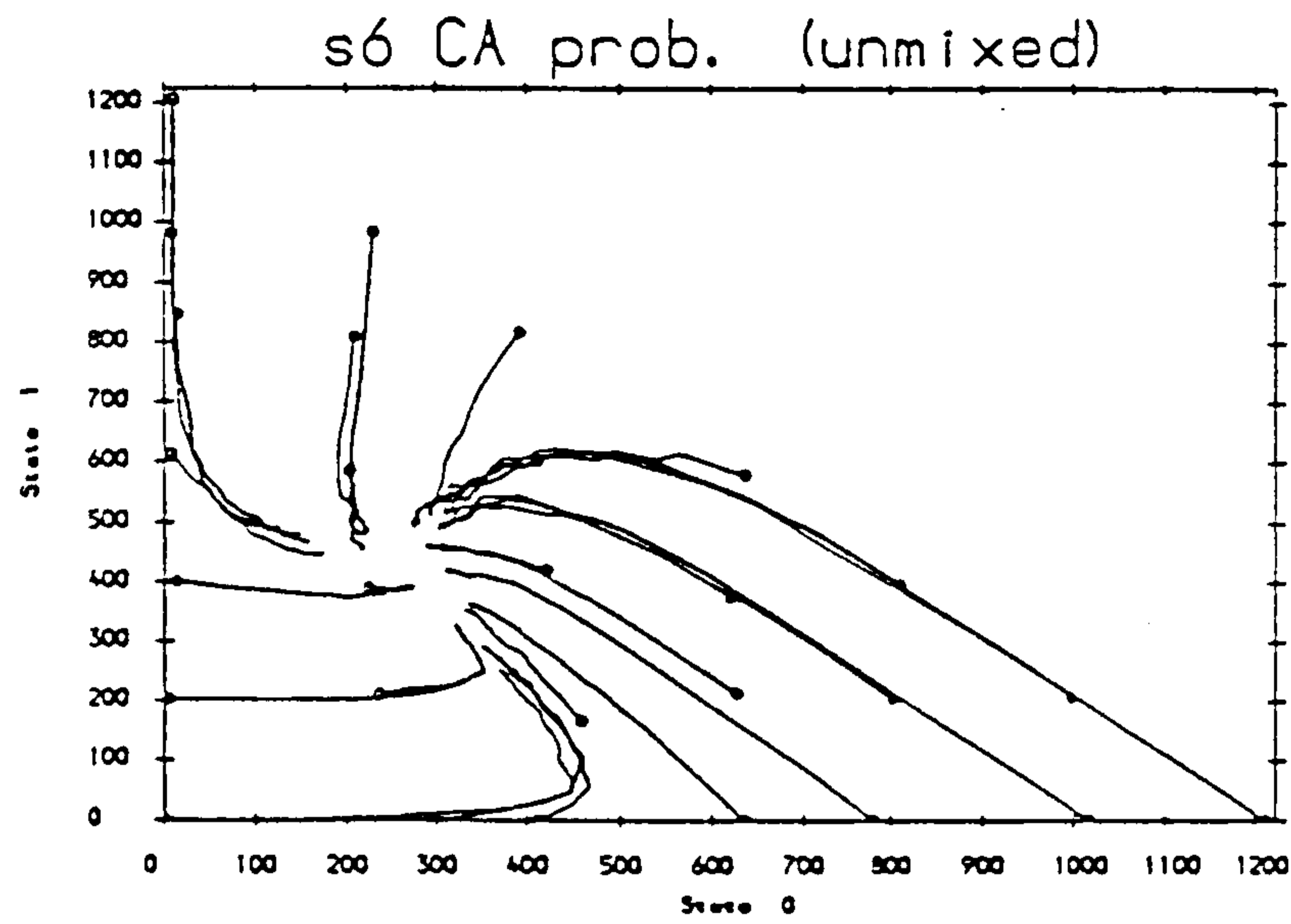


Figure 4.12: Threshold systems phase diagrams.  
(e) System 7: Probabilistic.

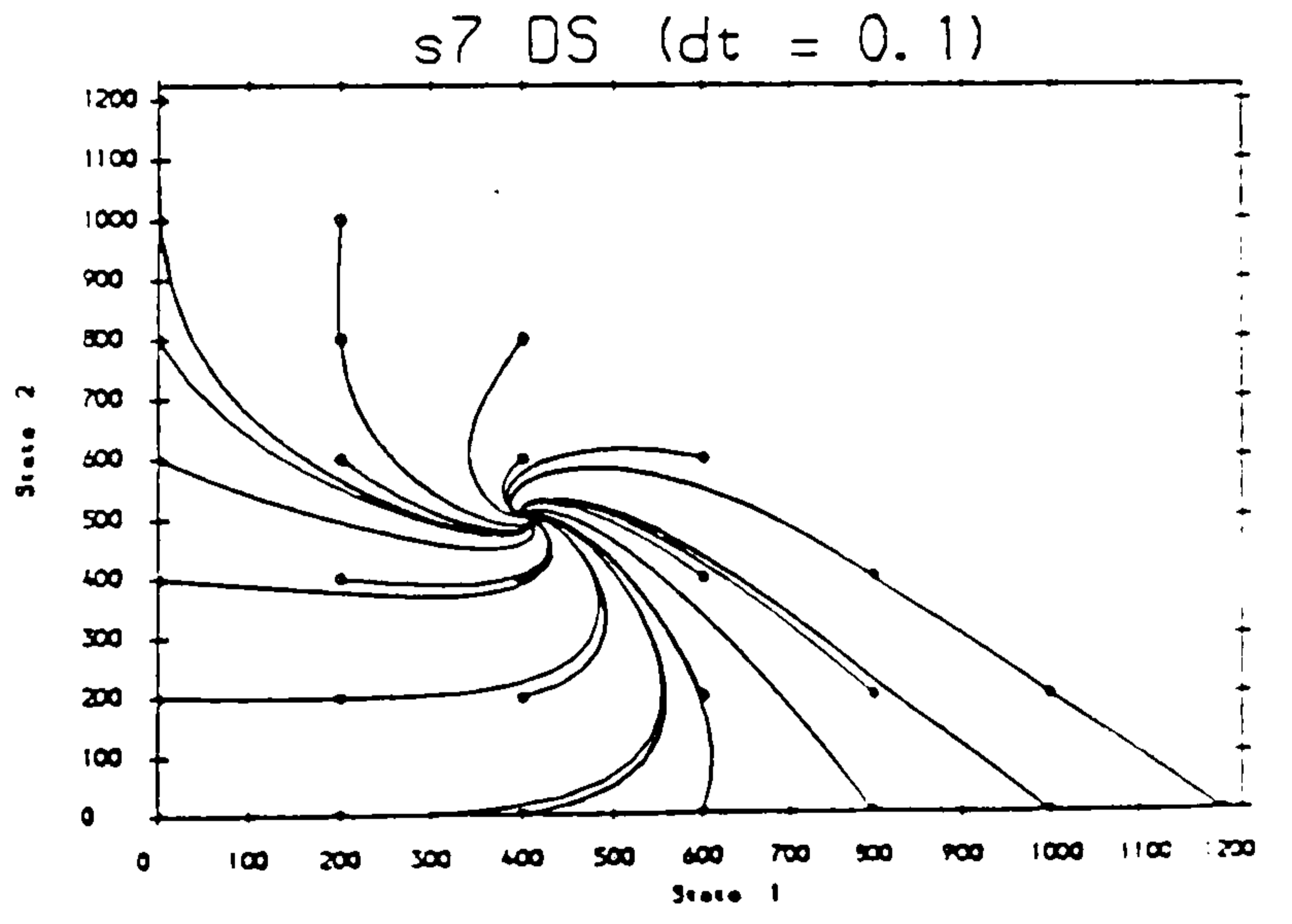
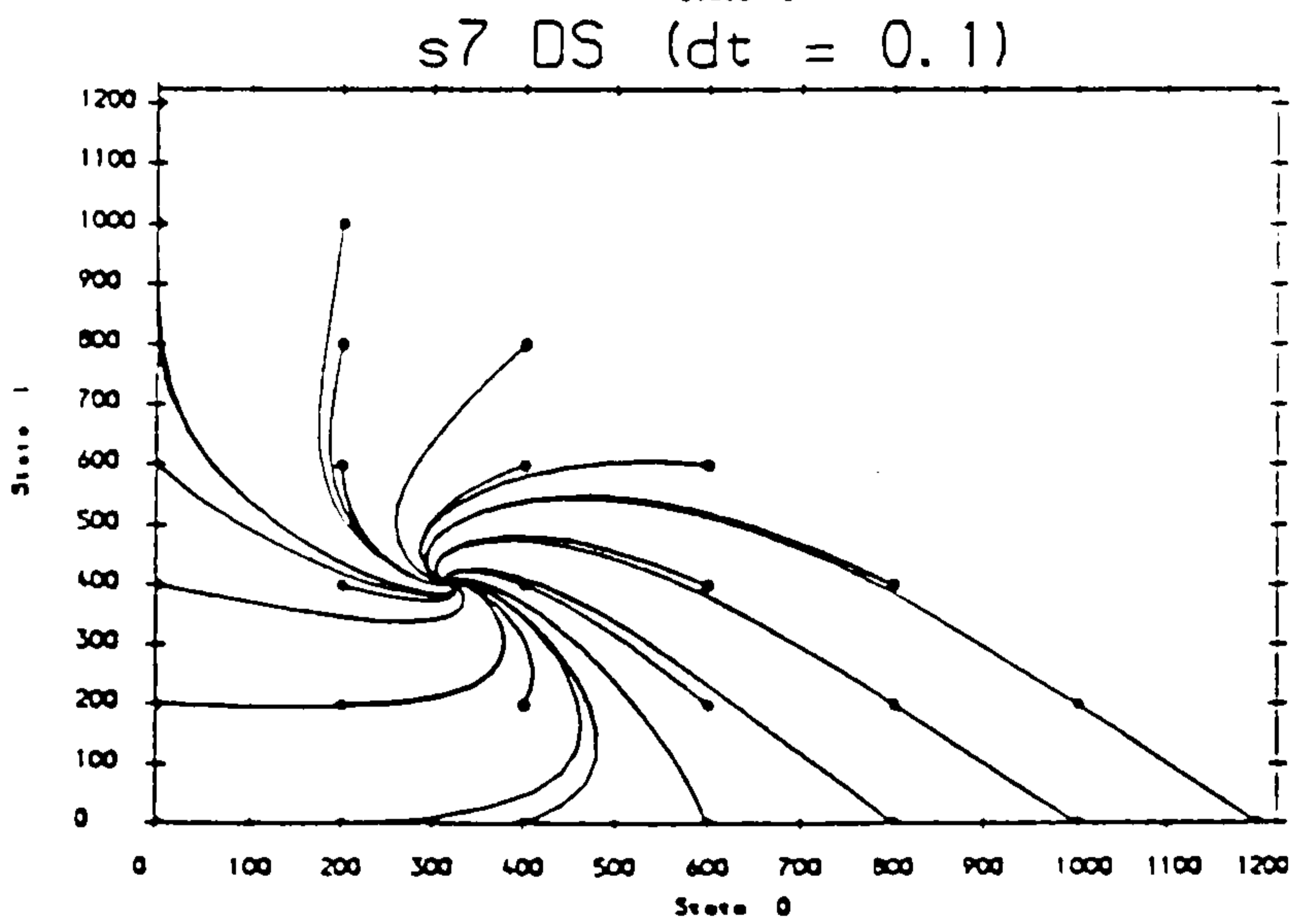
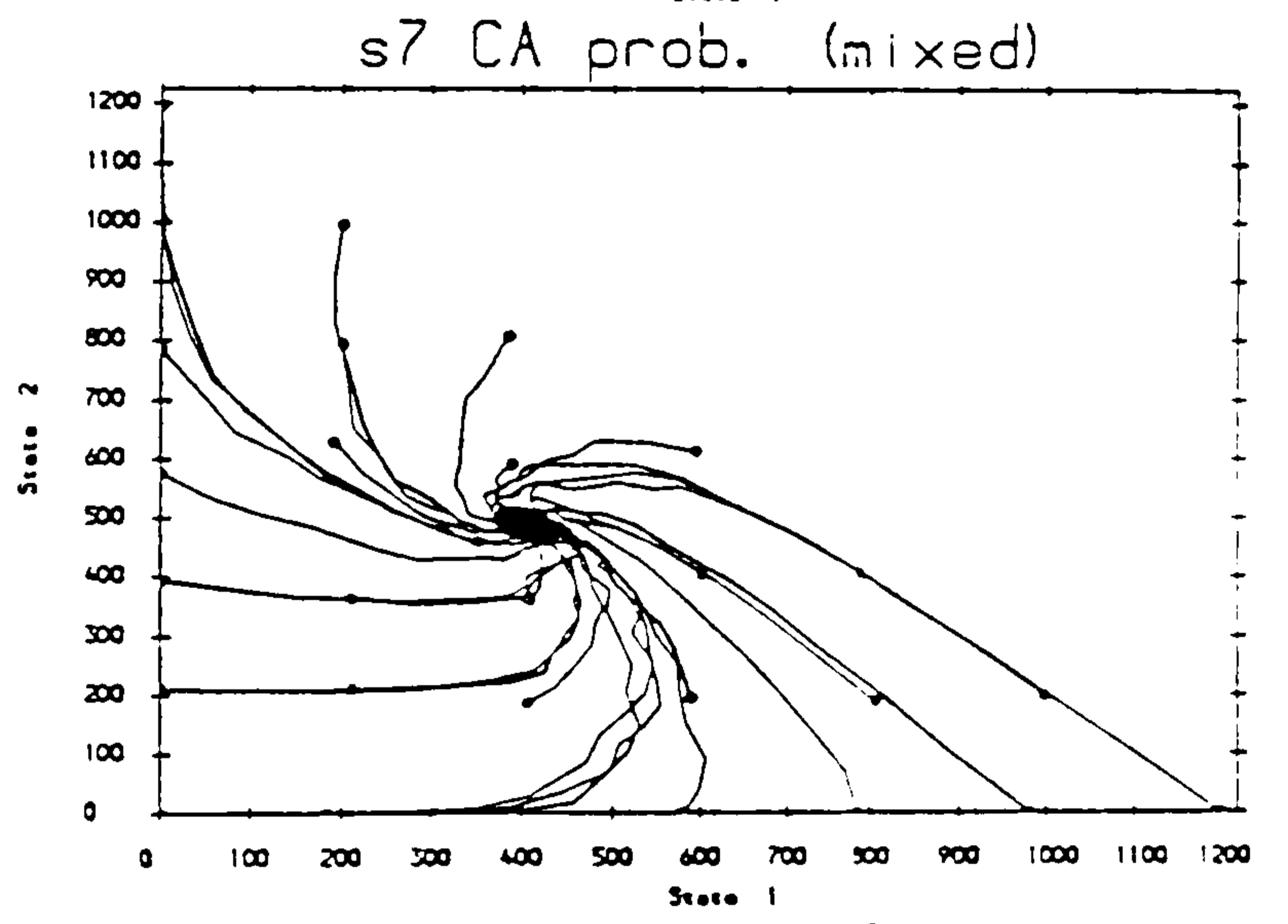
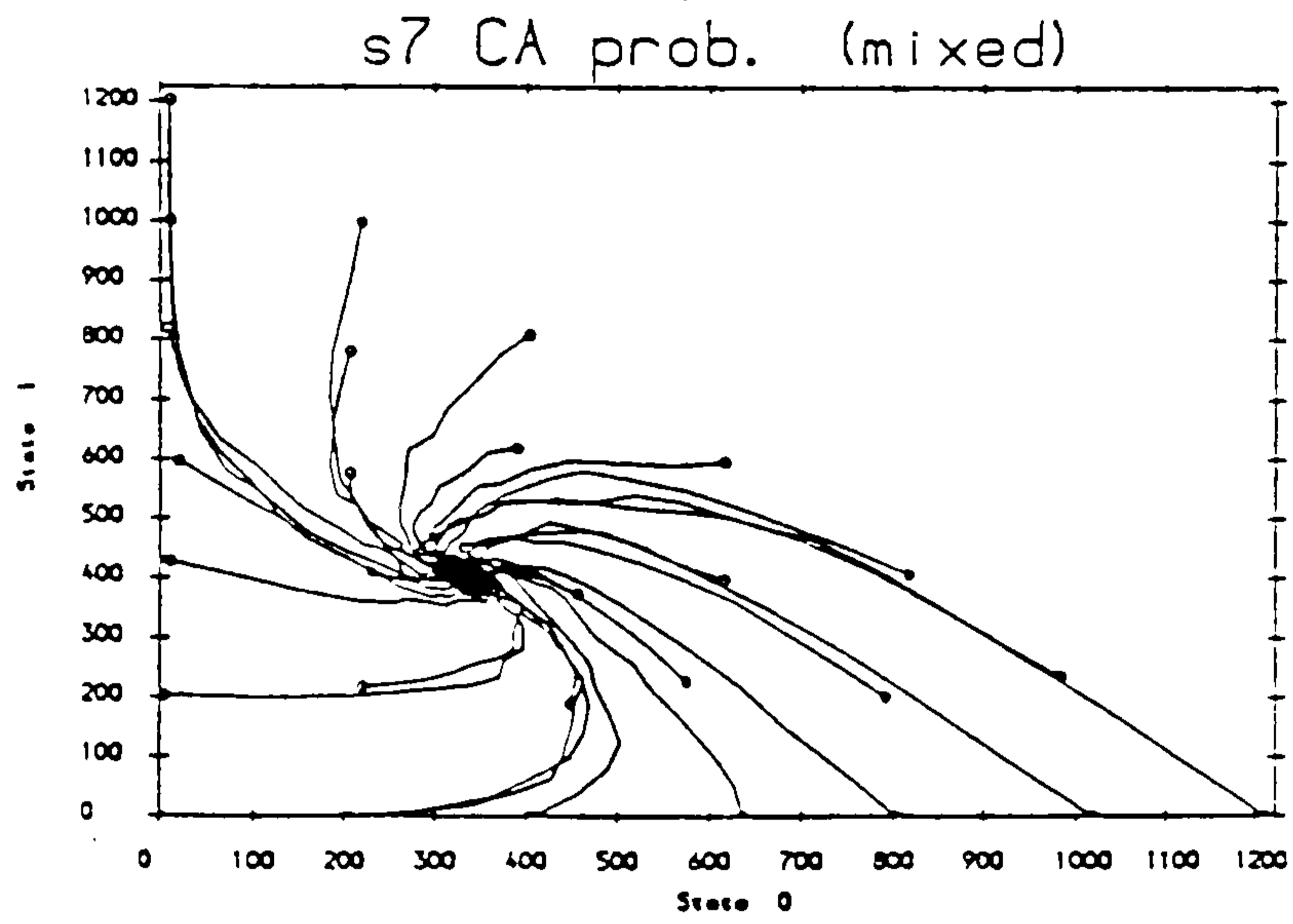
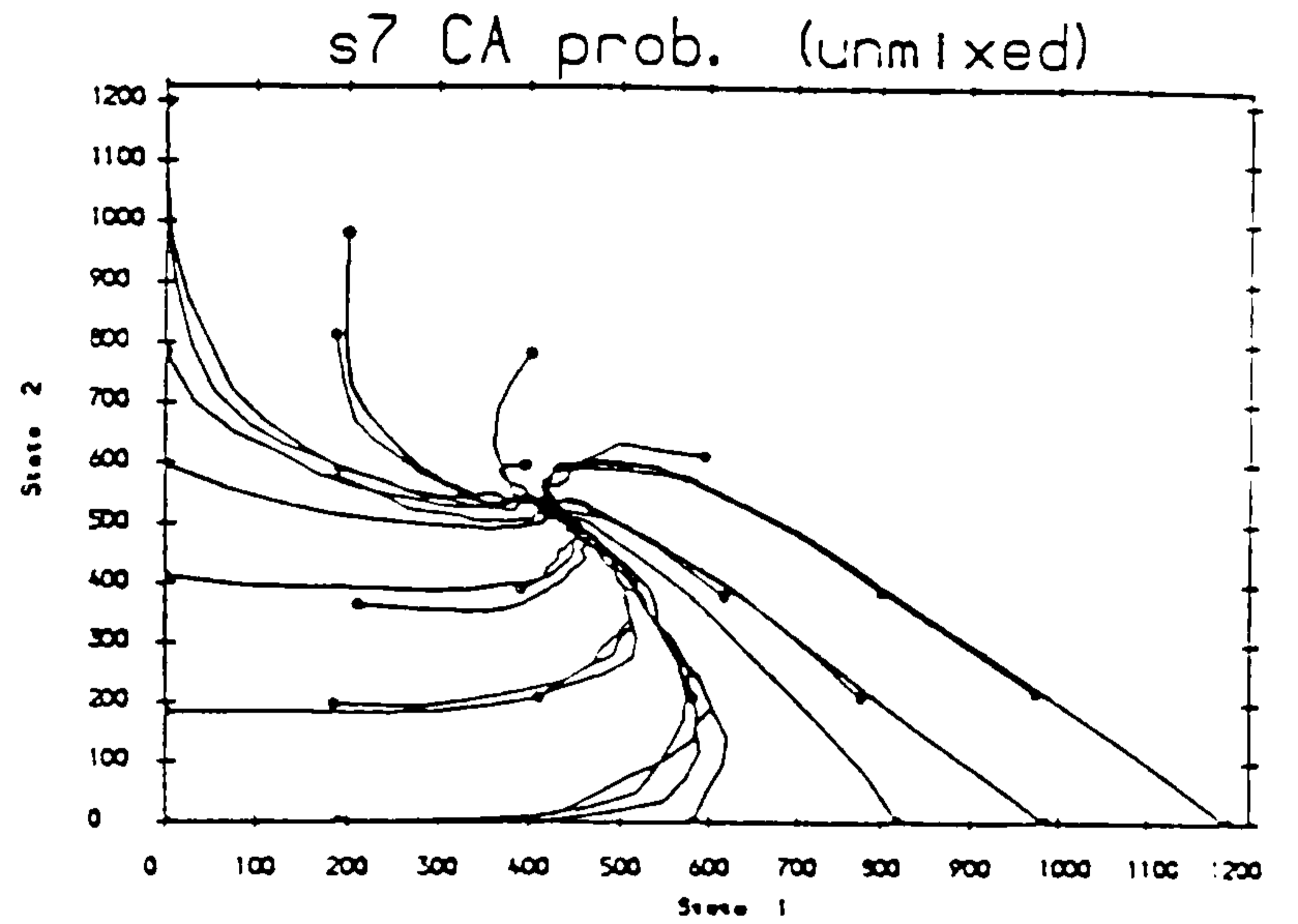
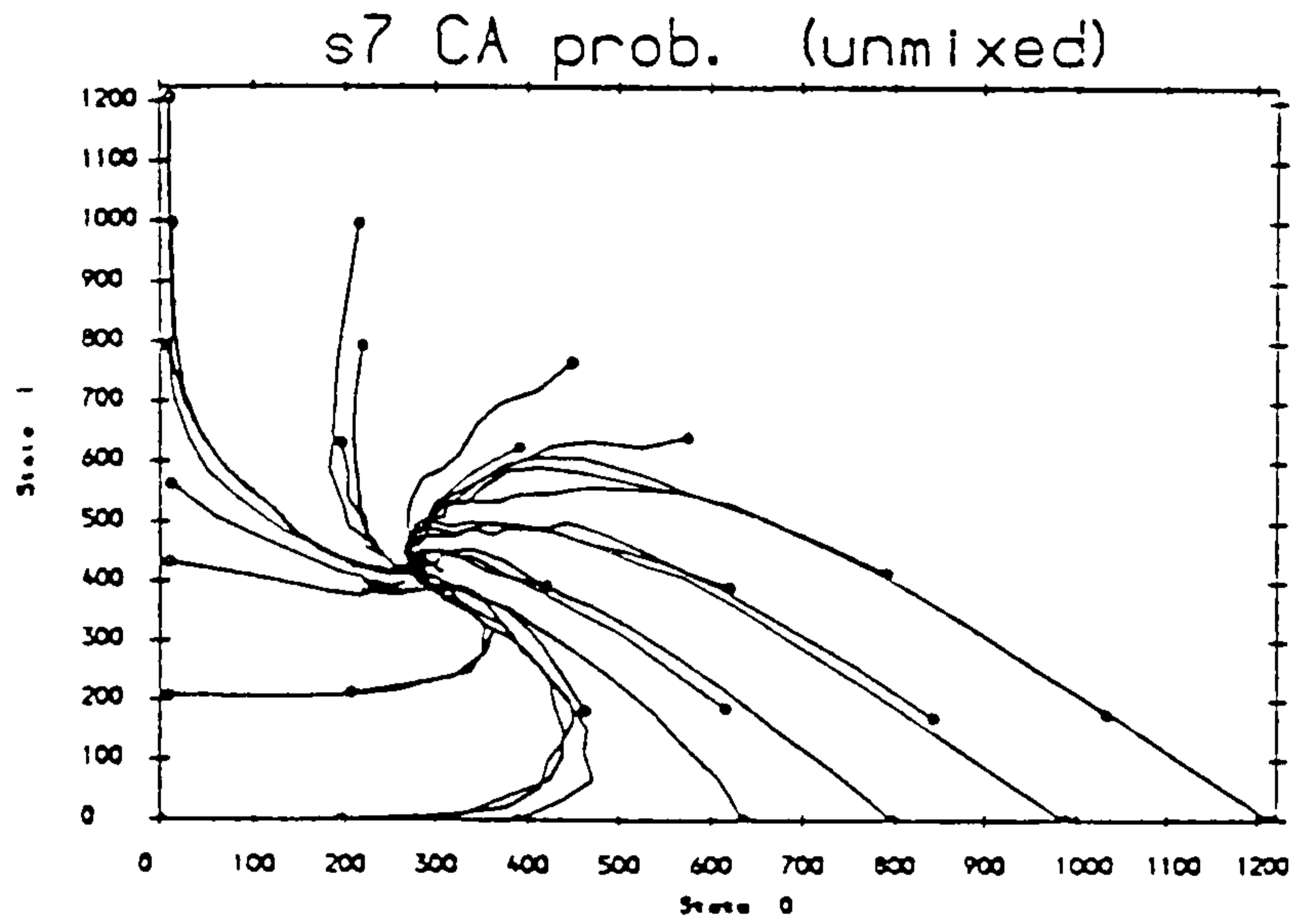


Figure 4.12: Threshold systems phase diagrams.  
(f) System 8: Probabilistic.

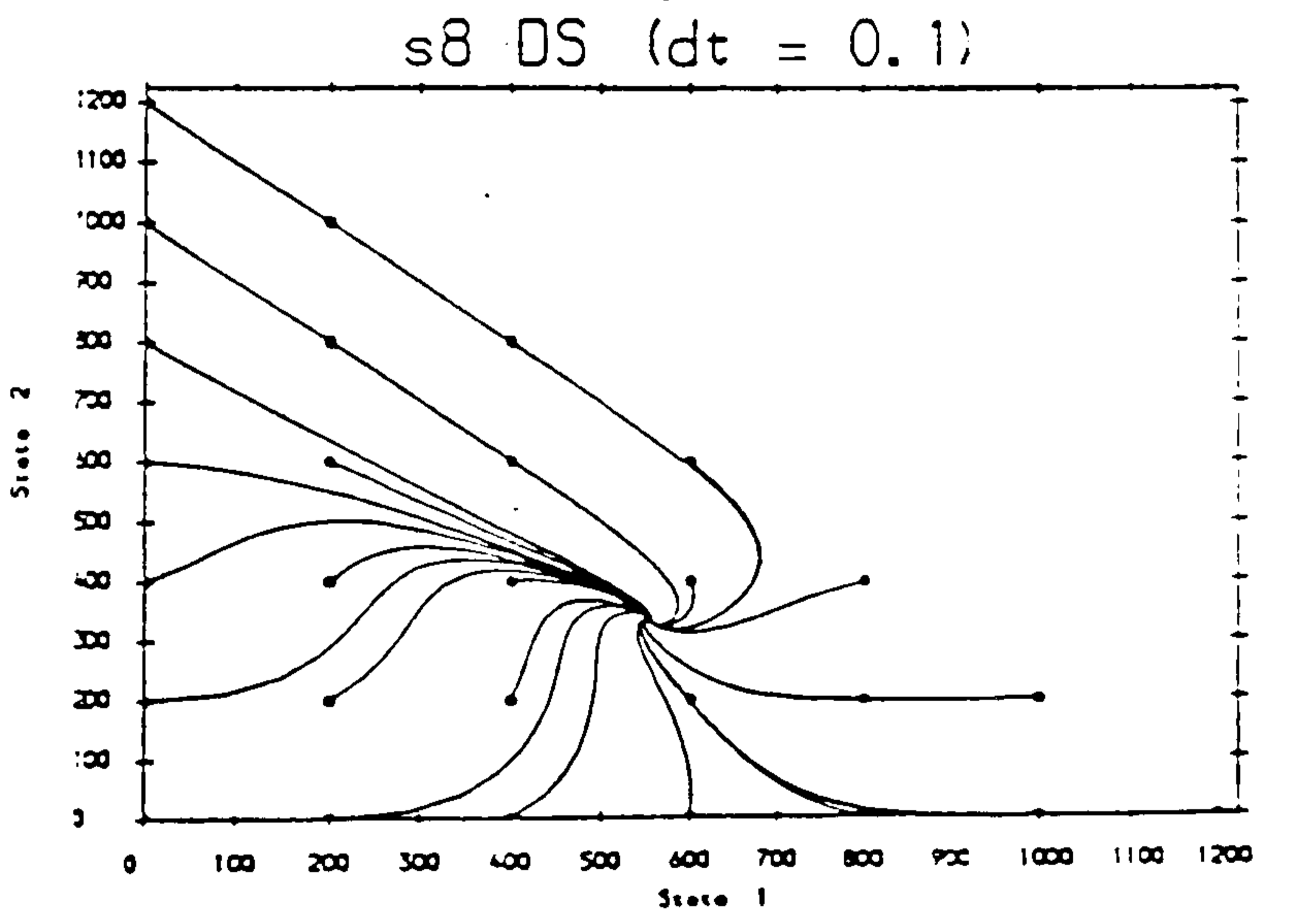
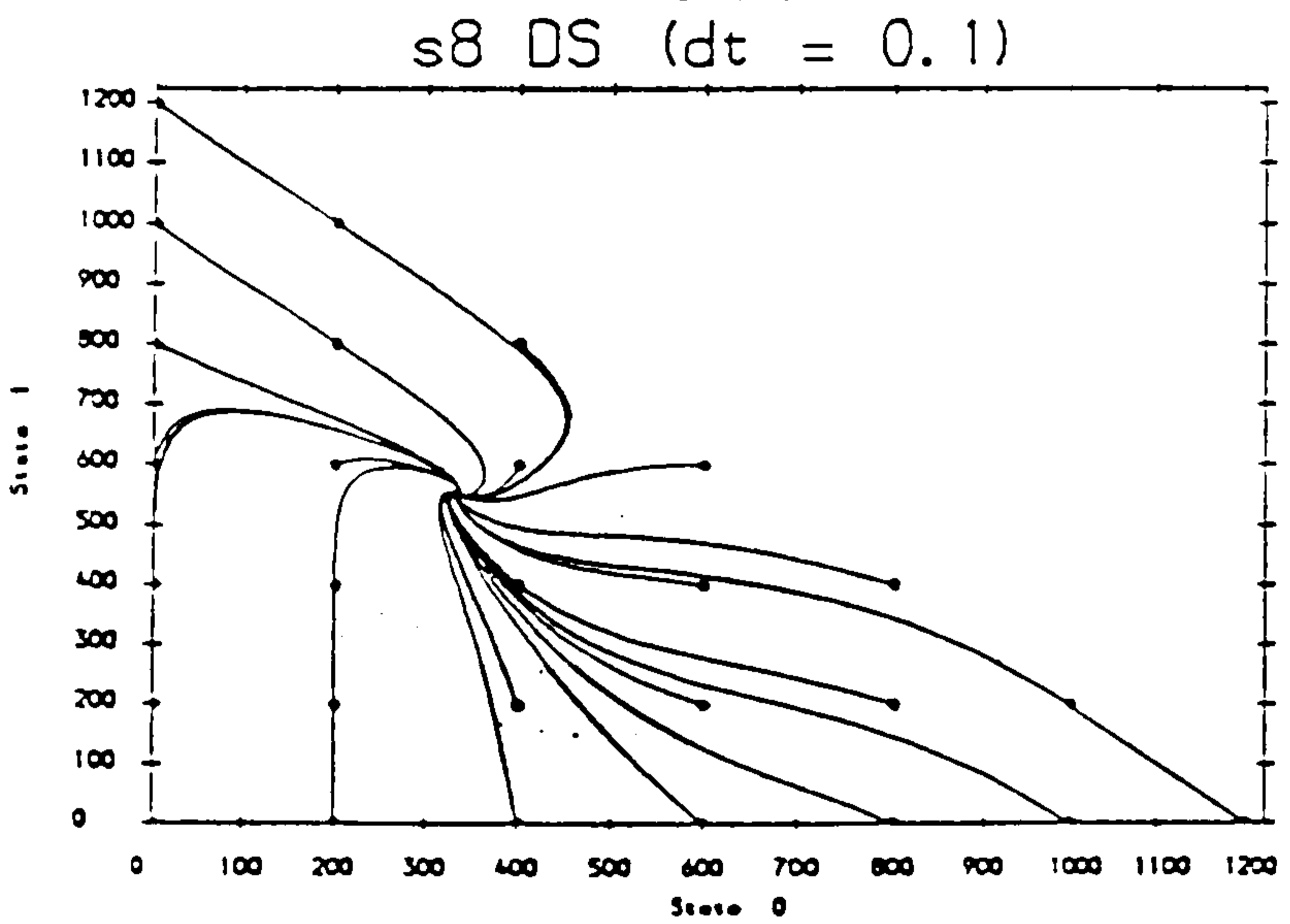
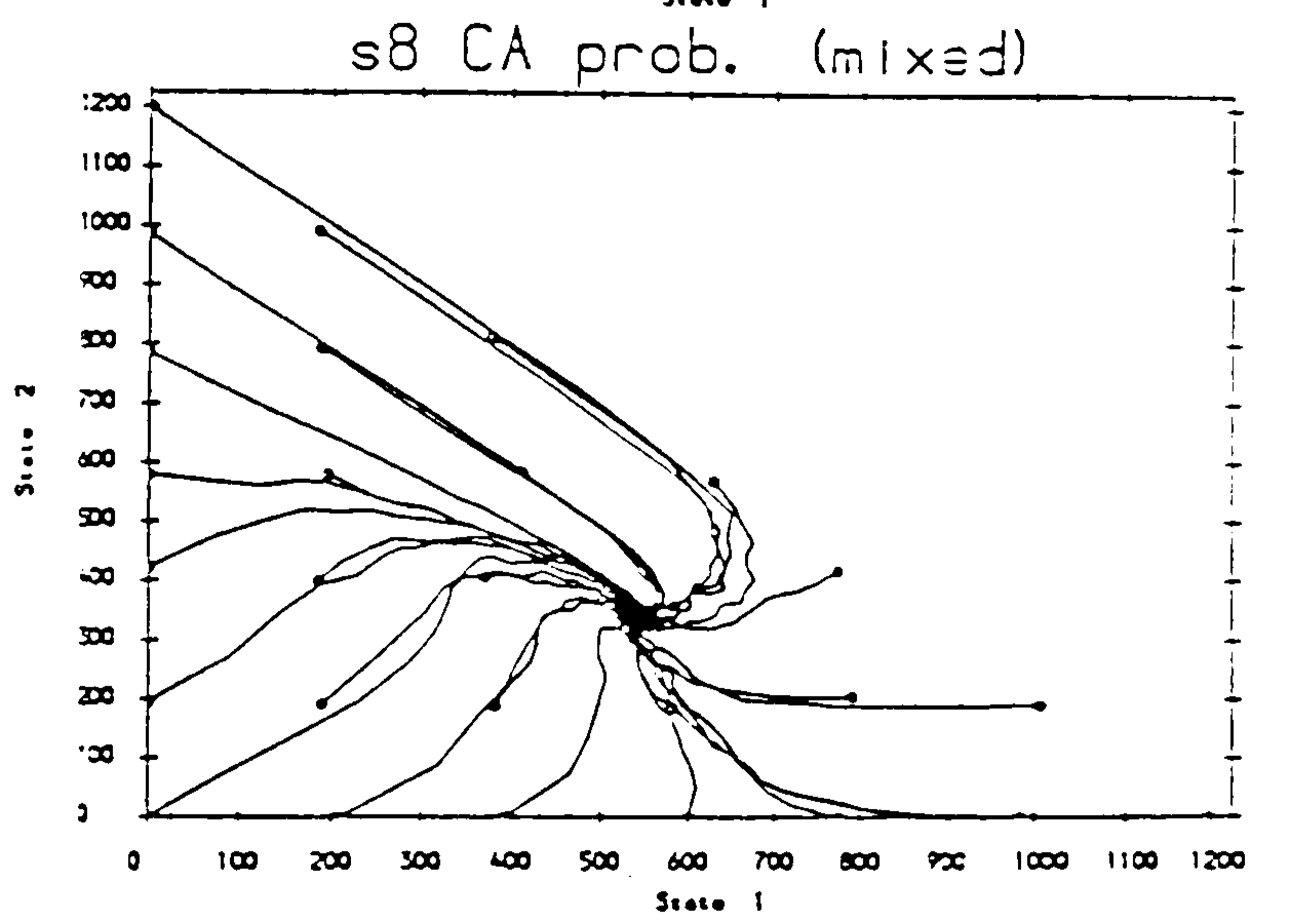
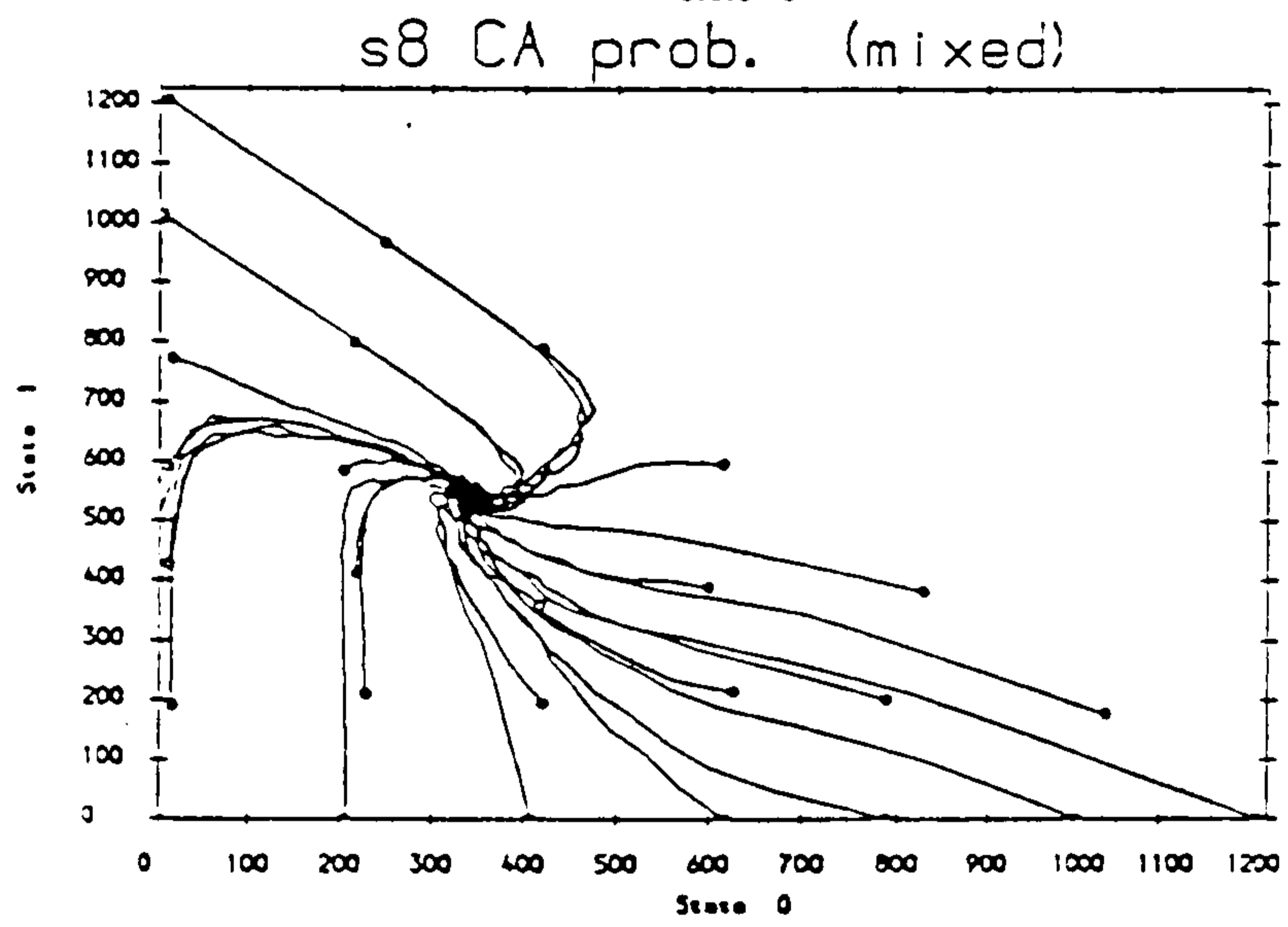
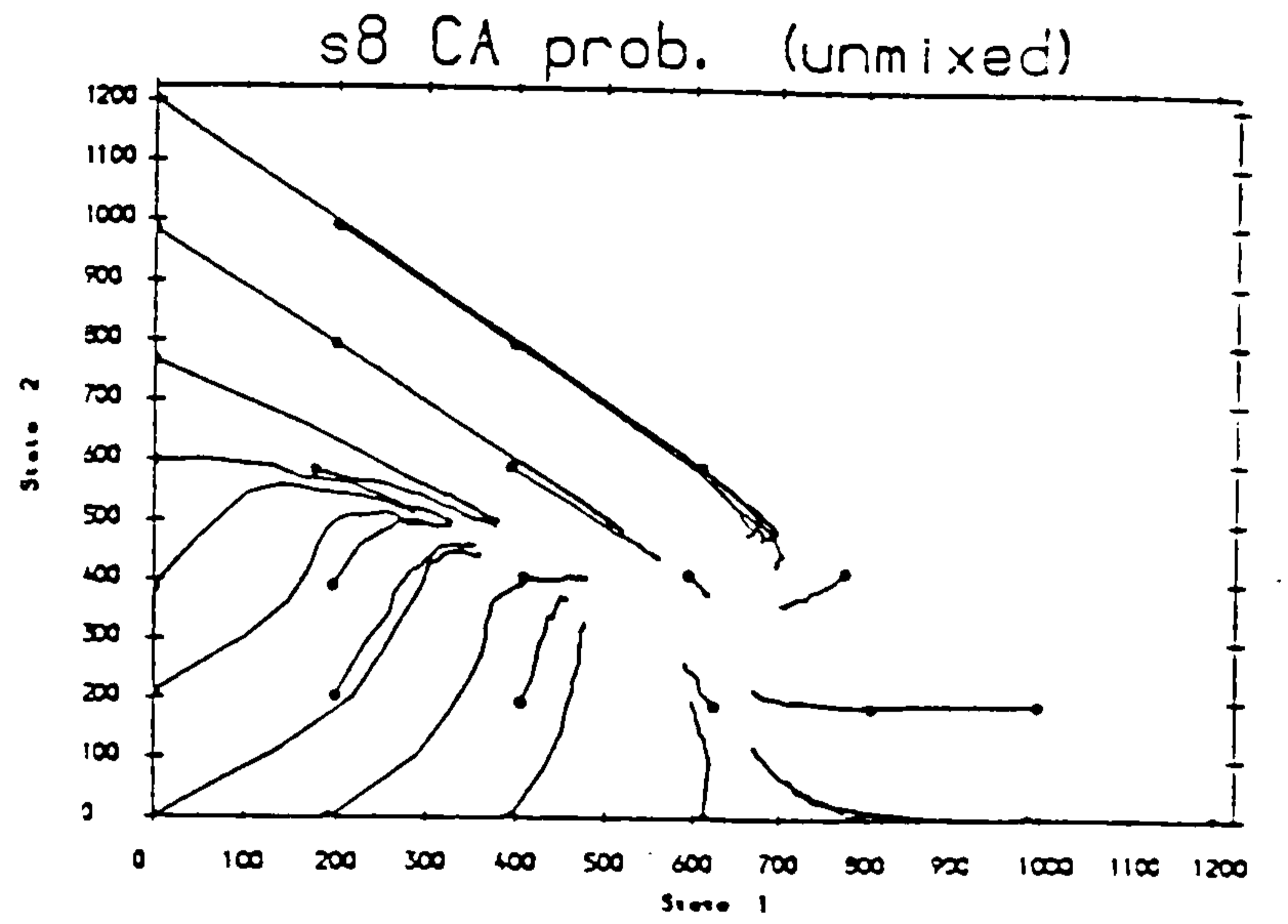
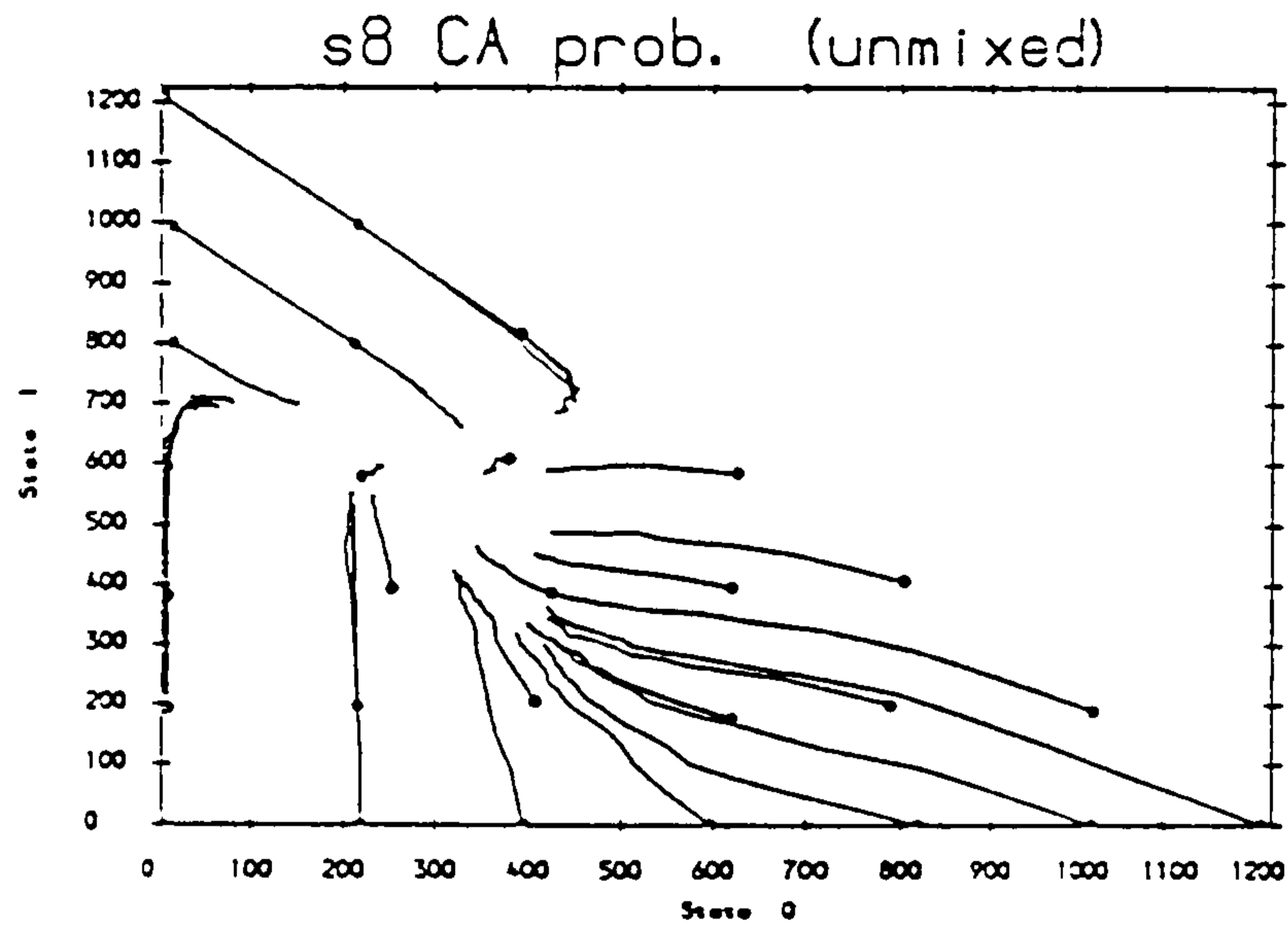
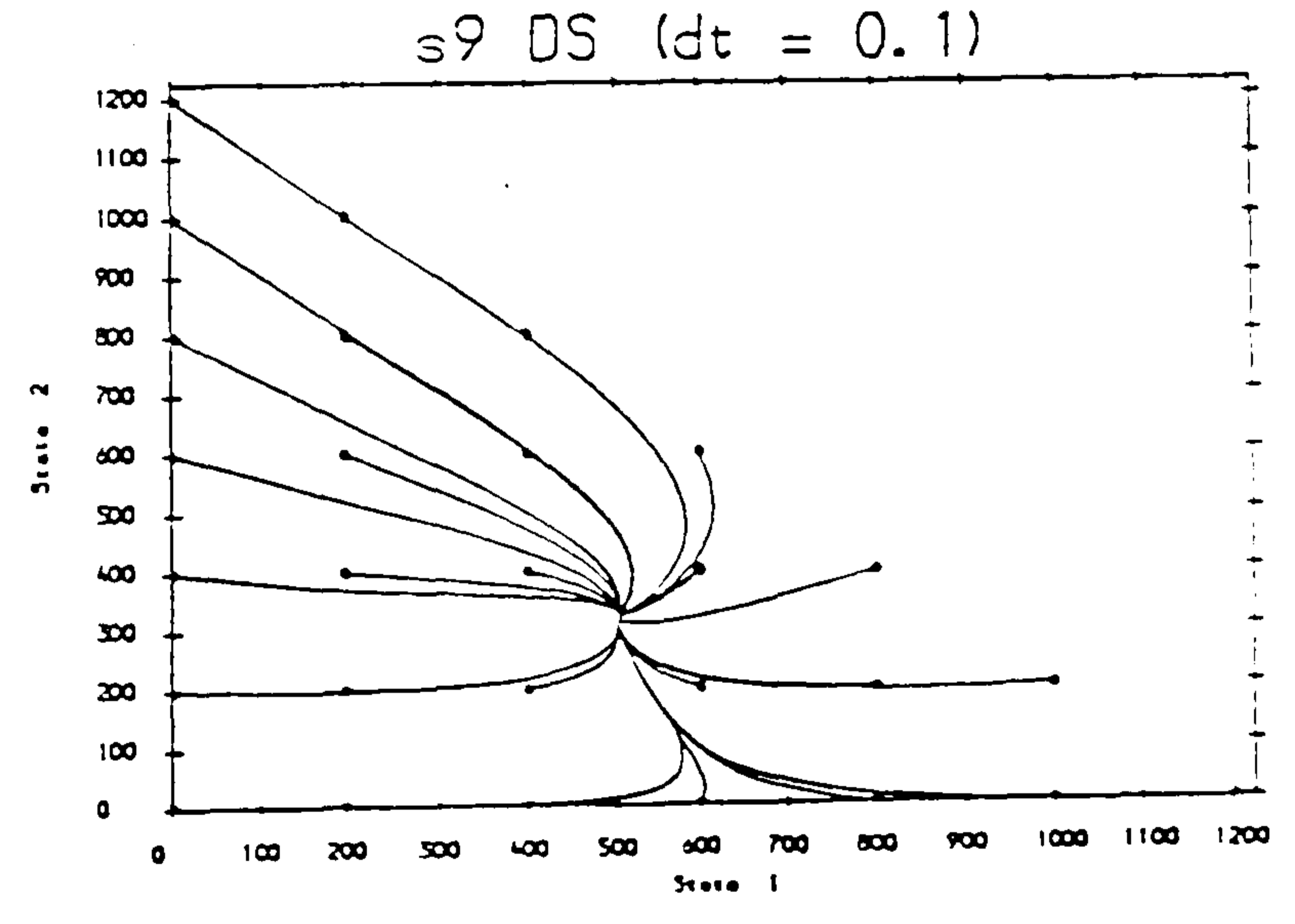
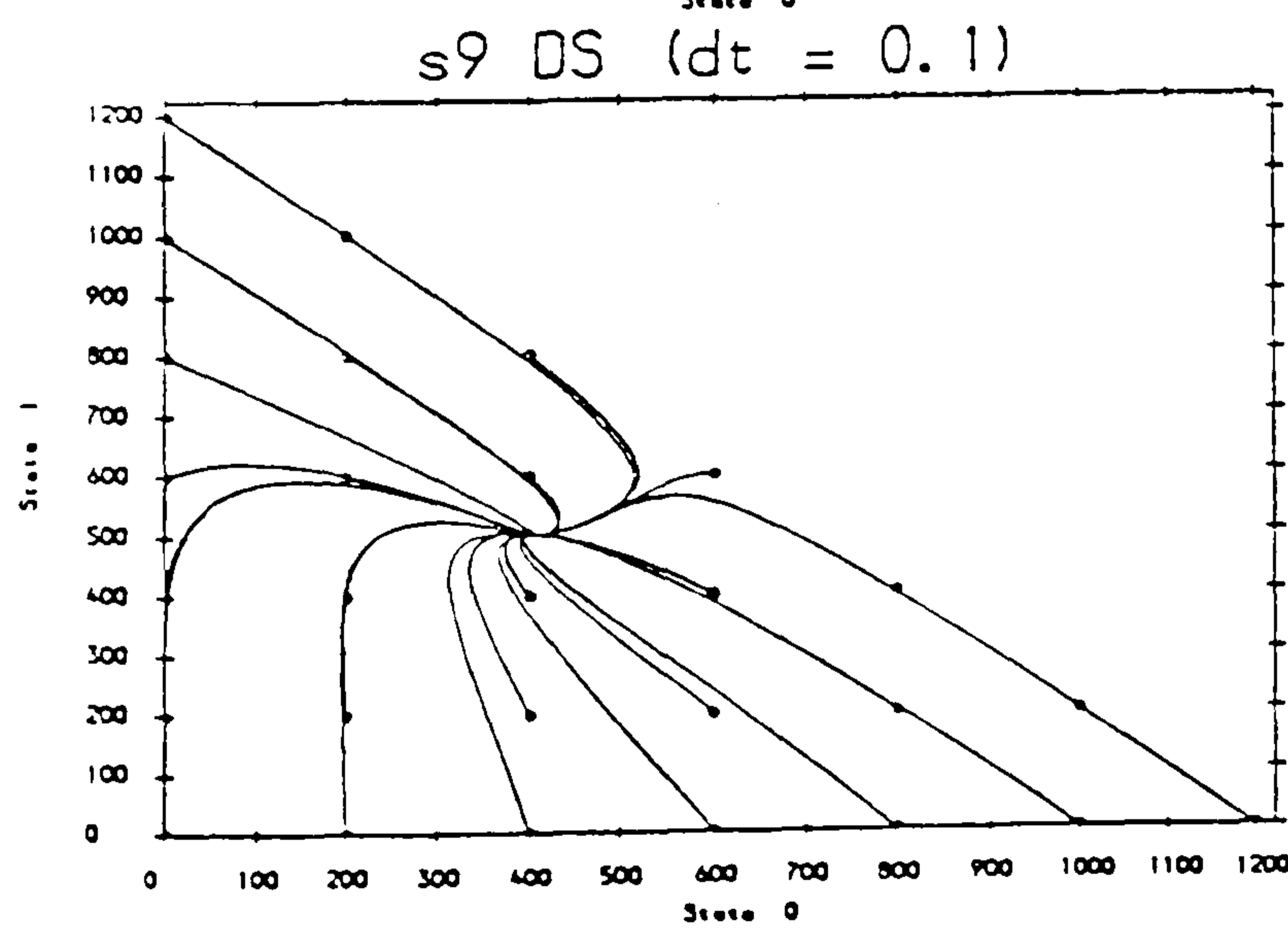
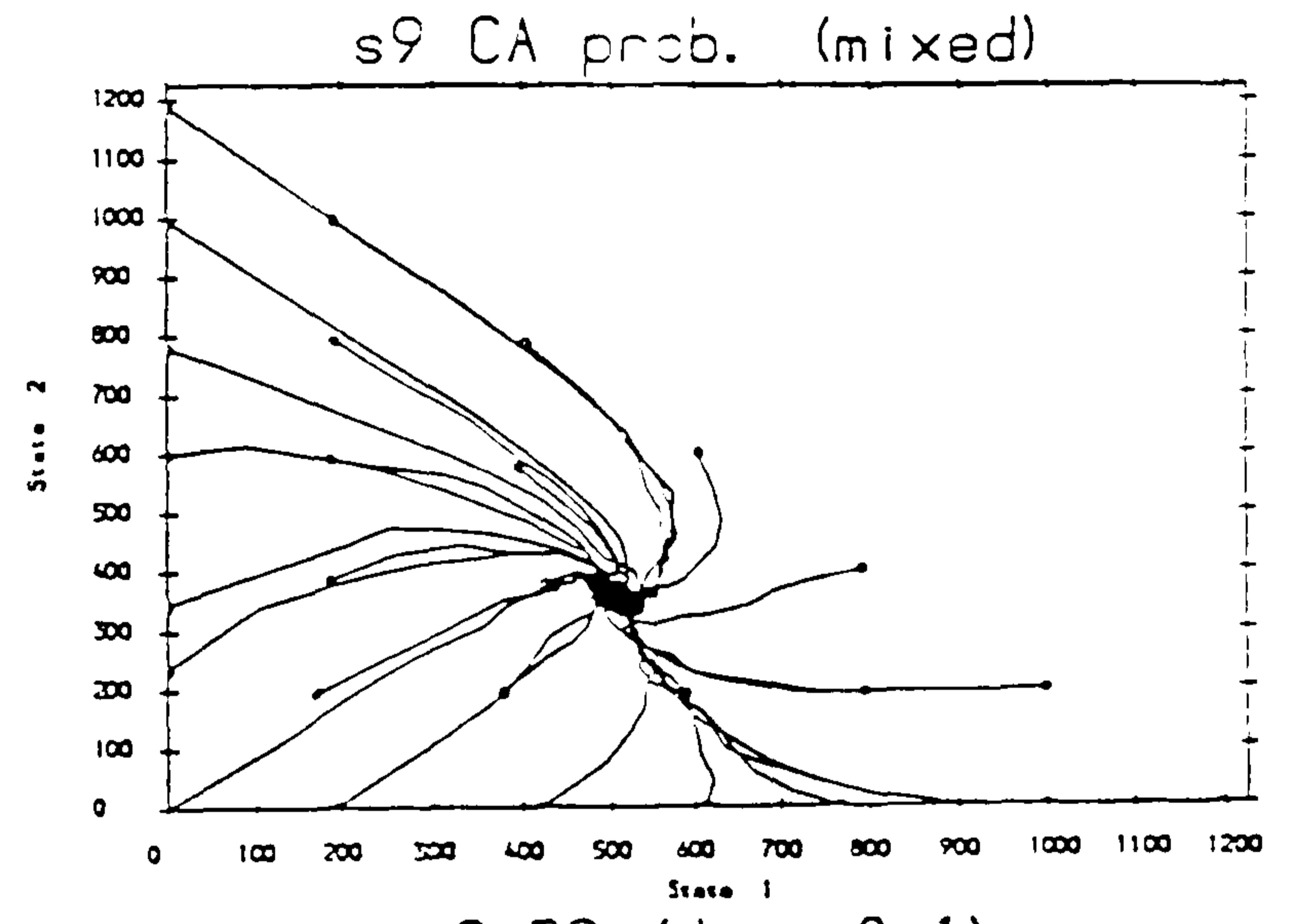
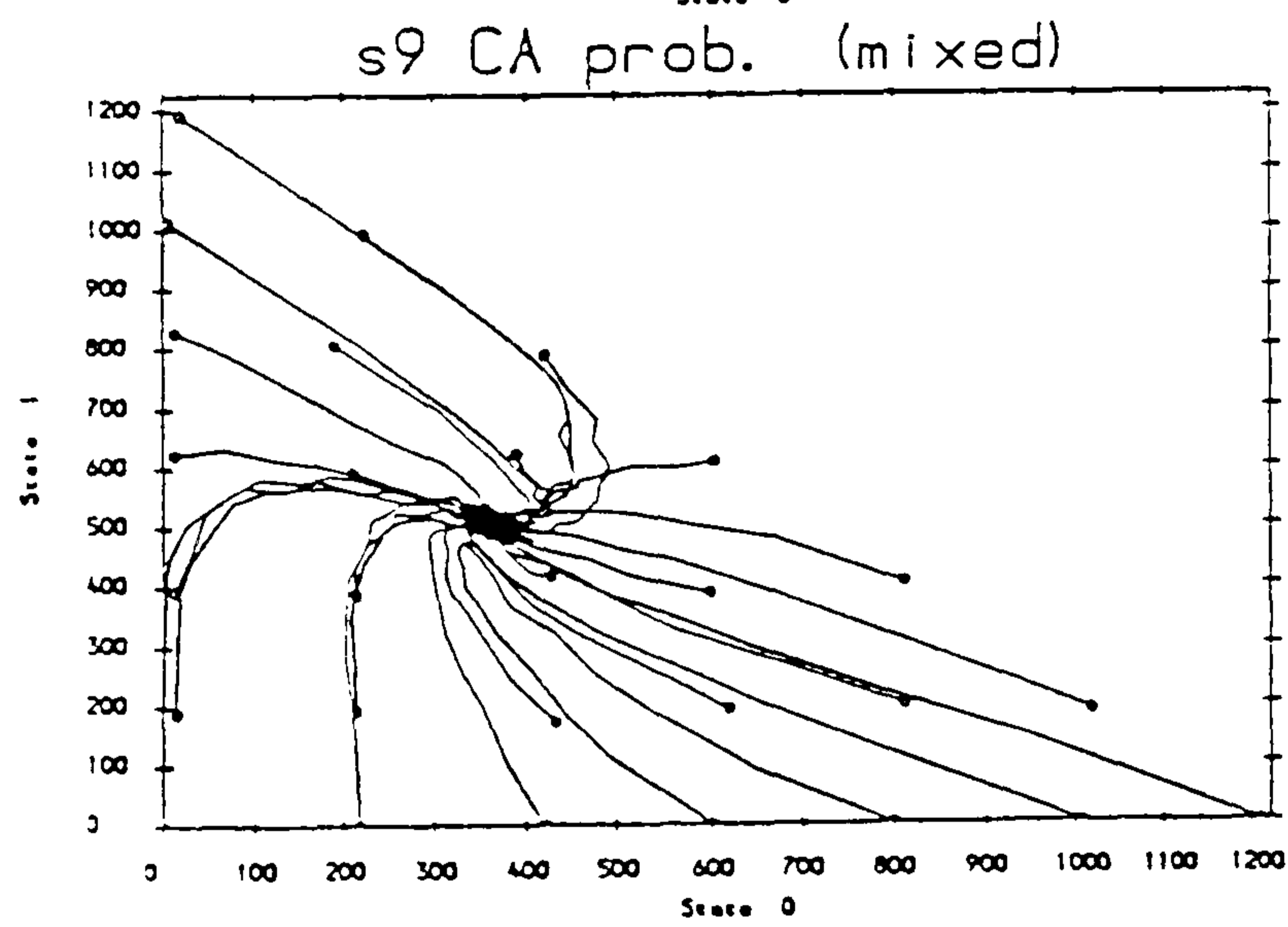
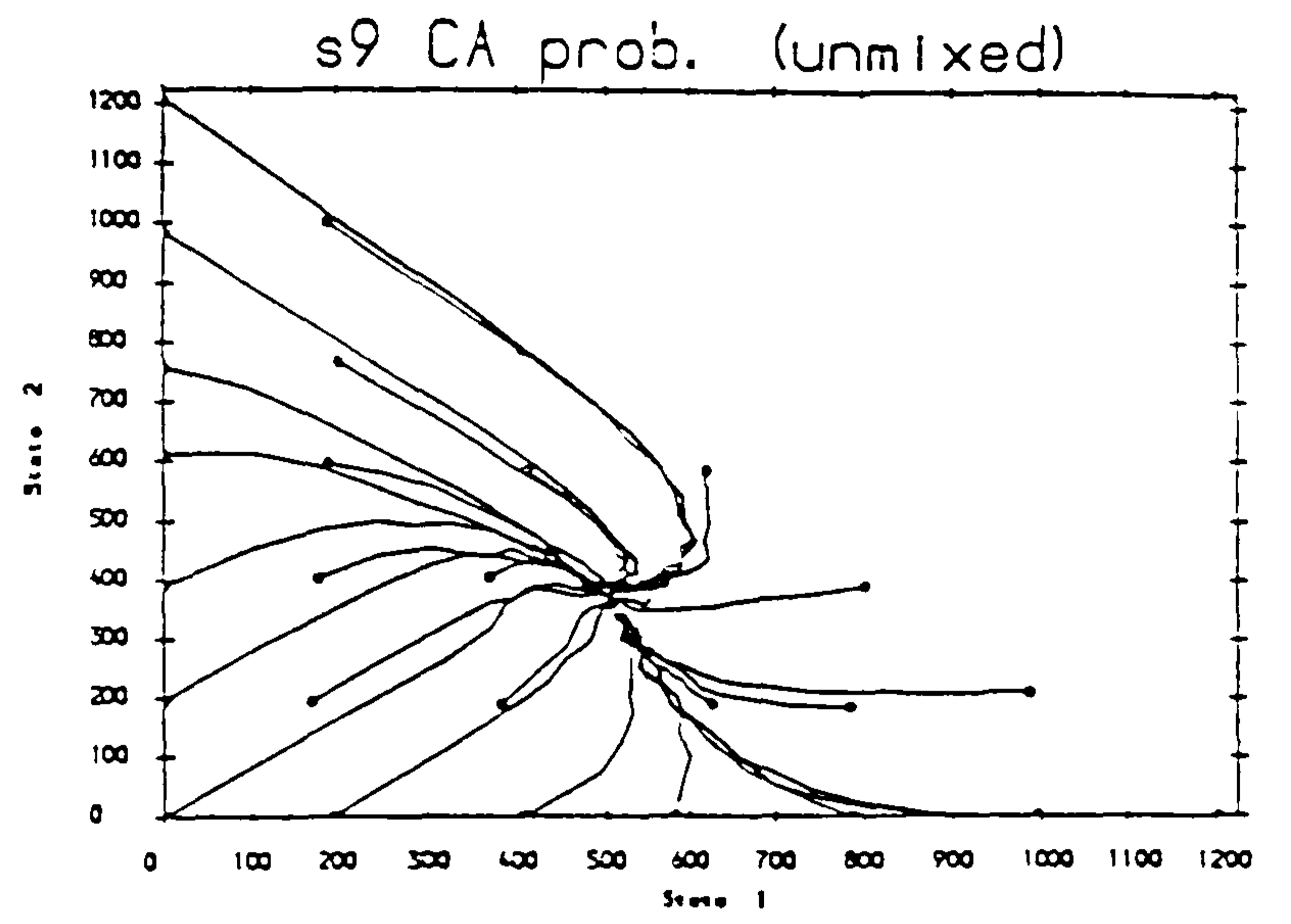
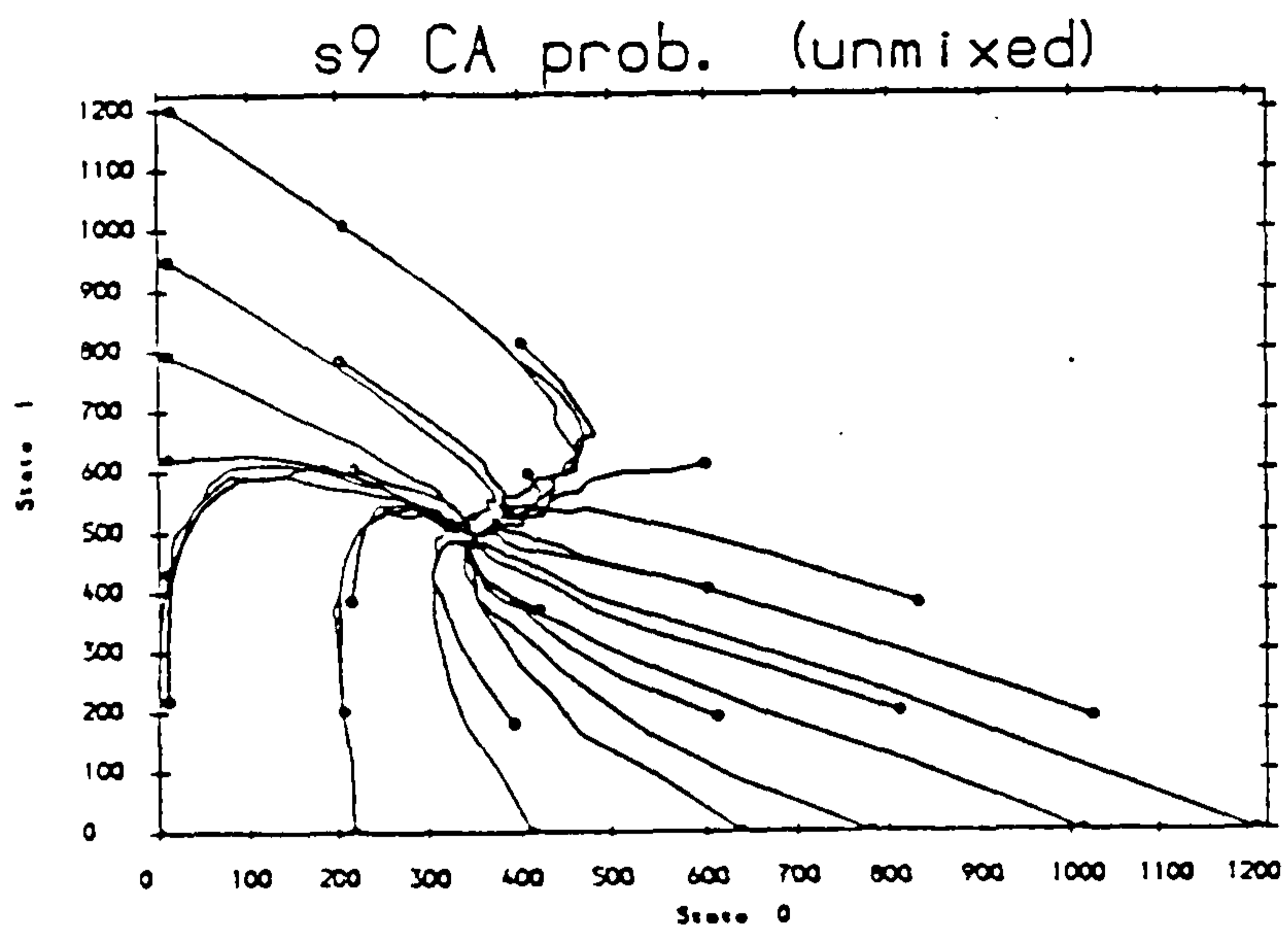




Figure 4.12: Threshold systems phase diagrams.  
(g) System 9: Probabilistic.



vector field maps in Figure 4.11. The trajectories are straight lines of slope  $-1$  because an increase in one state must be balanced by an equivalent decrease in the other state. Finally we can see from the rules that  $s_0$  changes to  $s_1$  with a greater probability than the reverse change and so more initial states lead to survival of  $s_1$  than  $s_0$ .

System 4 is our first non-trivial three state system. We can perform a qualitative analysis of the expected behaviour by considering the state transition diagram. Firstly we can see that state  $s_0$  will be favoured over the other states because it only requires a neighbourhood count of 1 or more cells in state  $s_1$  in order for its population to increase at the expense of state  $s_2$ . Now given that the  $s_0$  population will tend to be large we can see that this will favour state  $s_2$  over state  $s_1$  since the condition of 5 or more cells in state  $s_0$  will be more likely than 3 or more cells in state  $s_2$ . Therefore we can predict the limiting distribution will consist of a large  $s_0$  population and small  $s_1$  and  $s_2$  populations of which the  $s_2$  population is likely to be the larger. In order to determine the behaviour of this system in a more quantitative manner we can apply techniques from dynamical systems theory. The most important question is does the system reach a stable limit point or an oscillatory limit cycle. This can be determined by applying *Bendixons Negative Criterion* [85, 88],

There are no closed paths in a simply-connected domain of the phase plane on which  $\delta\dot{x}/\delta x + \delta\dot{y}/\delta y$  is of one sign.

Applying this to system 4 we obtain the following differentials,

$$\frac{\delta\dot{n}_0}{\delta n_0} = -P_{0,1}(3, n_2) \quad 4.4$$

$$\frac{\delta\dot{n}_1}{\delta n_1} = -P_{1,2}(5, n_0) \quad 4.5$$

$$\frac{\delta\dot{n}_2}{\delta n_2} = -P_{2,0}(1, n_1) \quad 4.6$$

So we see that all the differentials are always negative since the probabilities are always positive. The Bendixon expression is thus always negative and so we expect no closed paths in the phase space. The system must therefore behave chaotically or tend to a set of limit points.

We may determine the behaviour of the system near a limit point by the method of linearisation[85]. Using this method we assume that the system is approximately linear near the limit point. We can then write the equivalent linear system as follows. Firstly we shift the origin to the limit point giving us the new system variables  $m_0$ ,  $m_1$  and  $m_2$ .

The linear system is then,

$$\dot{m}_0 = a_0 m_0 + b_0 m_1 + c_0 m_2 \quad 4.7$$

$$\dot{m}_1 = a_1 m_0 + b_1 m_1 + c_1 m_2 \quad 4.8$$

$$\dot{m}_2 = a_2 m_0 + b_2 m_1 + c_2 m_2 \quad 4.9$$

where,

$$a_i = \frac{\delta \dot{m}_i(0,0,0)}{\delta m_0} \quad 4.10$$

$$b_i = \frac{\delta \dot{m}_i(0,0,0)}{\delta m_1} \quad 4.11$$

$$c_i = \frac{\delta \dot{m}_i(0,0,0)}{\delta m_2} \quad 4.12$$

Now for system 4 we obtain the following expressions for  $a_i$ ,  $b_i$  and  $c_i$ ,

$$\begin{aligned} a_0 &= -P_{0,1}(3, m_2) & b_0 &= P_{0,1}(3, m_2) & c_0 &= 0 \\ a_1 &= 0 & b_1 &= -P_{1,2}(5, m_0) & c_1 &= P_{1,2}(5, m_0) \\ a_2 &= P_{2,0}(1, m_1) & b_2 &= 0 & c_2 &= -P_{2,0}(1, m_1) \end{aligned} \quad 4.13$$

Taking solutions to this linear system to be of the form,



$$m_i = r_i \exp(\lambda t) \quad 4.14$$

We therefore obtain three simultaneous equations,

$$(a_0 - \lambda)r_0 + b_0 r_1 + c_0 r_2 = 0 \quad 4.15$$

$$a_1 r_0 + (b_1 - \lambda)r_1 + c_1 r_2 = 0 \quad 4.16$$

$$a_2 r_0 + b_2 r_1 + (c_2 - \lambda)r_2 = 0 \quad 4.17$$

which have non-trivial solutions if and only if,

$$\begin{bmatrix} (a_0 - \lambda) & b_0 & c_0 \\ a_1 & (b_1 - \lambda) & c_1 \\ a_2 & b_2 & (c_2 - \lambda) \end{bmatrix} = 0 \quad 4.18$$

This then gives us a cubic for  $\lambda$ ,

$$\lambda^3 - (a_0 + b_1)\lambda^2 + (a_0 b_1)\lambda - (a_1 b_2 c_0) \quad 4.19$$

This can be solved using Cardan's Formula[89] for the roots of a cubic equation which tells us that equation 4.19 has three real roots none of which are equal. We can therefore say that the limit point is a node[85].

Another method of determining the type of behaviour near a limit point is to calculate the index of the limit point[85]. This can be defined for a two-dimensional system as the number of  $2\pi$  rotations of the vector field  $(\dot{x}, \dot{y})$  along a smooth, closed curve about the equilibrium point in the anti-clockwise direction. In practice we can relax the restriction for the curve to be smooth and take a square about the equilibrium point[85]. Applying this method to system 4 gives us an index for the limit point of 1.

If we now consider the actual behaviour of system 4 on the phase diagram we can study the relationship between the CA and the DS. The unmixed CA shows distinct cyclic

behaviour superimposed on the evolution to the limiting region. The reason for this is apparent if we consider the vector field map which shows a distinct cyclic pattern. When mixing is applied to the CA the oscillations are considerably reduced. This is due to the destruction of highly active patterned regions which normally form on the CA lattice. Furthermore the evolution does not proceed as far as for the unmixed CA. This can also be understood from the vector field map, because the oscillations are much smaller for the unmixed CA the system settles more quickly into the region where the rate of evolution is very small. Once in this region the evolution of the system rapidly halts. The DS behaviour is almost identical to that for the mixed CA as we would expect from the analysis in Chapter 3. However the evolution of the DS can proceed further because the rates of change of the state populations can decrease smoothly to zero whereas the CA will stabilise when the state of the lattice no longer allows state transitions. This will not necessarily be at the limiting values of the state populations but will be within some limiting region around the DS limit point. The size of the limiting region will depend on the degree of inhomogeneity of the lattice as it approaches its limiting state and on the relative sizes of the state populations. If the lattice is highly inhomogeneous then certain regions may cease evolving due to absence of a critical cell state, a similar argument applies if a critical cell state population is very low.

System 4 is also demonstrates well the equivalence established between reducing the time step  $\Delta t$  for the DS and reducing the rule action probability of the CA. As can be seen the effect is consistent for both the CA and the DS.

We now come to a group of three systems which are structurally identical but which have different transition probability thresholds. These systems incorporate a form of negative feedback in that the larger the size of a state population the greater the rate of loss of

cells in that state. For low transition probability thresholds cells will easily change from one state to the next. Since the state transitions form a loop we would expect cyclic behaviour with the high population state transferring cyclically from one state to the next. Conversely for high thresholds where cells will tend to remain in their present state we would expect cells to be gradually distributed amongst the states until a stable situation is reached. Thus we expect a transition from limit point behaviour for system 5 to period 3 limit cycle behaviour for system 7 and indeed this is what we observe.

However if we apply Bendixons Negative Criterion we obtain the following

$$\frac{\delta \dot{n}_0}{\delta n_0} = -(P_{0,1}(T_{0,1}, n_0) + n_0 \frac{\delta P_{0,1}(T_{0,1}, n_0)}{\delta n_0}) \quad 4.20$$

$$\frac{\delta \dot{n}_1}{\delta n_1} = -(P_{1,2}(T_{1,2}, n_1) + n_1 \frac{\delta P_{1,2}(T_{1,2}, n_1)}{\delta n_1}) \quad 4.21$$

$$\frac{\delta \dot{n}_2}{\delta n_2} = -(P_{2,0}(T_{2,0}, n_2) + n_2 \frac{\delta P_{2,0}(T_{2,0}, n_2)}{\delta n_2}) \quad 4.22$$

These equations imply that no limit cycles are possible in the phase space of this group of systems. However Bendixon's Negative Criterion applies to continuous systems and our CA and their equivalent DS are discrete systems. If we examine the behaviour of the DS when the time step is made very small then we find that the systems all have limit point behaviour. The limit point being at the centre of the discrete systems limit cycle. We can understand this behaviour if we examine the vector field map. A typical system which exhibits this type of behaviour will have a vector field map on which the vectors or the rate of change of the system variables are high near the edges of the phase diagram, and decrease towards the central limit point. The vectors will also be parallel to the edges of the phase diagram, especially near the corners, but point progressively inwards as the central limit point is approached. Now if the value of  $\Delta t$  is such that the change in the



system variables causes the system to move further from the central limit point into a region of even greater rate of change then the system will tend to spiral to the edges of the phase diagram and settle into saturated oscillations. However if  $\Delta t$  is such that the change in the system variables moves the system into a region of lower rate of change then the system will tend to spiral inwards towards the central limit point. There will obviously be an intermediate range of  $\Delta t$  for which some initial points on the phase diagram lead to the saturated limit cycle and others which lead to the central limit point.

We can confirm the behaviour of these systems in their continuous form near the limit point by the method of linearisation used for system 4. The expressions we obtain for  $a_i$ ,  $b_i$  and  $c_i$  are as follows,

$$\begin{aligned}
 a_0 &= -P_{0,1}(T_{0,1}, m_0) & b_0 &= P_{0,1}(T_{0,1}, m_0) & c_0 &= 0 \\
 a_1 &= 0 & b_1 &= -P_{1,2}(T_{1,2}, m_1) & c_1 &= P_{1,2}(T_{1,2}, m_1) \\
 a_2 &= P_{2,0}(T_{2,0}, m_2) & b_2 &= 0 & c_2 &= -P_{2,0}(T_{2,0}, m_2)
 \end{aligned} \tag{4.13}$$

The relationship between the coefficients is identical to that for system 4 so the analysis proceeds in exactly the same way. For systems 5, 6 and 7 this leads to the conclusion that the limit point is a centre.

Furthermore, in the same way as for system 4 we can check the limit point behaviour by calculating the index of the limit point.

This group of systems also shows the excellent agreement between the behaviour of the CA and the equivalent dynamical system. This is especially true for system 7 for which both the CA and DS change from cyclic behaviour to limit point behaviour as the effective time step is reduced.

We now come to a group of two systems which are very similar to the previous group

except that an extra state transition has been introduced and the direction of the main transition loop has been reversed. We therefore expect similar behaviour to that for systems 5, 6 and 7. The secondary transition between states 0 and 1 will tend to increase the relative numbers of cells which remain in one of these two states. This is confirmed by the shift in the positions of the central limit point relative to the previous group of systems. Furthermore, the secondary transition will introduce a period two cycle which will be superimposed on the main period three cycle. On examining the phase diagrams we observe that these points are indeed confirmed. The secondary transition has indeed complicated the limit cycle structure which shows strong period two behaviour combined with the original period three limit cycle. This is especially true for system 9 where the period 2 cycle is the more pronounced. Comparison of the trajectories which spiral into the limit point in the previous group and this group show that the trajectories for this group do indeed spiral in the opposite direction due to the reversing of the transition loop.

Applying Bendixon's Negative Criterion to this group reveals the same problem as with the previous group. The equations we obtain are as follows,

$$\frac{\delta \dot{n}_0}{\delta n_0} = -(P_{0,2}(T_{0,2}, n_0) + n_0 \frac{\delta P_{0,2}(T_{0,2}, n_0)}{\delta n_0} + P_{0,1}(T_{0,1}, n_0) + n_0 \frac{\delta P_{0,1}(T_{0,1}, n_0)}{\delta n_0}) \quad 4.24$$

$$\frac{\delta \dot{n}_1}{\delta n_1} = -(P_{1,0}(T_{1,0}, n_1) + n_1 \frac{\delta P_{1,0}(T_{1,0}, n_1)}{\delta n_1}) \quad 4.25$$

$$\frac{\delta \dot{n}_2}{\delta n_2} = -(P_{2,1}(T_{2,1}, n_2) + n_2 \frac{\delta P_{2,1}(T_{2,1}, n_2)}{\delta n_2}) \quad 4.26$$

These equations again imply that no limit cycles are possible in the phase space of this group of systems. However, as noted for the previous group, Bendixon's Negative Criterion applies to continuous systems. If we examine the continuous phase diagrams of

systems 8 and 9 we find that they do indeed evolve to limit points.

The agreement between the CA and DS is once again excellent.

In summary therefore we can state that the state population dynamics of threshold CA can be modelled very accurately by our dynamical system formulation.

All of the systems described above have either limit cycle or limit point behaviour. They therefore belong to classes 1 or 2 of Wolframs classification system. These are the two simplest CA classes, we must therefore now consider the two more complex CA classes, 3 and 4. Firstly we will consider Class 4 because, as was noted in Chapter 1, Class 4 can be considered an intermediate stage between the periodic behaviour of Class 2 and the chaotic behaviour of Class 3.

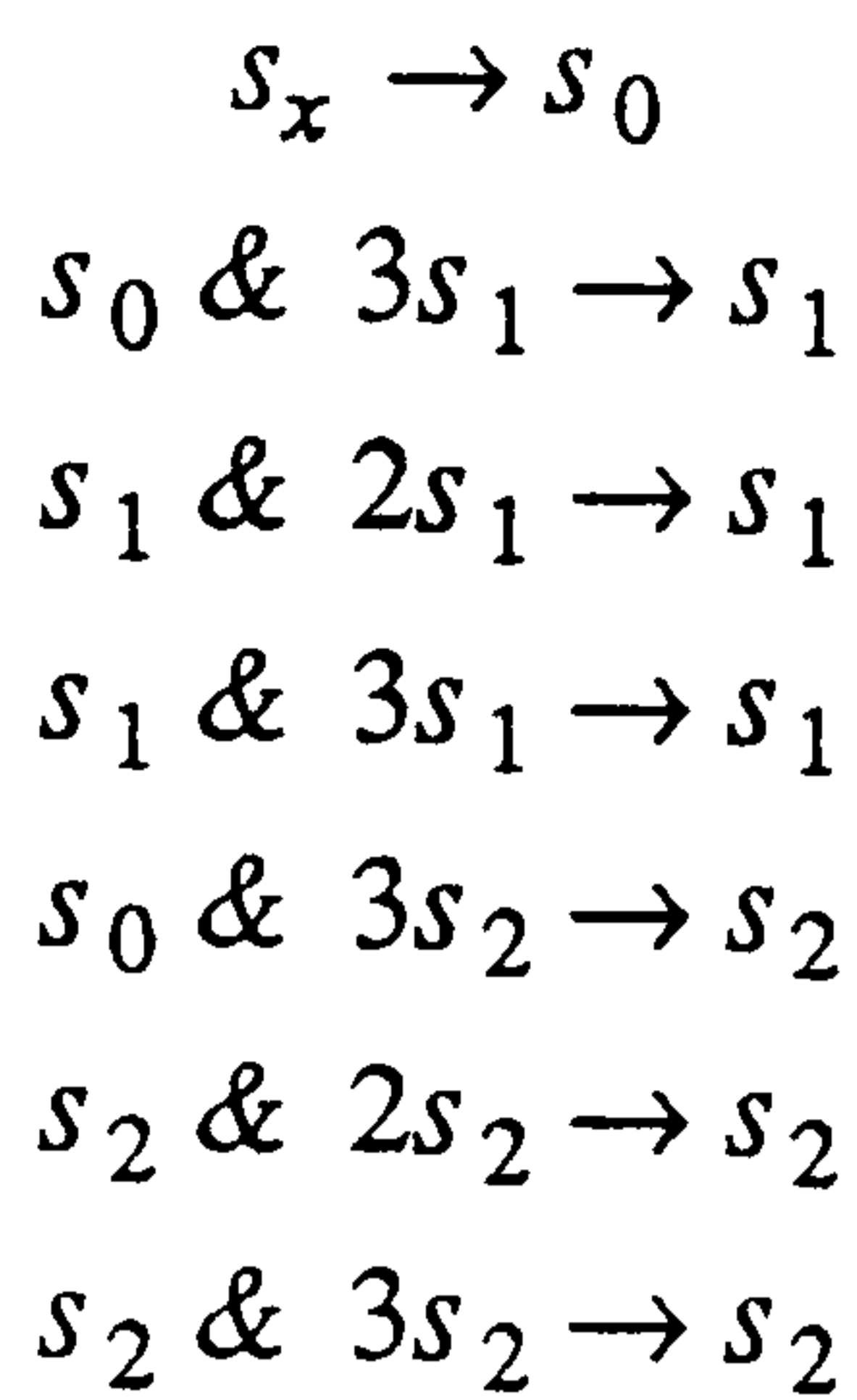
Now Class 4 CA are very rare especially in dimensions greater than one[44]. Thus our study of Class 4 CA will be based on the Twolife CA introduced in Chapter 2 together with simple variations of this CA. These systems are summarised in Figure 4.13, their state transition diagrams in Figure 4.14, vector field diagrams in Figure 4.15 and phase diagrams in Figure 4.16.

System A is the original Twolife CA. For system B the transitions from the quiescent state  $s_0$  to the two population states  $s_1$  and  $s_2$  depend on the number of cells in the other population state rather than on the state to which the transition is occurring. System C has the same alteration to the  $s_0$  to  $s_1$  and  $s_2$  transitions and the active population state survival rule becomes a transition between the active population states. System D has the latter alteration to system C but the former alteration is removed. The similarities and differences of these four systems are more easily compared by studying the state transition diagrams.



Figure 4.13: Class 4 systems summary (a).

**System A**

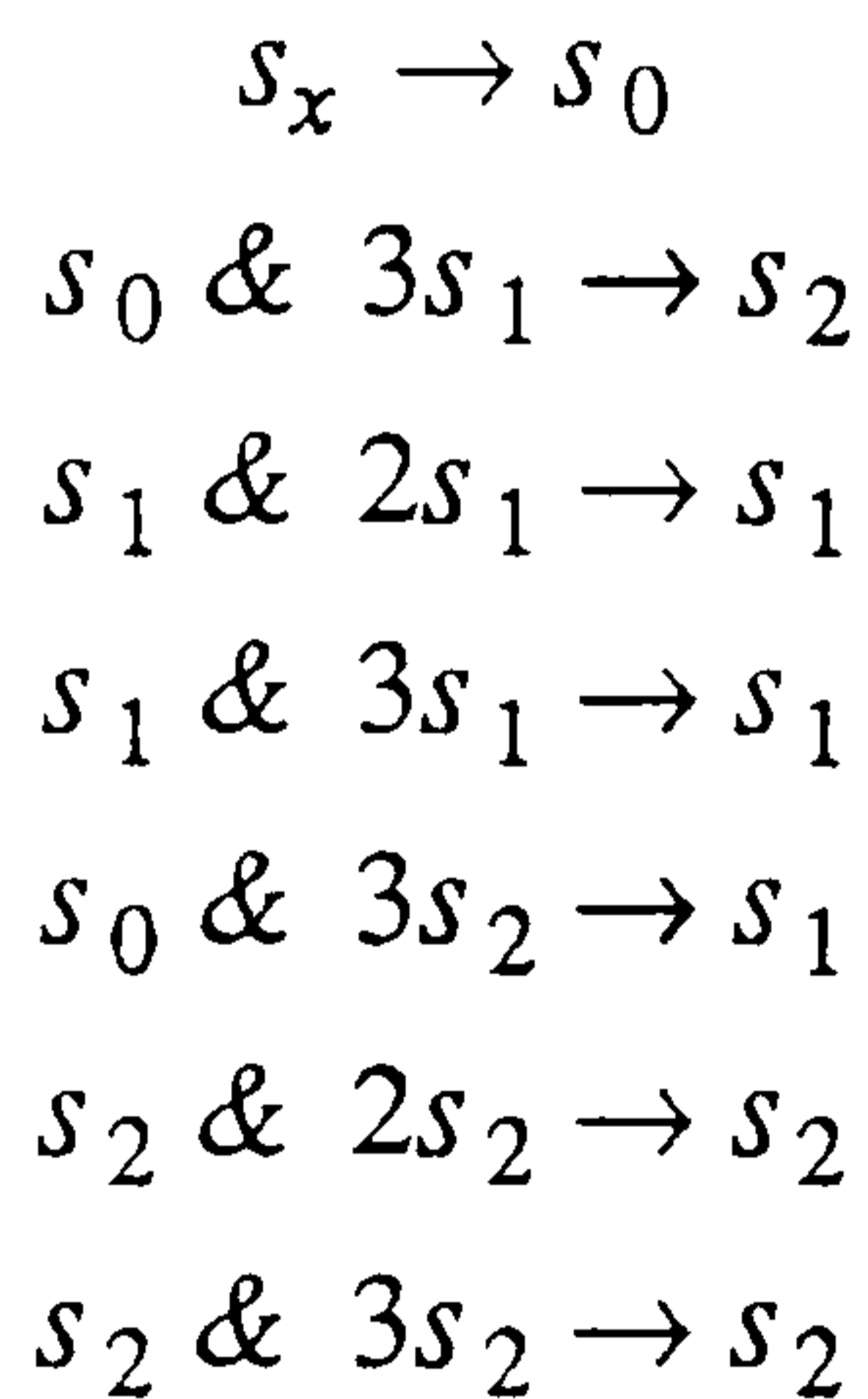


$$\dot{n}_0 = n_1 - P_{1,1}(2,n_1)n_1 - P_{1,1}(3,n_1)n_1 + n_2 - P_{2,2}(2,n_2)n_2 - P_{2,2}(3,n_2)n_2 - P_{0,1}(3,n_1)n_0 - P_{0,2}(3,n_2)n_0$$

$$\dot{n}_1 = P_{0,1}(3,n_1)n_0 - n_1 + P_{1,1}(2,n_1)n_1 + P_{1,1}(3,n_1)n_1$$

$$\dot{n}_2 = P_{0,2}(3,n_2)n_0 - n_2 + P_{2,2}(2,n_2)n_2 + P_{2,2}(3,n_2)n_2$$

**System B**



$$\dot{n}_0 = n_1 - P_{1,1}(2,n_1)n_1 - P_{1,1}(3,n_1)n_1 + n_2 - P_{2,2}(2,n_2)n_2 - P_{2,2}(3,n_2)n_2 - P_{0,1}(3,n_2)n_0 - P_{0,2}(3,n_1)n_0$$

$$\dot{n}_1 = P_{0,1}(3,n_2)n_0 - n_1 + P_{1,1}(2,n_1)n_1 + P_{1,1}(3,n_1)n_1$$

$$\dot{n}_2 = P_{0,2}(3,n_1)n_0 - n_2 + P_{2,2}(2,n_2)n_2 + P_{2,2}(3,n_2)n_2$$

Figure 4.13: Class 4 systems summary (b).

**System C**

$$s_x \rightarrow s_0$$

$$s_0 \& 3s_1 \rightarrow s_2$$

$$s_1 \& 2s_1 \rightarrow s_2$$

$$s_1 \& 3s_1 \rightarrow s_2$$

$$s_0 \& 3s_2 \rightarrow s_1$$

$$s_2 \& 2s_2 \rightarrow s_1$$

$$s_2 \& 3s_2 \rightarrow s_1$$

$$\dot{n}_0 = n_1 - P_{1,2}(2,n_1)n_1 - P_{1,2}(3,n_1)n_1 + n_2 - P_{2,1}(2,n_2)n_2 - P_{2,1}(3,n_2)n_2 - P_{0,1}(3,n_2)n_0 - P_{0,2}(3,n_1)n_0$$

$$\dot{n}_1 = P_{0,1}(3,n_2)n_0 + P_{2,1}(2,n_2)n_2 + P_{2,1}(3,n_2)n_2 - n_1 + P_{1,2}(2,n_1)n_1 + P_{1,2}(3,n_1)n_1$$

$$\dot{n}_2 = P_{0,2}(3,n_1)n_0 + P_{1,2}(2,n_1)n_1 + P_{1,2}(3,n_1)n_1 - n_2 + P_{2,1}(2,n_2)n_2 + P_{2,1}(3,n_2)n_2$$

**System D**

$$s_x \rightarrow s_0$$

$$s_0 \& 3s_1 \rightarrow s_1$$

$$s_1 \& 2s_1 \rightarrow s_2$$

$$s_1 \& 3s_1 \rightarrow s_2$$

$$s_0 \& 3s_2 \rightarrow s_2$$

$$s_2 \& 2s_2 \rightarrow s_1$$

$$s_2 \& 3s_2 \rightarrow s_1$$

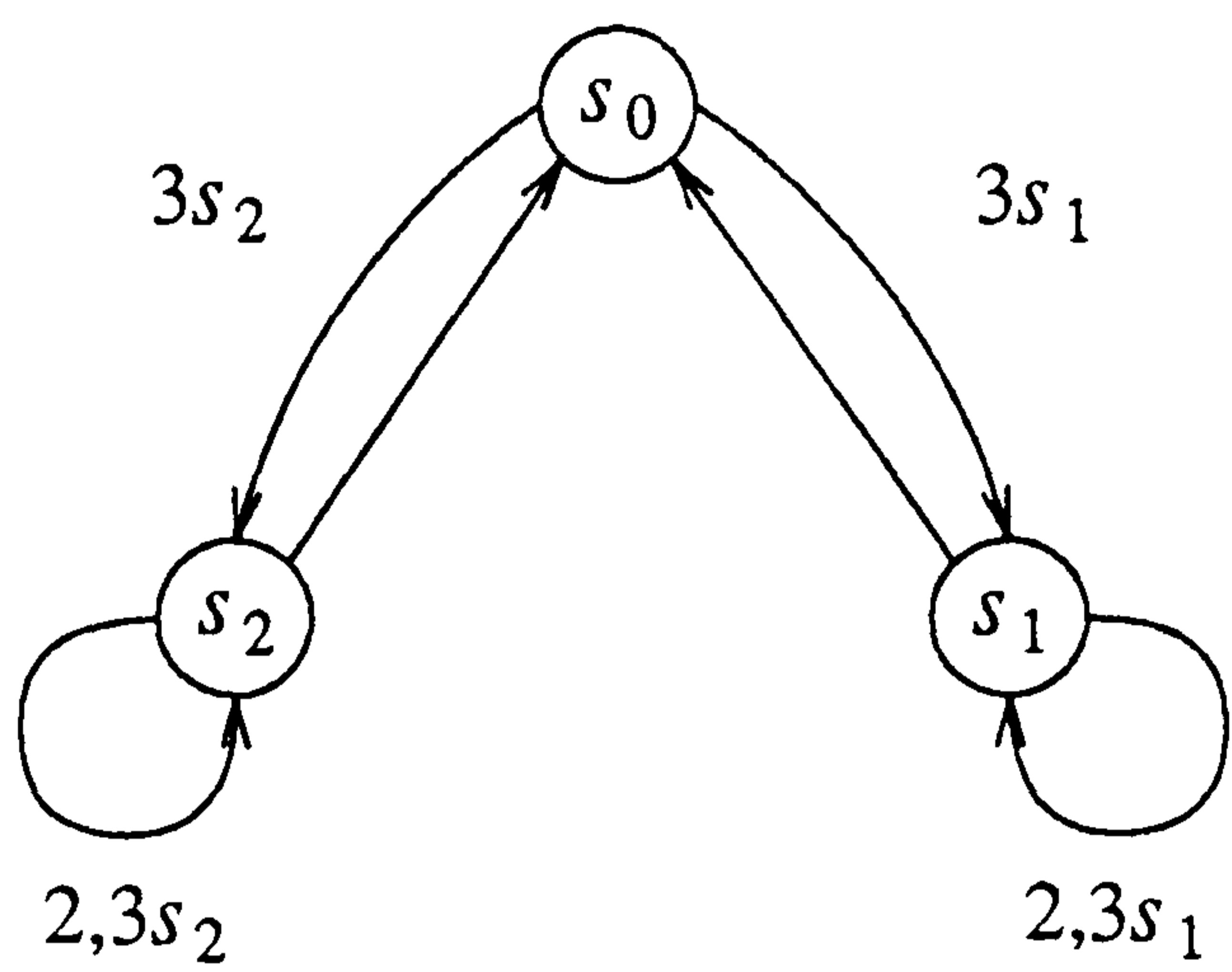
$$\dot{n}_0 = n_1 - P_{1,2}(2,n_1)n_1 - P_{1,2}(3,n_1)n_1 + n_2 - P_{2,1}(2,n_2)n_2 - P_{2,1}(3,n_2)n_2 - P_{0,1}(3,n_1)n_0 - P_{0,2}(3,n_2)n_0$$

$$\dot{n}_1 = P_{0,1}(3,n_1)n_0 + P_{2,1}(2,n_2)n_2 + P_{2,1}(3,n_2)n_2 - n_1 + P_{1,2}(2,n_1)n_1 + P_{1,2}(3,n_1)n_1$$

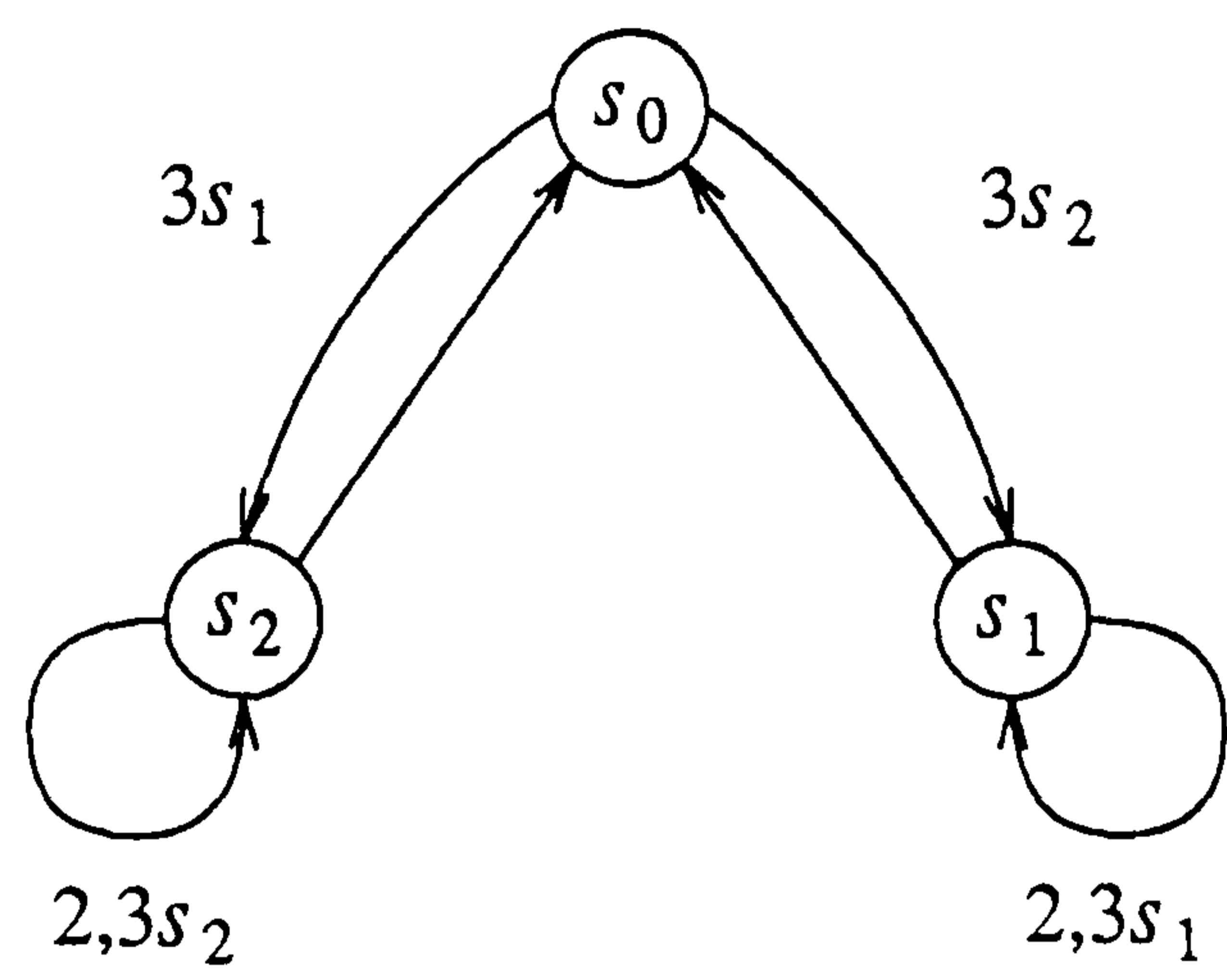
$$\dot{n}_2 = P_{0,2}(3,n_2)n_0 + P_{1,2}(2,n_1)n_1 + P_{1,2}(3,n_1)n_1 - n_2 + P_{2,1}(2,n_2)n_2 + P_{2,1}(3,n_2)n_2$$

Figure 4.14: Class 4 systems state transition diagrams.

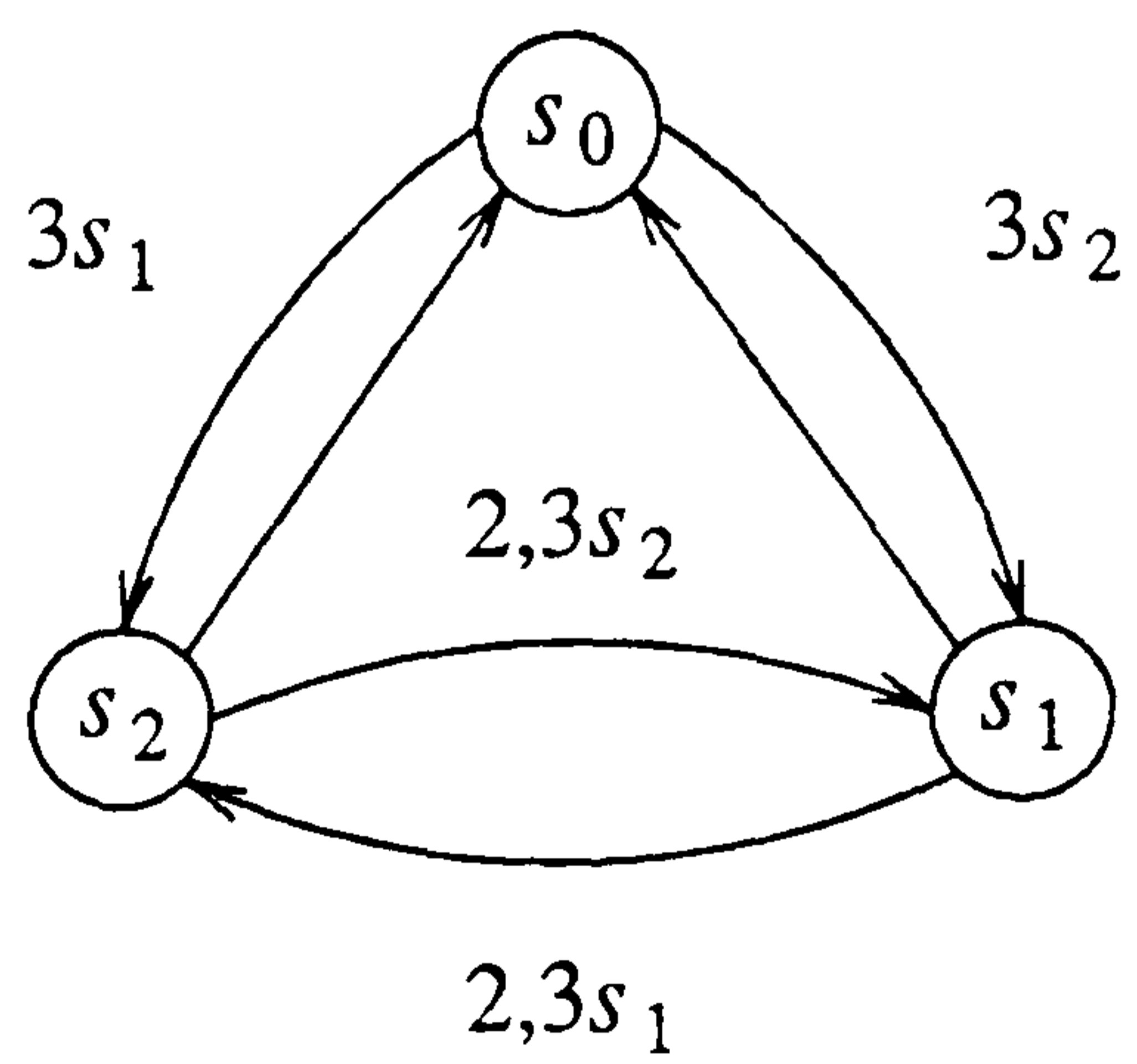
System A



System B



System C



System D

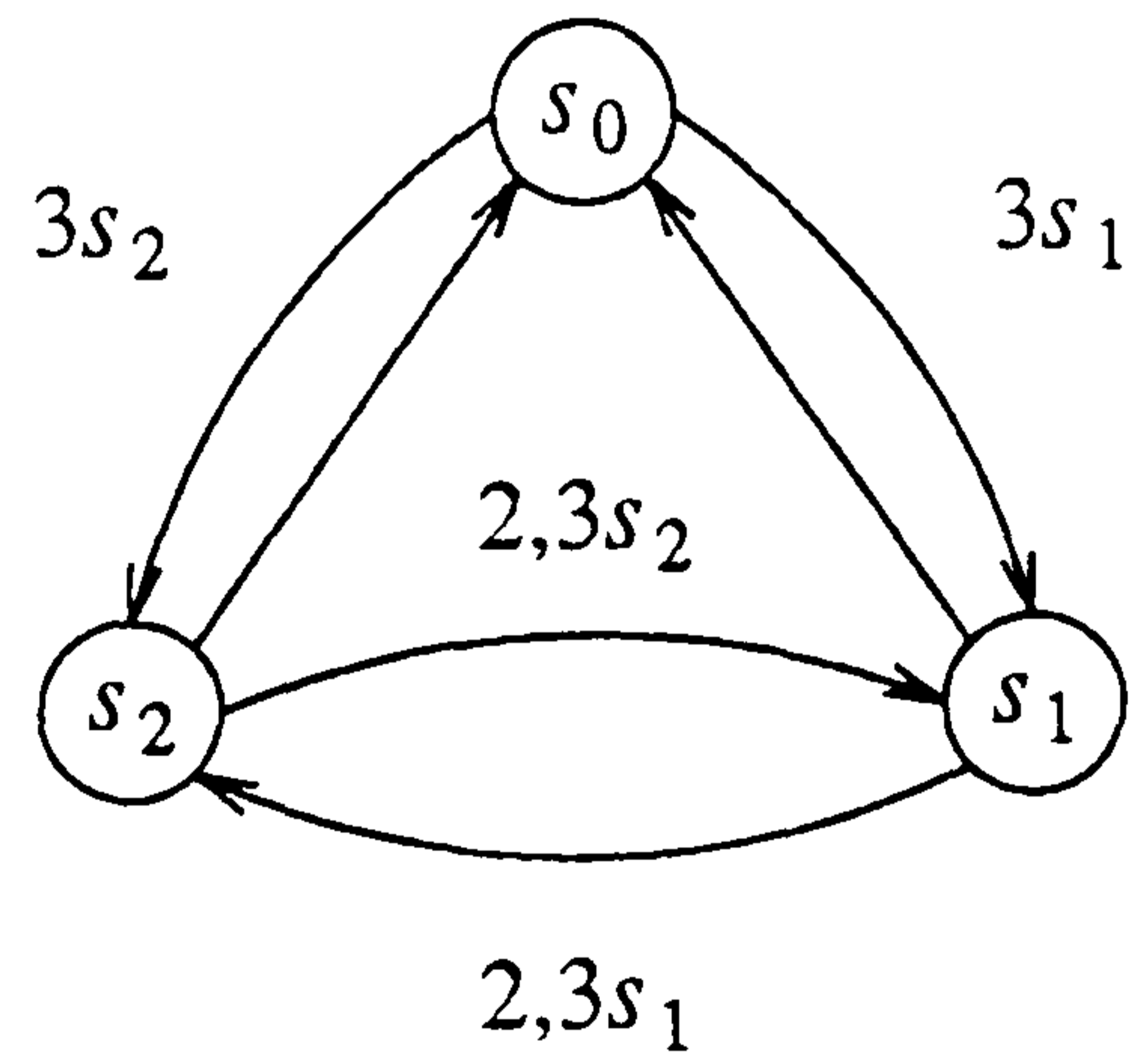




Figure 4.15: Class 4 vector field diagrams.  
(a) System A.

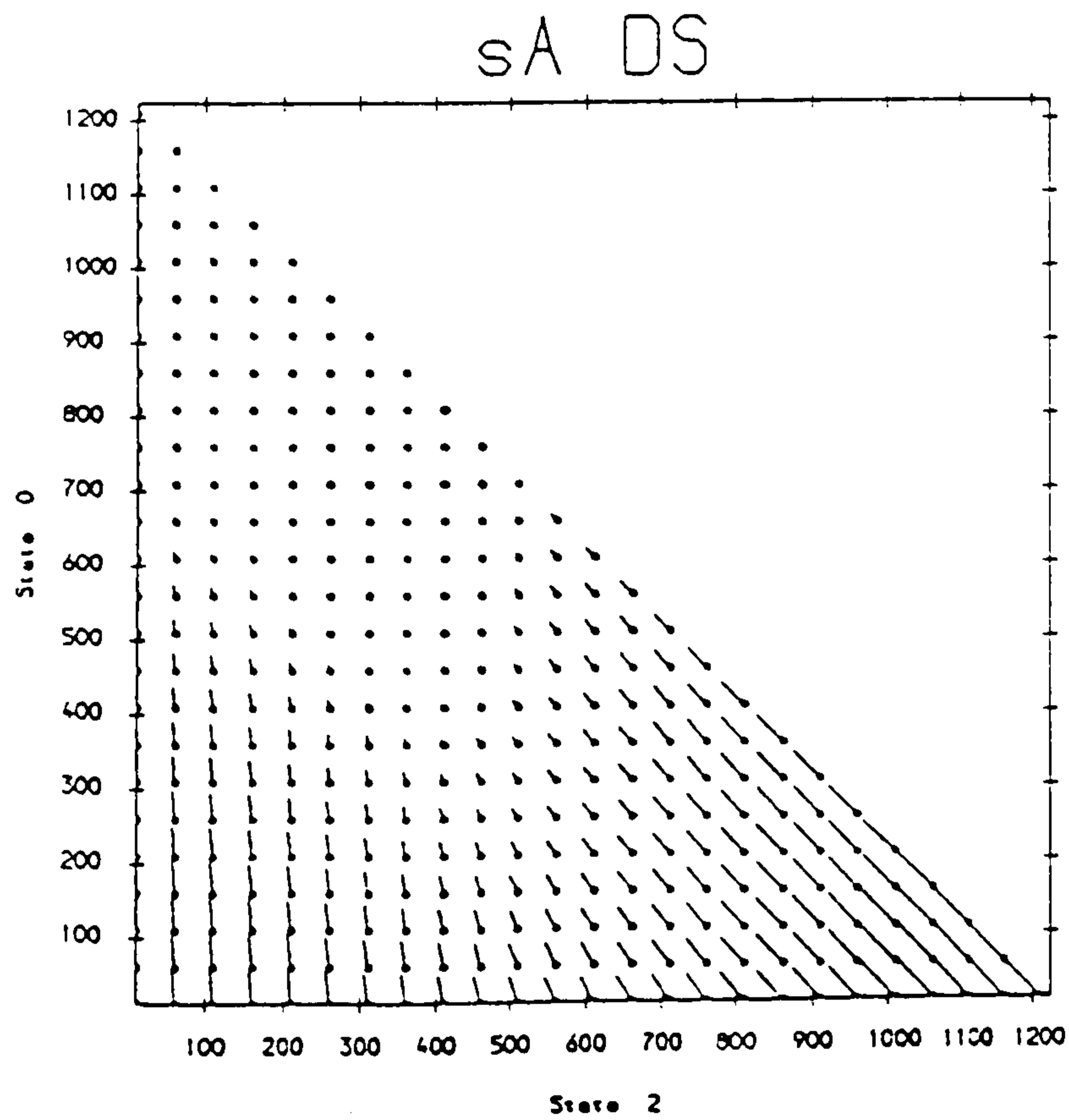
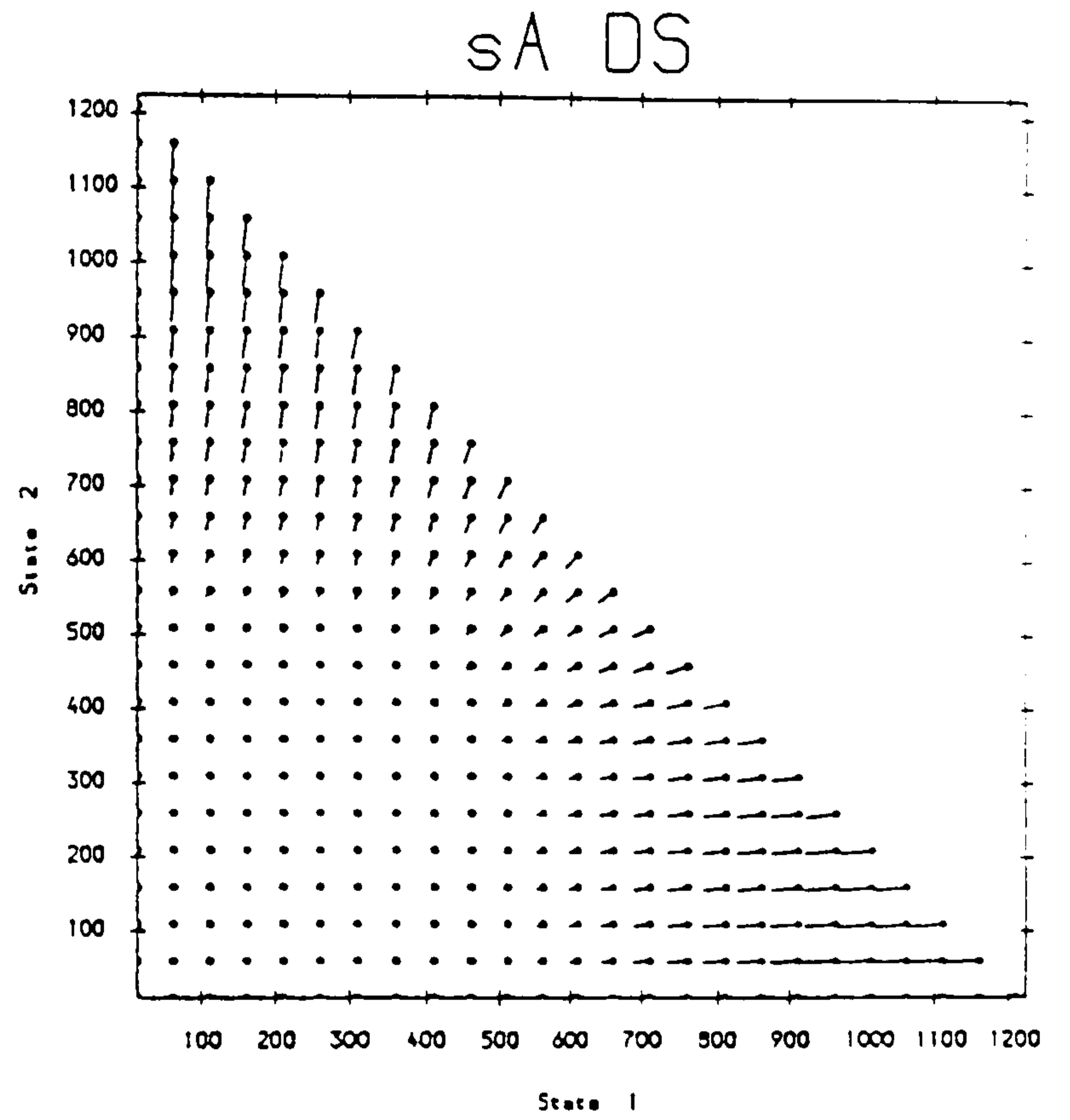
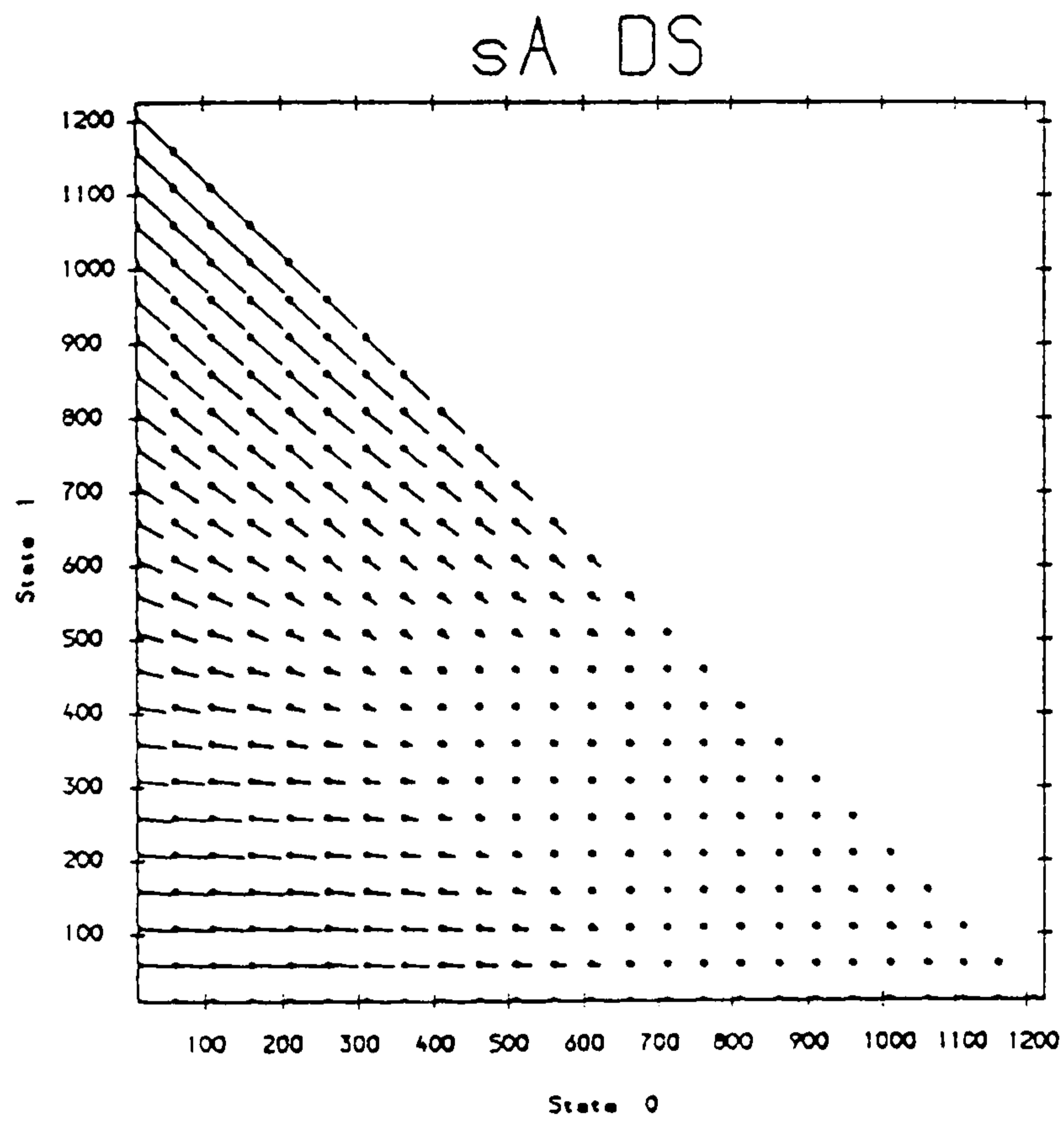


Figure 4.15: Class 4 vector field diagrams.  
(b) System B.

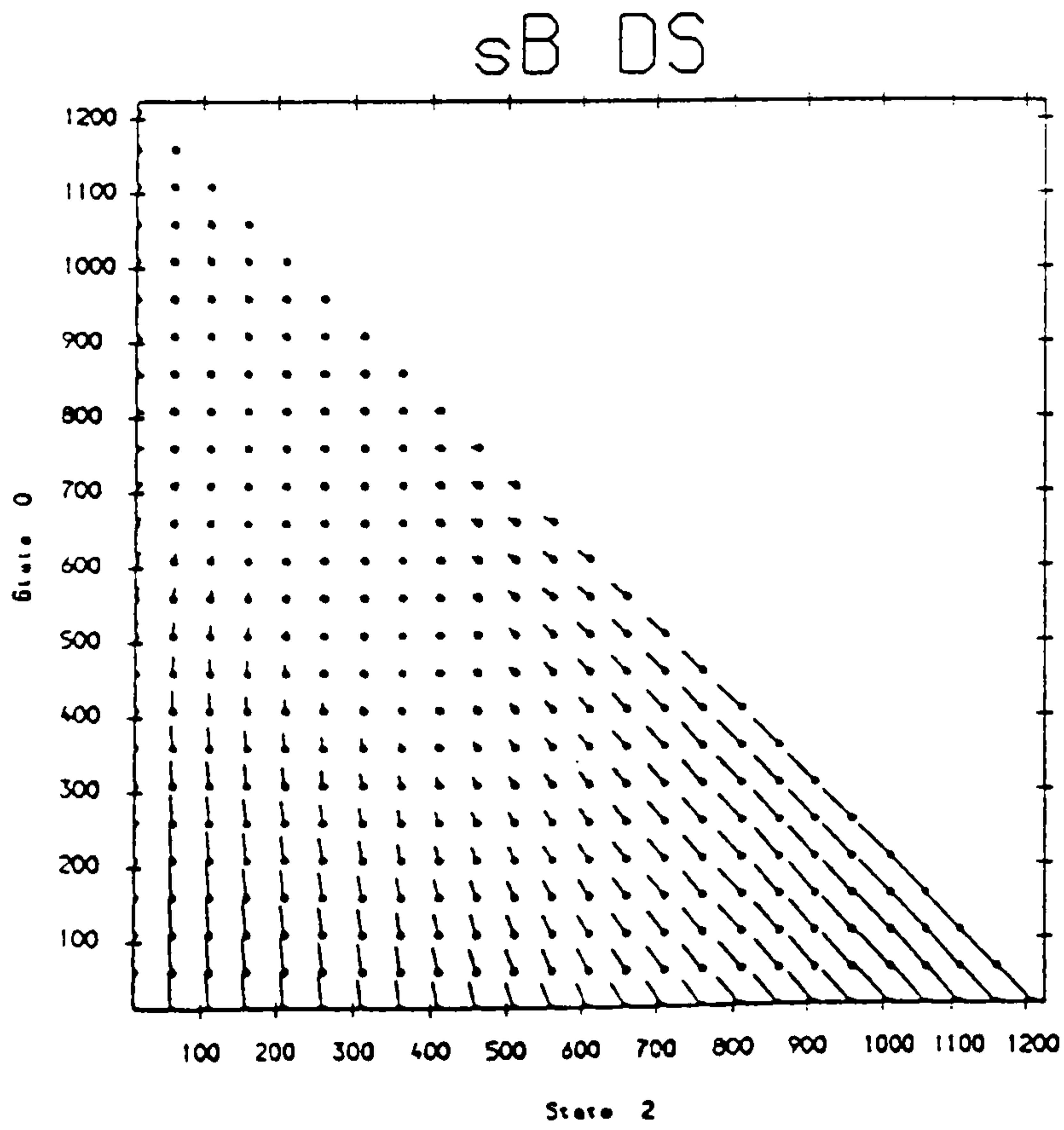
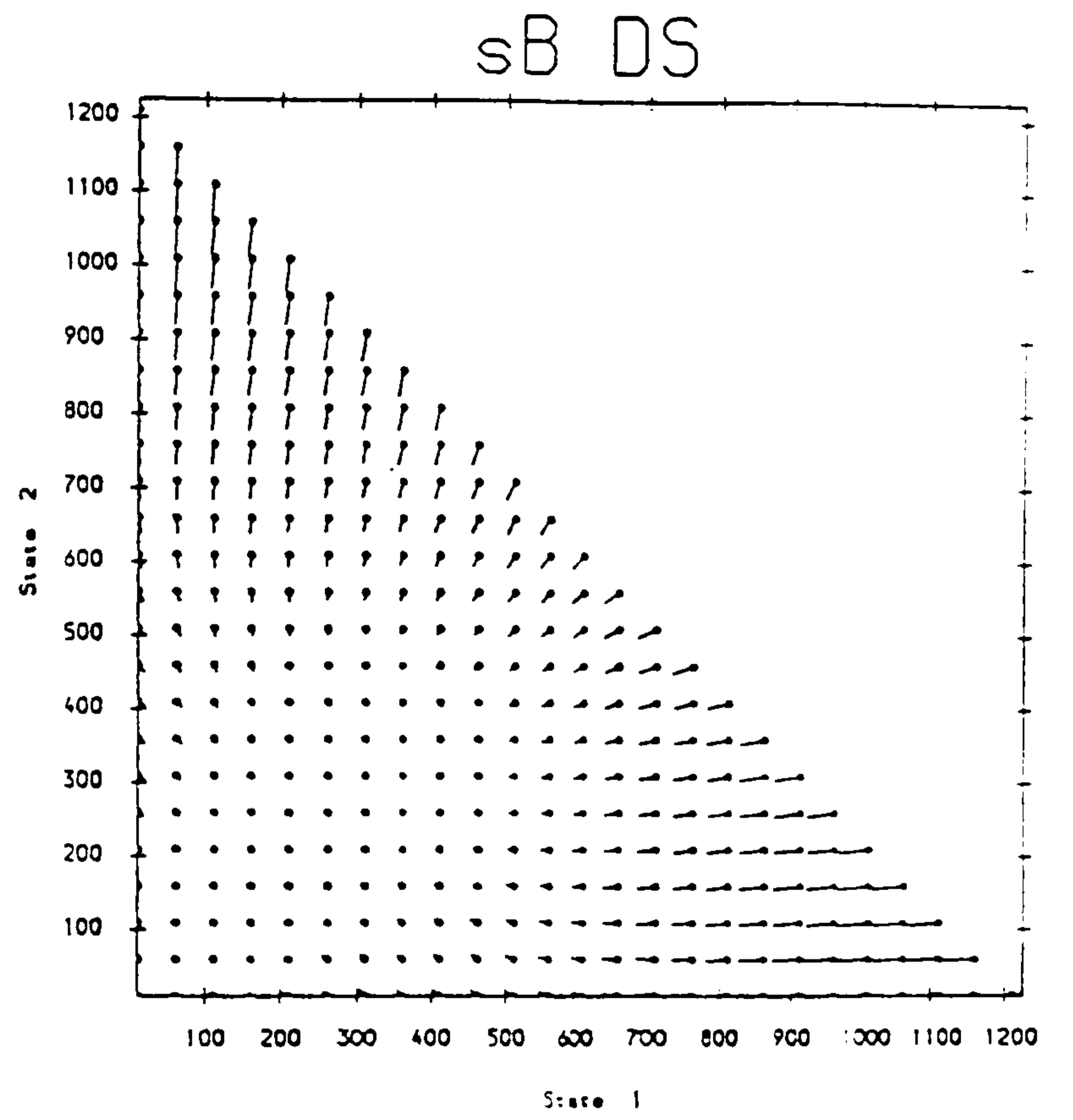
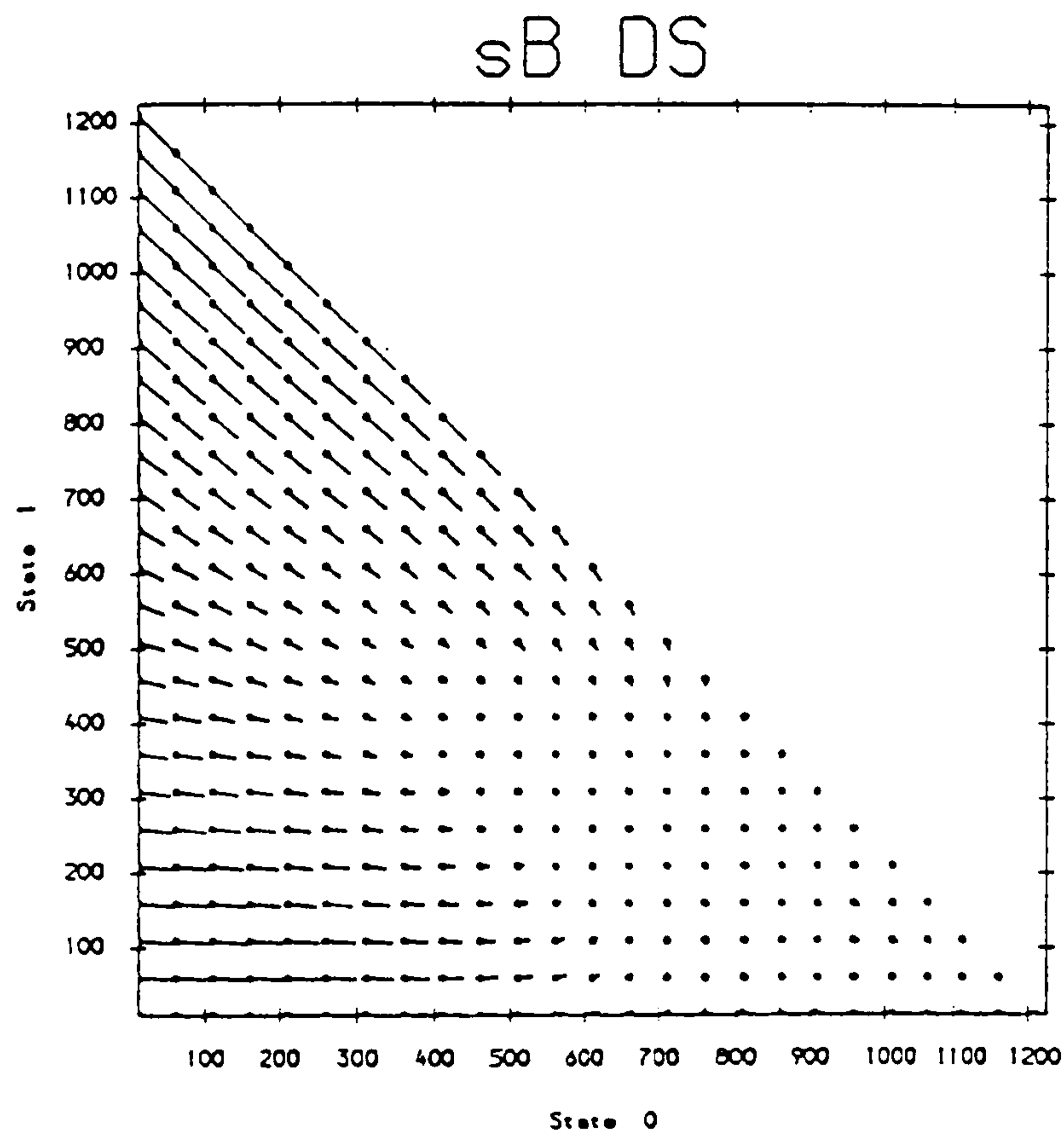


Figure 4.15: Class 4 vector field diagrams.  
(c) System C.

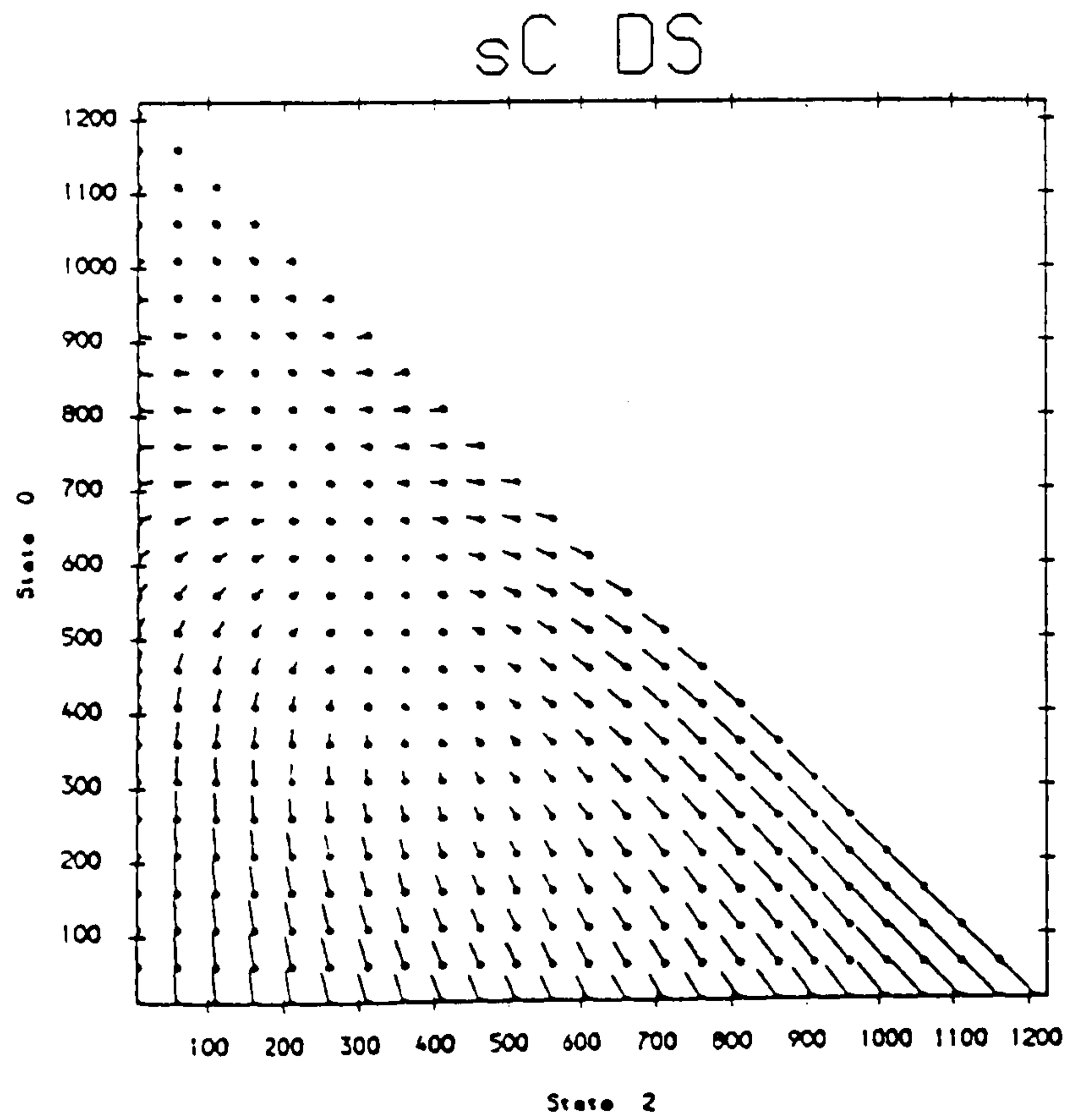
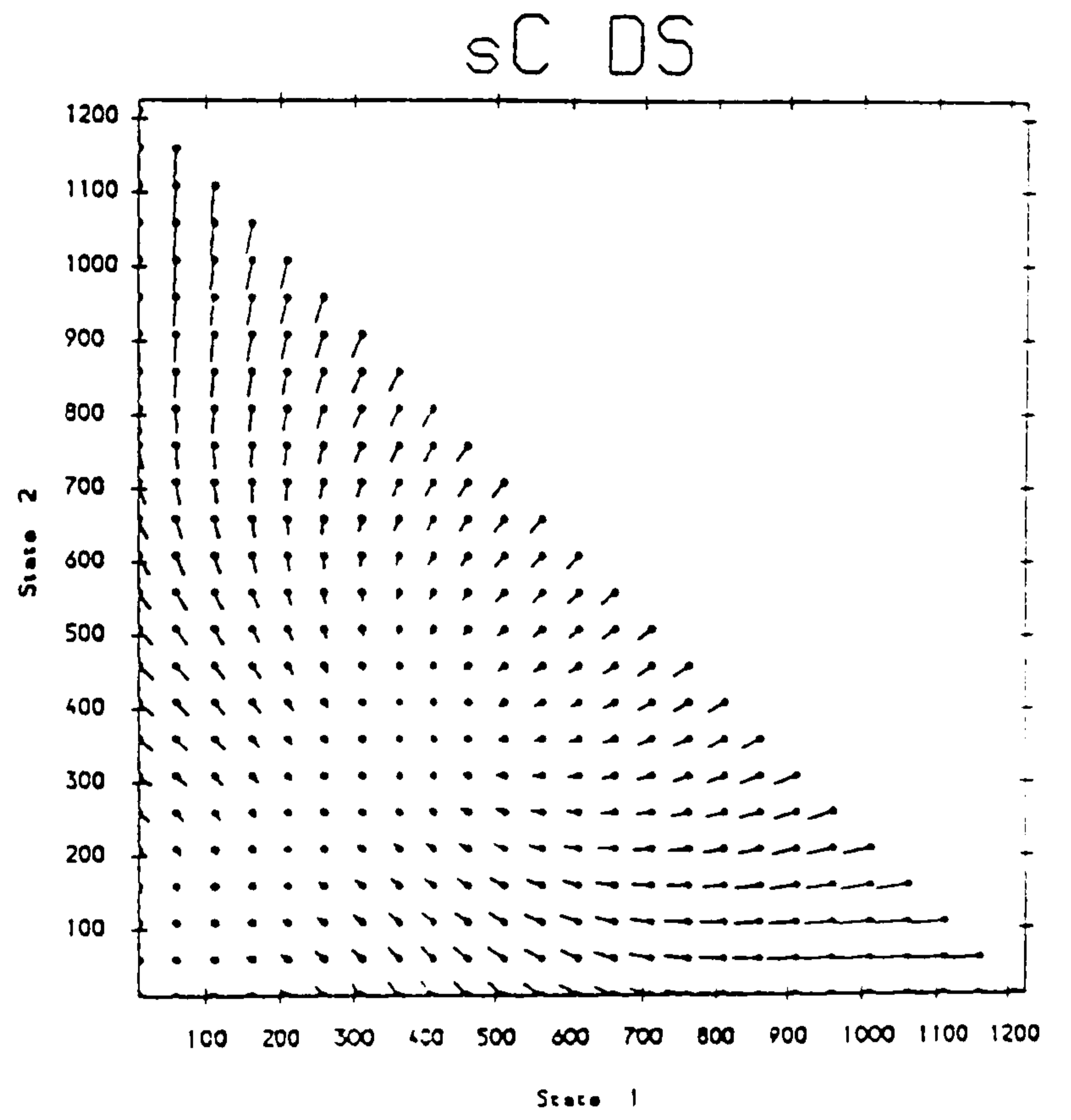
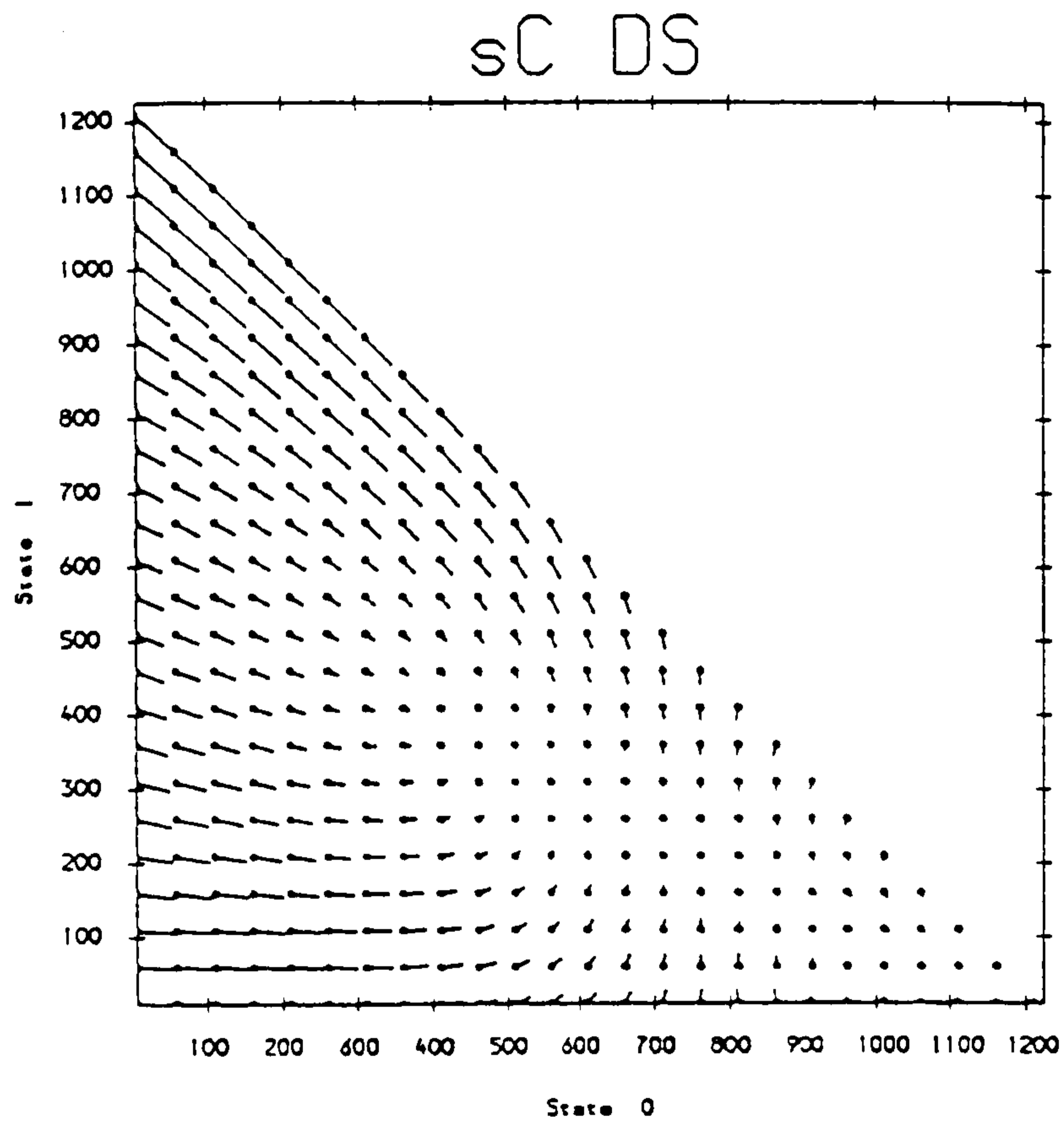




Figure 4.15: Class 4 vector field diagrams.  
(d) System D.

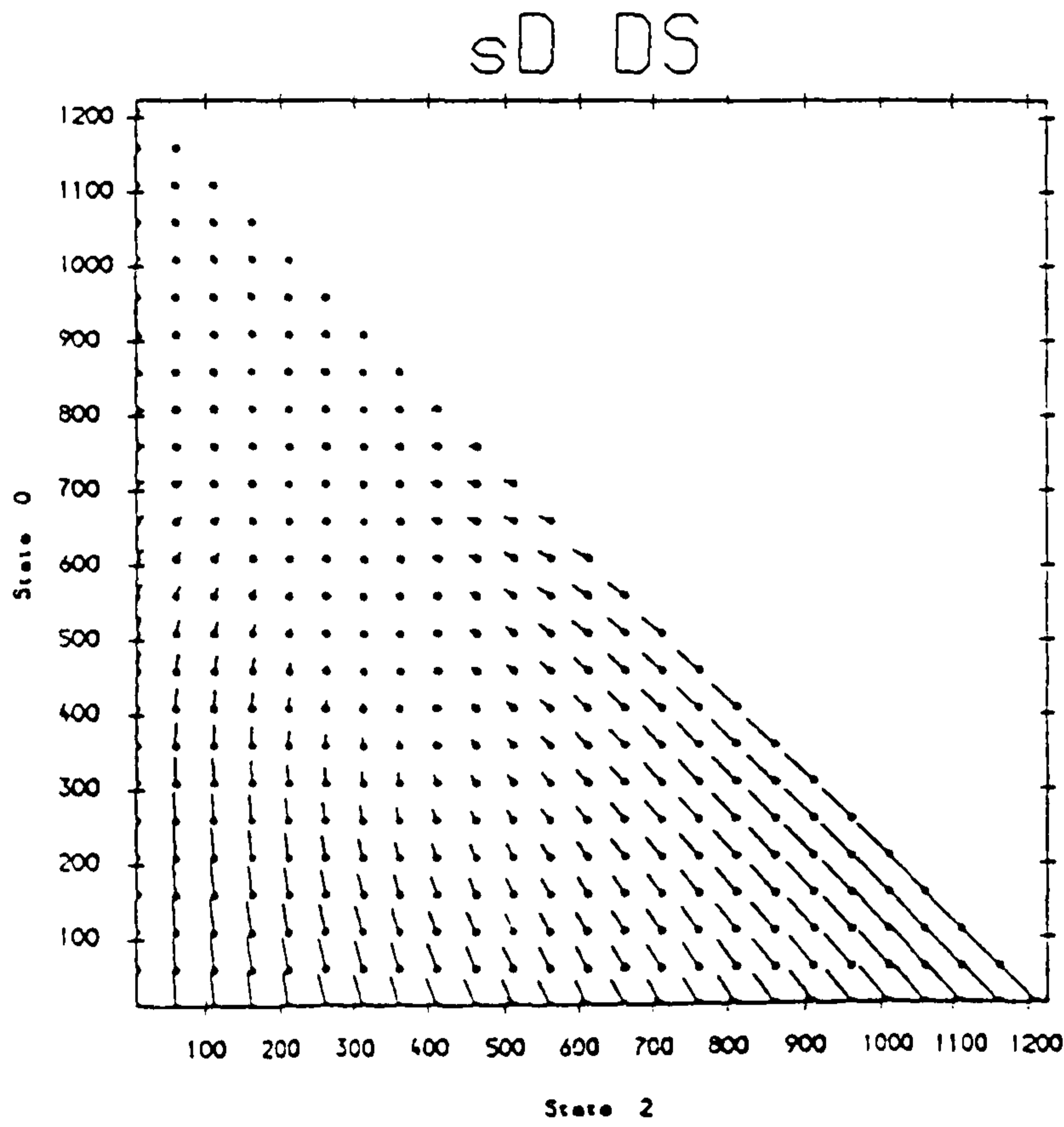
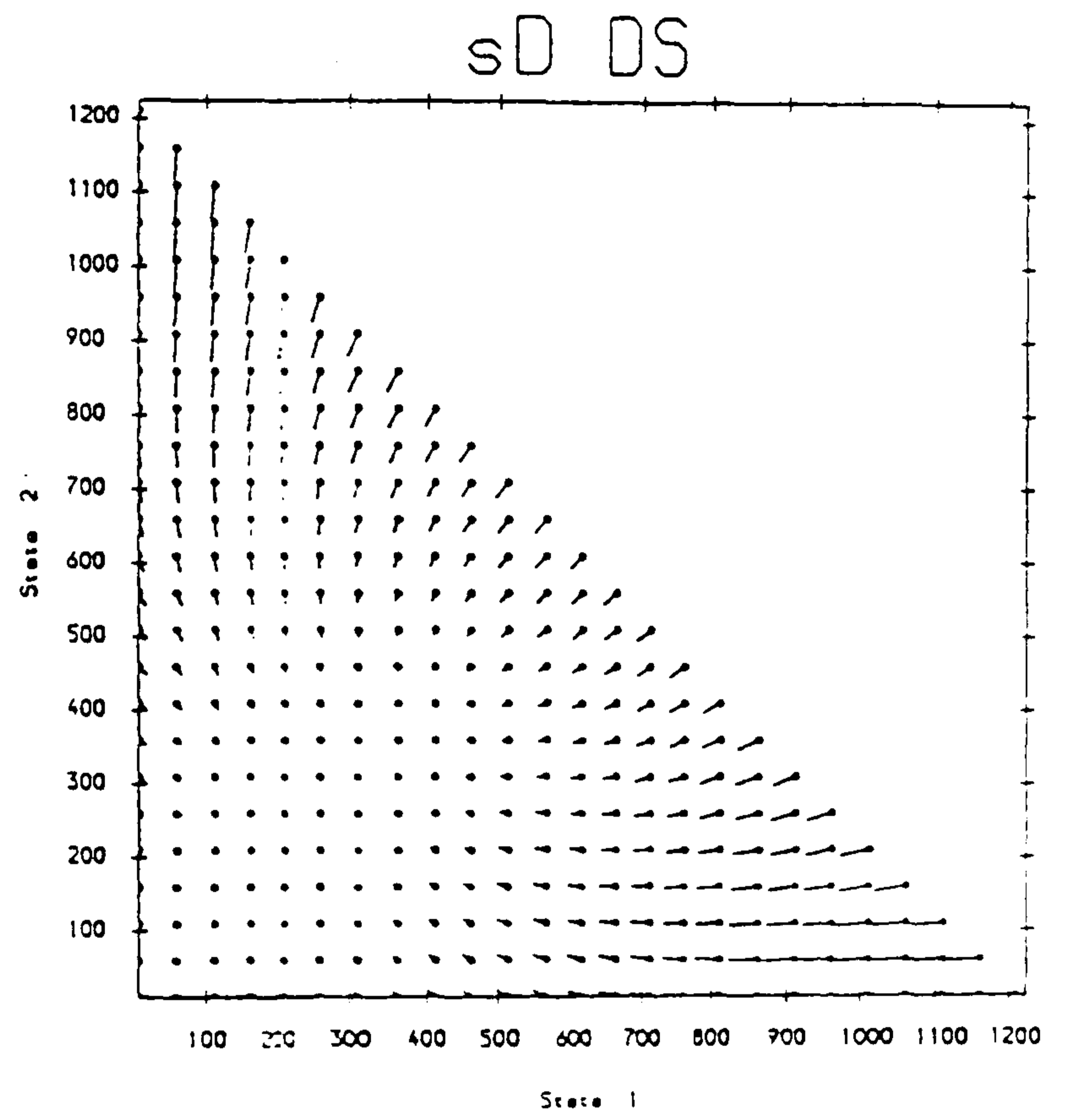
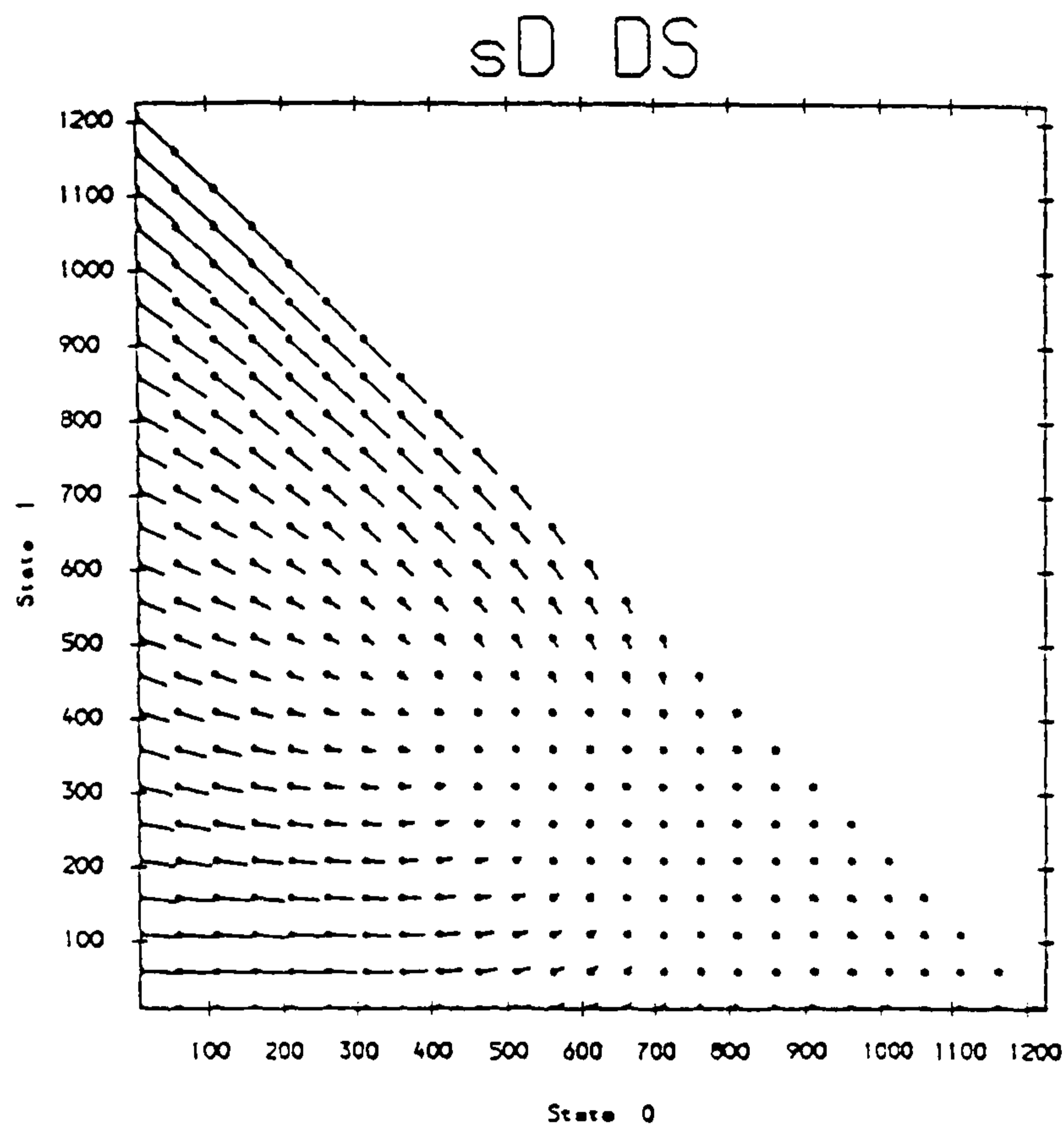


Figure 4.16: Class 4 phase diagrams.  
(a) System A: Deterministic.

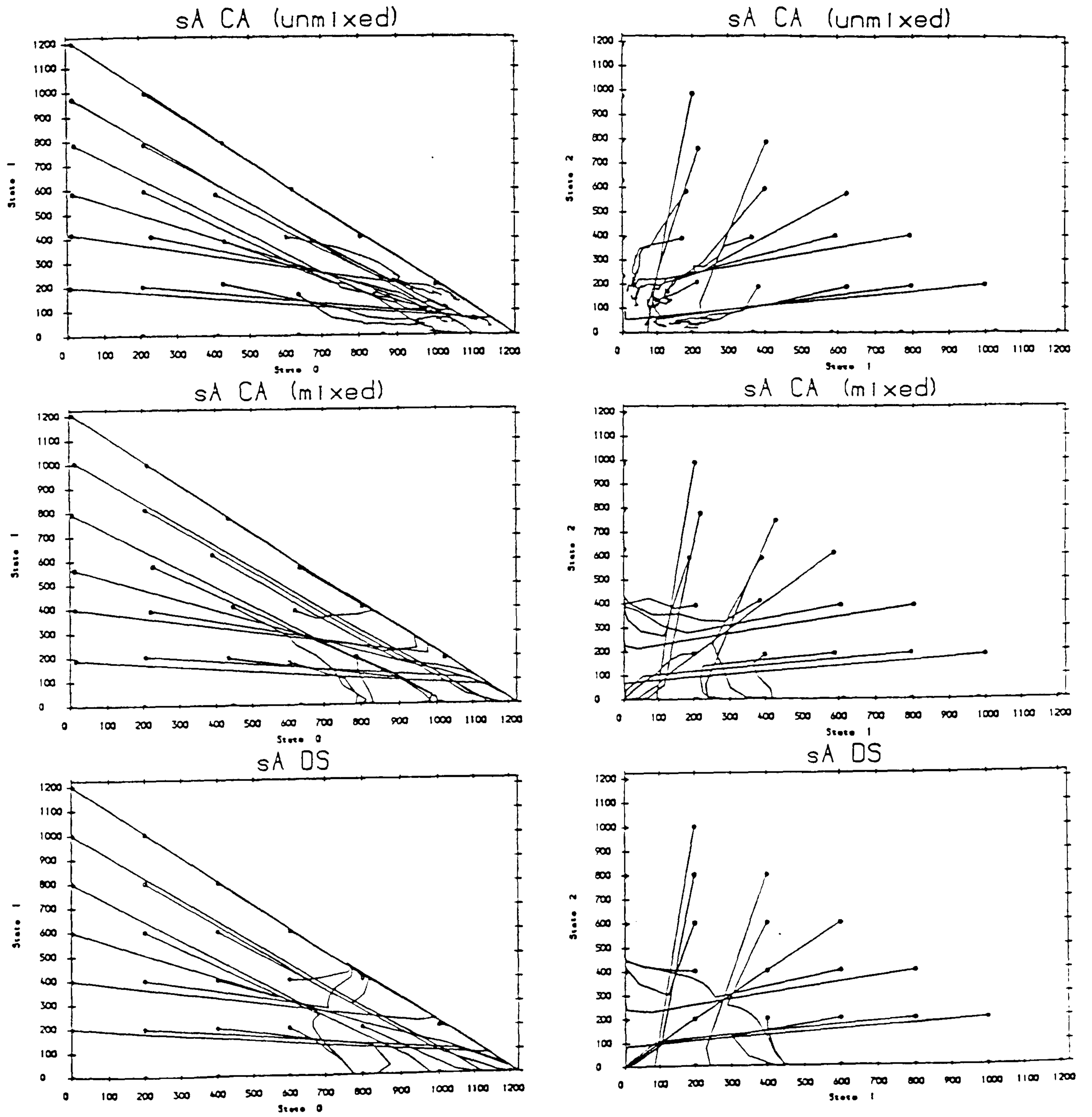


Figure 4.16: Class 4 phase diagrams.  
(b) System B: Deterministic.

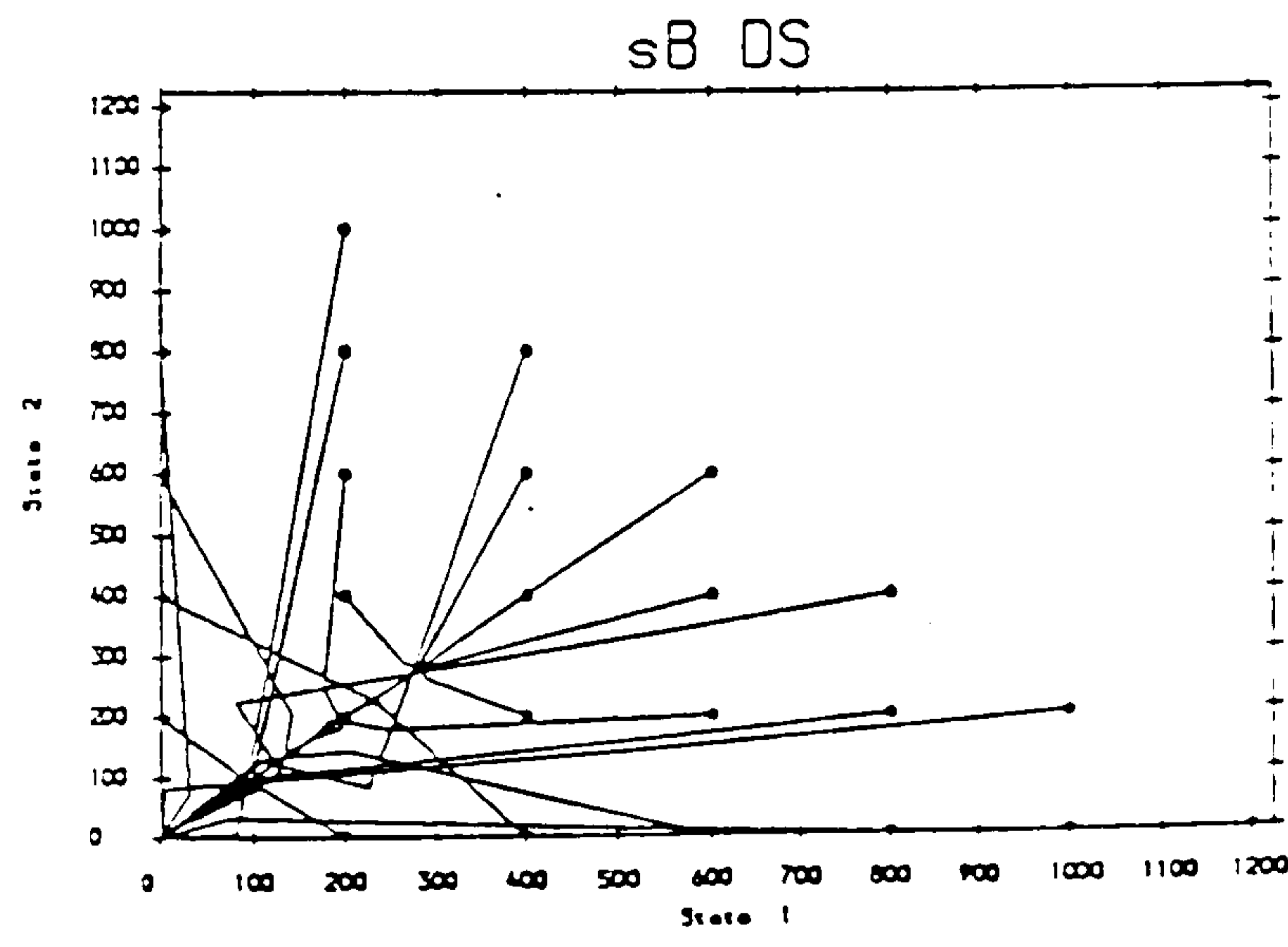
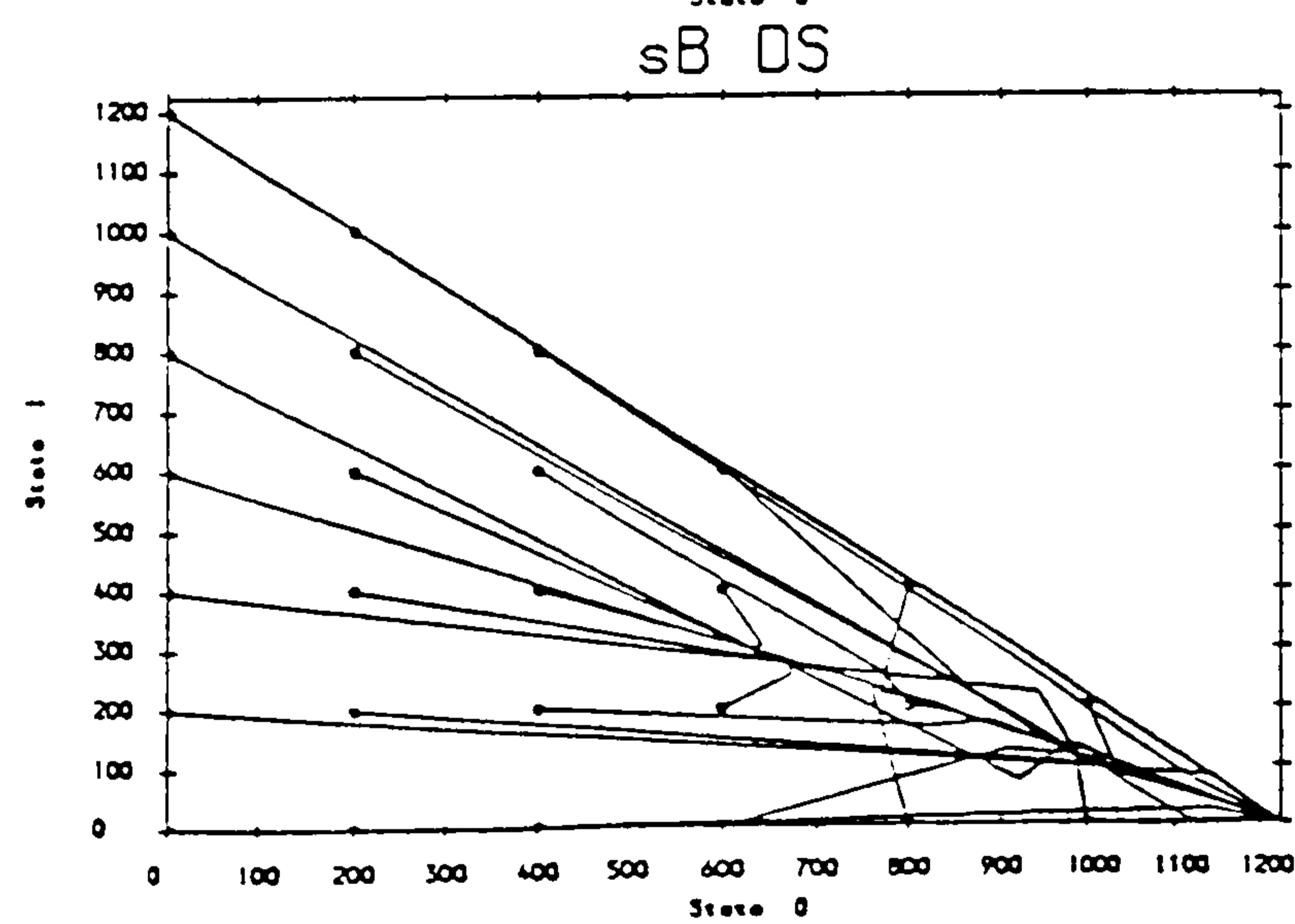
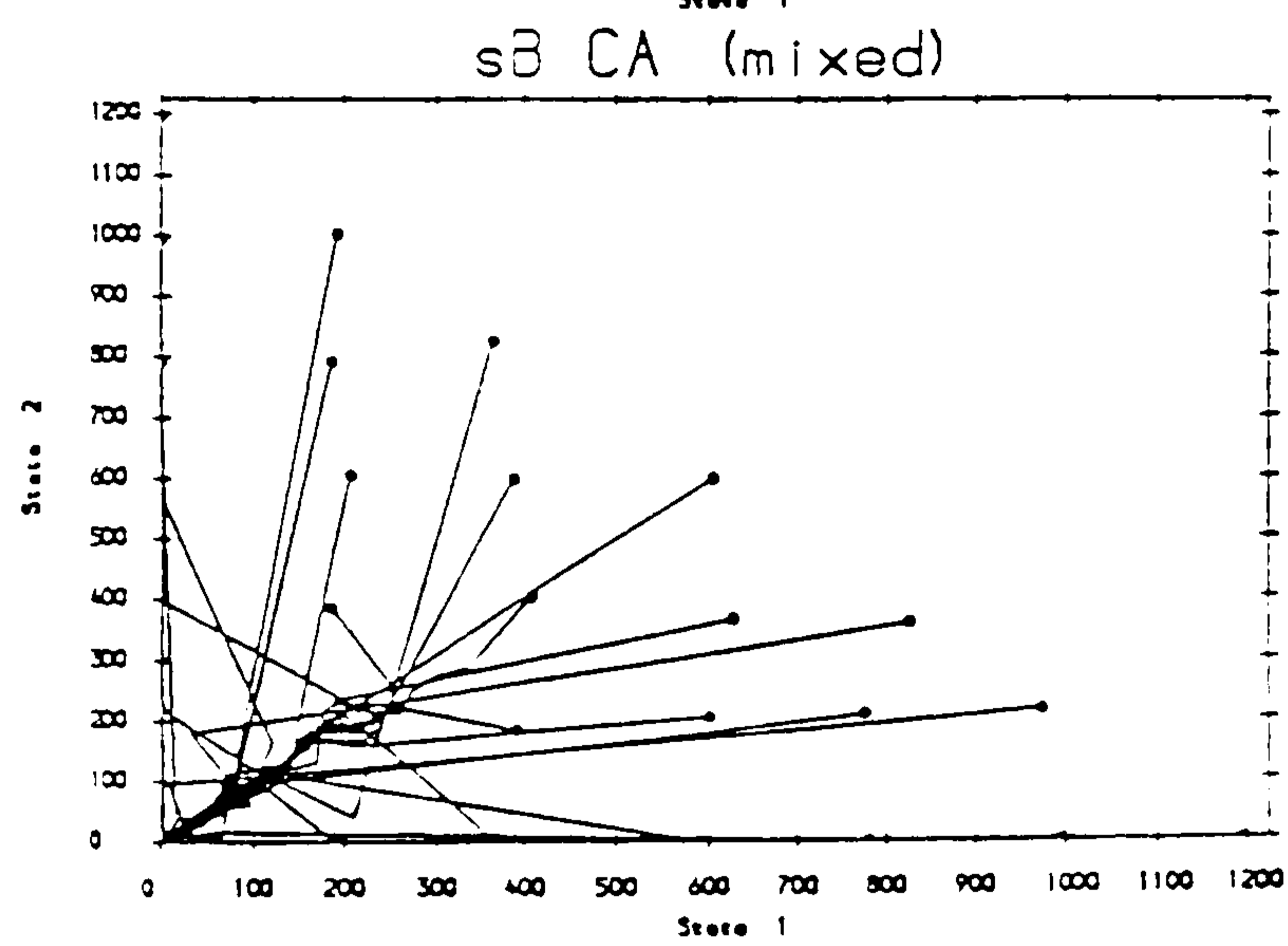
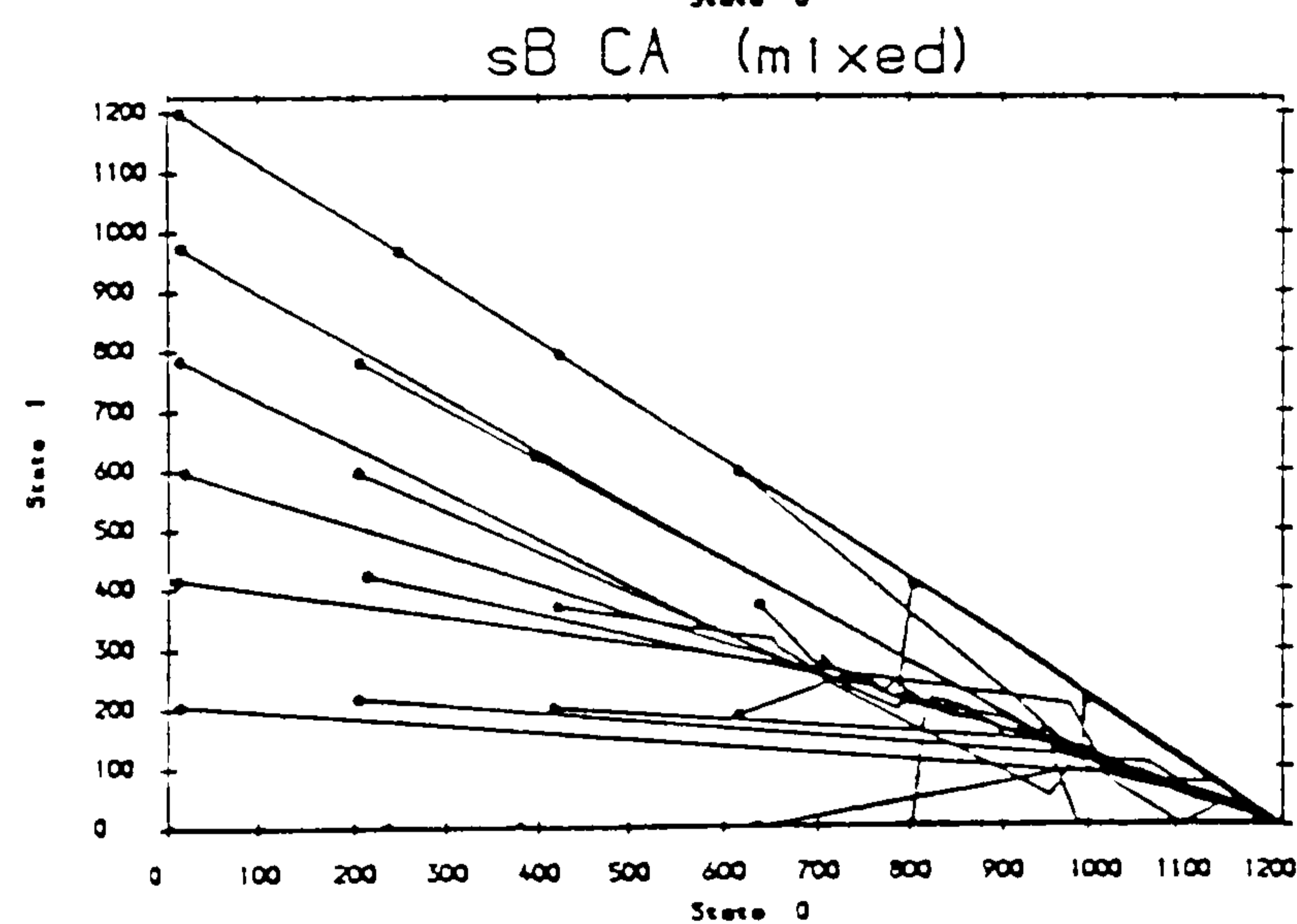
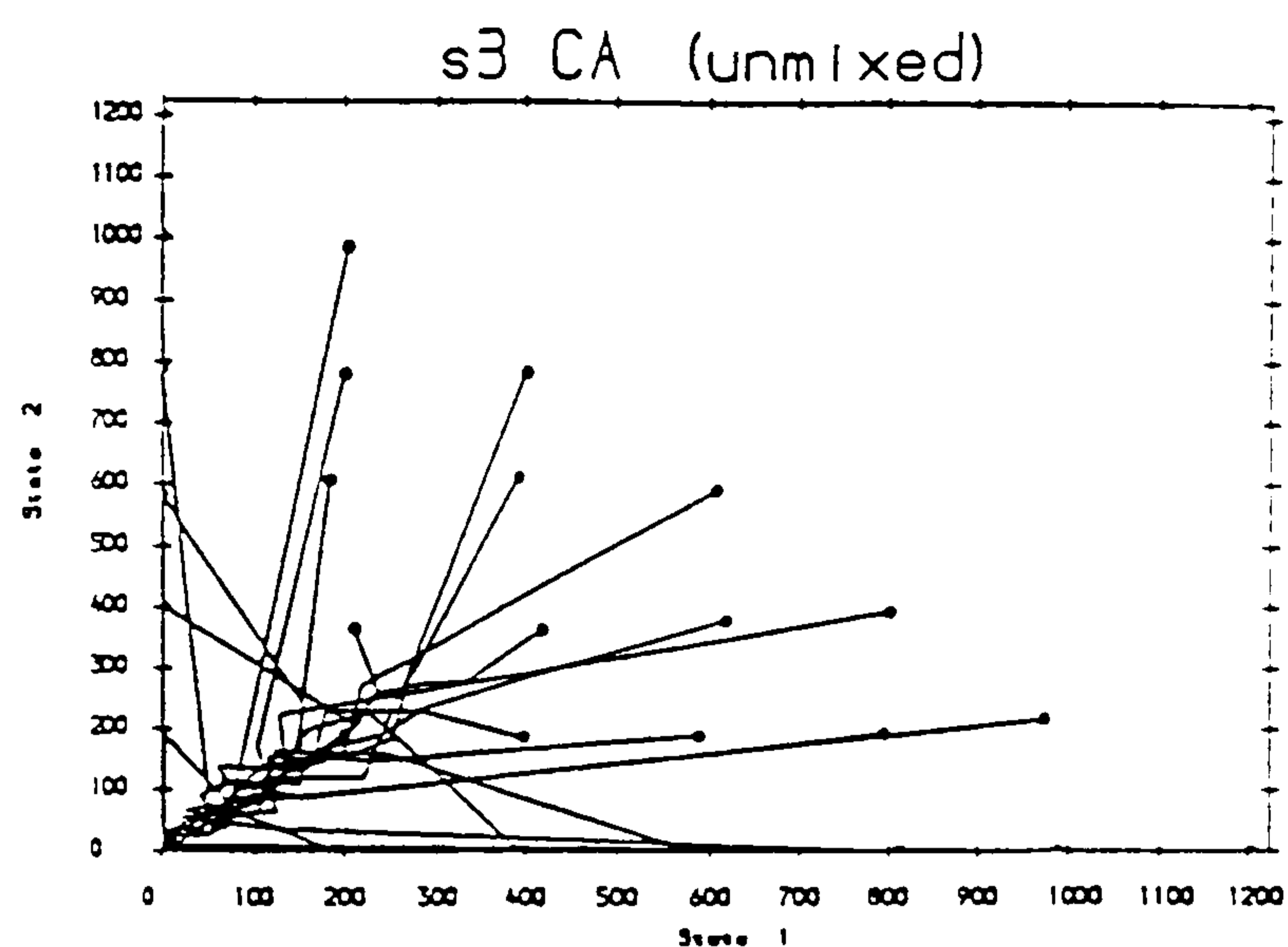
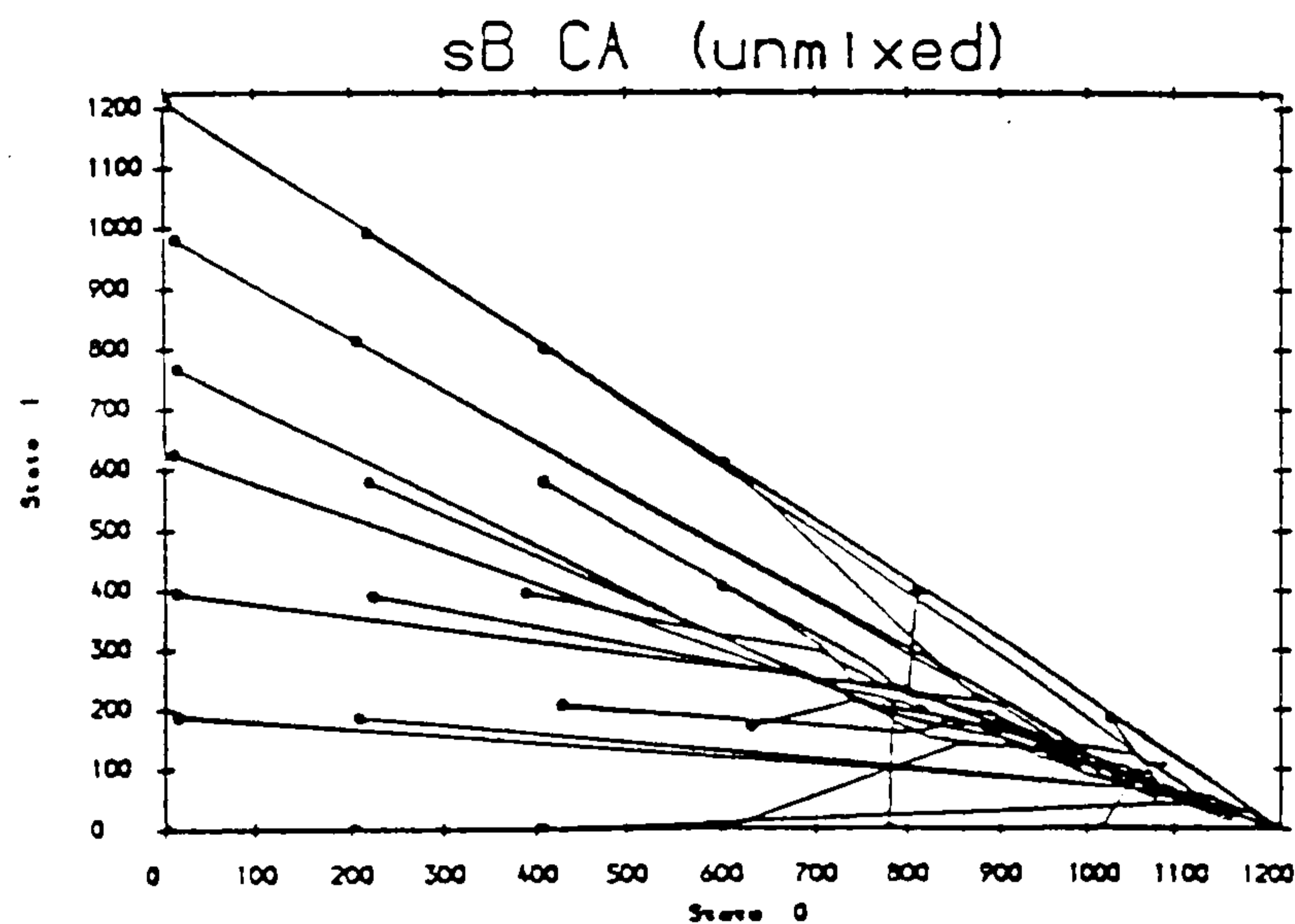




Figure 4.16: Class 4 phase diagrams.  
(c) System C: Deterministic.

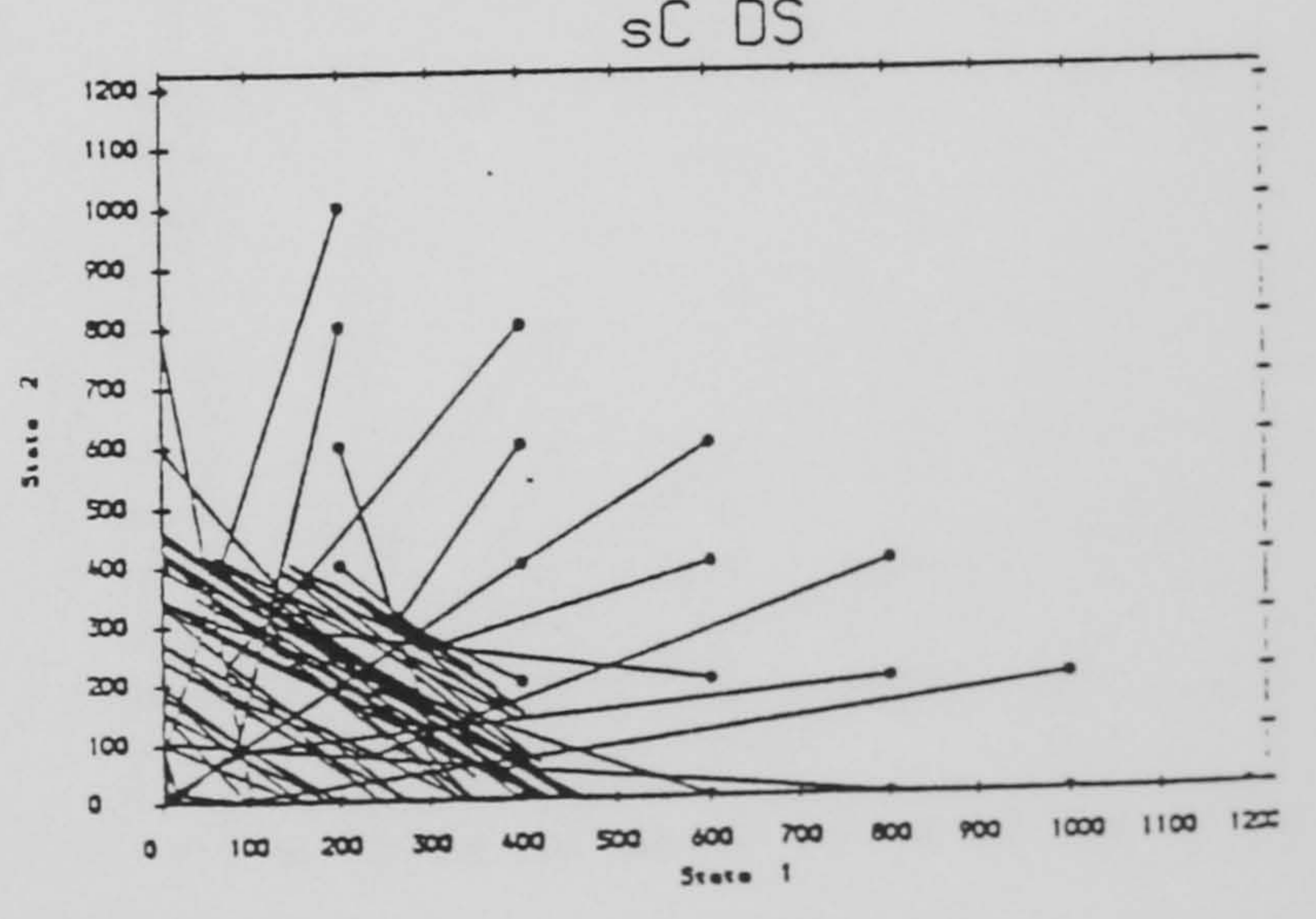
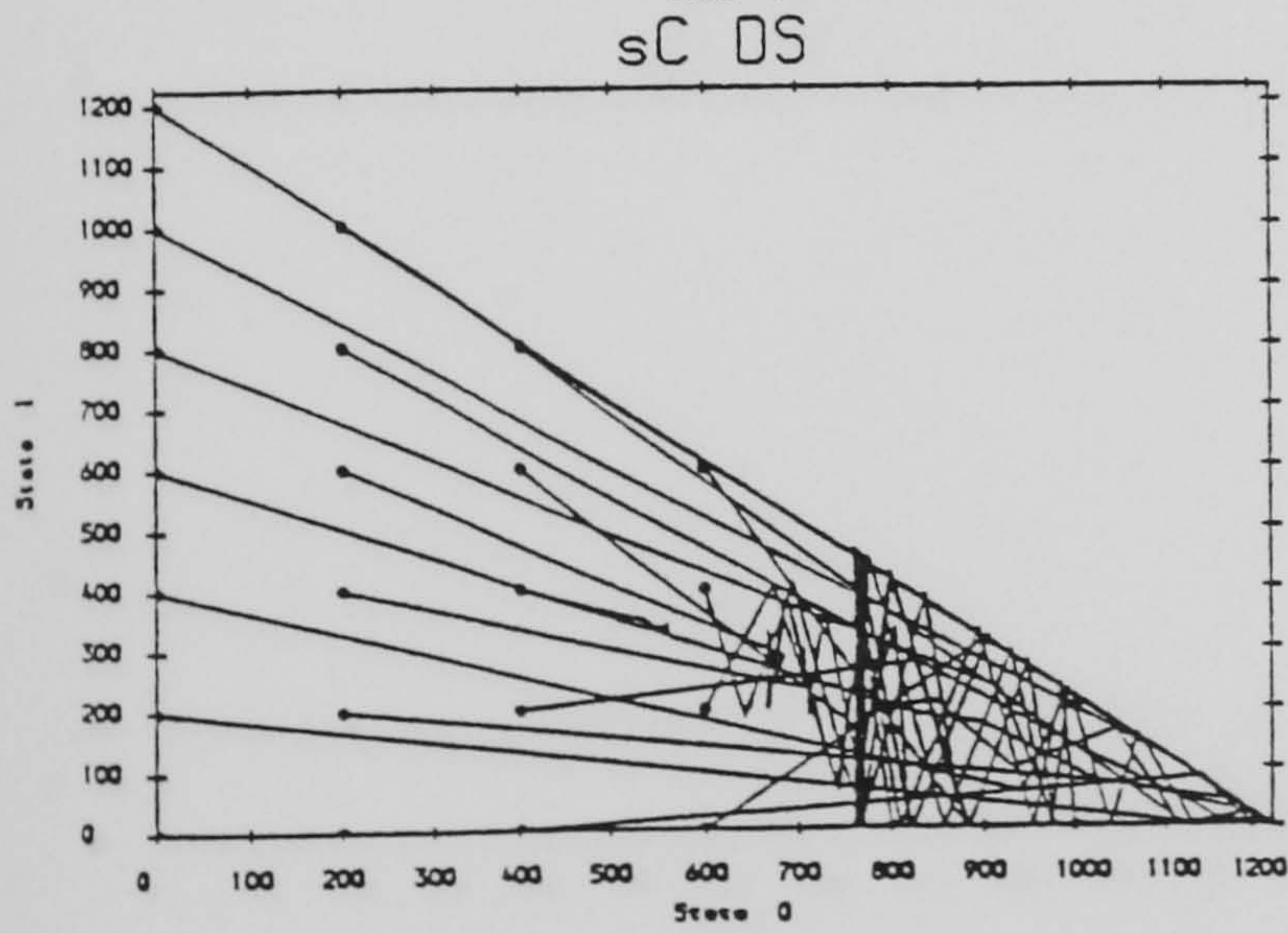
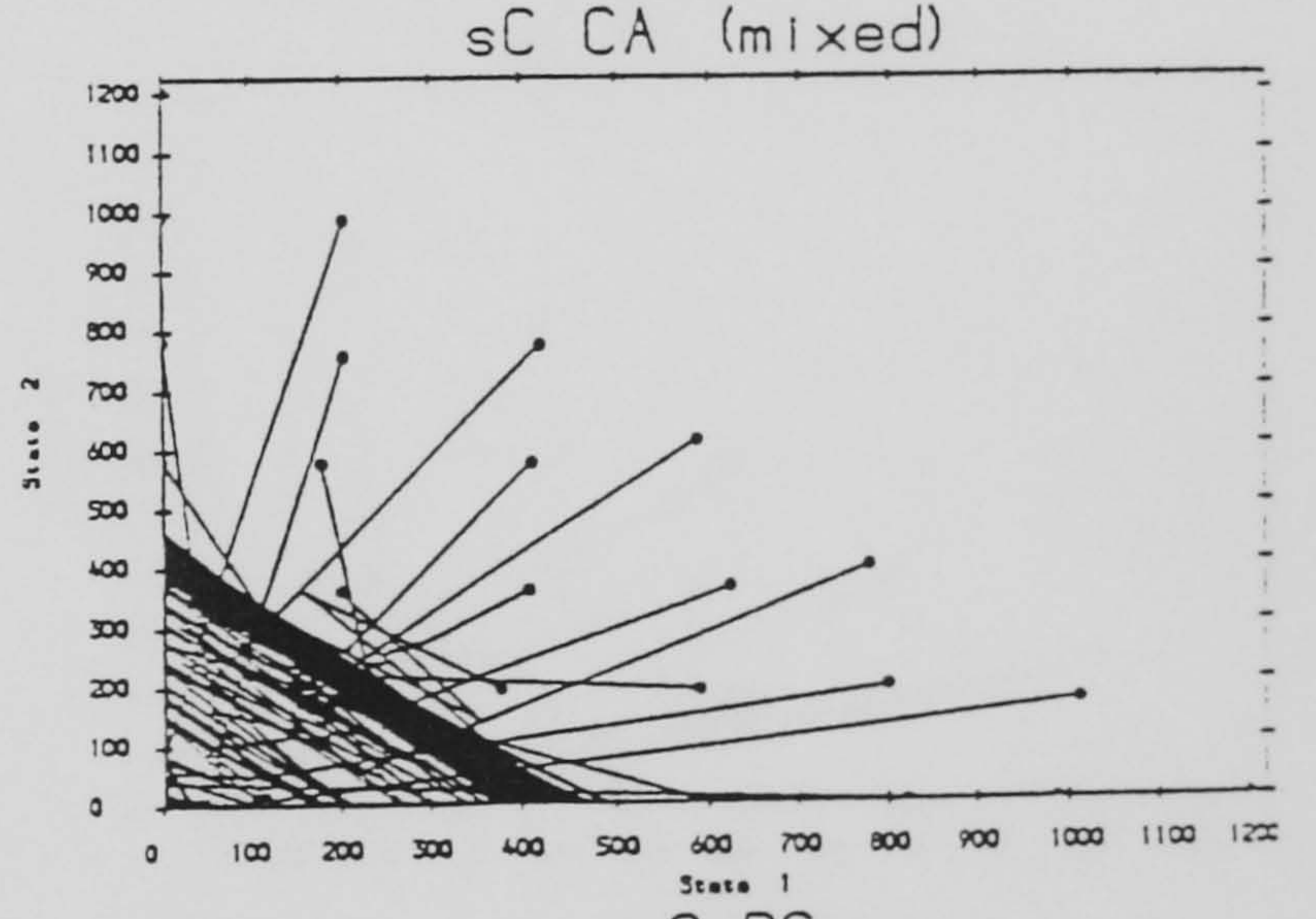
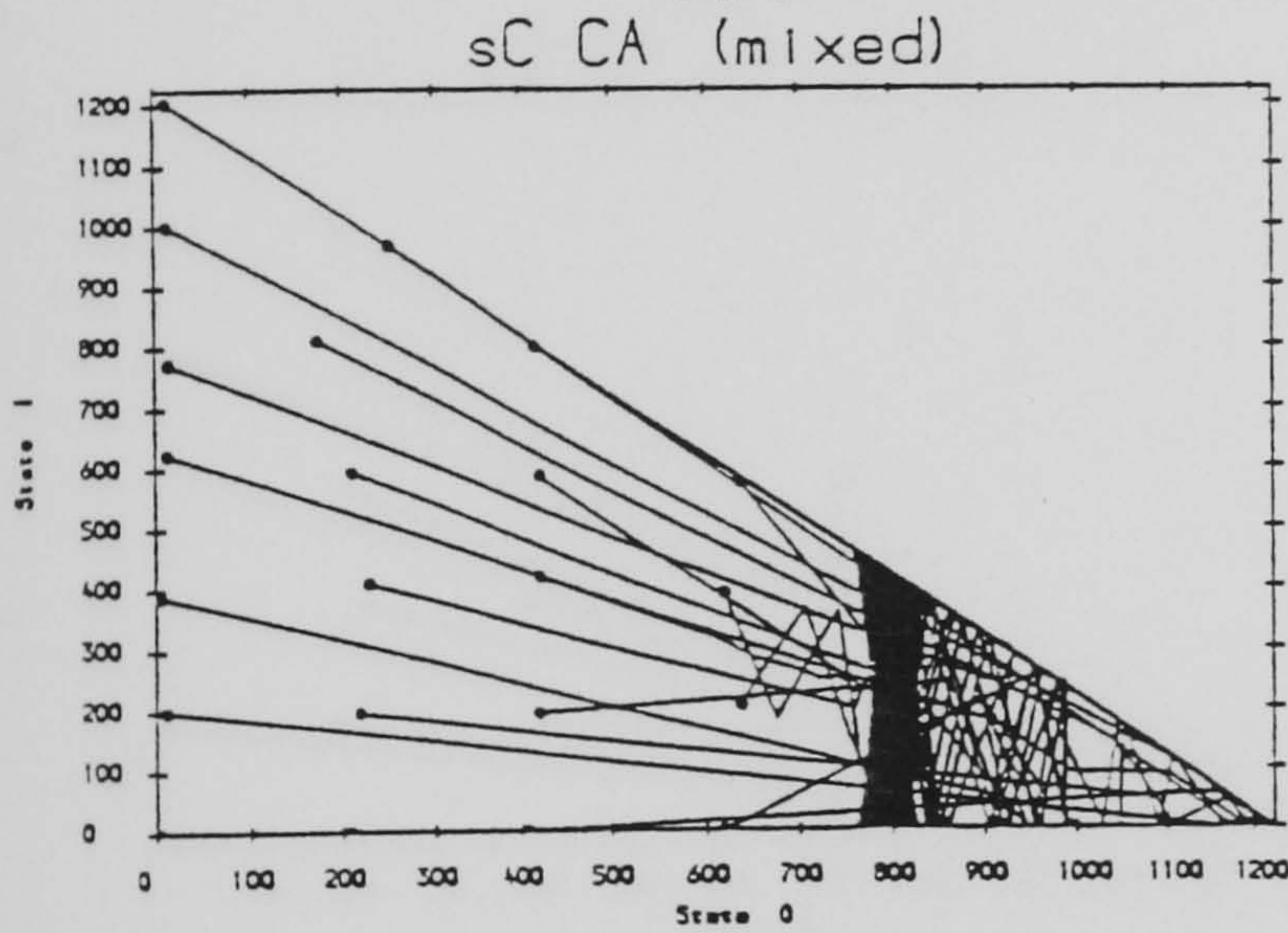
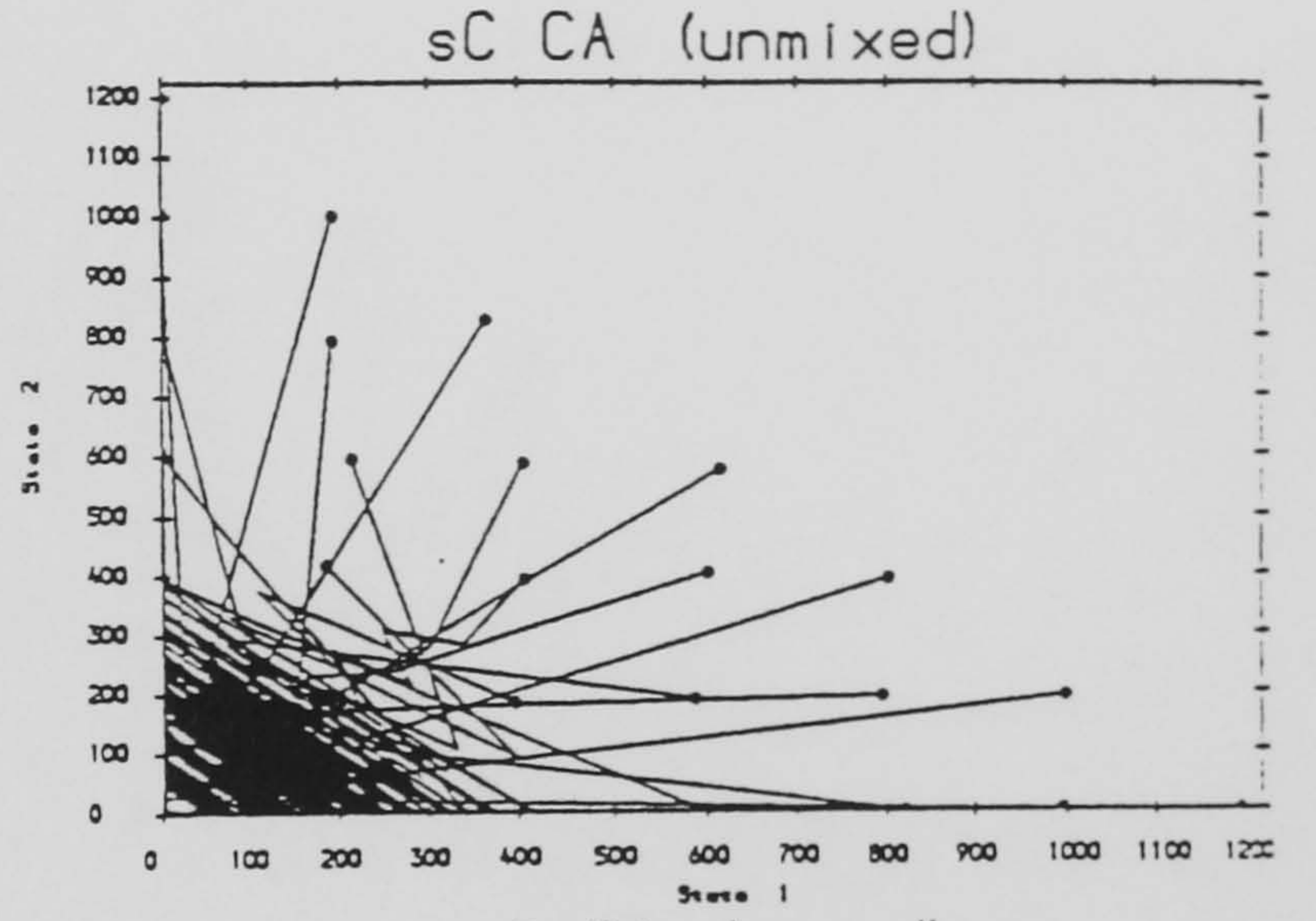
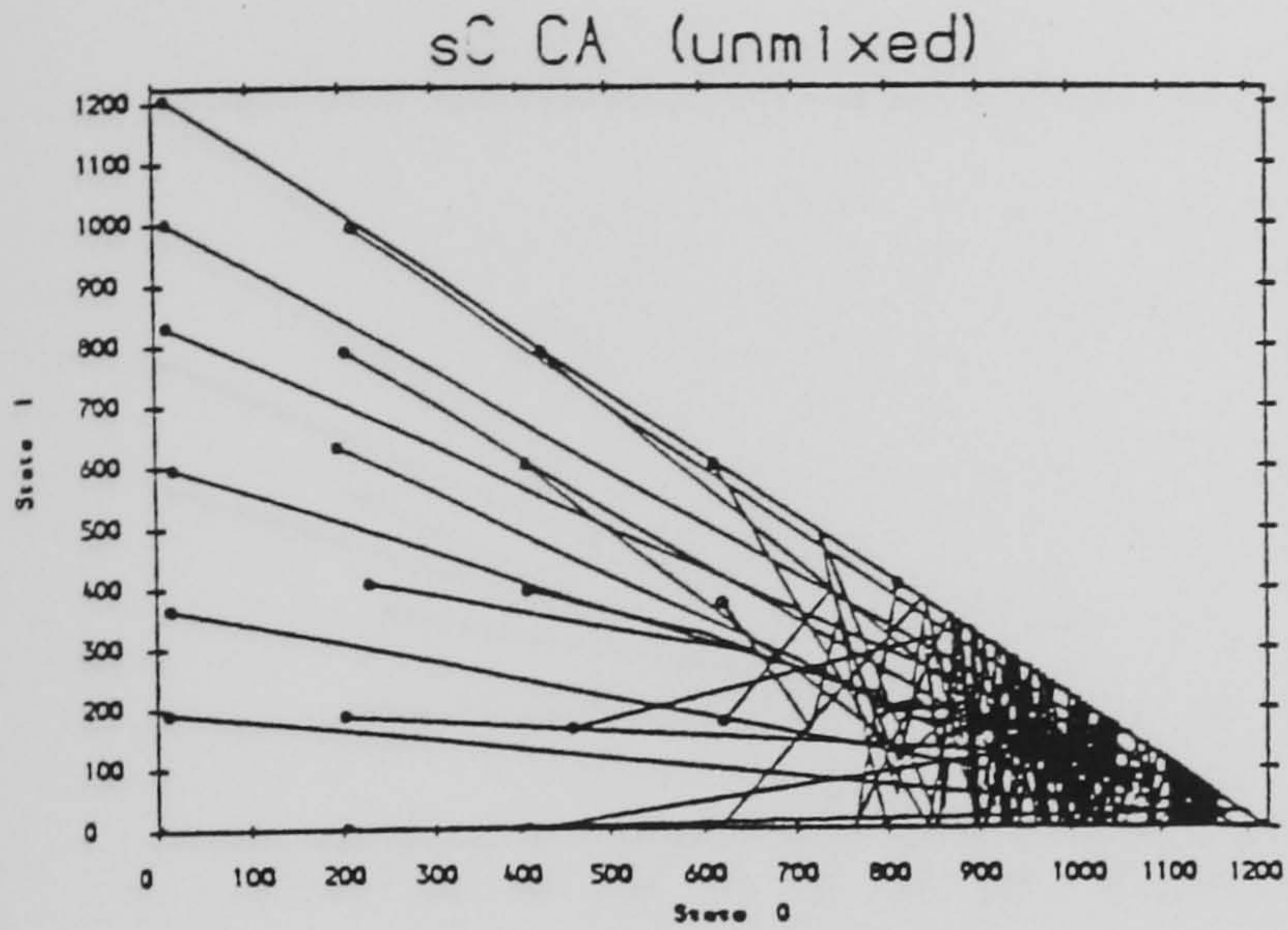




Figure 4.16: Class 4 phase diagrams.  
(d) System D: Deterministic.

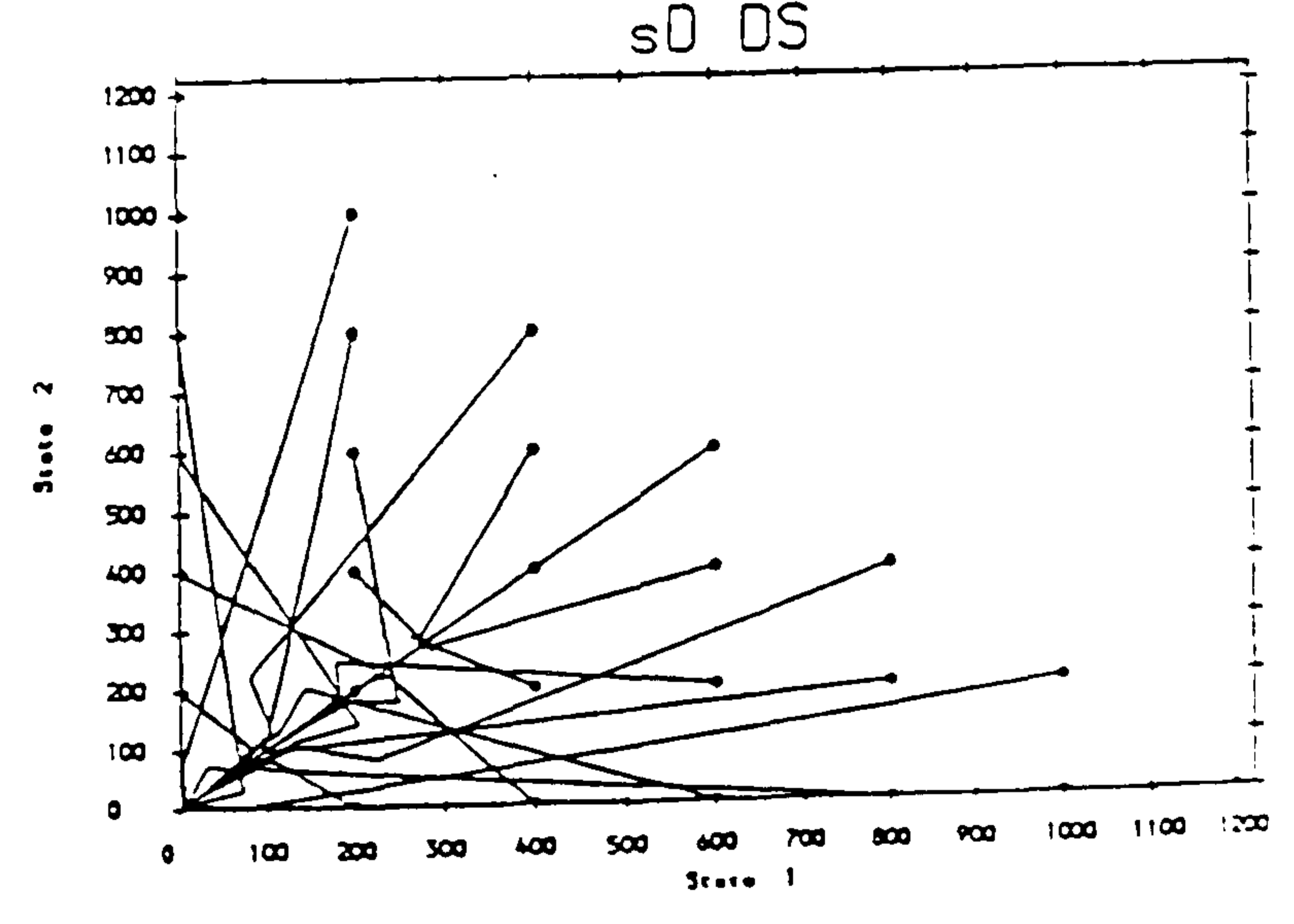
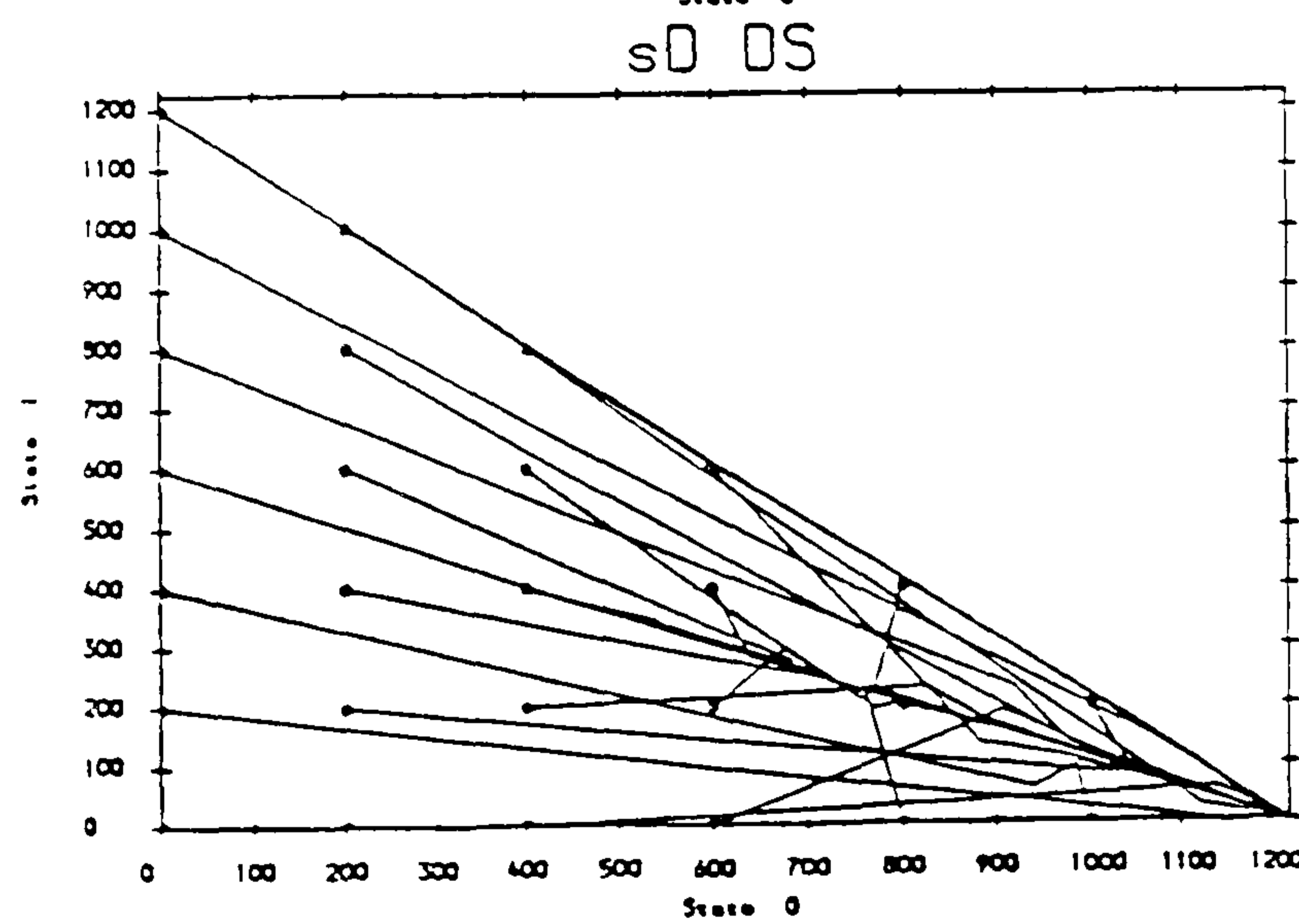
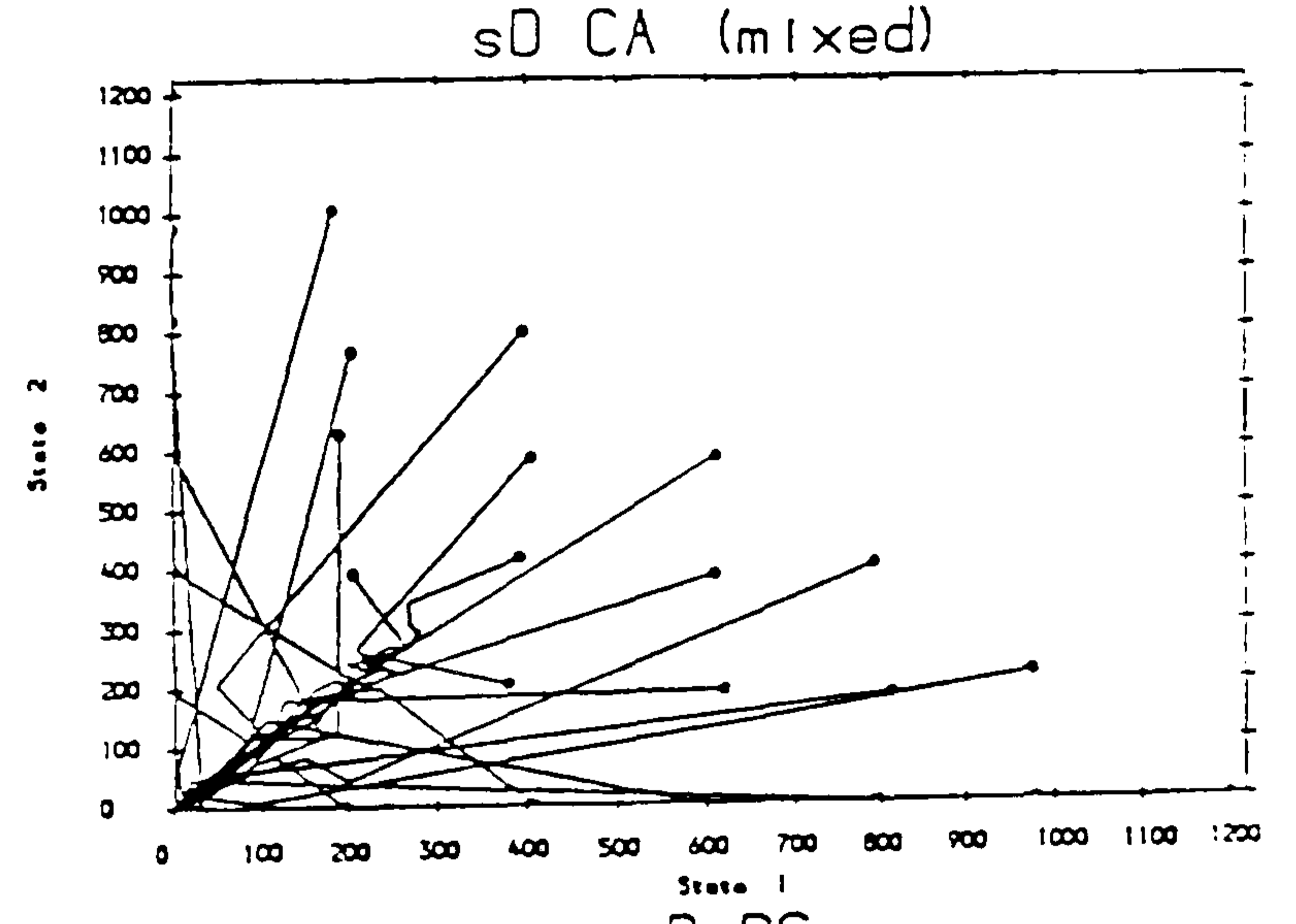
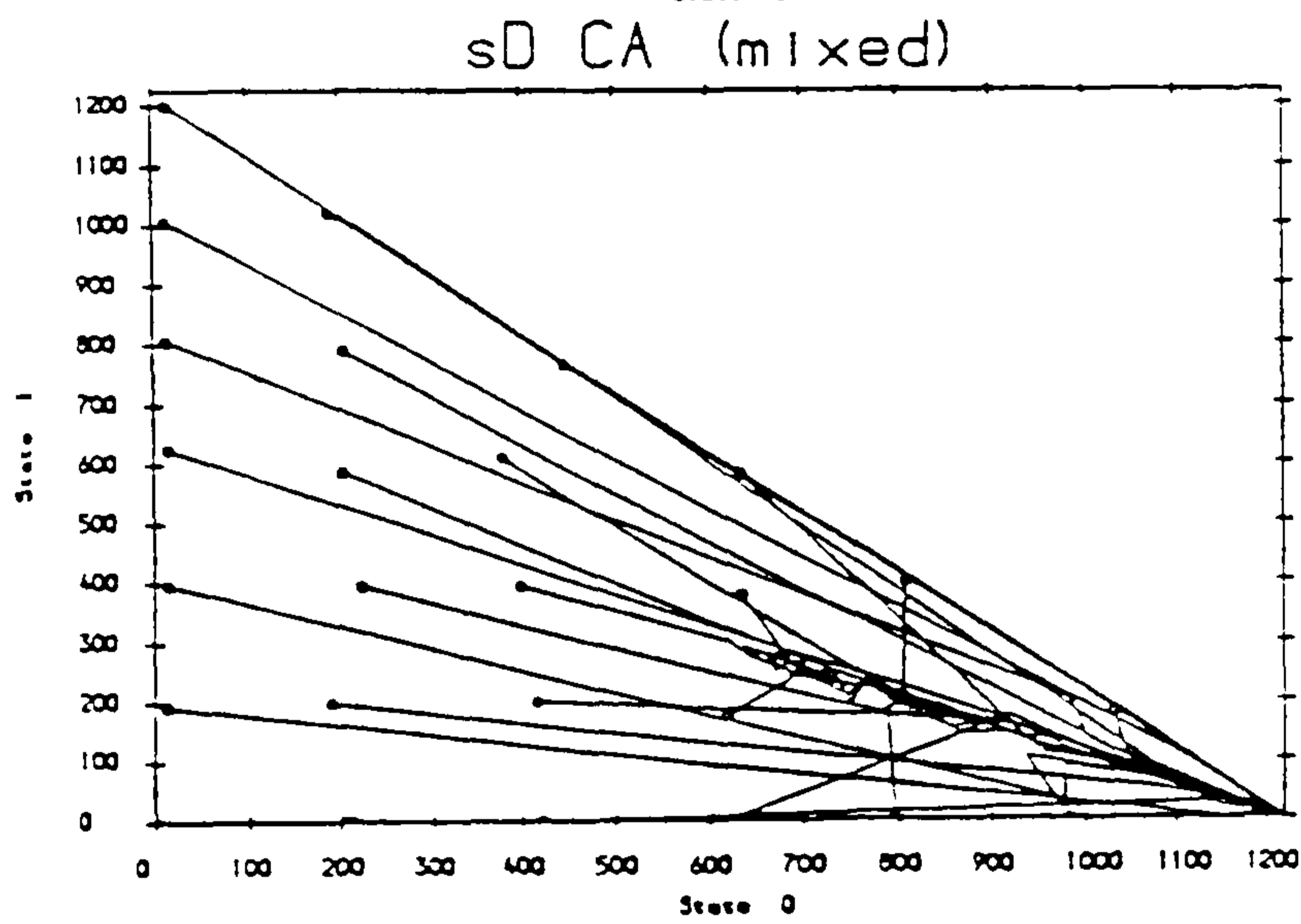
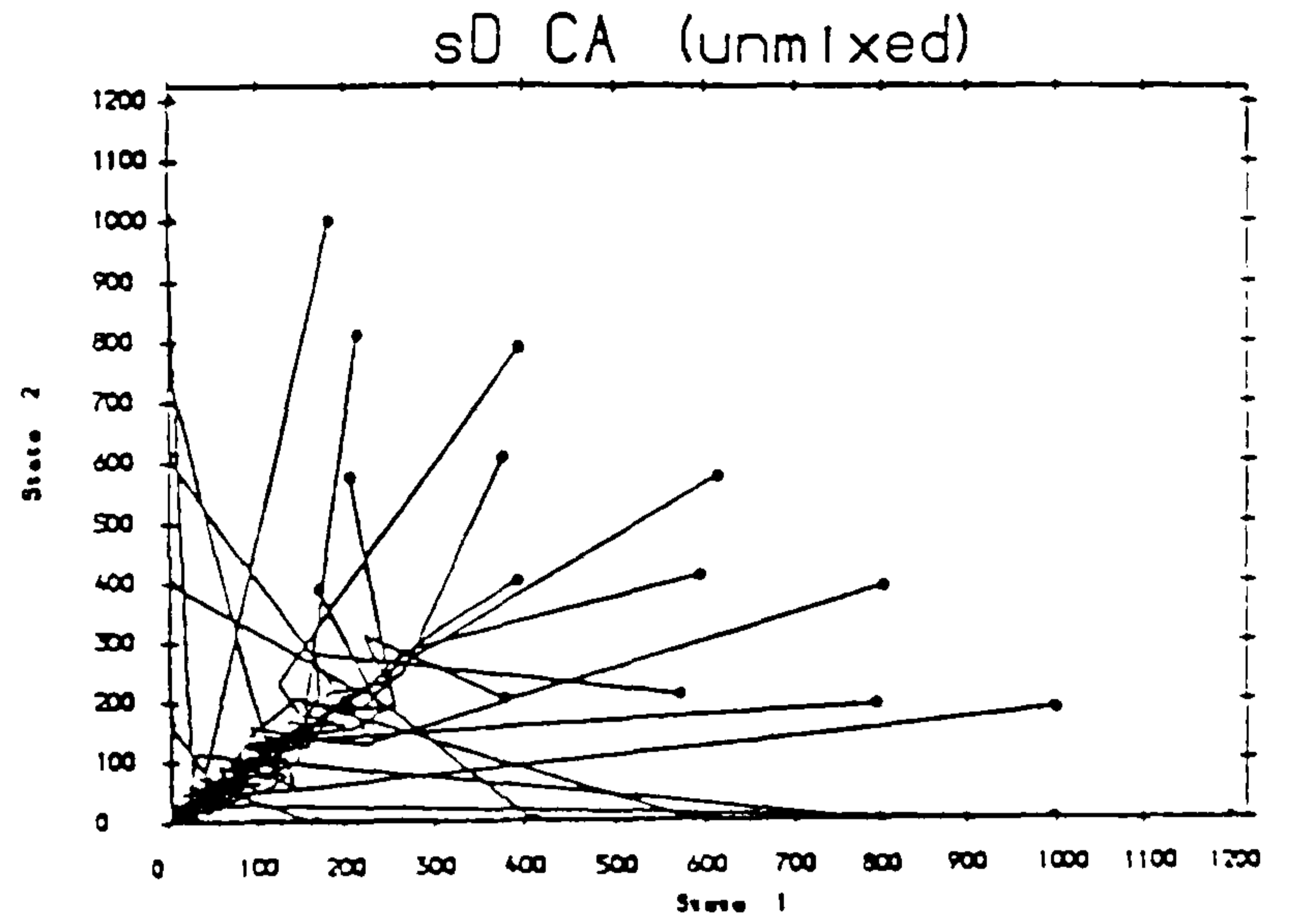
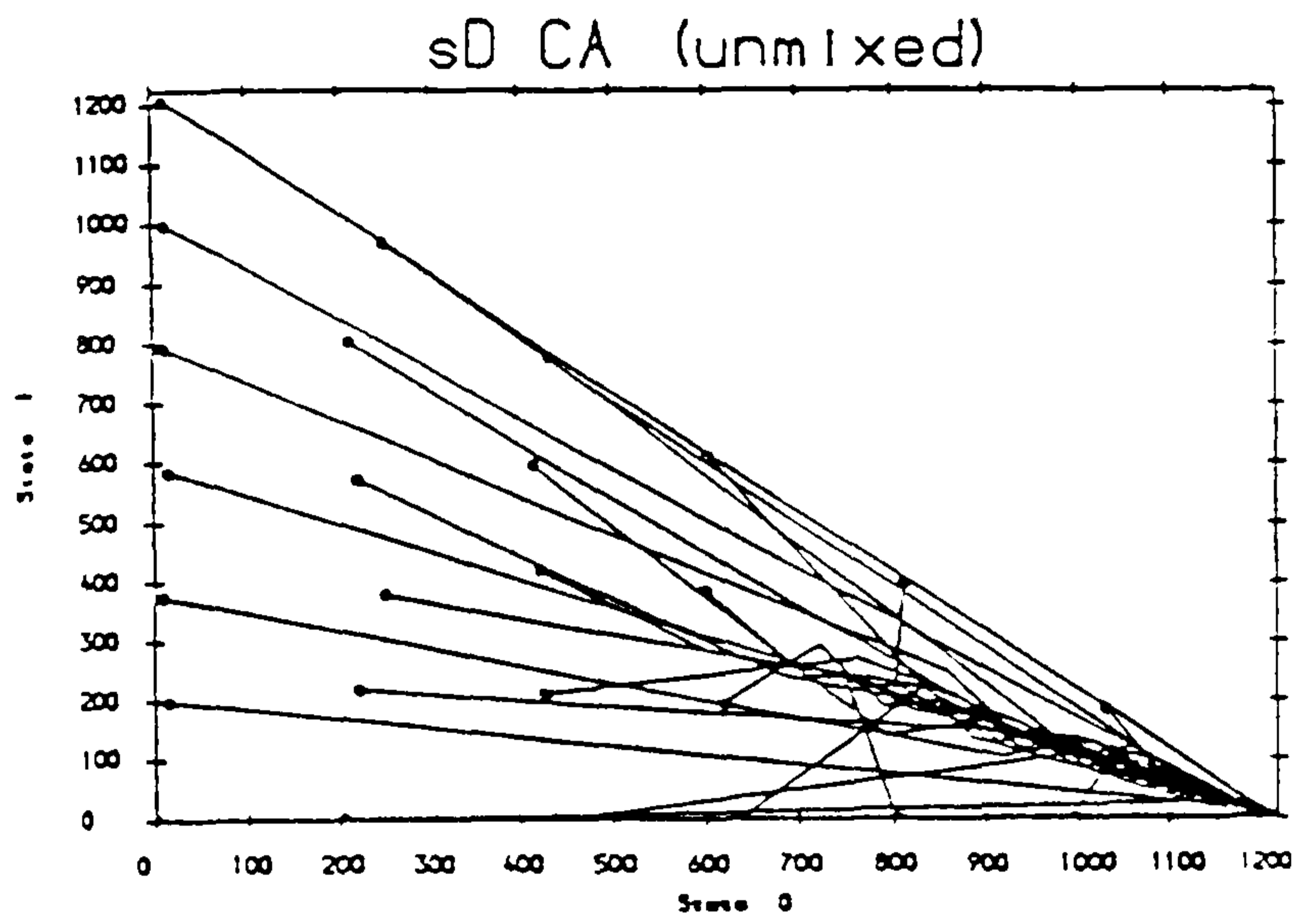


Figure 4.16: Class 4 phase diagrams.  
 (a) System A: Probabilistic.

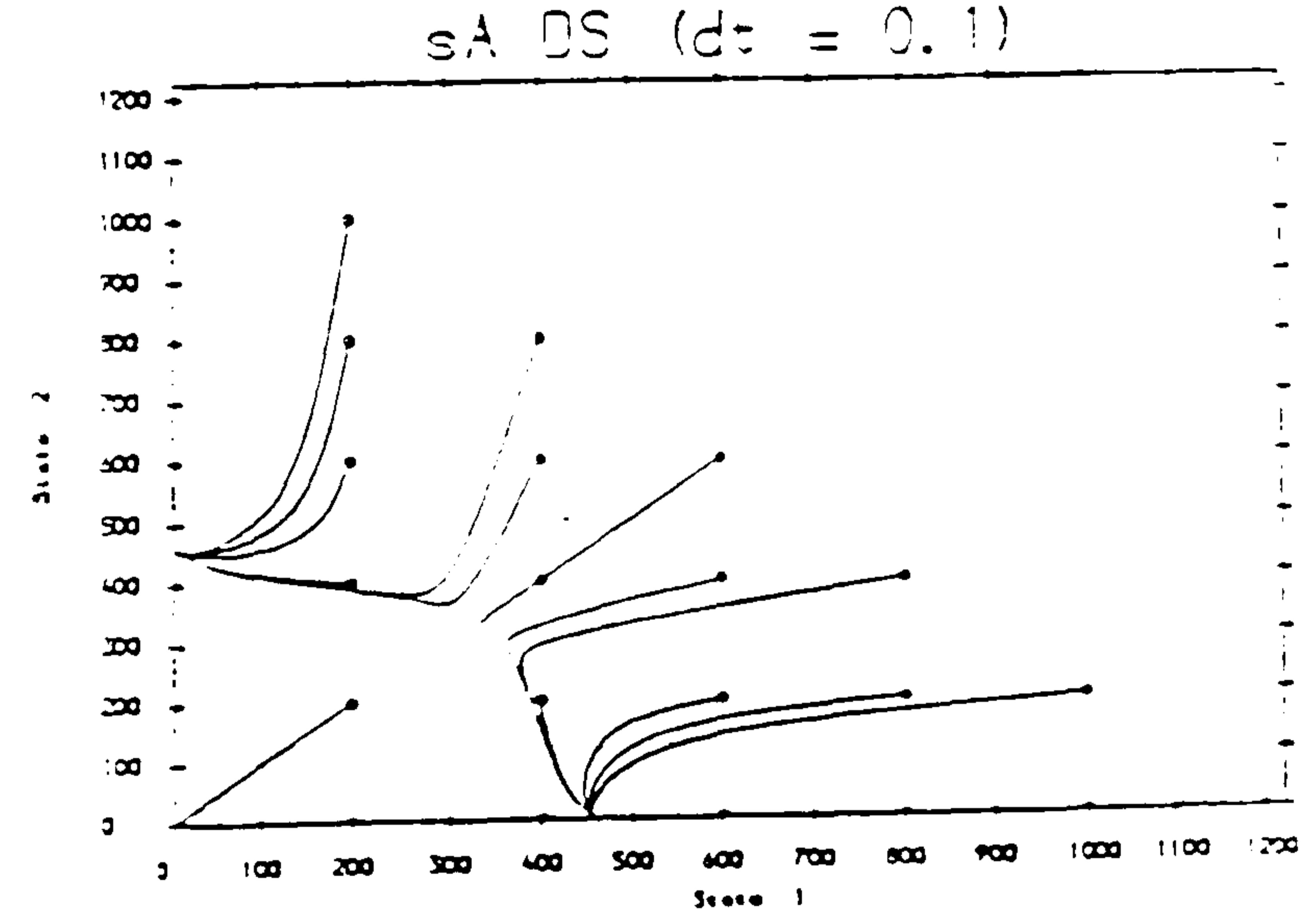
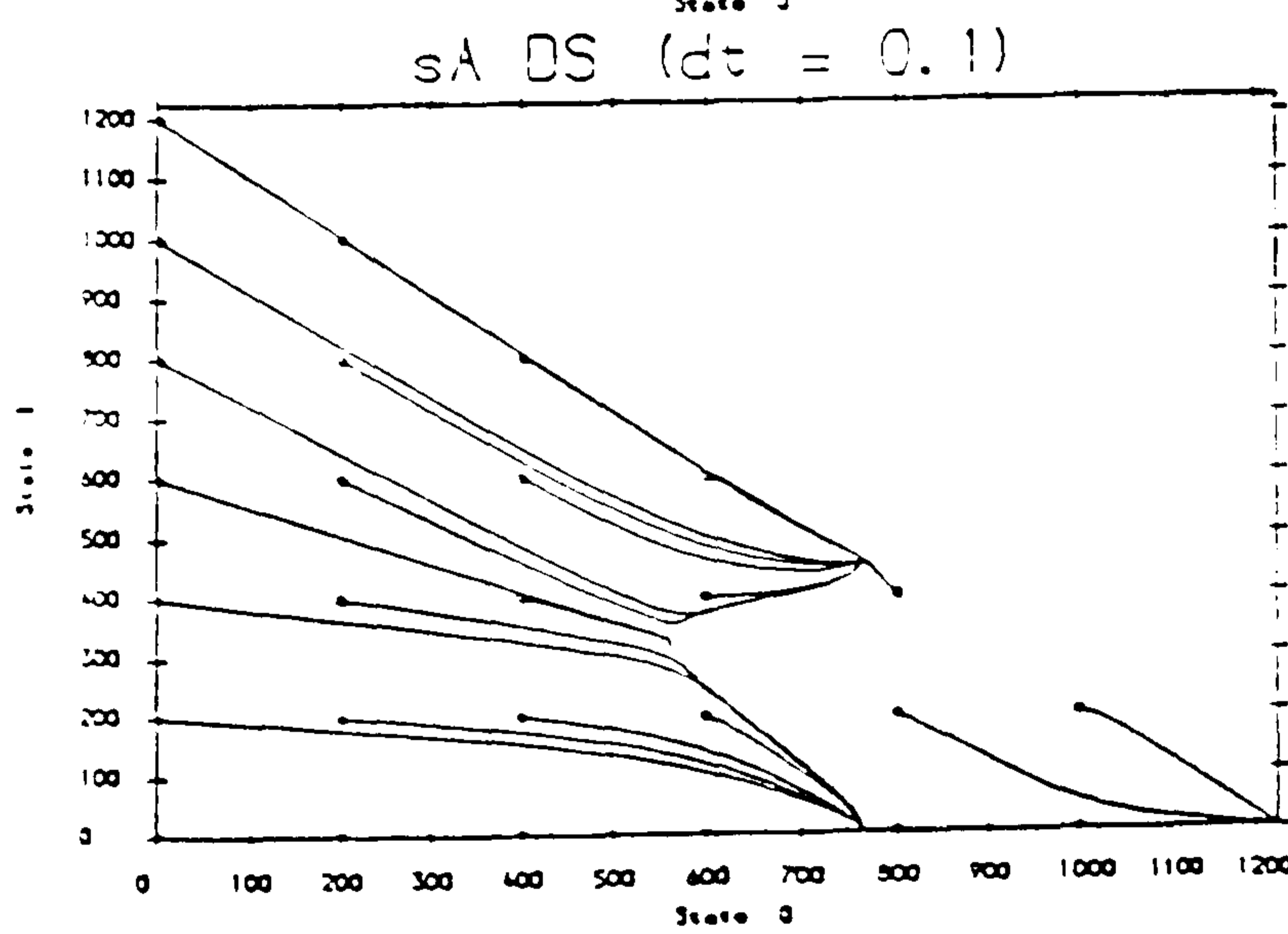
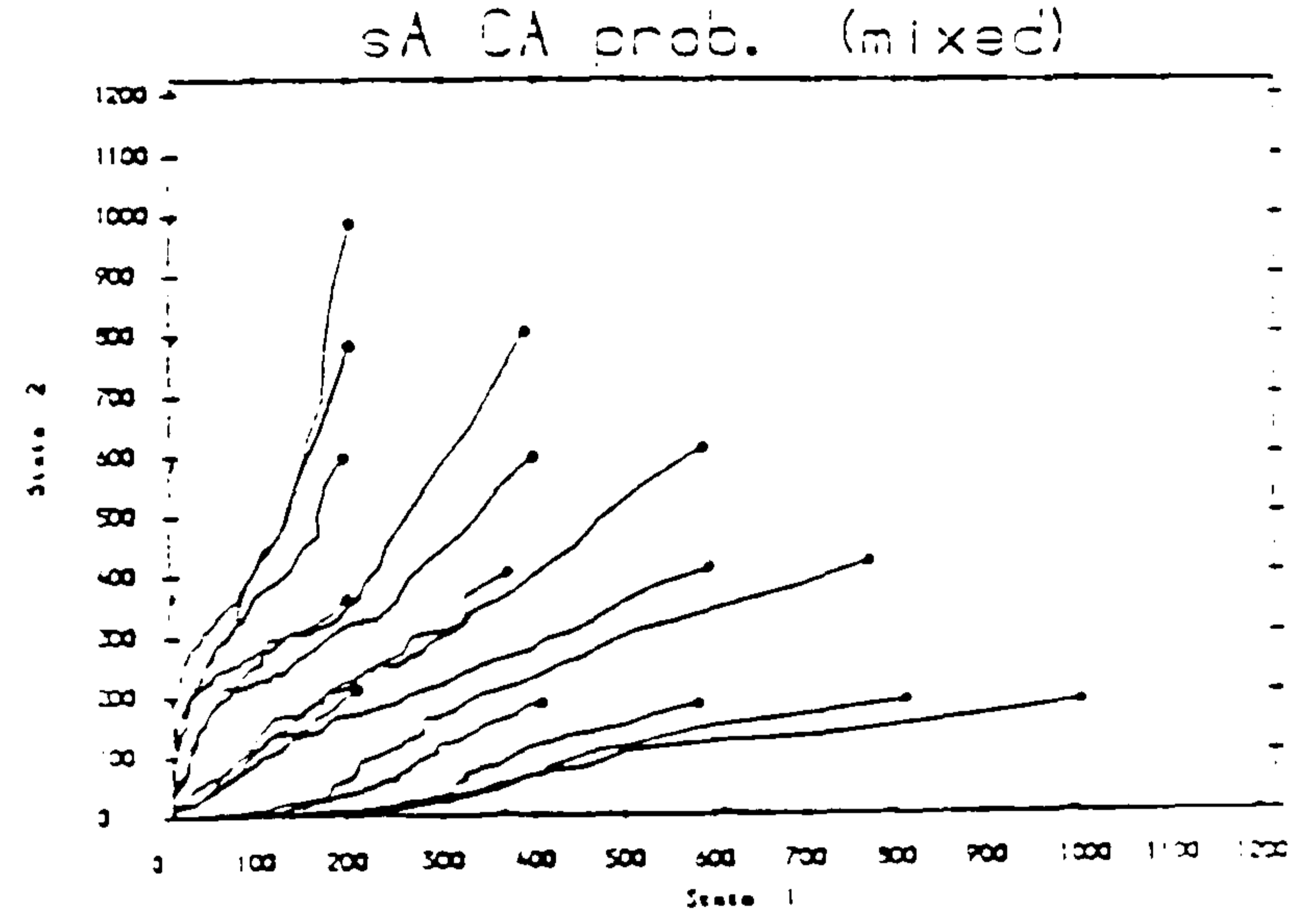
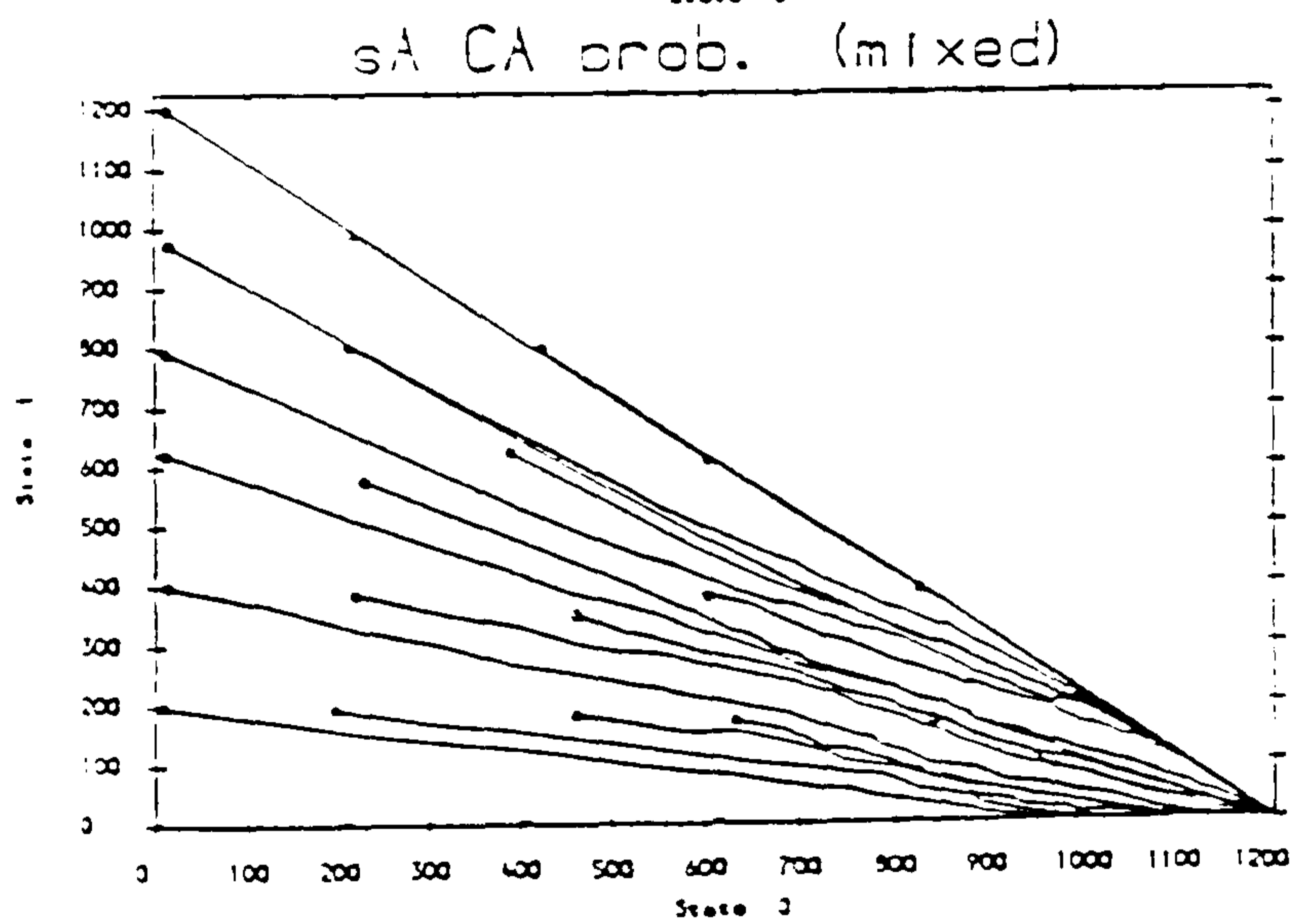
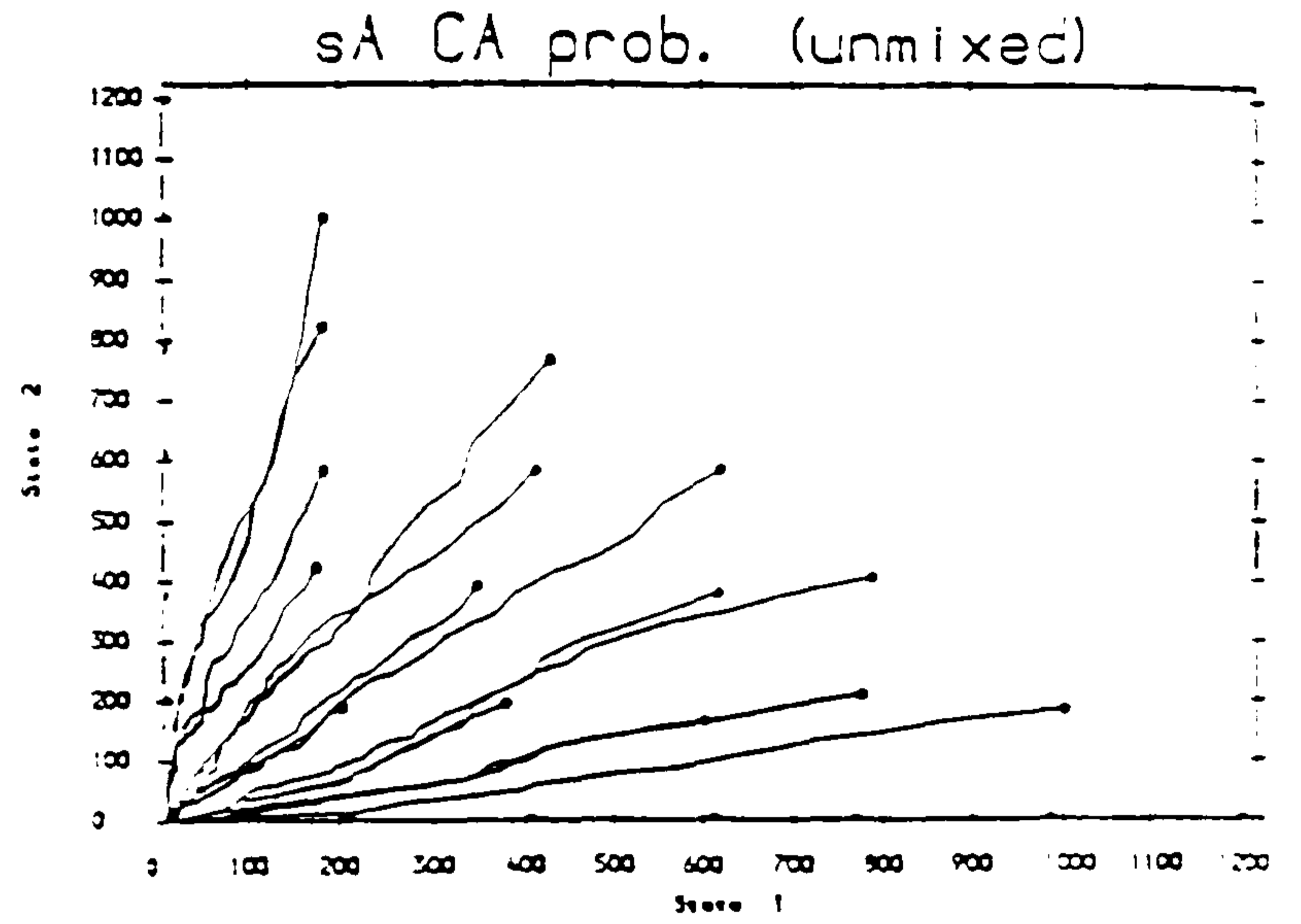
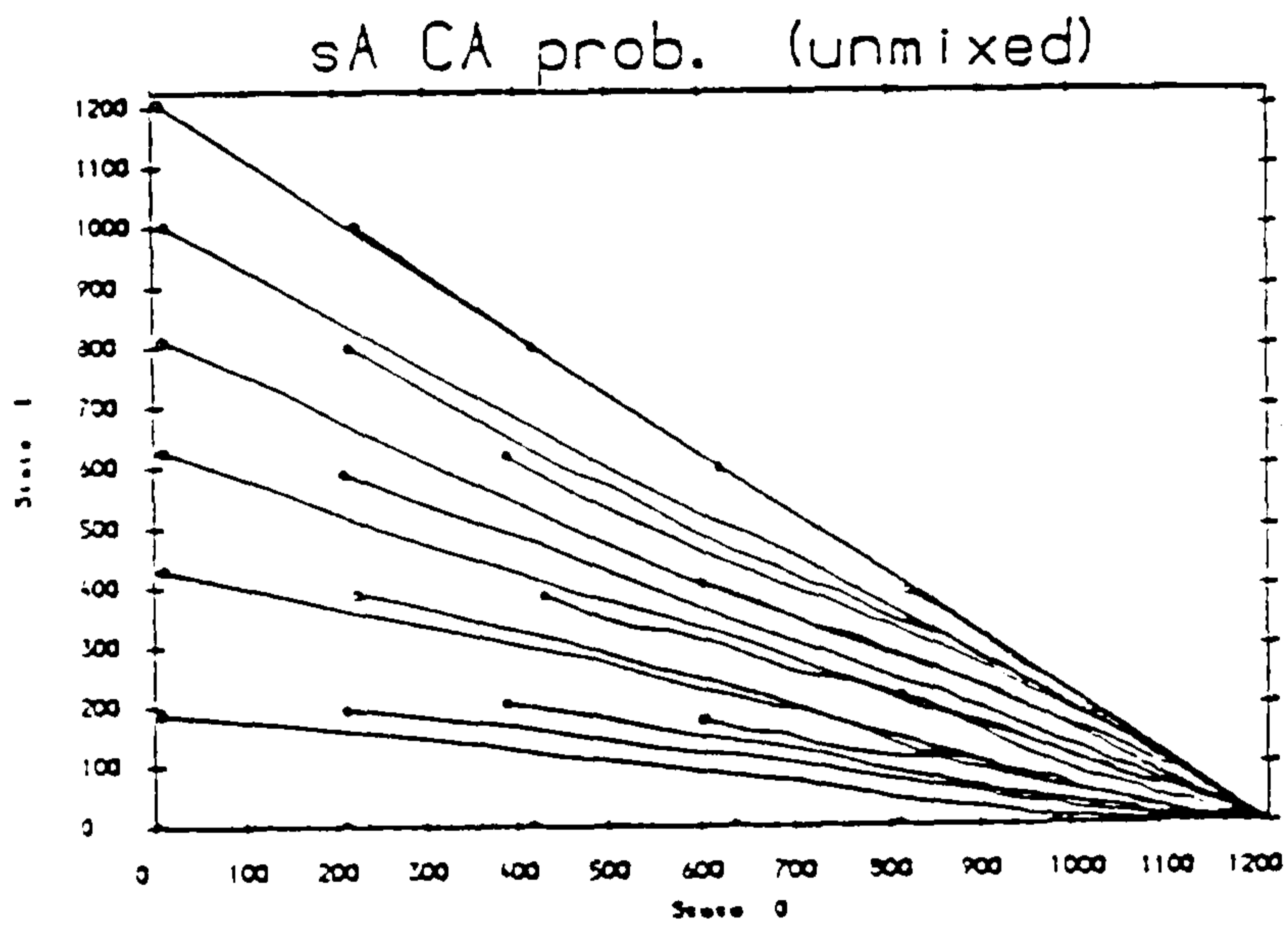




Figure 4.16: Class 4 phase diagrams.  
(b) System B: Probabilistic.

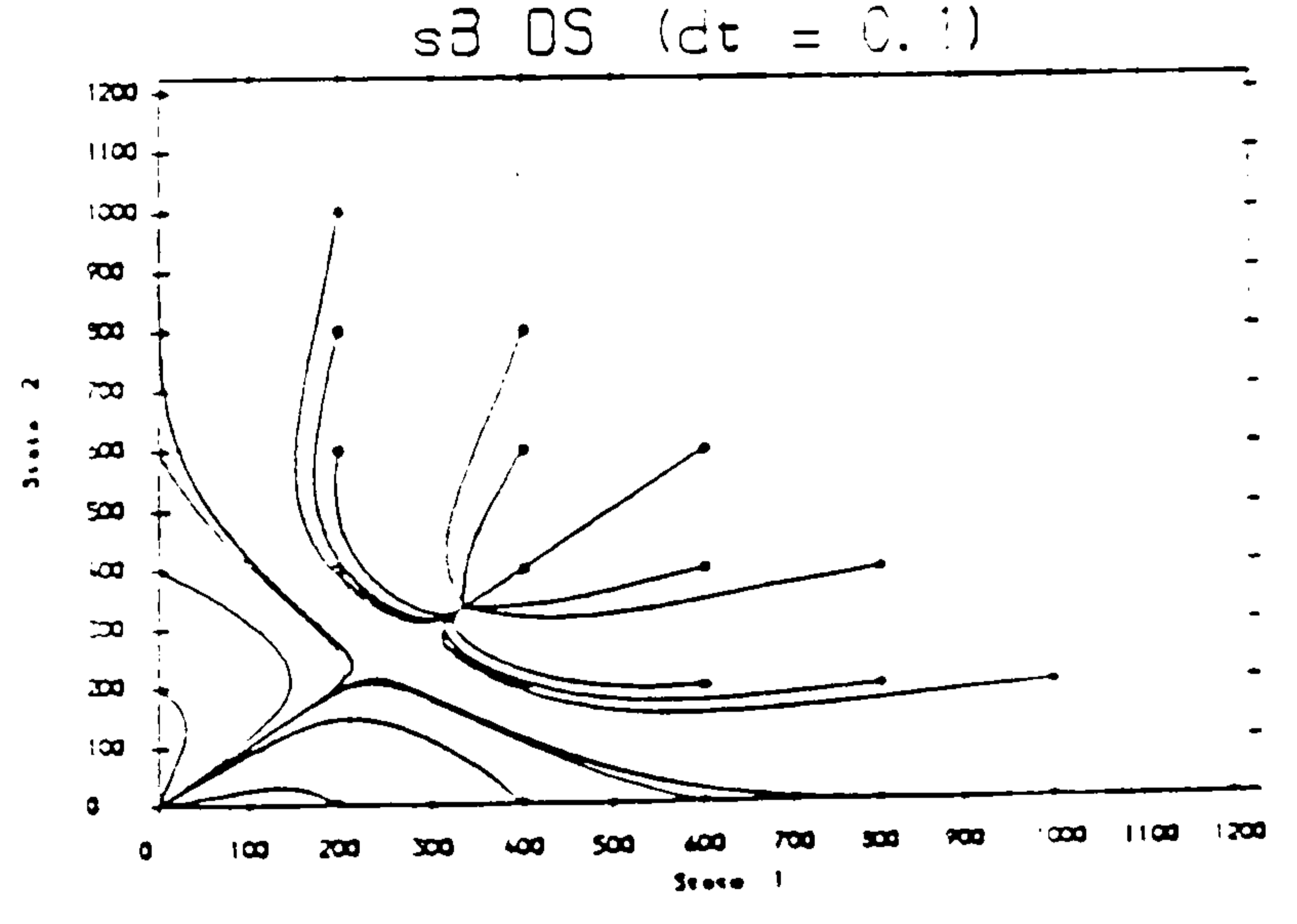
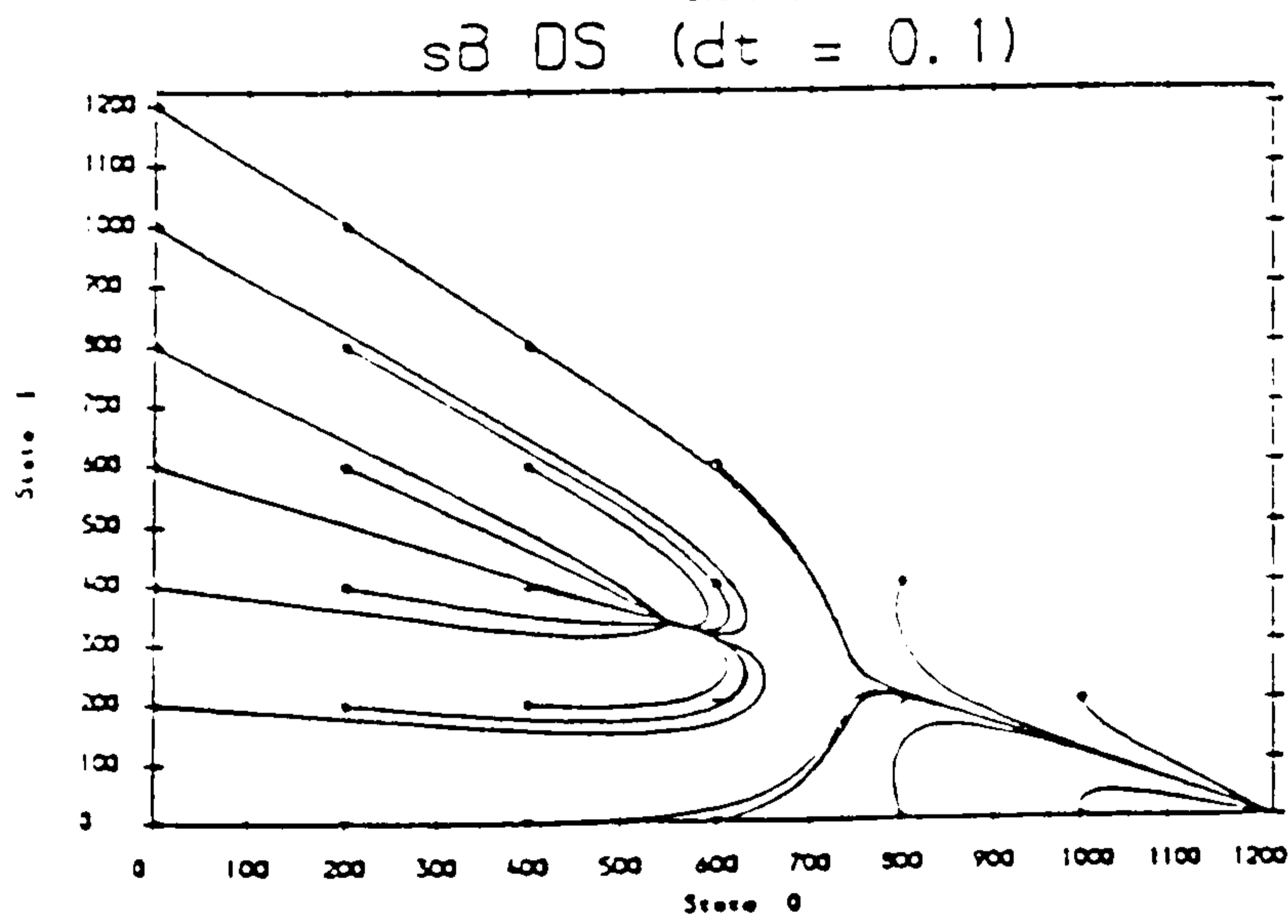
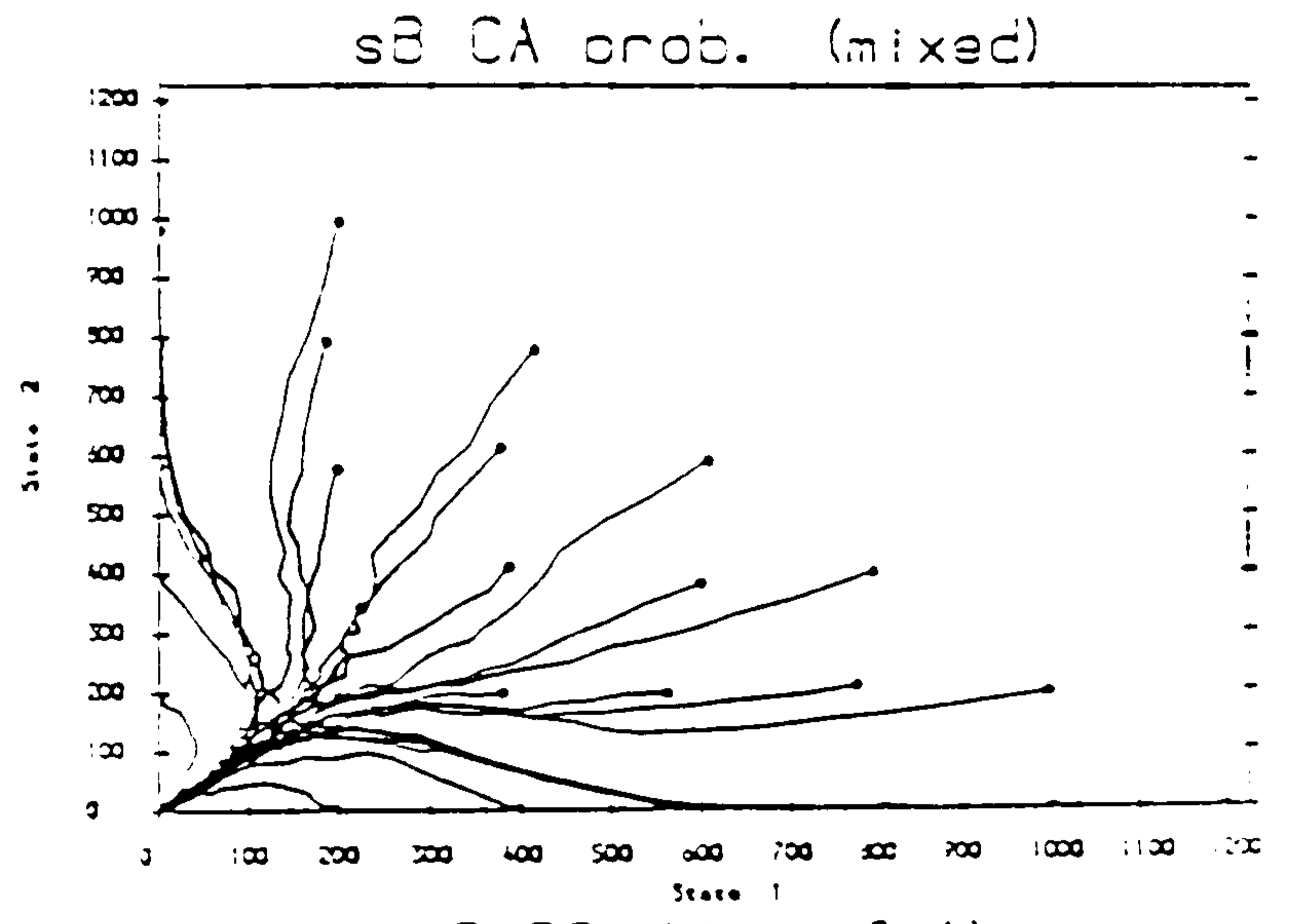
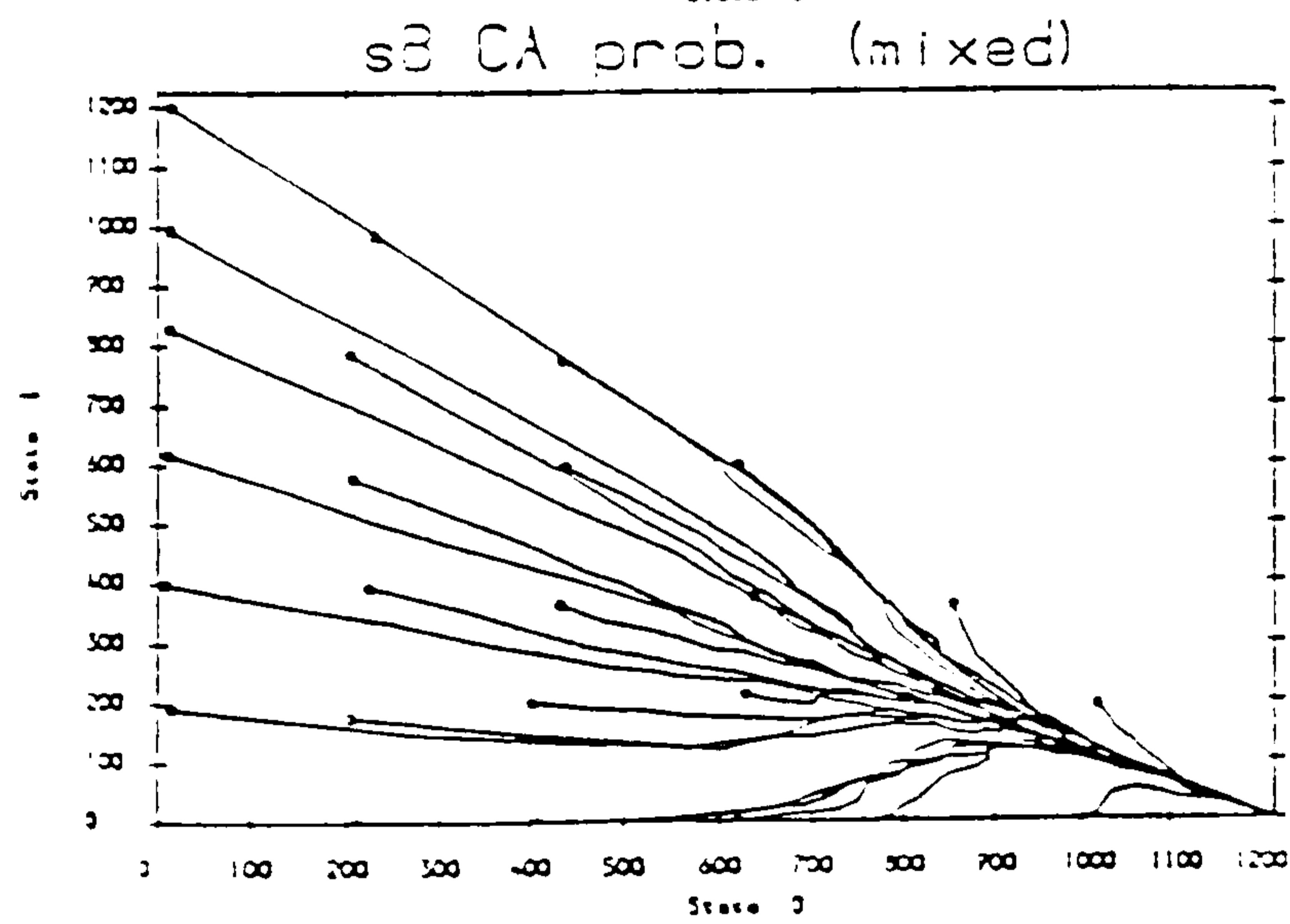
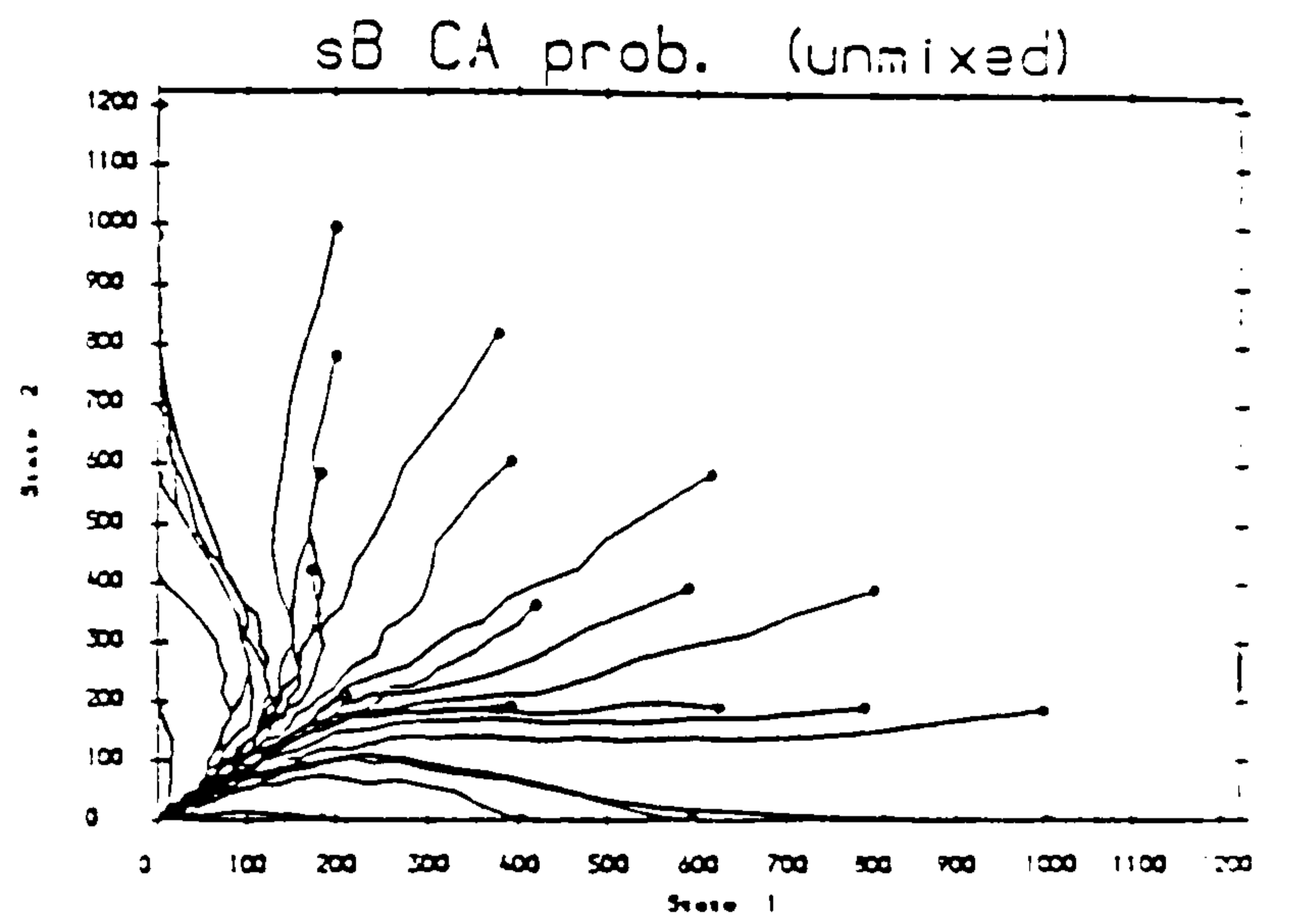
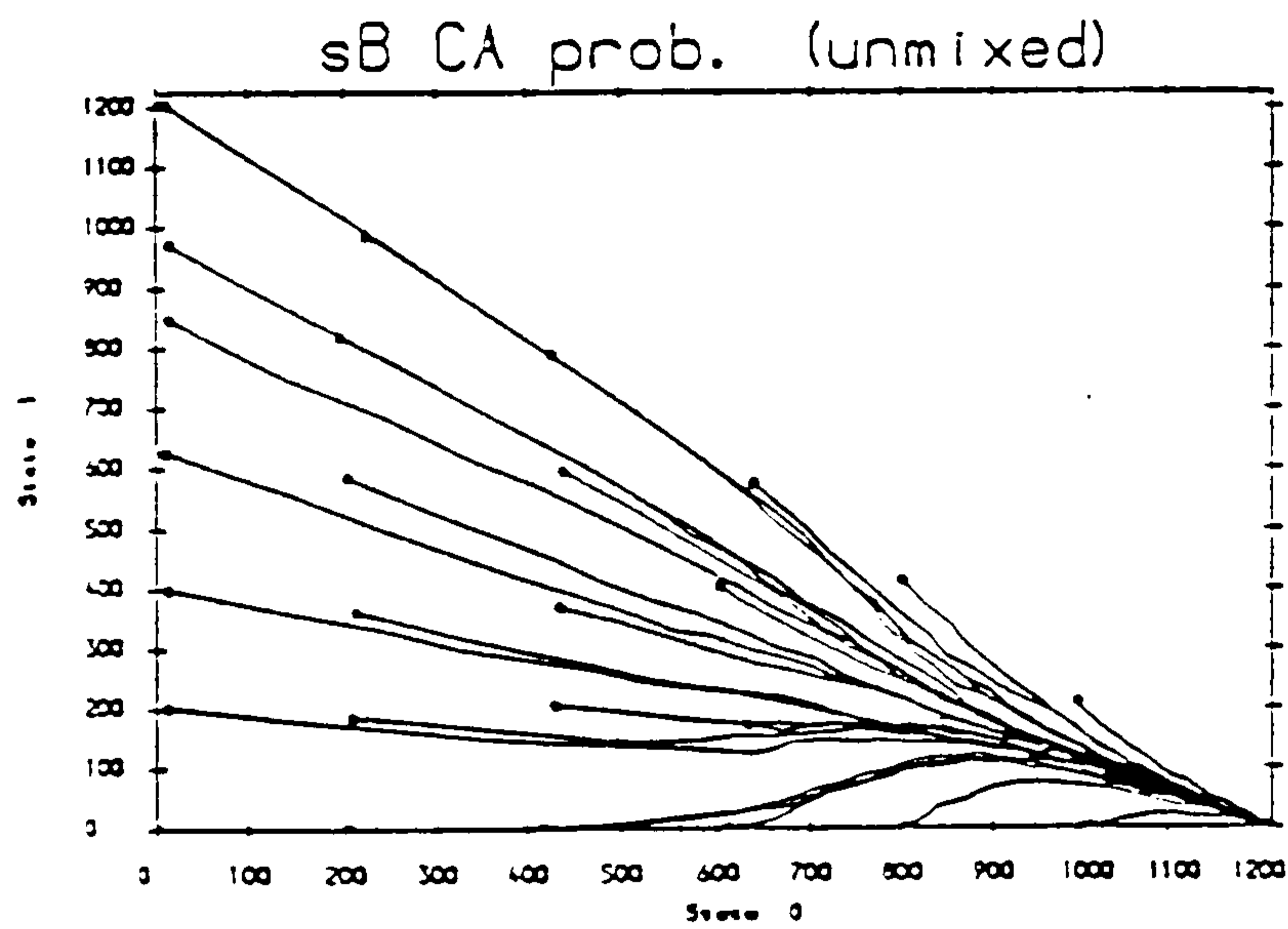


Figure 4.16: Class 4 phase diagrams.  
(c) System C: Probabilistic.

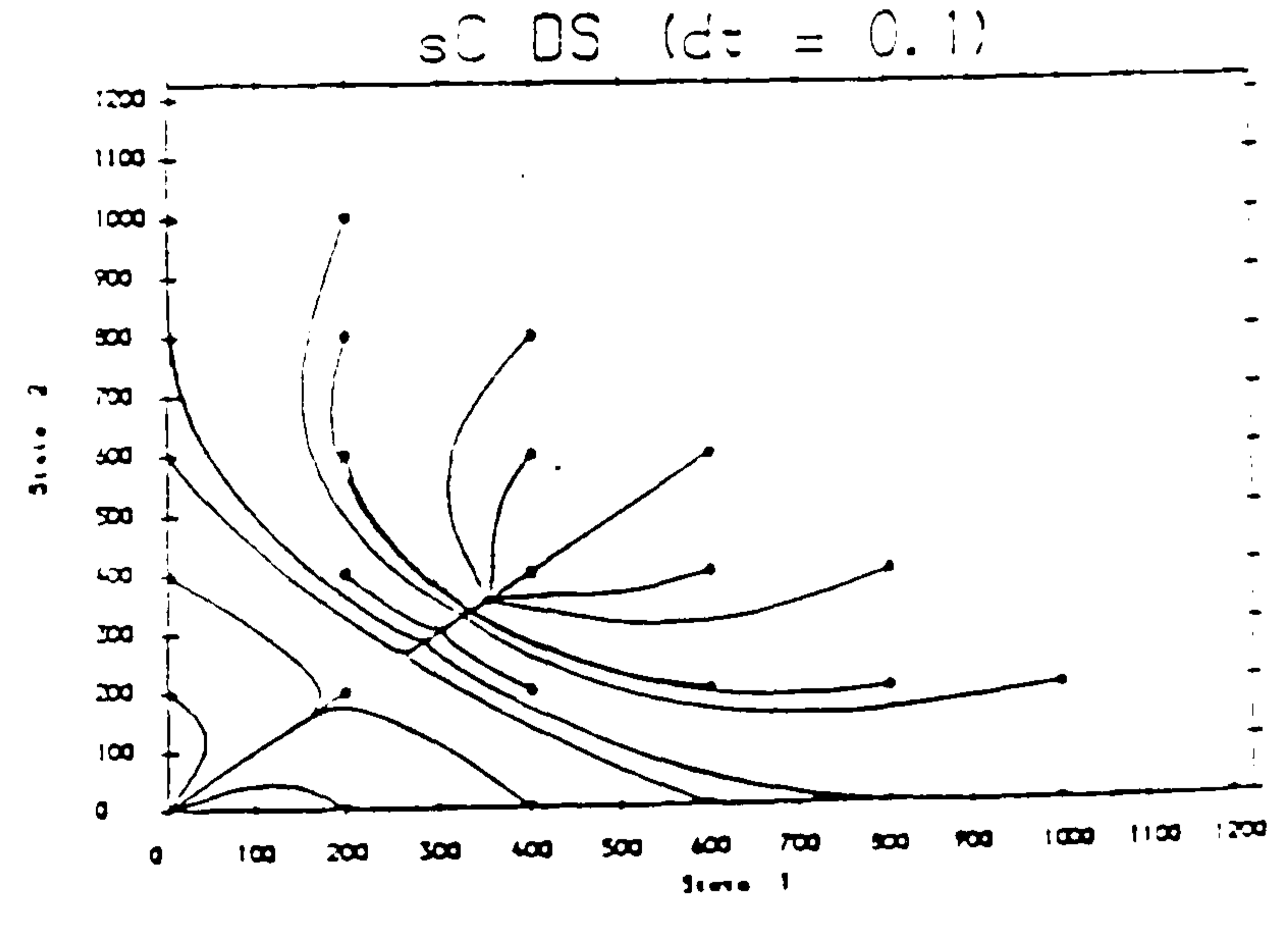
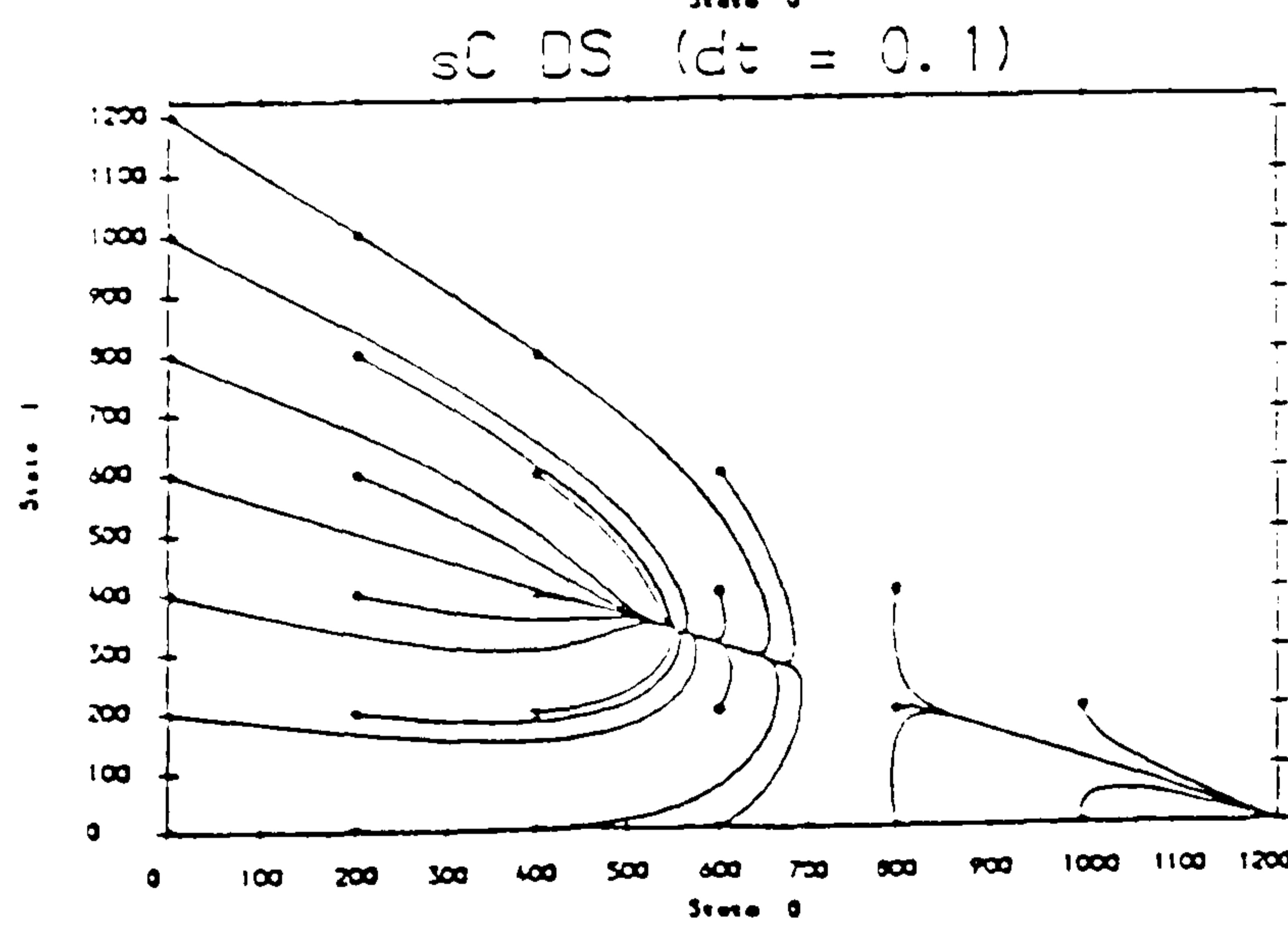
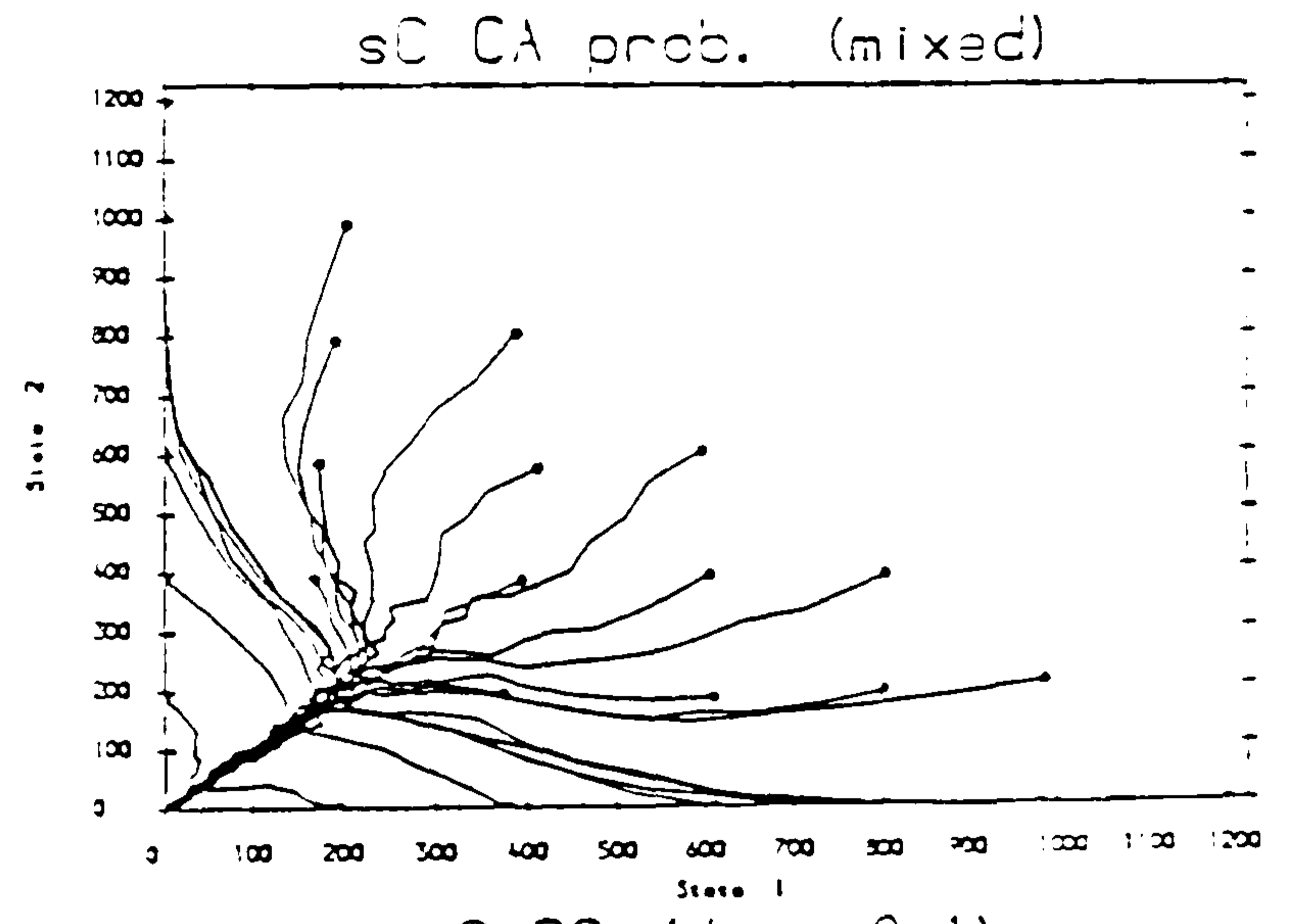
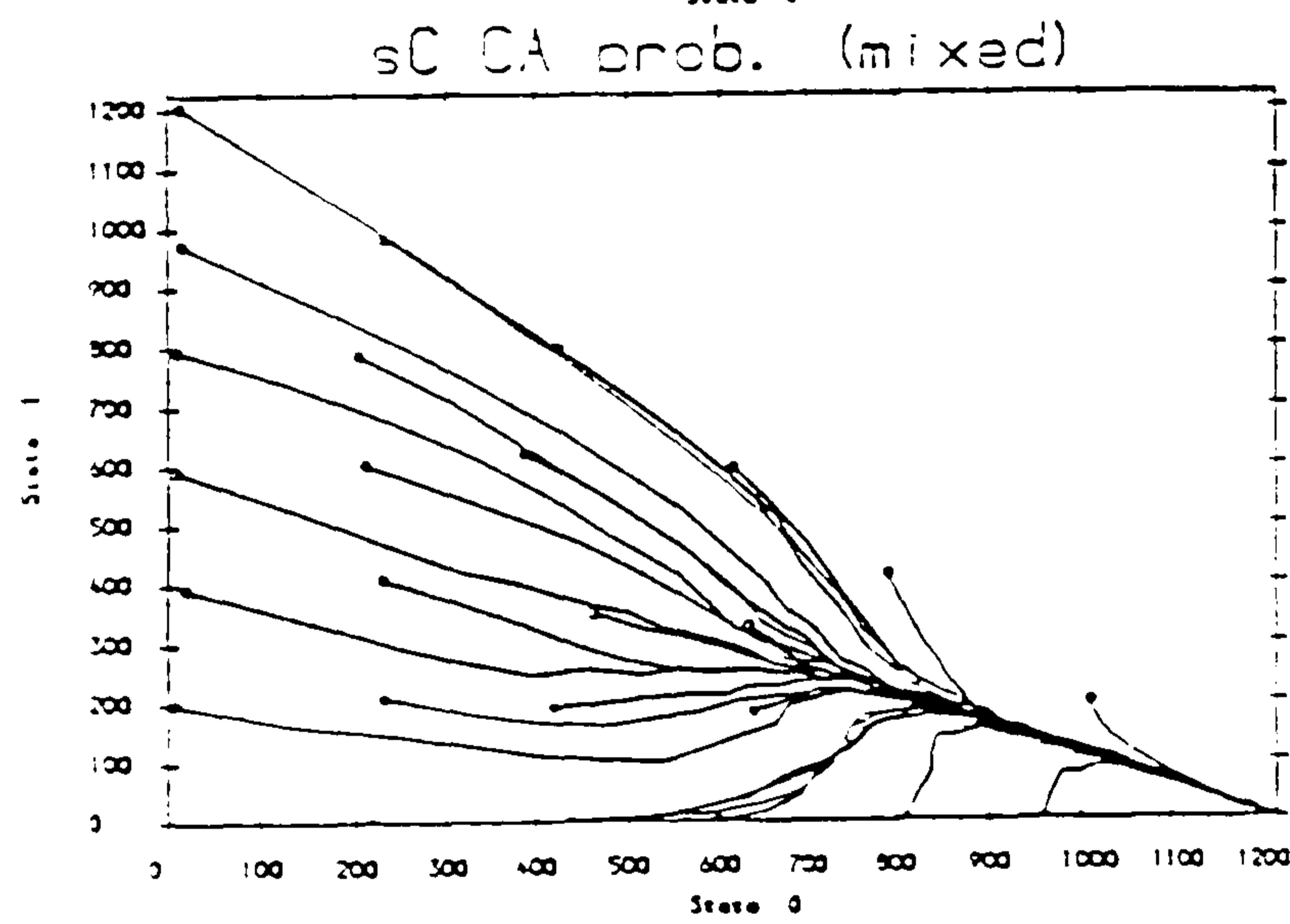
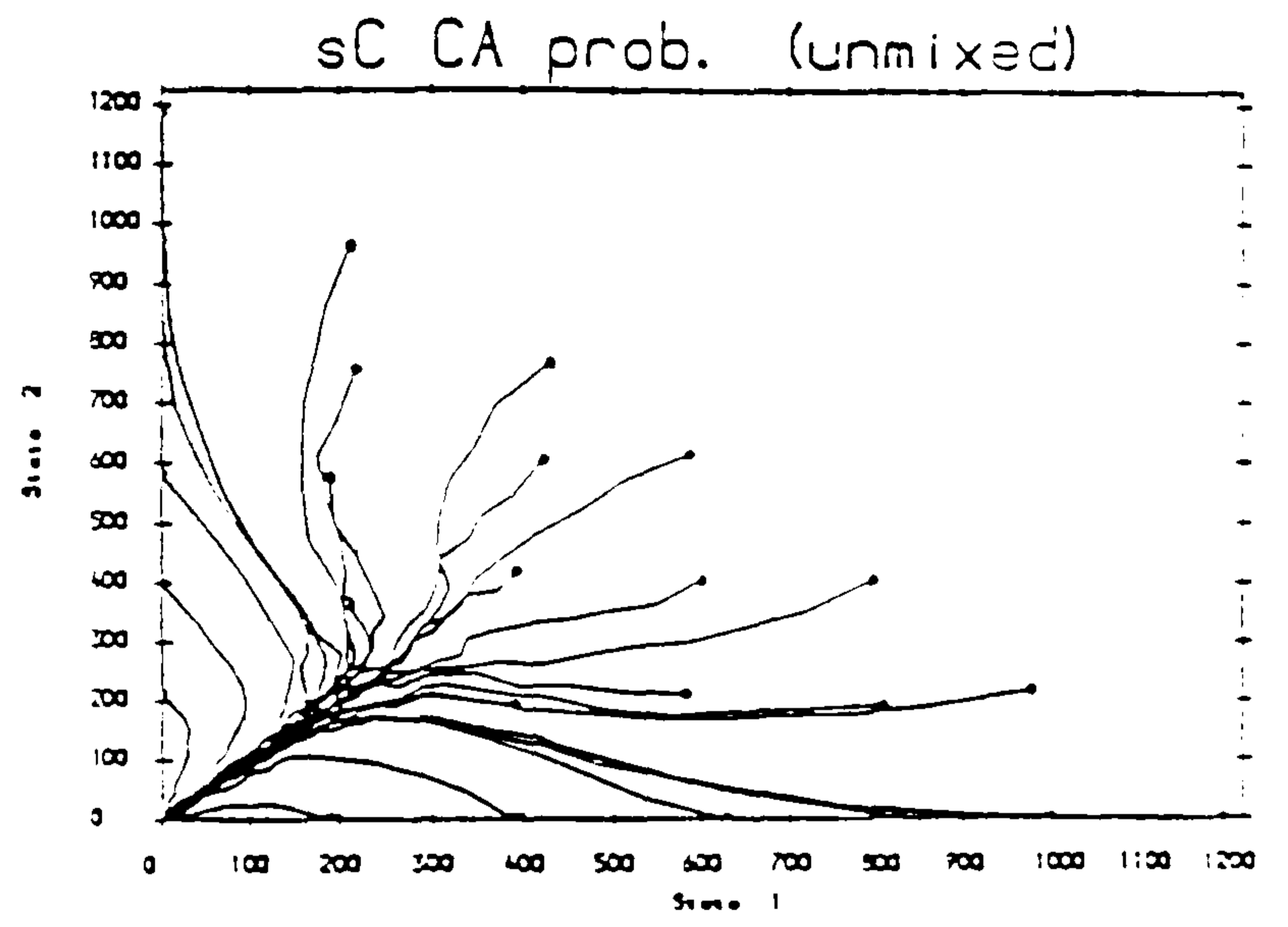
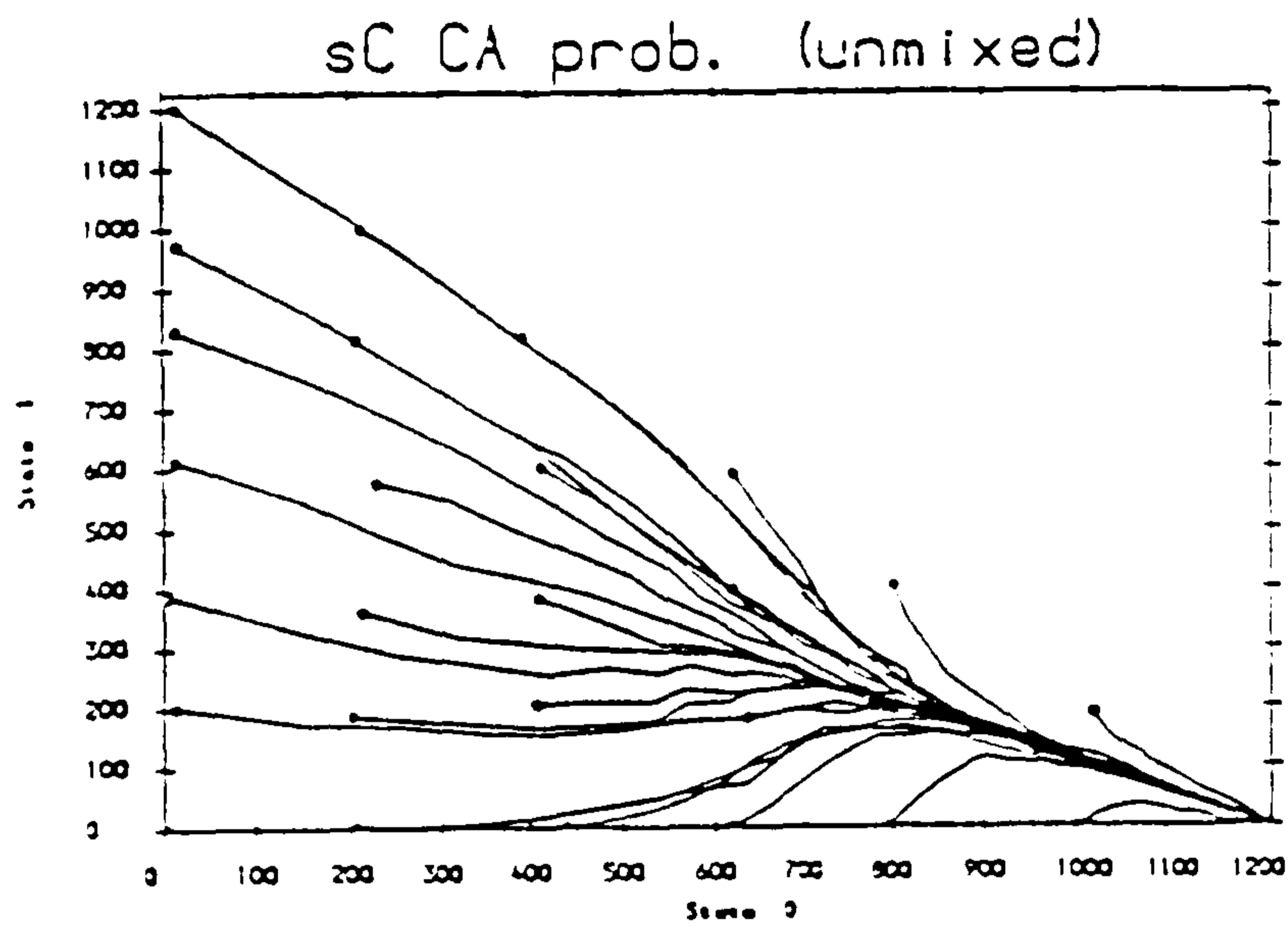
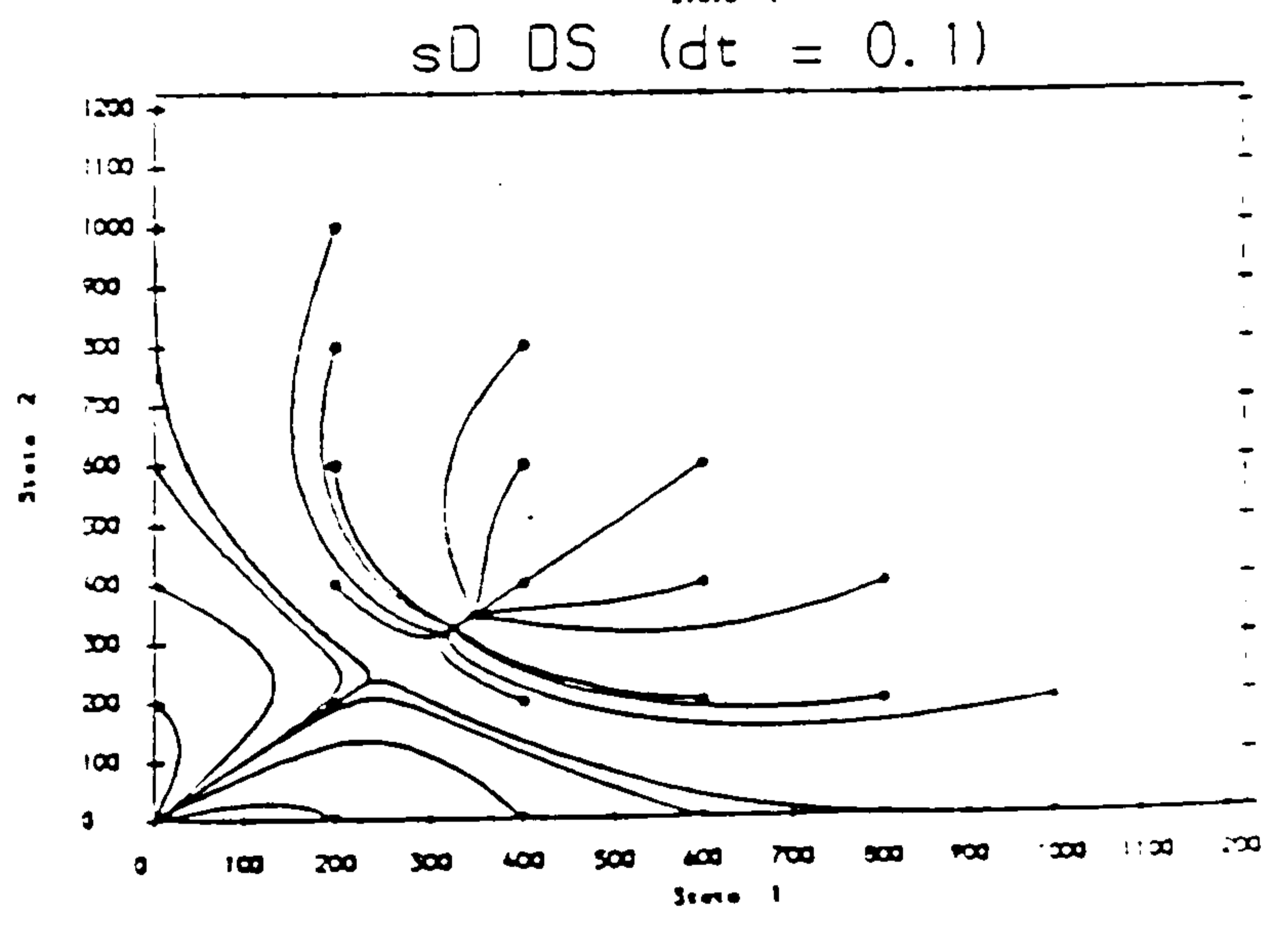
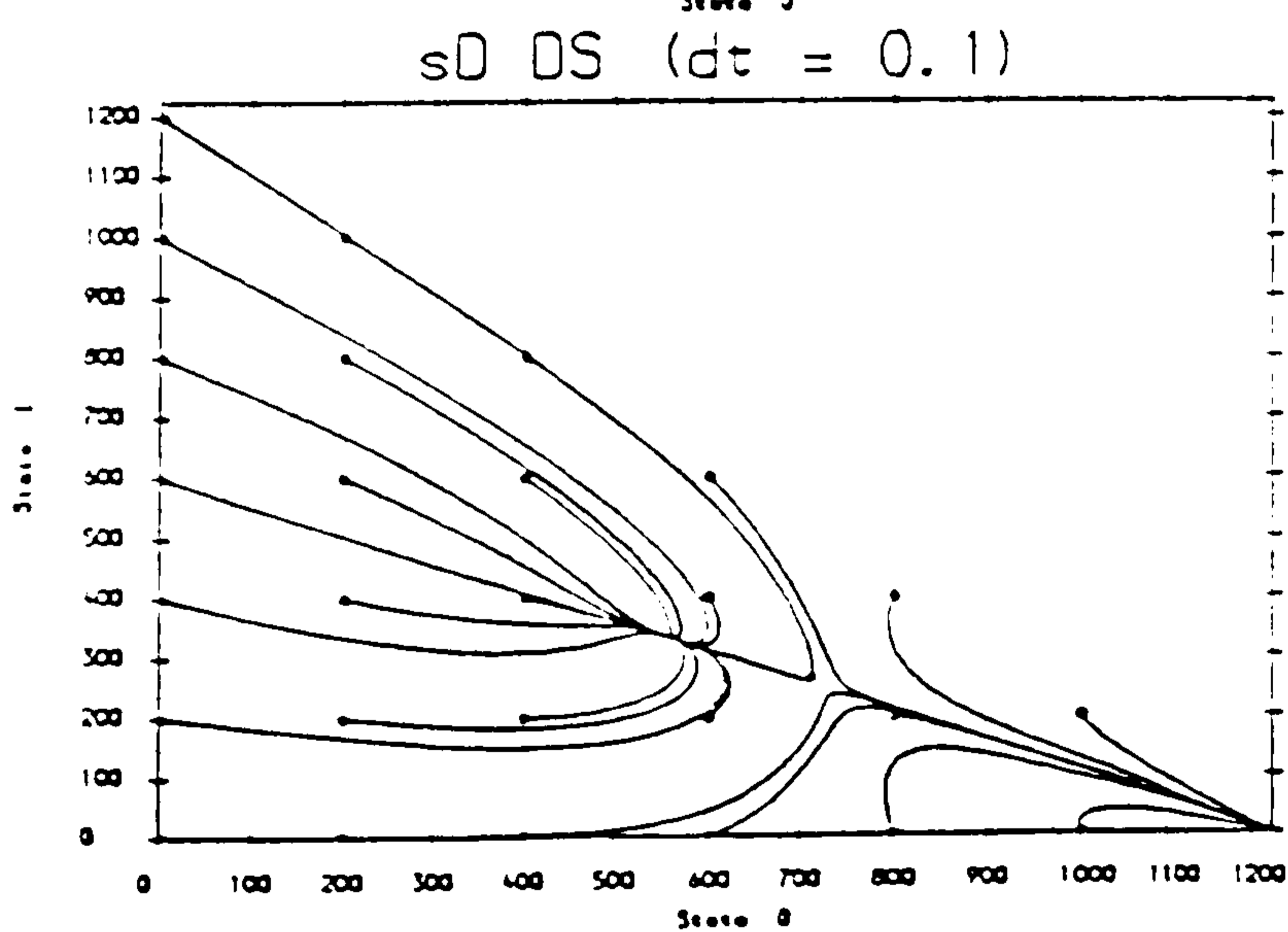
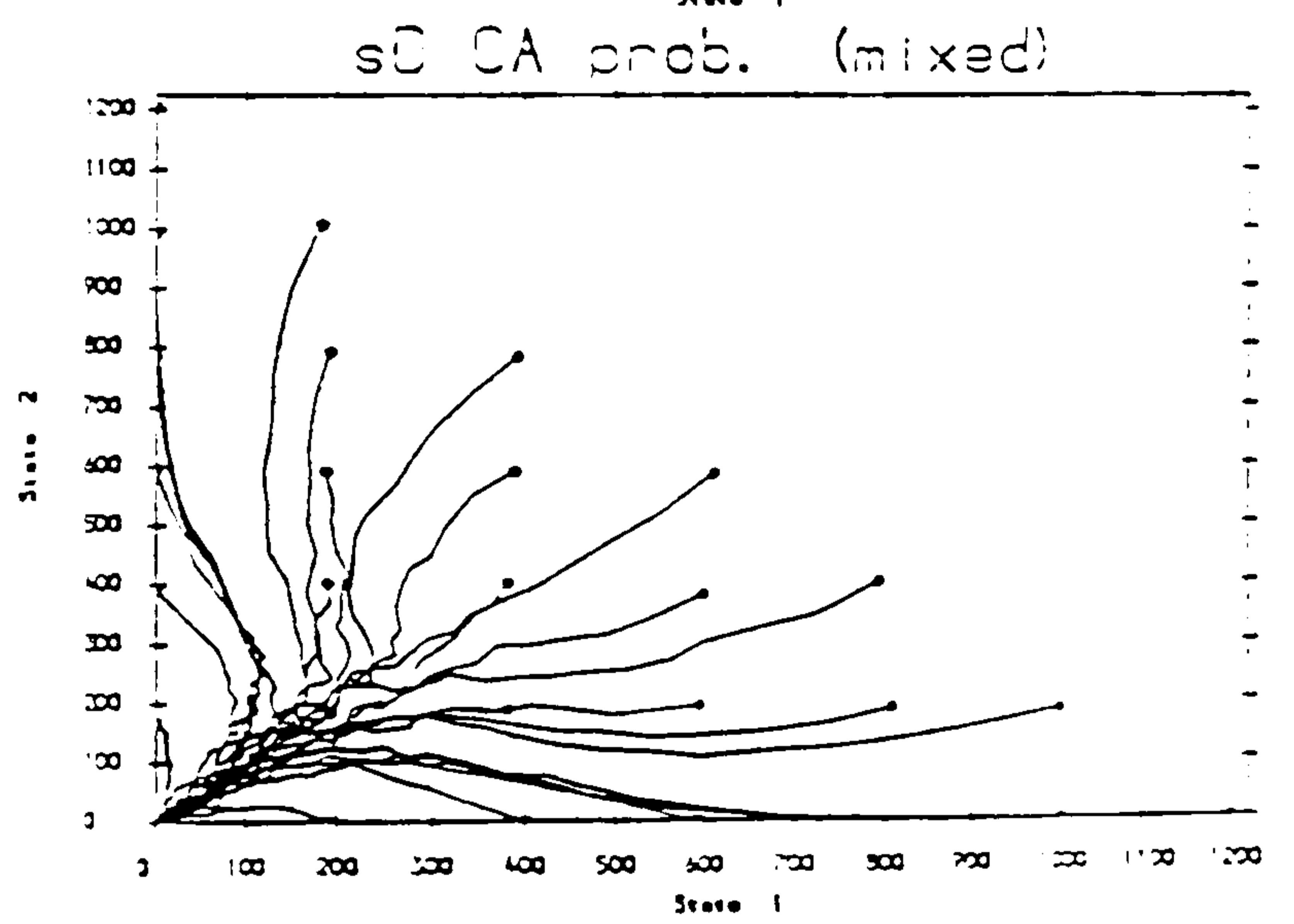
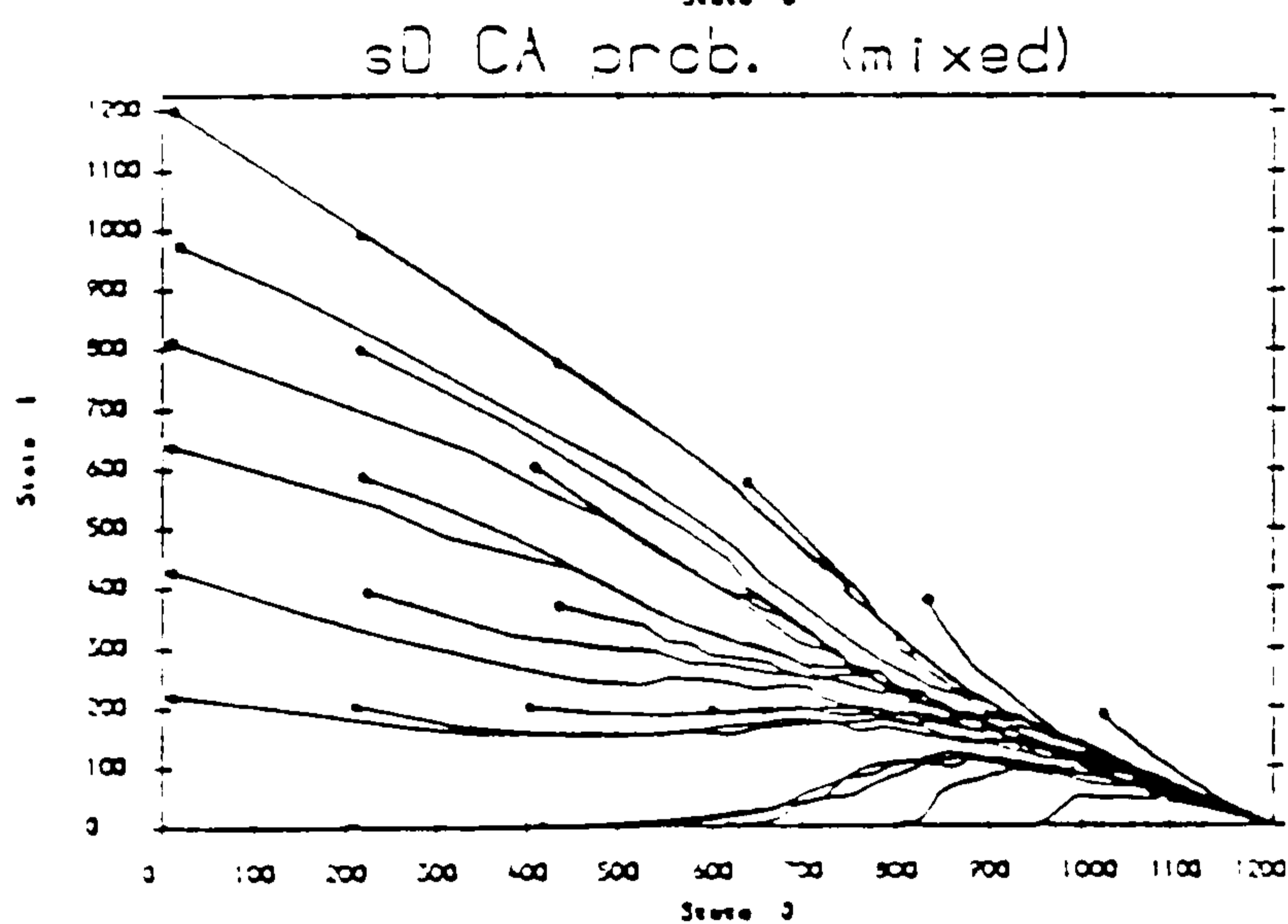
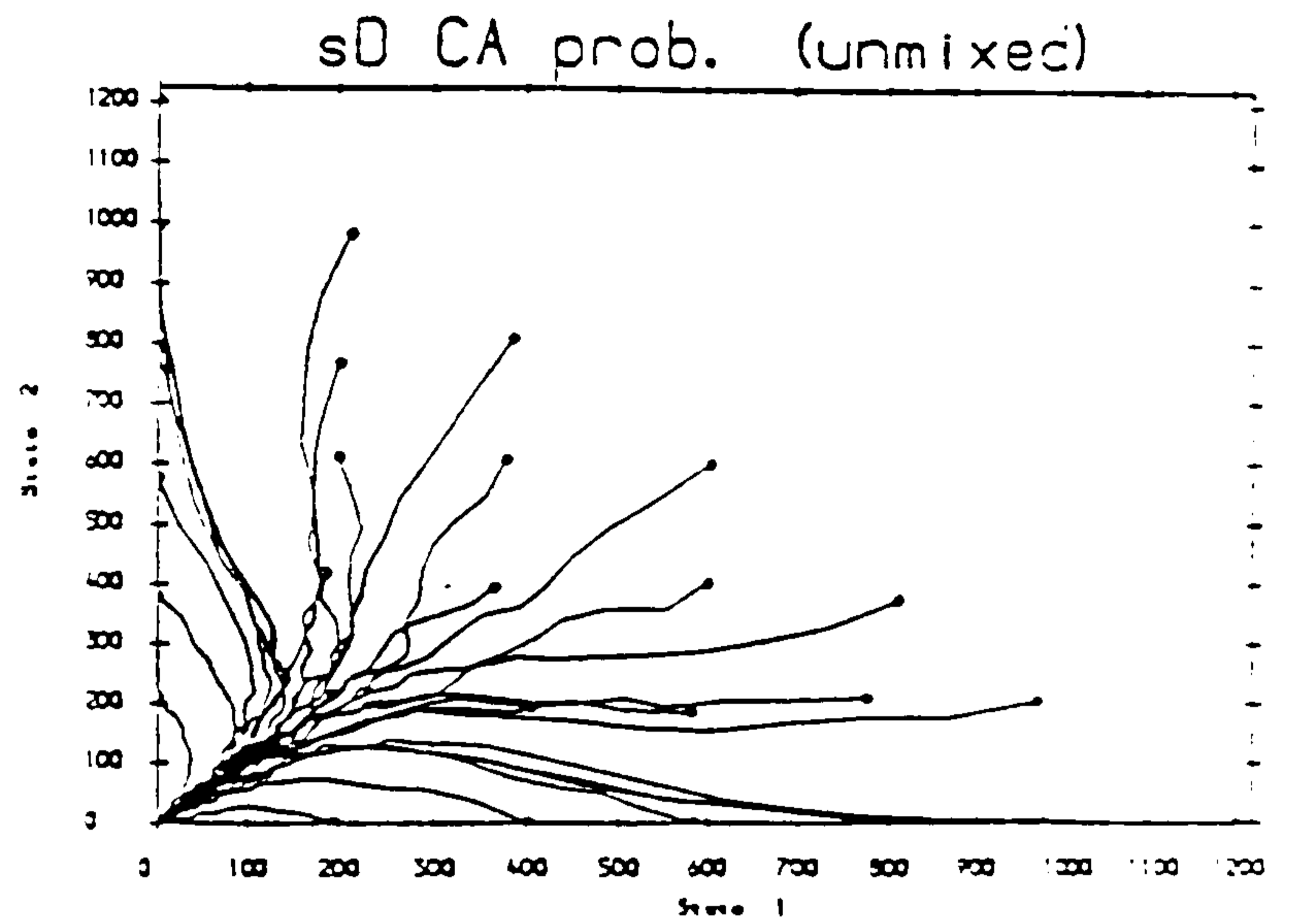
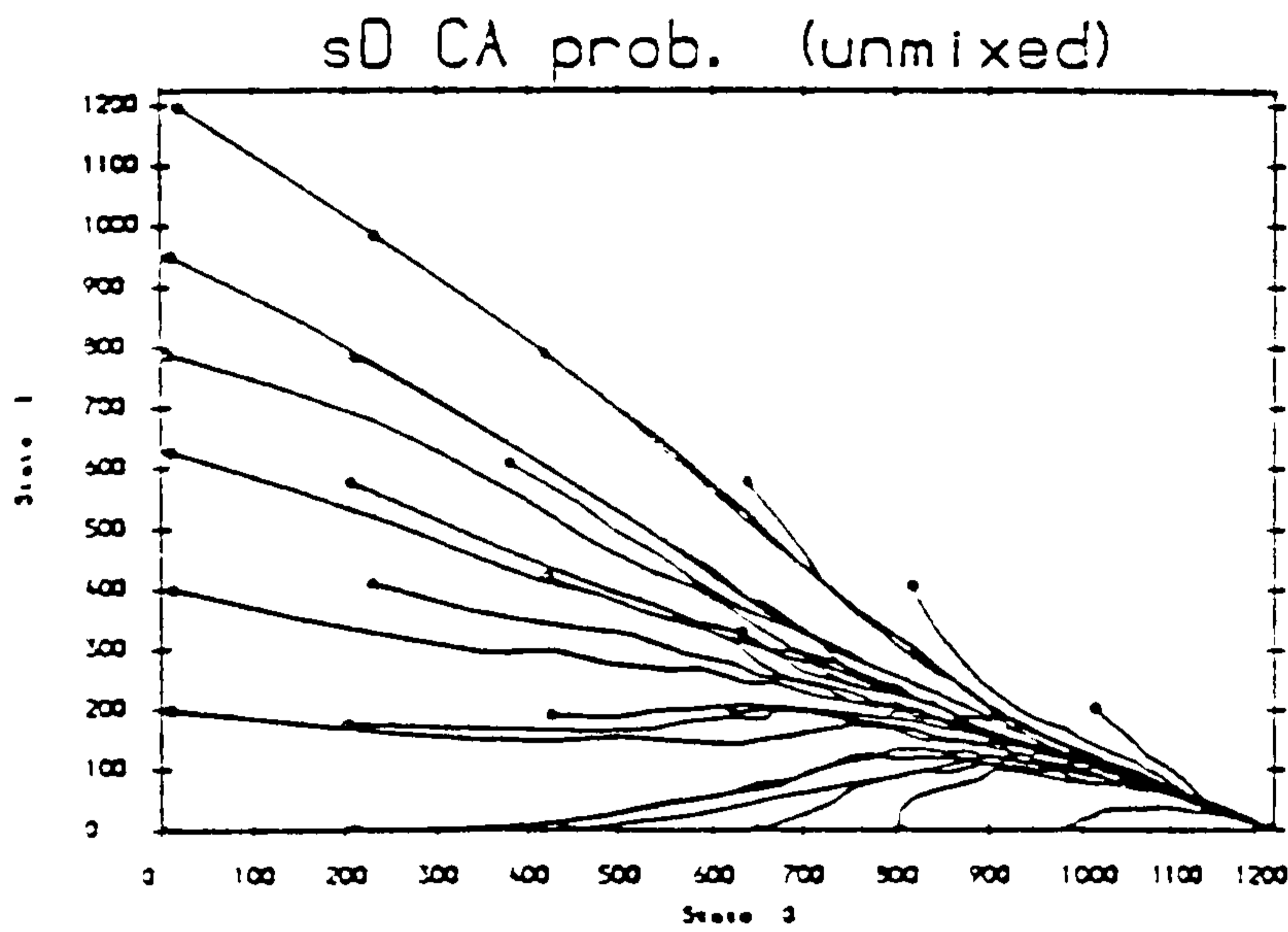


Figure 4.16: Class 4 phase diagrams.  
(d) System D: Probabilistic.





The first point to note about this group of systems is that although system A is definitely Class 4 and we have only made very simple alterations to obtain systems B, C and D these latter three systems are not full Class 4 but are partially Class 1 or 2.

System A when unmixed shows typical LIFE behaviour. Initial points outside the region of state densities which supports the universal computation behaviour rapidly evolve towards this region. Once inside this region the trajectories have a complex “random walk” behaviour which is the result of the periodic increase and decrease of active cells as the typical configurations of LIFE interact. When the lattice is mixed as it evolves we see immediately that the “random walk” property of the trajectories has disappeared. This is because the typical LIFE configurations are destroyed before they can migrate across the lattice and interact. We also notice that two new limiting regions have been created. These new limit regions reflect the fact that when one of the active state populations becomes very small it will die out completely because when the lattice is mixed no neighbourhood will contain more than one cell of that state and at least two cells in a neighbourhood are required for the population to grow. In the unmixed system, as the population size decreases it reaches a point where the probability of a small group of cells being able to grow significantly is very high and so the population never completely dies out. Thus one observes that although the unmixed system initially tends towards these new limiting regions it eventually moves away. The dynamical system behaviour is very similar to that for the mixed system except that it is symmetrical with respect to the two active state. The probabilistic systems are similar except that the mixed system does not exhibit the new limiting regions. This is because the probabilistic evolution destroys the larger populations ability to grow so that it cannot stabilise at the value of the new limiting regions. The probabilistic DS allows more trajectories to reach these new limits because the evolutionary steps on the phase diagram are not large enough to take them

passed these limits to the domain of attraction of the original limit point (in which both active populations are extinct).

The first point to note about system B is that it does not exhibit the characteristic “random walk” of its trajectories which is typical of a Class 4 CA. Indeed all trajectories tend to the limit point at which both active populations are extinct. The mixed system does not exhibit the new limiting regions which system A did because the two active populations are no longer independent as they were for system A. Instead the rate of growth of each depends on the size of the other population, this leads to a new limiting region at which the two populations are equal. The effect of this cross-coupling and the resultant limit point are most obvious on the probabilistic phase diagrams, although only the DS exhibits the limit point, as for system A.

System C exhibits distinct oscillations which are simply due to the rules, which in system A produced the survival of the active states, in this system cause a cell to switch from one active state to the other. The effect of mixing is to allow the active populations to reach their true equilibrium values of approximately 450. An interesting point is that the probabilistic systems appear very different, but this is only because the probabilistic rules mean that all the active cells don't switch at the same time so the size of the oscillations are considerably reduced. It will be noted that because of this the probabilistic behaviour is very similar to that for system B.

Finally system D behaves in all respects in an almost identical manner to system B. If the state transition diagrams for these two systems are compared it will be seen that each has a single pair of cross coupling transitions. These have similar effects on the self supporting LIFE behaviour, which is the disruption of the two cooperating processes of creation of active cells and the maintenance of active cells when the relevant neighbourhood



conditions occur. This in turn causes the active populations to rapidly die out. Comparing this with system C we see that here we have both cross couplings and this leaves the cooperating processes partially intact. Consider cells in state  $s_1$ , even though cells in this state which would normally remain unaltered now change to state  $s_2$ , cells in state  $s_0$  which would normally change to state  $s_1$ , based on the distribution of  $s_1$  cells in their neighbourhood, now also change to  $s_2$ . However, the new neighbourhood distribution of a given cell will not be identical to that which system A would have produced and so the LIFE type behaviour is lost. Instead we obtain the highly coupled behaviour which system C exhibits.

In general we can see that the DS system models well the behaviour of these CA systems, only failing to capture the universal computation effects of true class 4 CA. If the CA lattice is mixed, destroying the universal computation behaviour then the agreement is excellent. With the probabilistic systems we see that the DS model can support limit points which do not occur for the CA. However, we can see that the CA are affected by these “hidden” limit points and thus the DS can give us an insight into the underlying reasons for certain characteristics of the CA behaviour.

We now need to consider class 3 type systems which have limiting behaviour analogous to the strange attractors of dynamical systems theory. We will consider four systems of this type, firstly system K which is derived from system A and has a very simple rule set. The last three systems E, F and G are similar to system K but with transitions added between states  $s_1$  and  $s_2$ . Their state transition diagrams are identical and they only differ in the exact conditions of the neighbourhood which cause the various transitions. They were derived from examples of class 3 CA given by Packard and Wolfram[46]. These systems are summarised in Figure 4.17, their state transition diagrams in Figure 4.18,



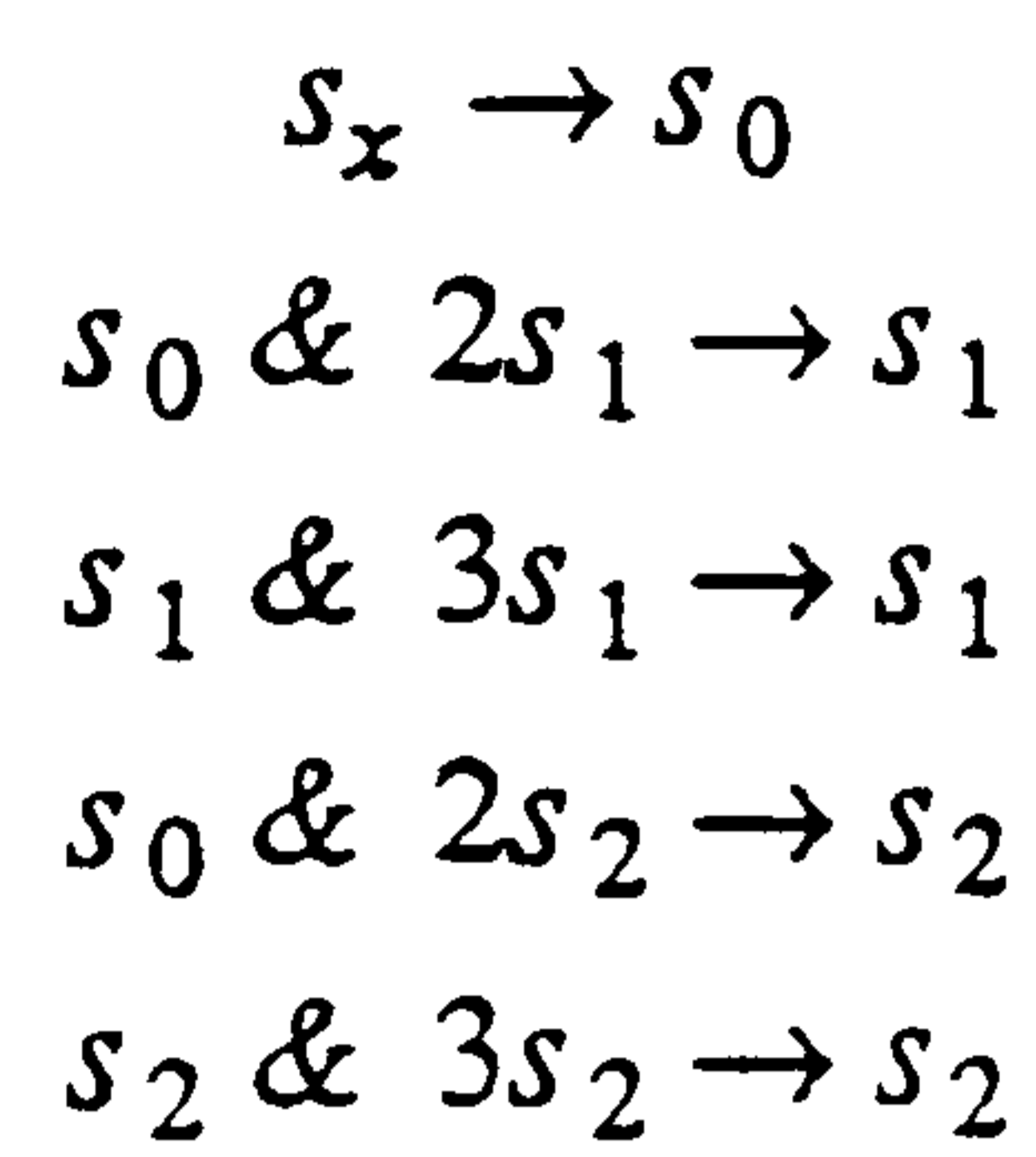
vector field diagrams in Figure 4.19 and phase diagrams in Figure 4.20.

The rule set for system K contains a potential conflict in that state  $s_0$  may change to state  $s_1$  or  $s_2$  depending on the state of the neighbourhood. However it is possible for both of the neighbourhood conditions for these transitions to be met simultaneously. In this case the transition to state  $s_2$  is chosen. This causes state  $s_2$  to be treated preferentially and results this state surviving while  $s_1$  dies out completely. The DS does not contain this conflict however and can treat both states equally, this results in a limiting state in which the populations of states  $s_1$  and  $s_2$  are equal. This problem could be dealt with in two ways. Firstly, when a rule conflict occurs the rule to apply could be chosen randomly with each having equal probability. Alternatively, the possible rule conflicts could be modelled within the DS by calculating the probability of a rule conflict and adjusting the relevant transition probabilities based on this. System K is therefore a straight competition between states  $s_1$  and  $s_2$  for “empty” cells. The probabilistic evolutions do not have the bias which is evident in the normal phase diagrams.

The state transition diagrams for systems E, F and G have very distinct two-transition loops. We would therefore expect, where the two transitions have similar probabilities and the probabilities are relatively high, for period two oscillations to appear on the phase diagram. This indeed occurs for system E, but for systems F and G the transition probabilities are too low for oscillations to be sustained and the systems settle to limit points. As with system K, these three systems all contain conflicts in the rules for creation of the active states from the quiescent state. However, systems E and F contain two possible conflicts one of which was resolved in favour of state  $s_1$ , while the other was resolved in favour of state  $s_2$ . This tends to equalise the effect of the conflicts and so the phase diagrams do not show noticeable bias. System G however has only one conflict

Figure 4.17: Class 3 systems summary (a).

**System K**



$$\dot{n}_0 = (1 - P_{1,1}(3, n_1))n_1 + (1 - P_{2,2}(3, n_2))n_2 - (P_{0,1}(2, n_1) + P_{0,2}(2, n_2))n_0$$

$$\dot{n}_1 = P_{0,1}(2, n_1)n_0 - (1 - P_{1,1}(3, n_1))n_1$$

$$\dot{n}_2 = P_{0,2}(2, n_2)n_0 - (1 - P_{2,2}(3, n_2))n_2$$

Figure 4.17: Class 3 systems summary (b).

**System E**

$$s_x \rightarrow s_0$$

$$s_1 \& 0s_1 \rightarrow s_1$$

$$s_2 \& 0s_2 \rightarrow s_2$$

$$s_x \& 1s_2 \rightarrow s_2$$

$$s_x \& 1s_1 \rightarrow s_1$$

$$s_1 \& 2s_1 \rightarrow s_1$$

$$s_2 \& 2s_2 \rightarrow s_2$$

$$s_x \& 3s_2 \rightarrow s_2$$

$$s_x \& 3s_1 \rightarrow s_1$$

$$s_1 \& 4s_1 \rightarrow s_1$$

$$s_2 \& 4s_2 \rightarrow s_2$$

$$(1 - P_{1,1}(0, n_1) - P_{1,1}(1, n_1) - P_{1,1}(2, n_1) - P_{1,1}(3, n_1)$$

$$- P_{1,1}(4, n_1) - P_{1,2}(1, n_2) - P_{1,2}(3, n_2))n_1$$

$$\dot{n}_0 = (1 - P_{2,2}(0, n_2) - P_{2,2}(1, n_2) - P_{2,2}(2, n_2) - P_{2,2}(3, n_2)$$

$$- P_{2,2}(4, n_2) - P_{2,1}(1, n_1) - P_{2,1}(3, n_1))n_2$$

$$- P_{0,1}(1, n_1)n_0 - P_{0,1}(3, n_1)n_0 - P_{0,2}(1, n_2)n_0 - P_{0,2}(3, n_2)n_0$$

$$P_{0,1}(1, n_1)n_0 + P_{0,1}(3, n_1)n_0 + P_{2,1}(1, n_1)n_2 + P_{2,1}(3, n_1)n_2$$

$$\dot{n}_1 = (1 - P_{1,1}(0, n_1) - P_{1,1}(1, n_1) - P_{1,1}(2, n_1) - P_{1,1}(3, n_1) - P_{1,1}(4, n_1))n_1$$

$$P_{0,2}(1, n_2)n_0 + P_{0,2}(3, n_2)n_0 + P_{1,2}(1, n_2)n_1 + P_{1,2}(3, n_2)n_1$$

$$\dot{n}_2 = (1 - P_{2,2}(0, n_2) - P_{2,2}(1, n_2) - P_{2,2}(2, n_2) - P_{2,2}(3, n_2) - P_{2,2}(4, n_2))n_2$$



Figure 4.17: Class 3 systems summary (c).

**System F**

$$s_x \rightarrow s_0$$

$$s_1 \& 0s_1 \rightarrow s_1$$

$$s_2 \& 0s_2 \rightarrow s_2$$

$$s_x \& 3s_1 \rightarrow s_1$$

$$s_x \& 3s_2 \rightarrow s_2$$

$$s_x \& 4s_2 \rightarrow s_2$$

$$s_x \& 4s_1 \rightarrow s_1$$

$$s_1 \& 5s_1 \rightarrow s_1$$

$$s_2 \& 5s_2 \rightarrow s_2$$

$$s_1 \& 6s_1 \rightarrow s_1$$

$$s_2 \& 6s_2 \rightarrow s_2$$

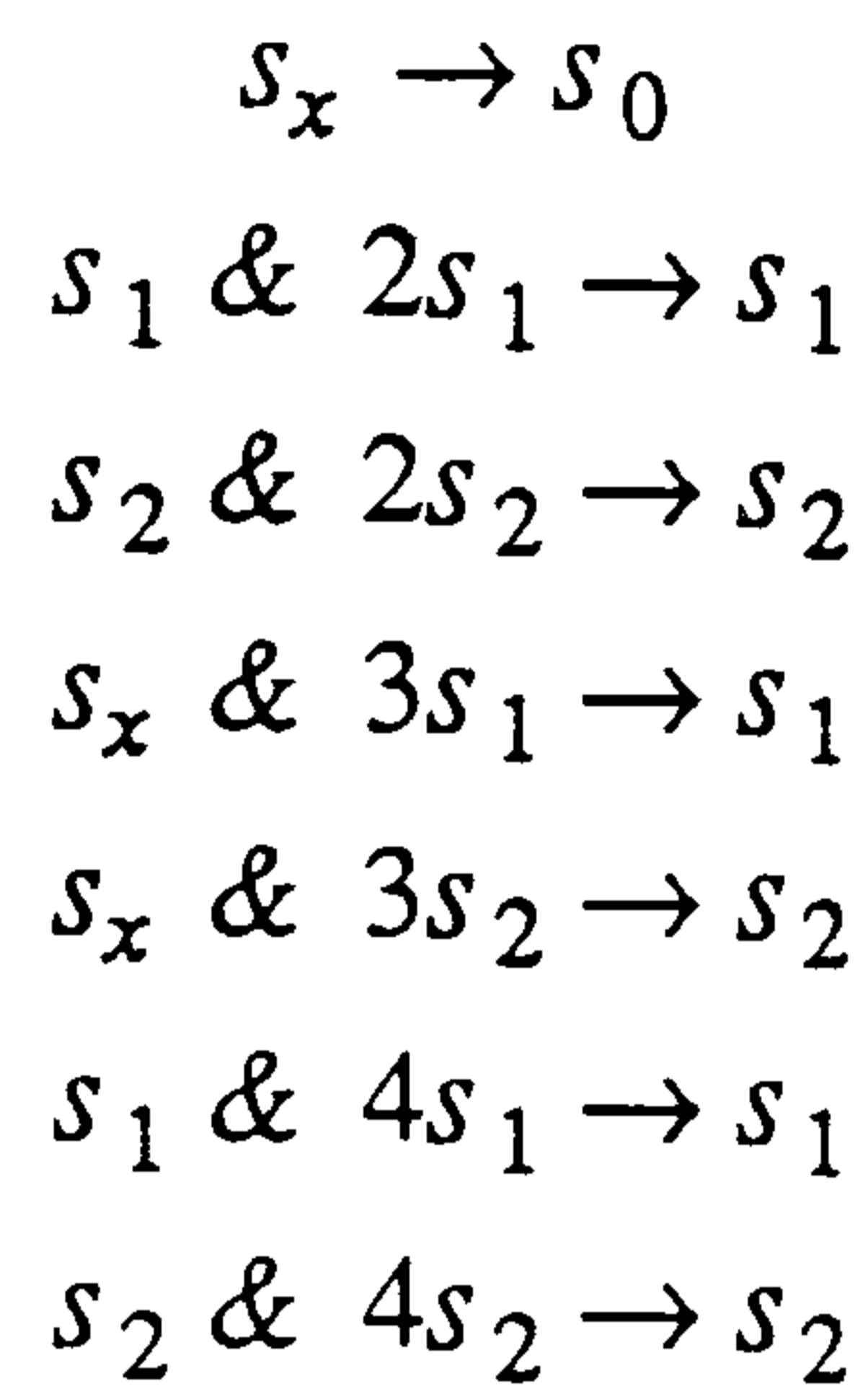
$$\begin{aligned} & (1 - P_{1,1}(0,n_1) - P_{1,1}(3,n_1) - P_{1,1}(4,n_1) - P_{1,1}(5,n_1) \\ & \quad - P_{1,1}(6,n_1) - P_{1,2}(3,n_2) - P_{1,2}(4,n_2))n_1 \\ \dot{n}_0 = & (1 - P_{2,2}(0,n_2) - P_{2,2}(3,n_2) - P_{2,2}(4,n_2) - P_{2,2}(5,n_2) \\ & \quad - P_{2,2}(6,n_2) - P_{2,1}(3,n_1) - P_{2,1}(4,n_1))n_2 \\ & - P_{0,1}(3,n_1)n_0 - P_{0,1}(4,n_1)n_0 - P_{0,2}(3,n_2)n_0 - P_{0,2}(4,n_2)n_0 \end{aligned}$$

$$\dot{n}_1 = \frac{P_{0,1}(3,n_1)n_0 + P_{0,1}(4,n_1)n_0 + P_{2,1}(3,n_1)n_2 + P_{2,1}(4,n_1)n_2}{(1 - P_{1,1}(0,n_1) - P_{1,1}(3,n_1) - P_{1,1}(4,n_1) - P_{1,1}(5,n_1) - P_{1,1}(6,n_1))n_1}$$

$$\dot{n}_2 = \frac{P_{0,2}(3,n_2)n_0 + P_{0,2}(4,n_2)n_0 + P_{1,2}(3,n_2)n_1 + P_{1,2}(4,n_2)n_1}{(1 - P_{2,2}(0,n_2) - P_{2,2}(3,n_2) - P_{2,2}(4,n_2) - P_{2,2}(5,n_2) - P_{2,2}(6,n_2))n_2}$$

Figure 4.17: Class 3 systems summary (d).

**System G**



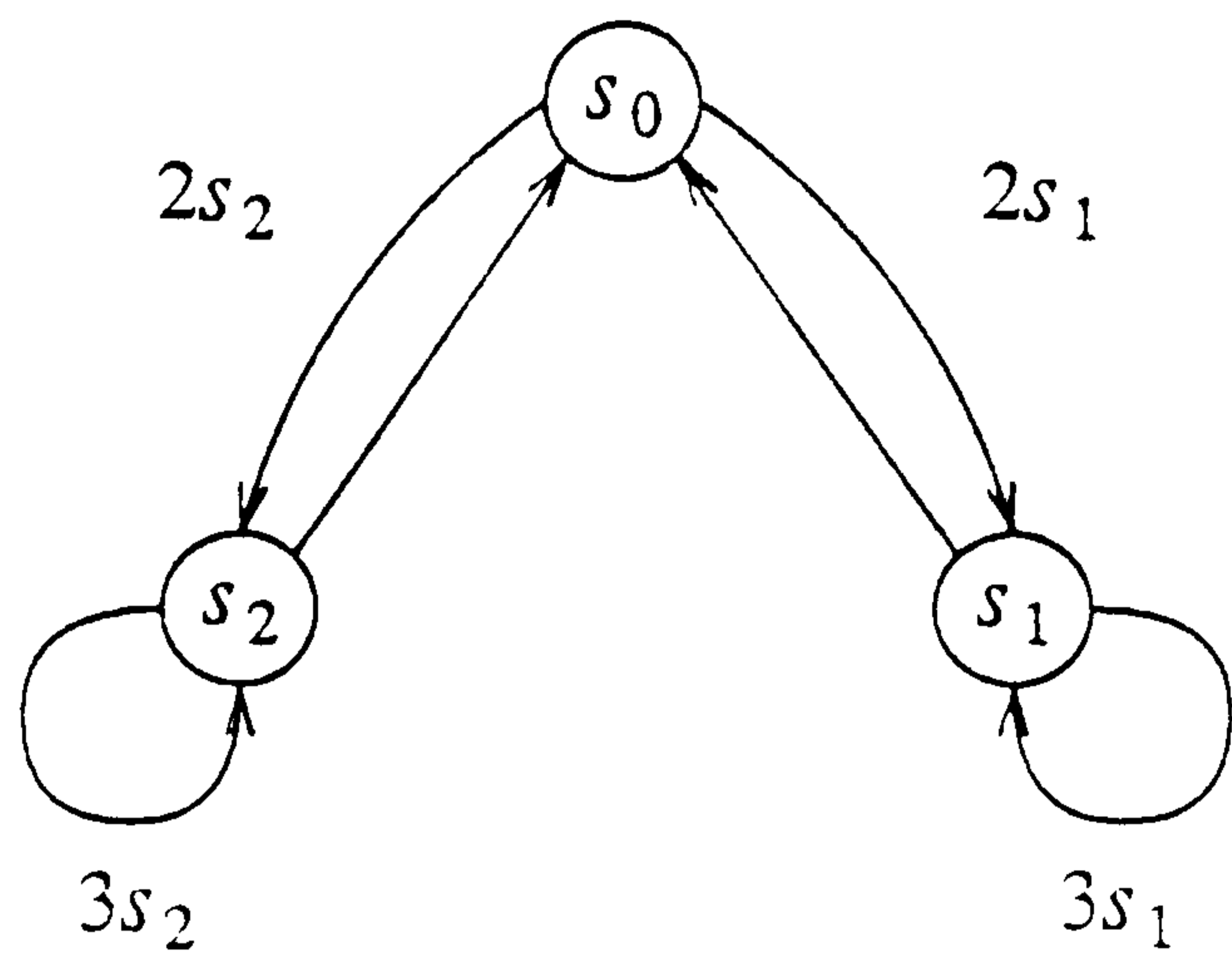
$$\begin{aligned}
 \dot{n}_0 &= (1 - P_{1,1}(2, n_1) - P_{1,1}(3, n_1) - P_{1,1}(4, n_1) - P_{1,2}(3, n_2))n_1 \\
 &\quad - P_{2,2}(2, n_2) - P_{2,2}(3, n_2) - P_{2,2}(4, n_2) - P_{2,1}(3, n_1))n_2 \\
 &\quad - P_{0,1}(3, n_1)n_0 - P_{0,2}(3, n_2)n_0
 \end{aligned}$$

$$\dot{n}_1 = \frac{P_{0,1}(3, n_1)n_0 + P_{2,1}(3, n_1)n_2}{(1 - P_{1,1}(2, n_1) - P_{1,1}(3, n_1) - P_{1,1}(4, n_1))n_1}$$

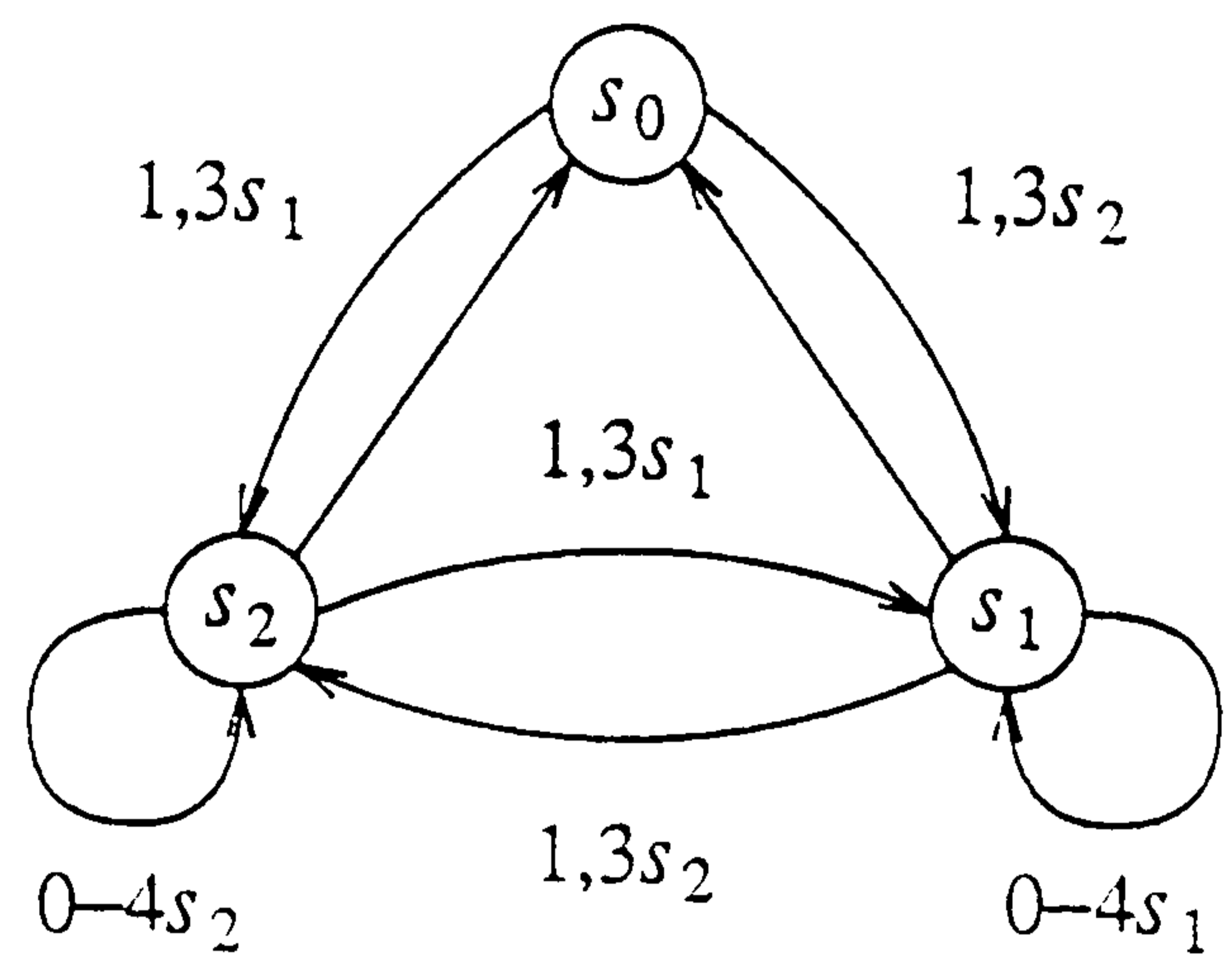
$$\dot{n}_2 = \frac{P_{0,2}(3, n_2)n_0 + P_{1,2}(3, n_2)n_1}{(1 - P_{2,2}(2, n_2) - P_{2,2}(3, n_2) - P_{2,2}(4, n_2))n_2}$$

Figure 4.18: Class 3 systems state transition diagrams.

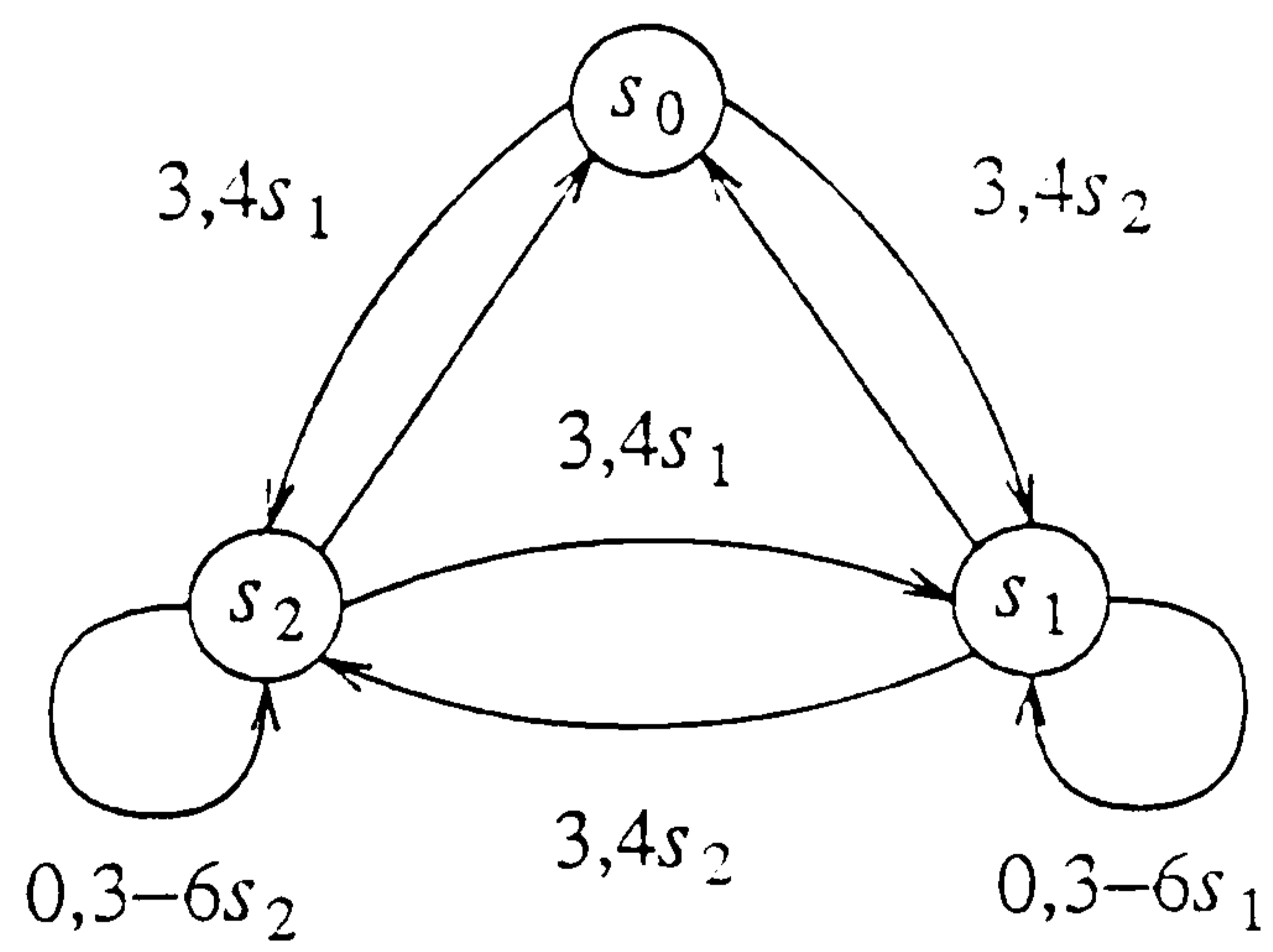
System K



System E



System F



System G

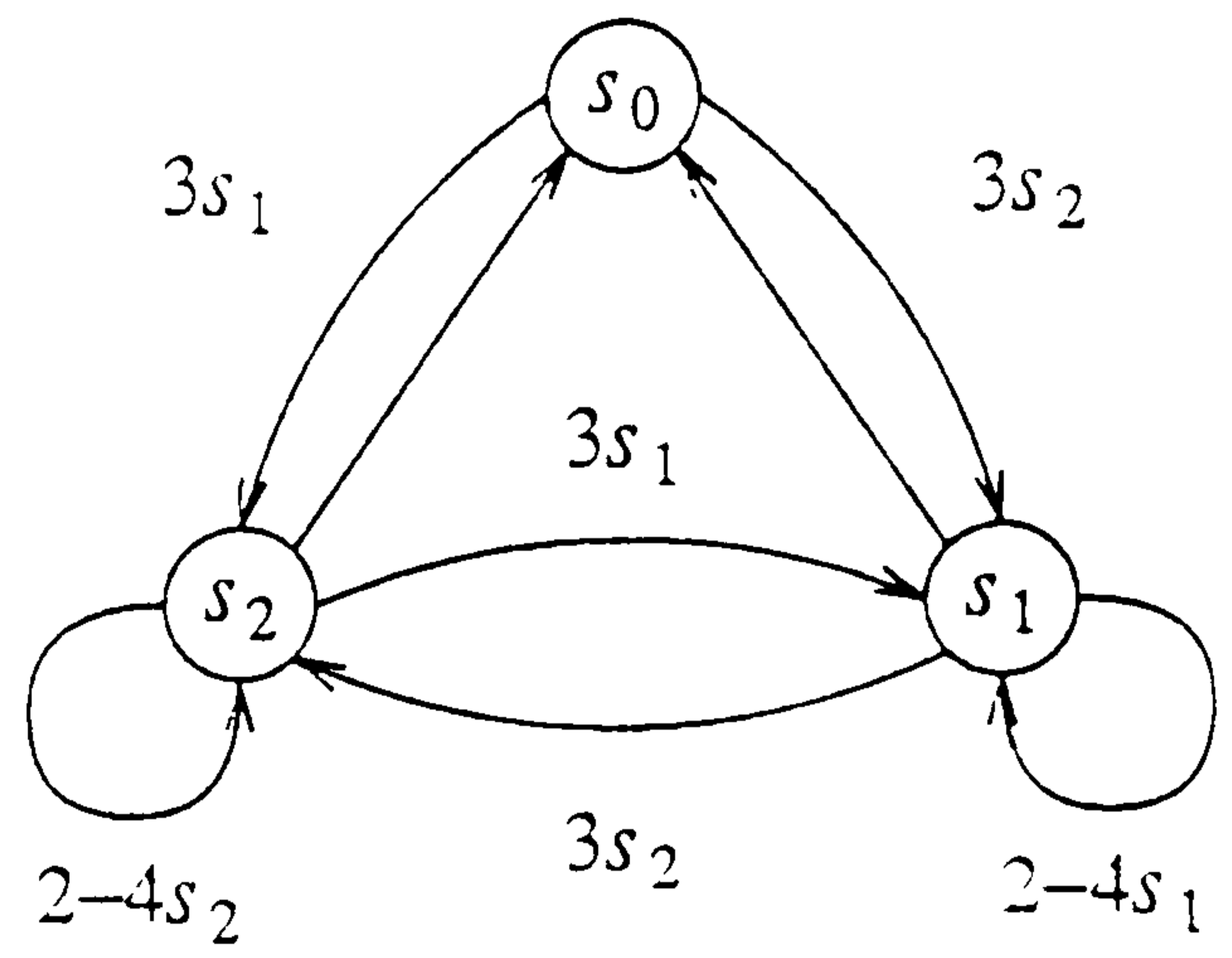




Figure 4.19: Class 3 vector field diagrams.  
(a) System K.

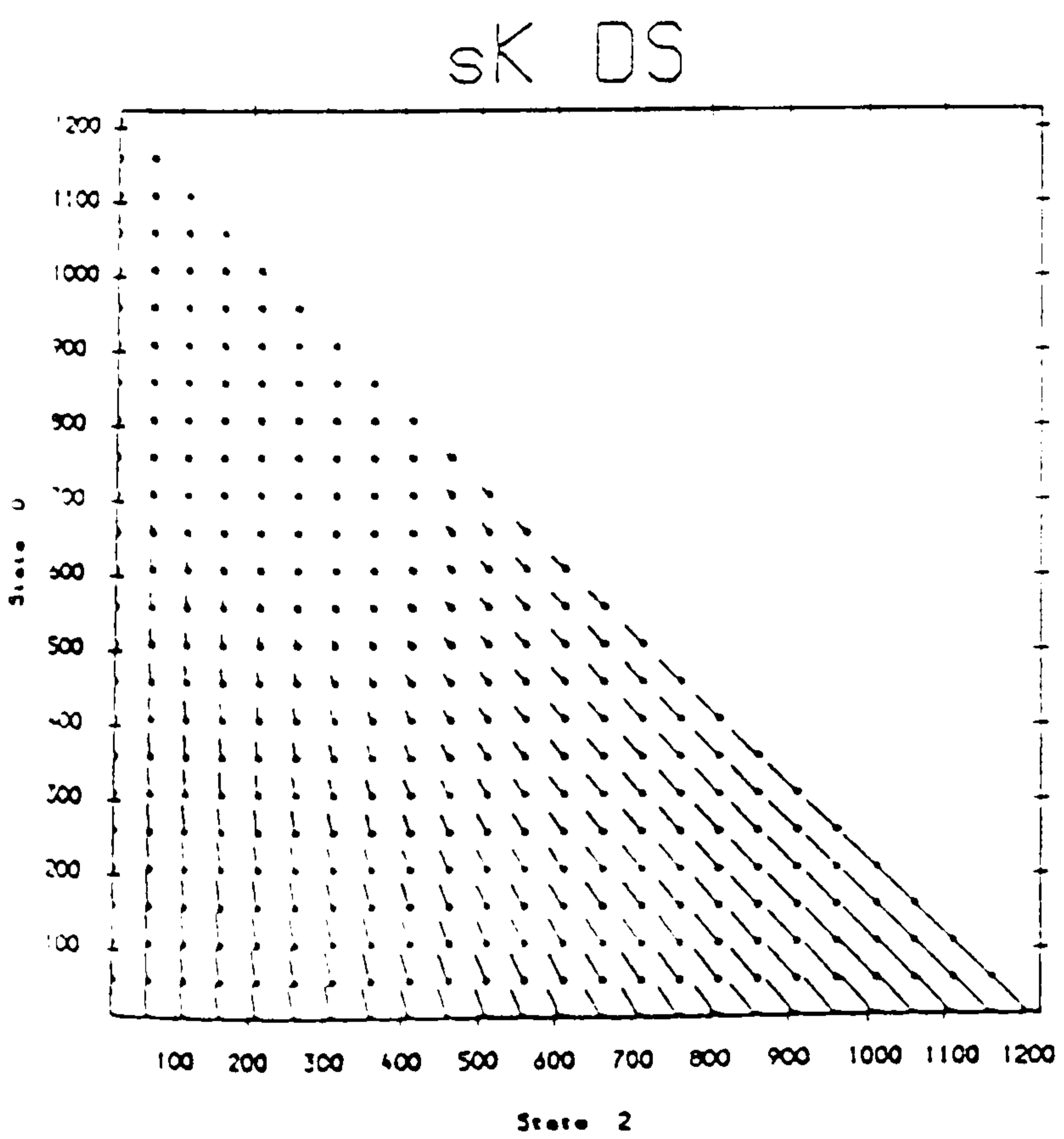
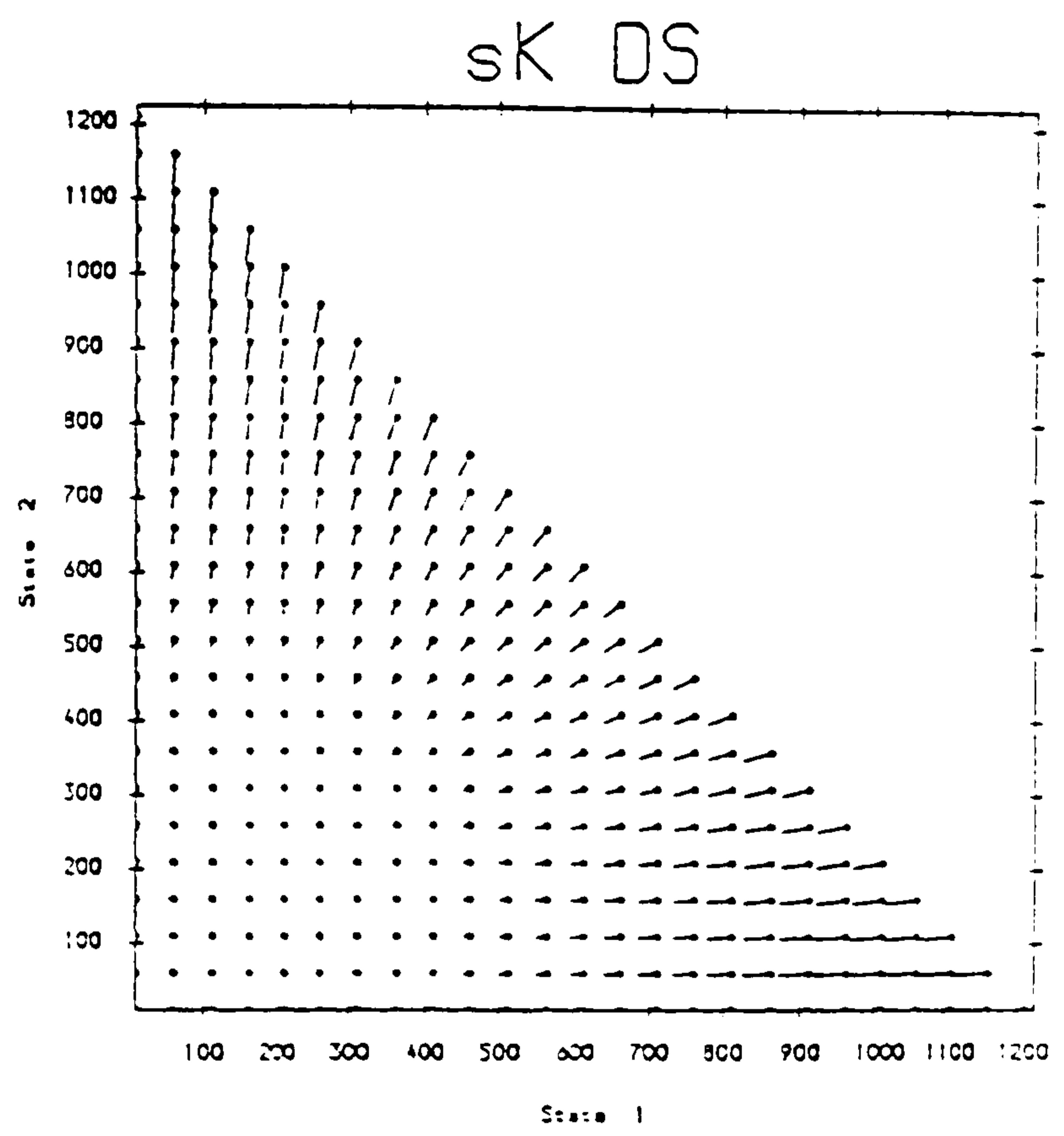
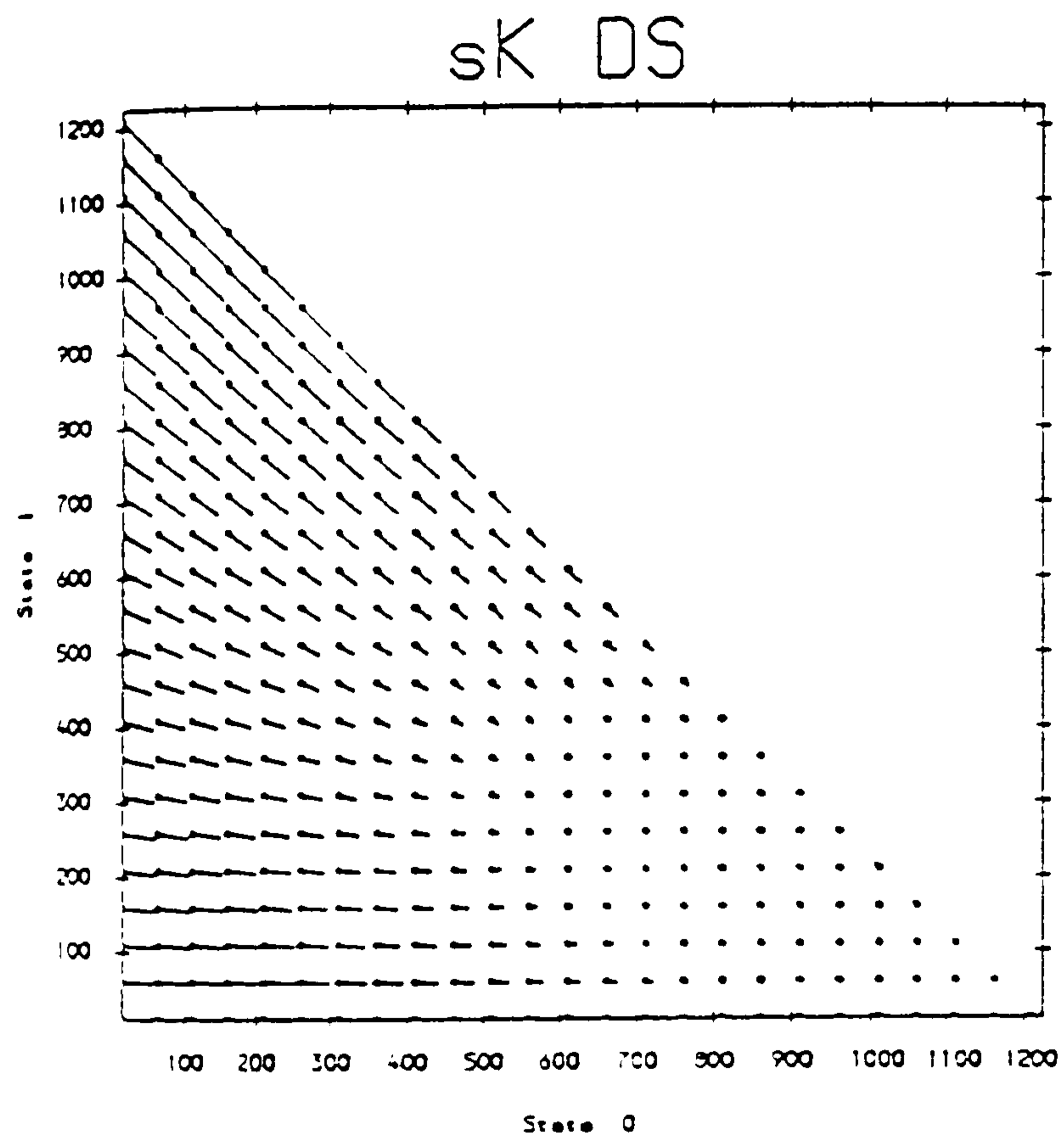


Figure 4.19: Class 3 vector field diagrams.  
(b) System E.

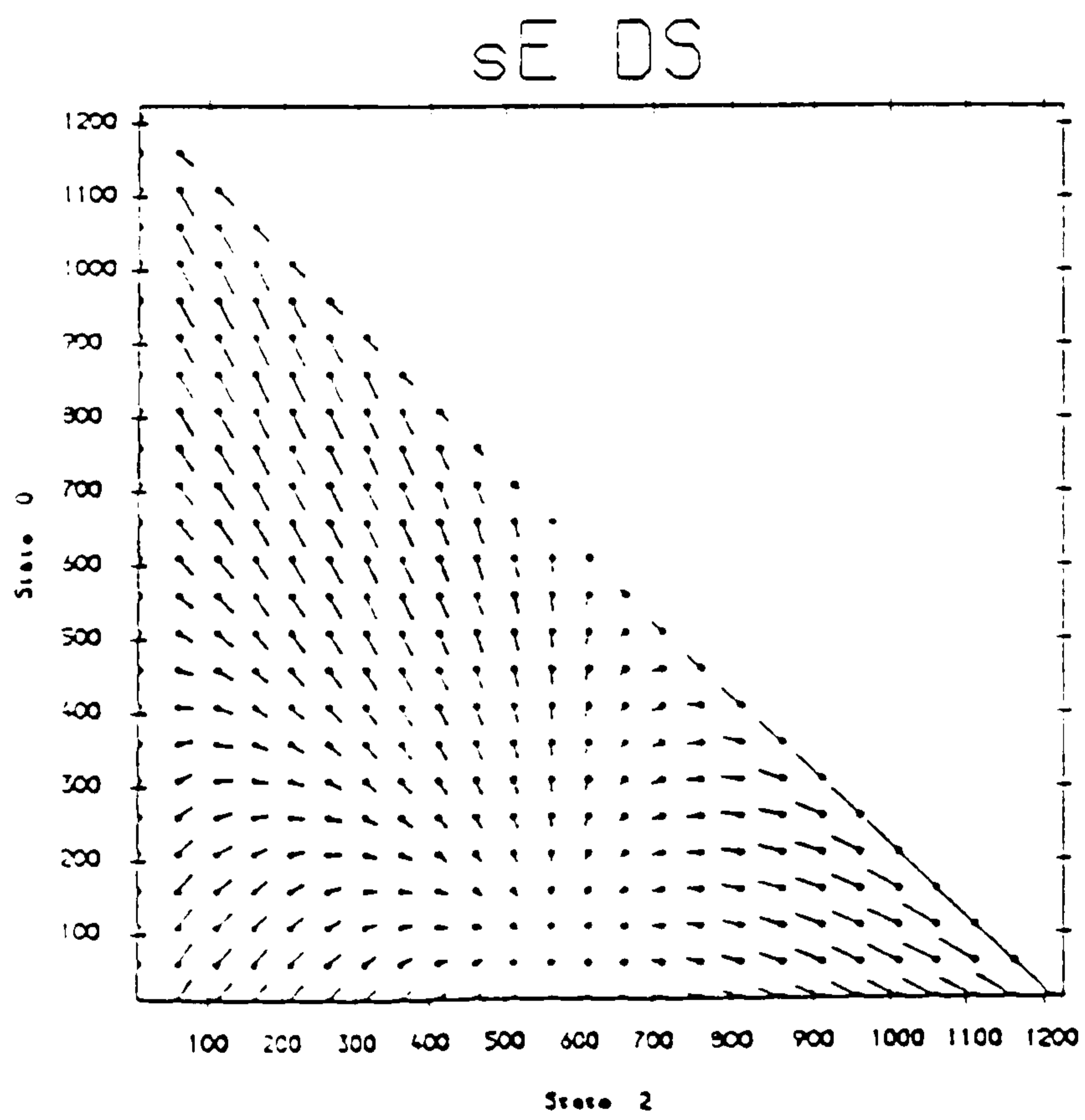
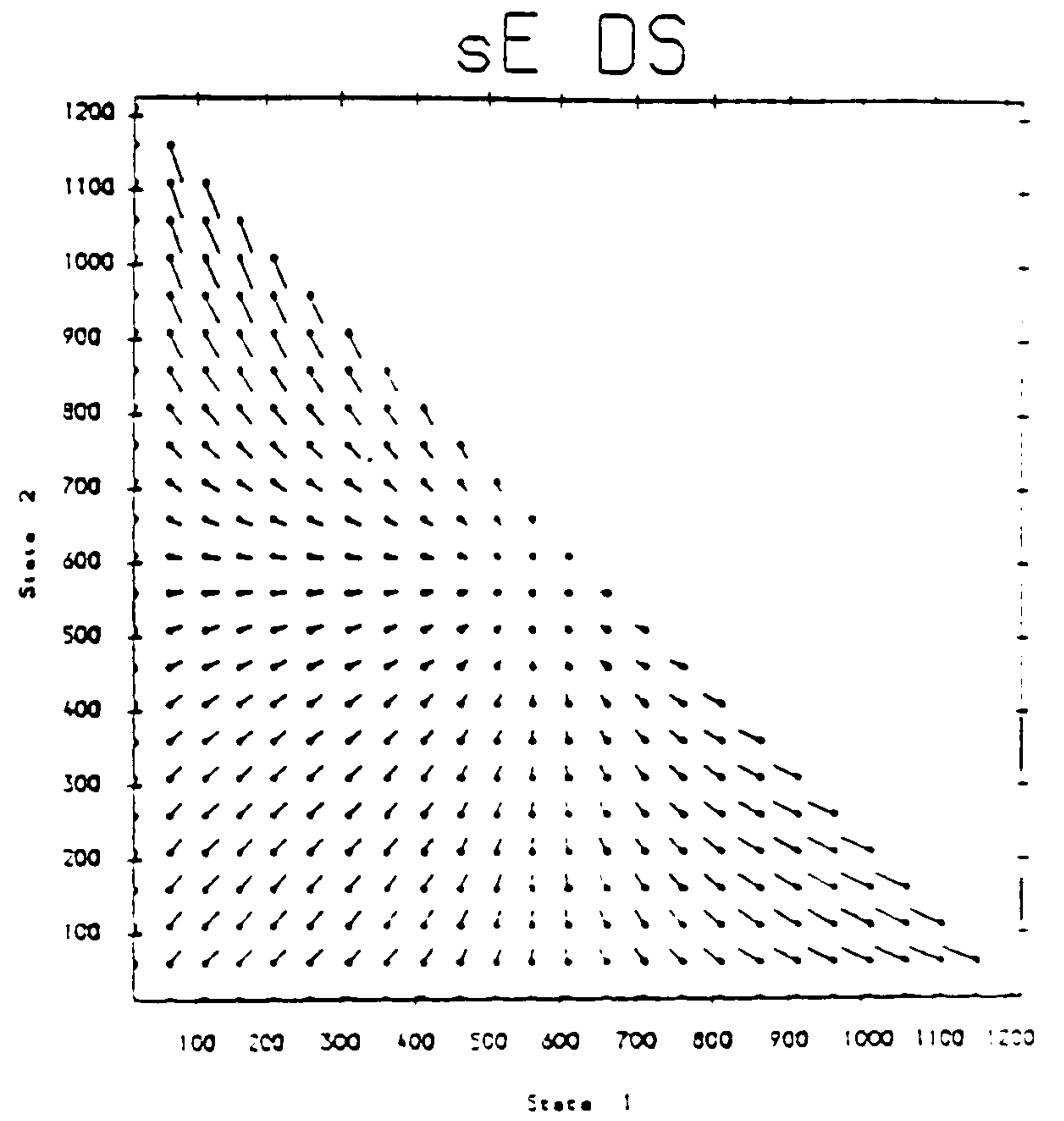
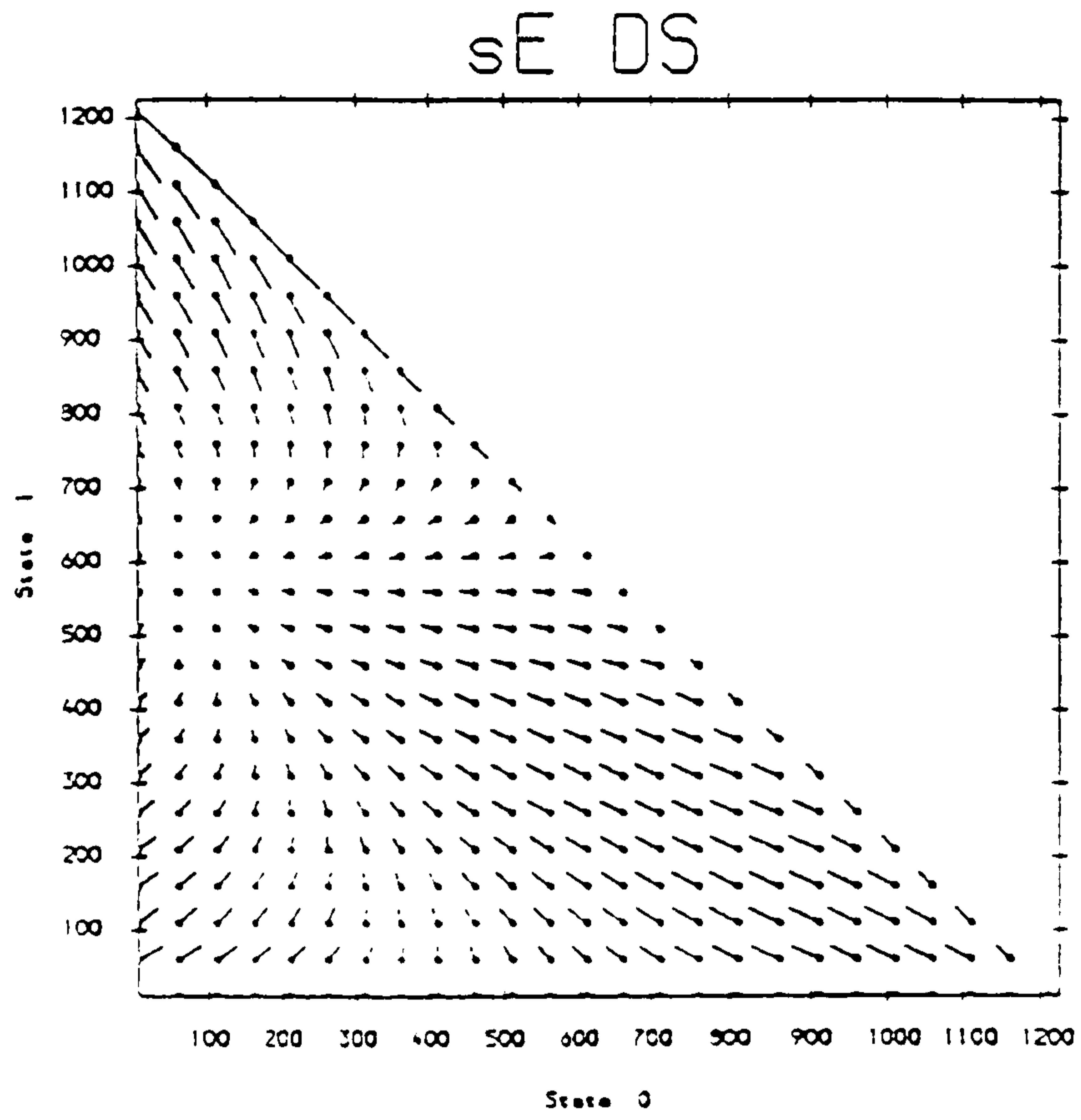


Figure 4.19: Class 3 vector field diagrams.  
(c) System F.

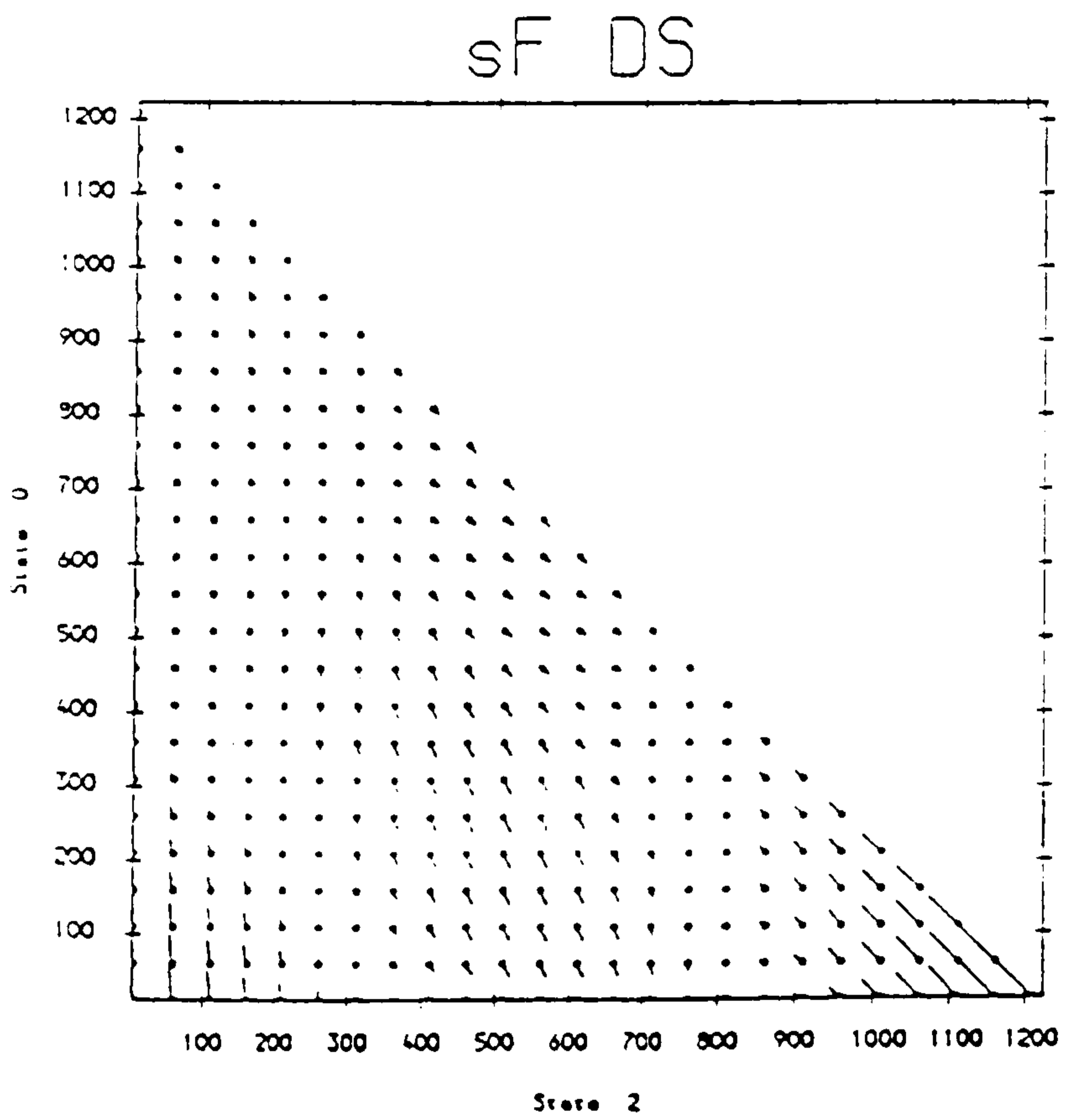
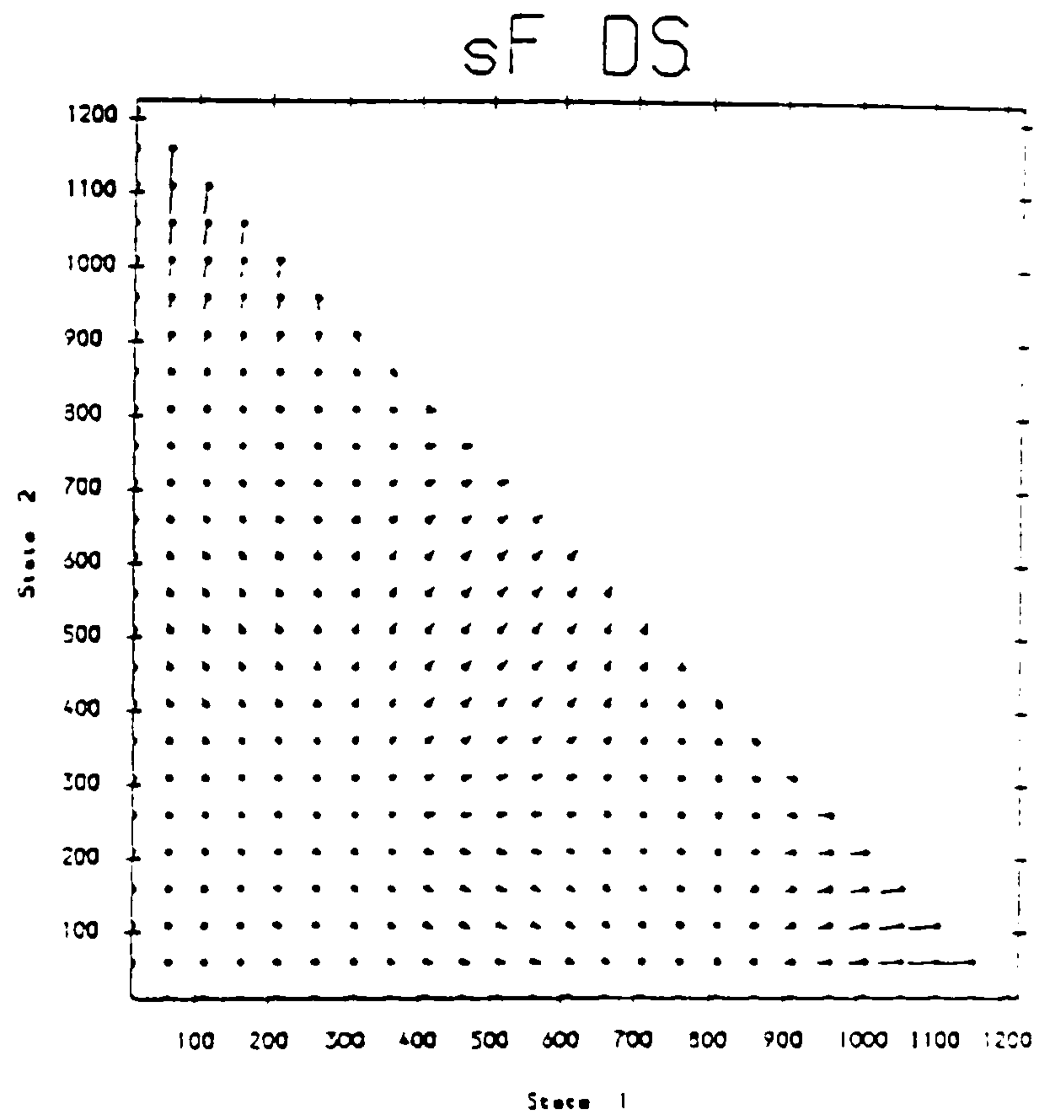
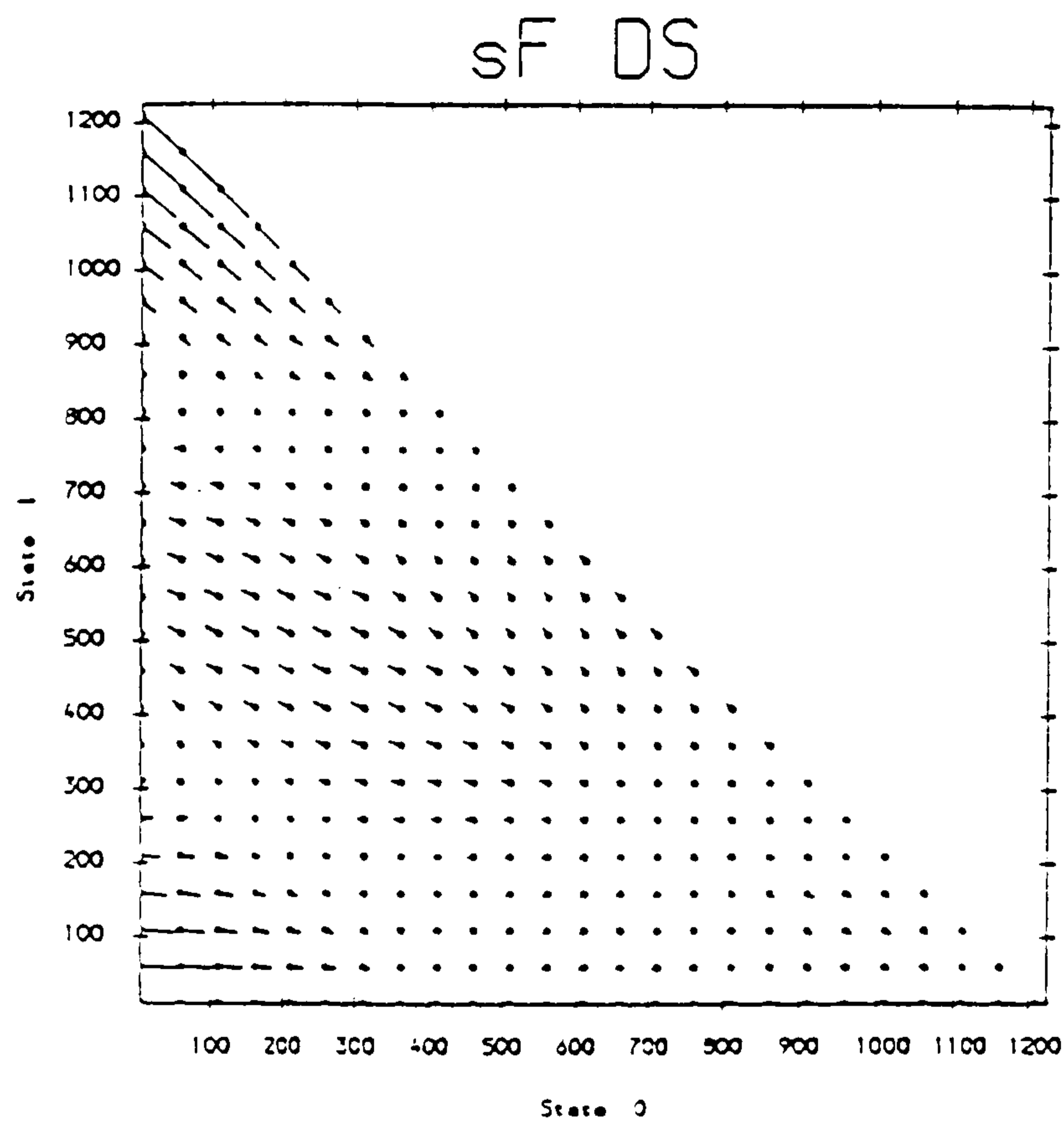




Figure 4.19: Class 3 vector field diagrams.  
(d) System G.

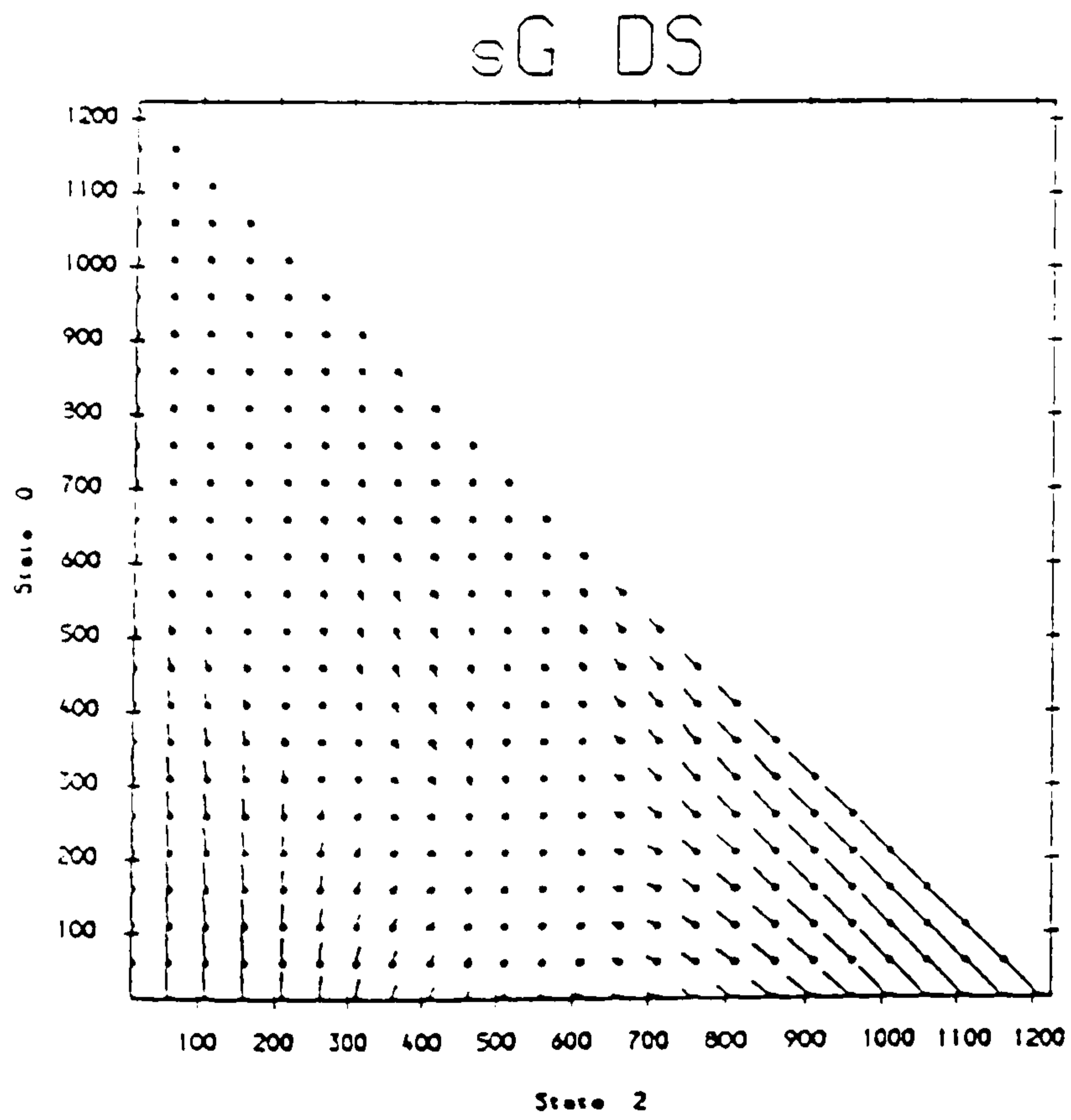
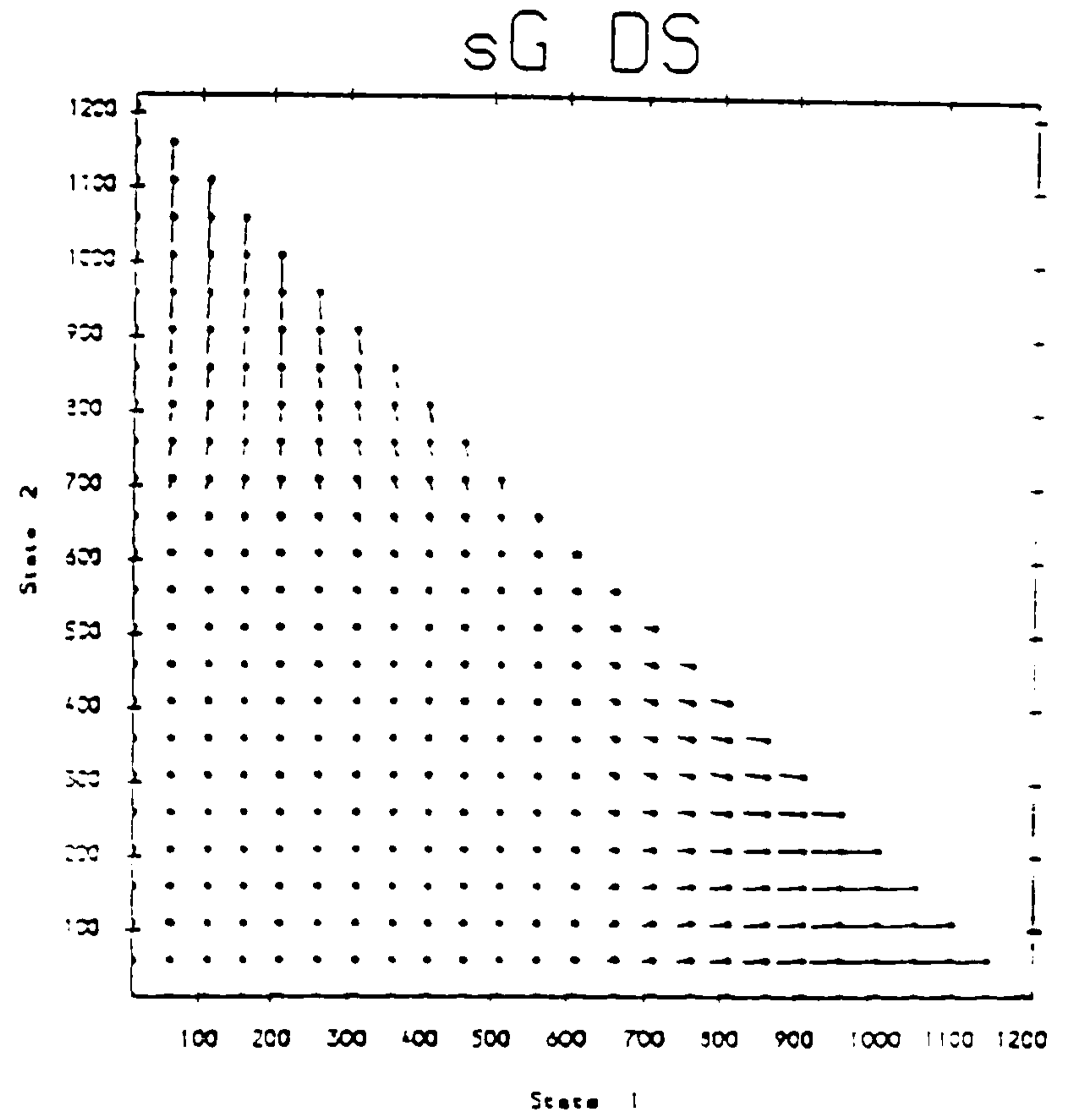
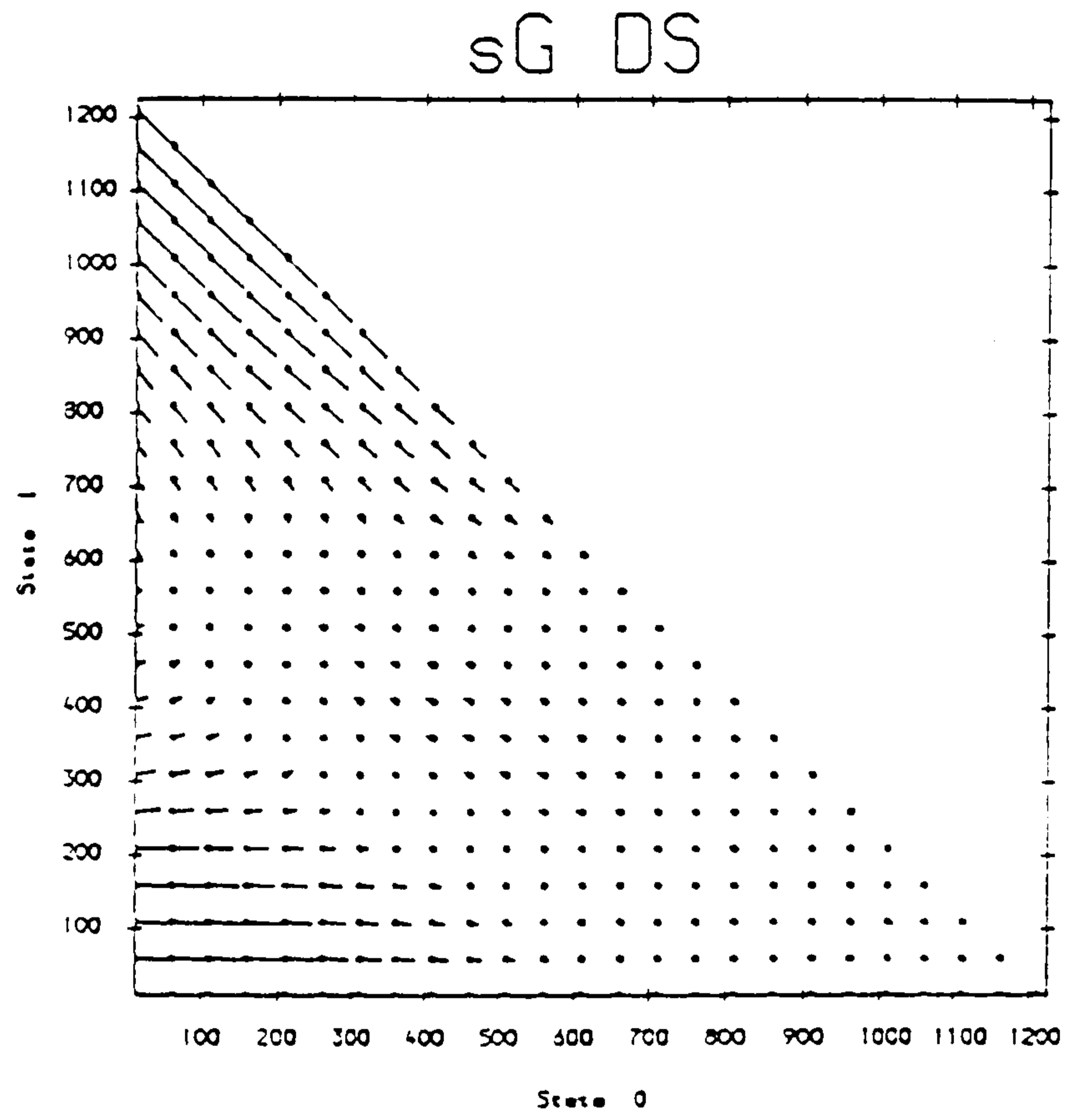


Figure 4.20: Class 3 phase diagrams.  
 (a) System K: Deterministic.

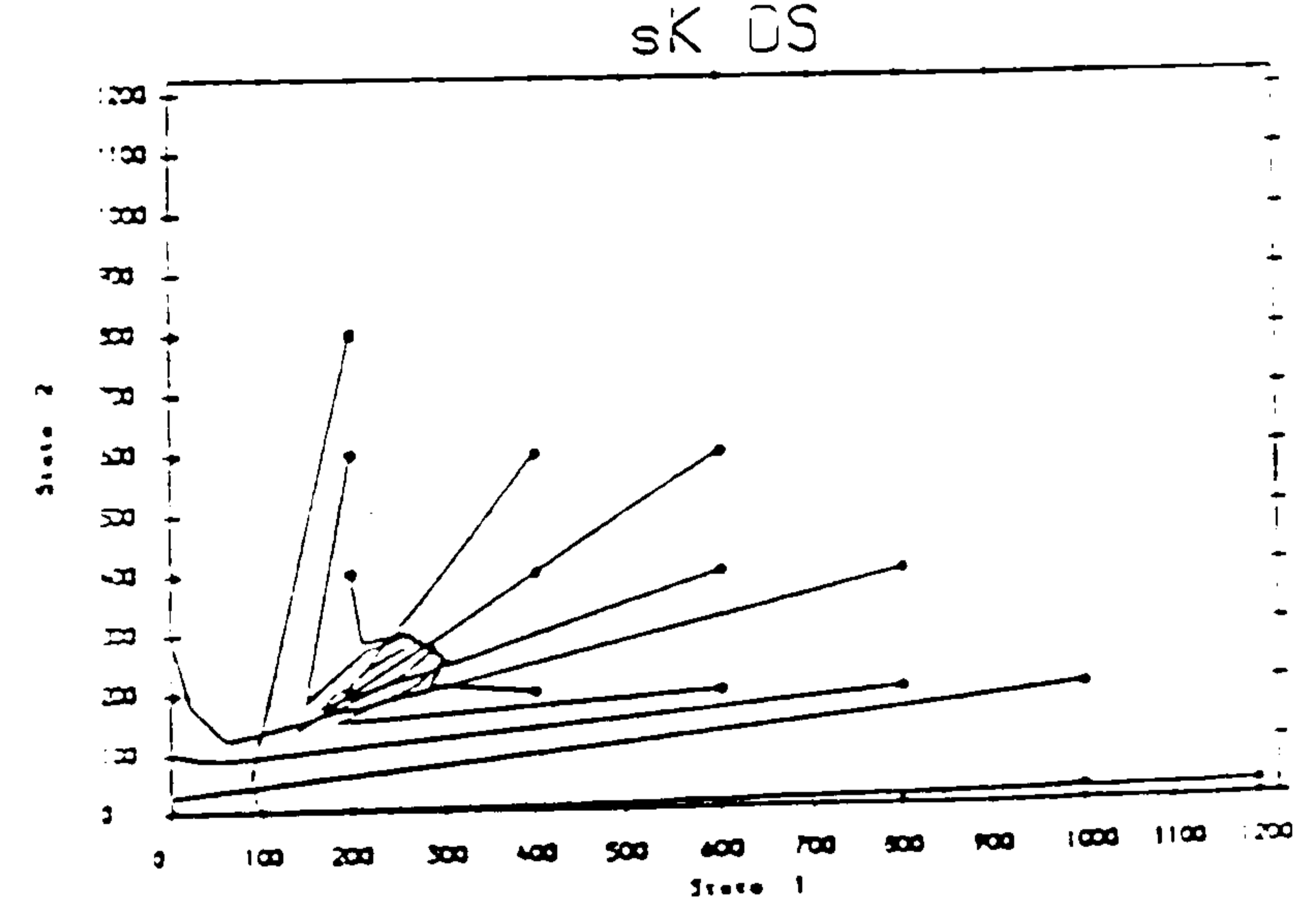
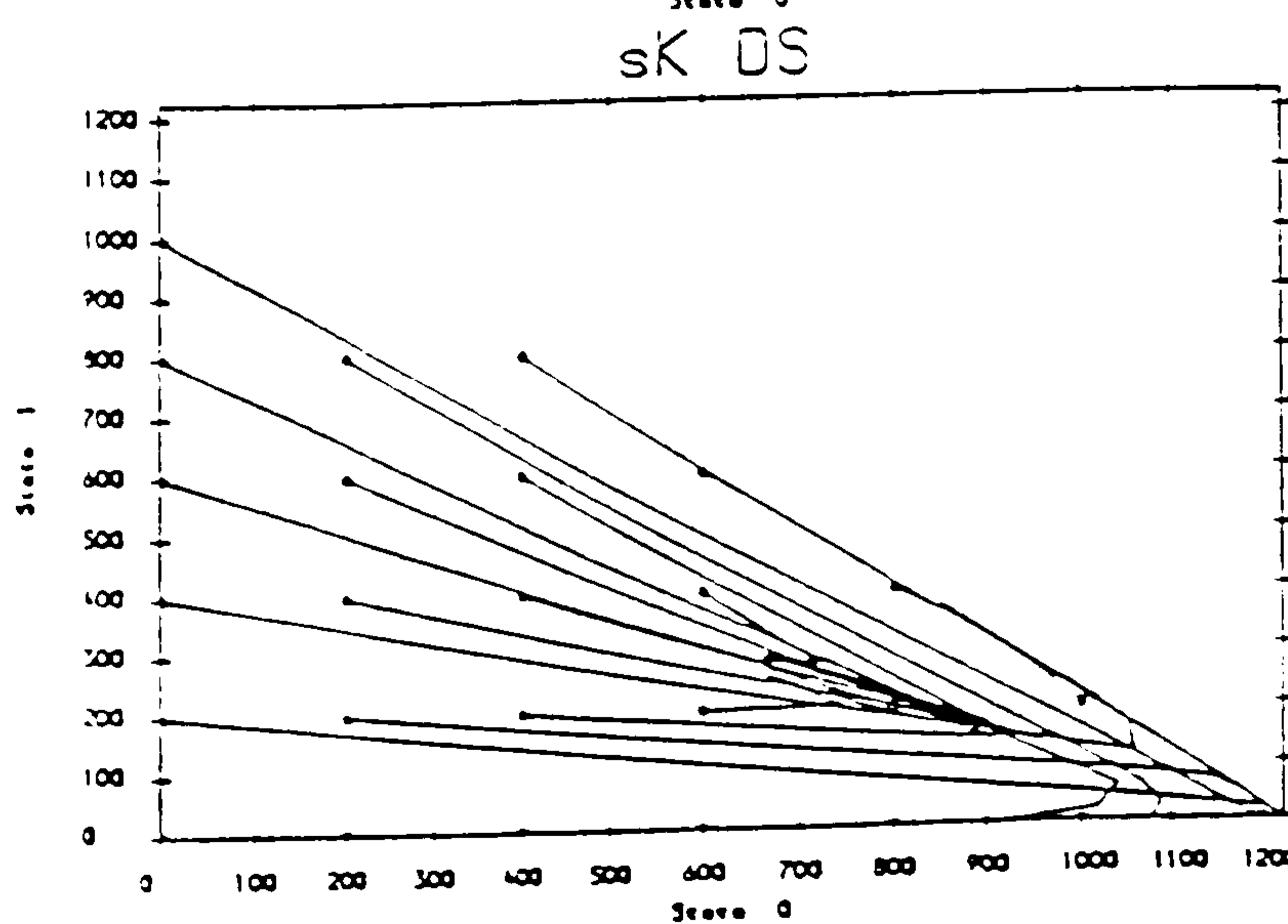
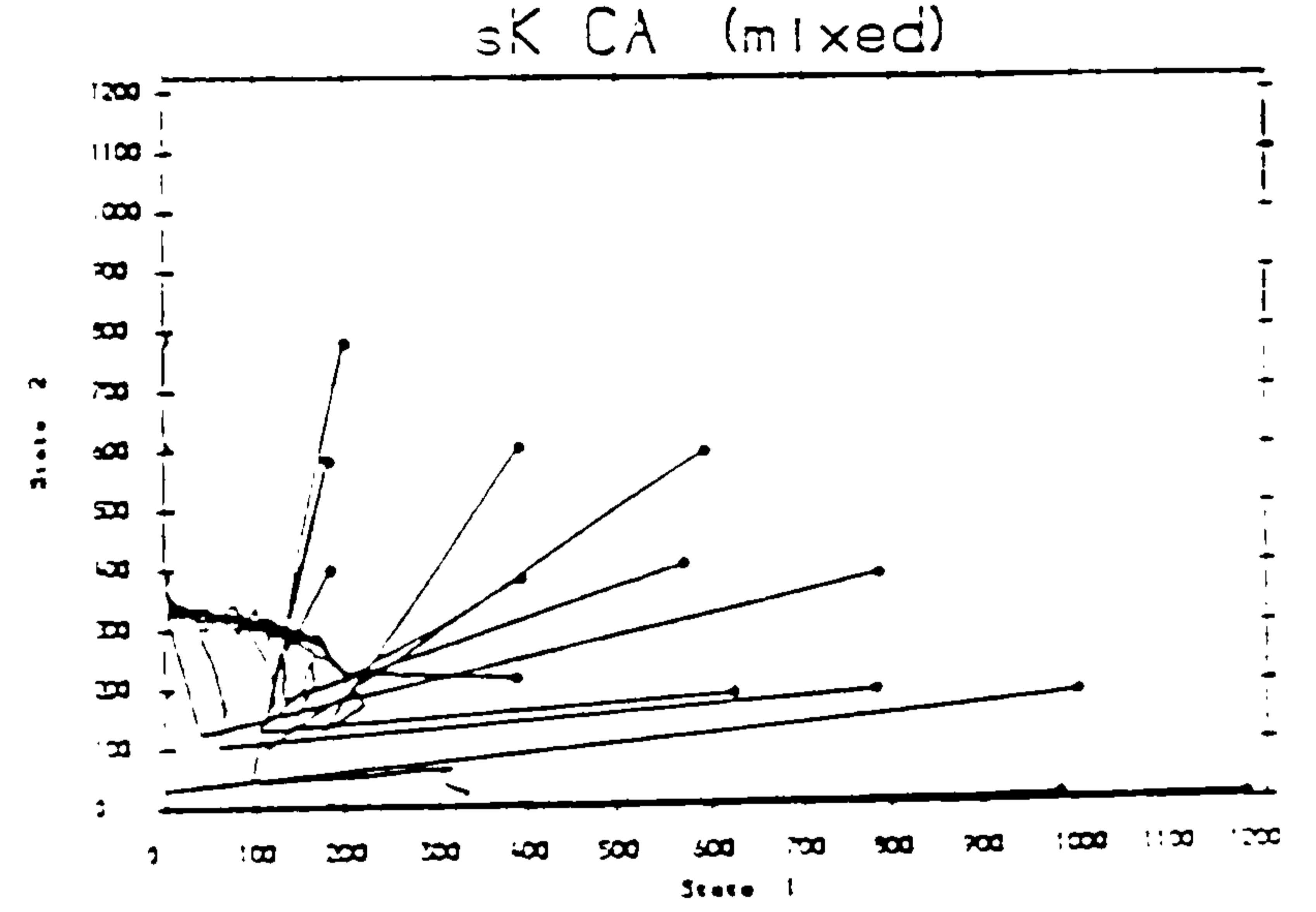
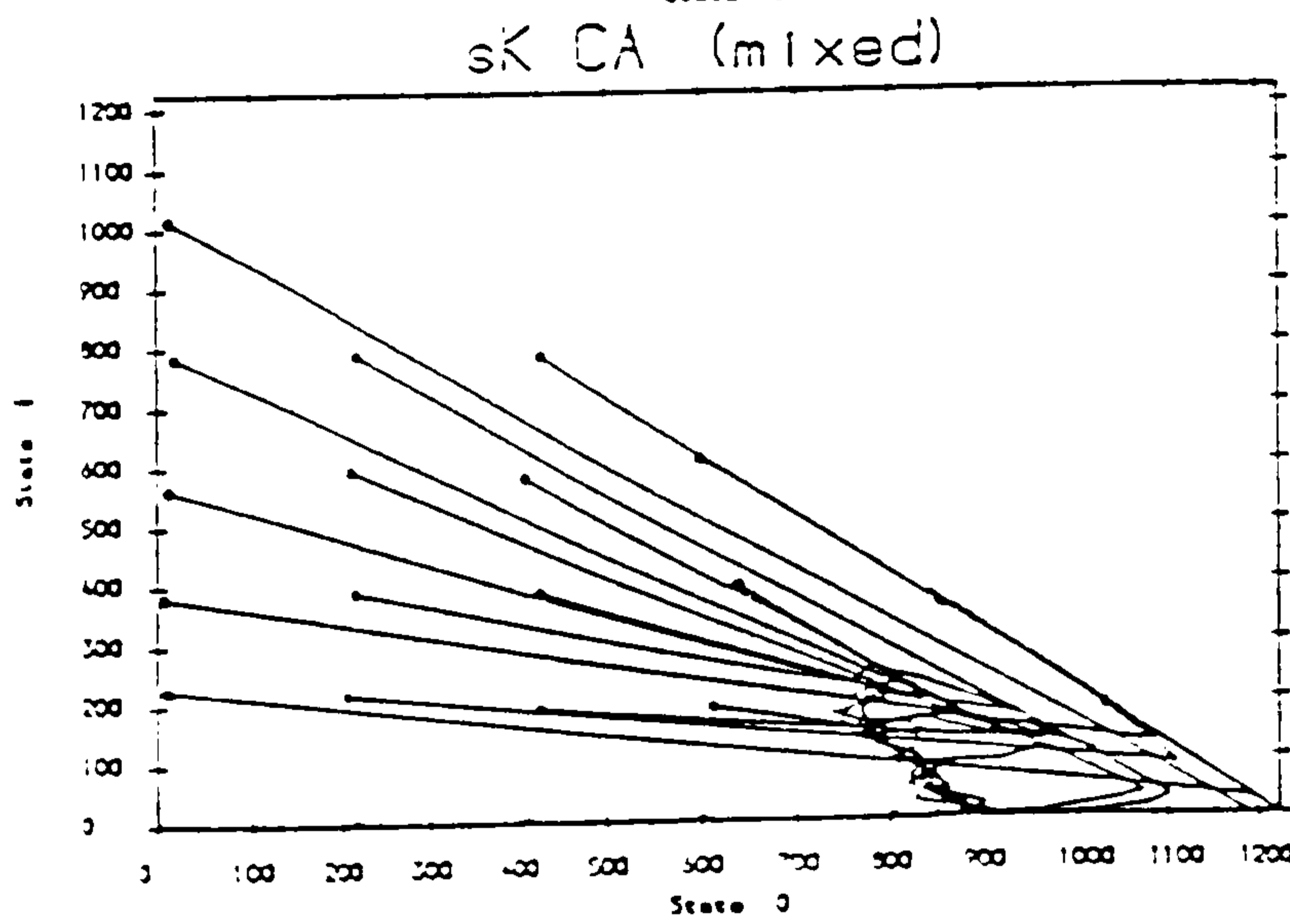
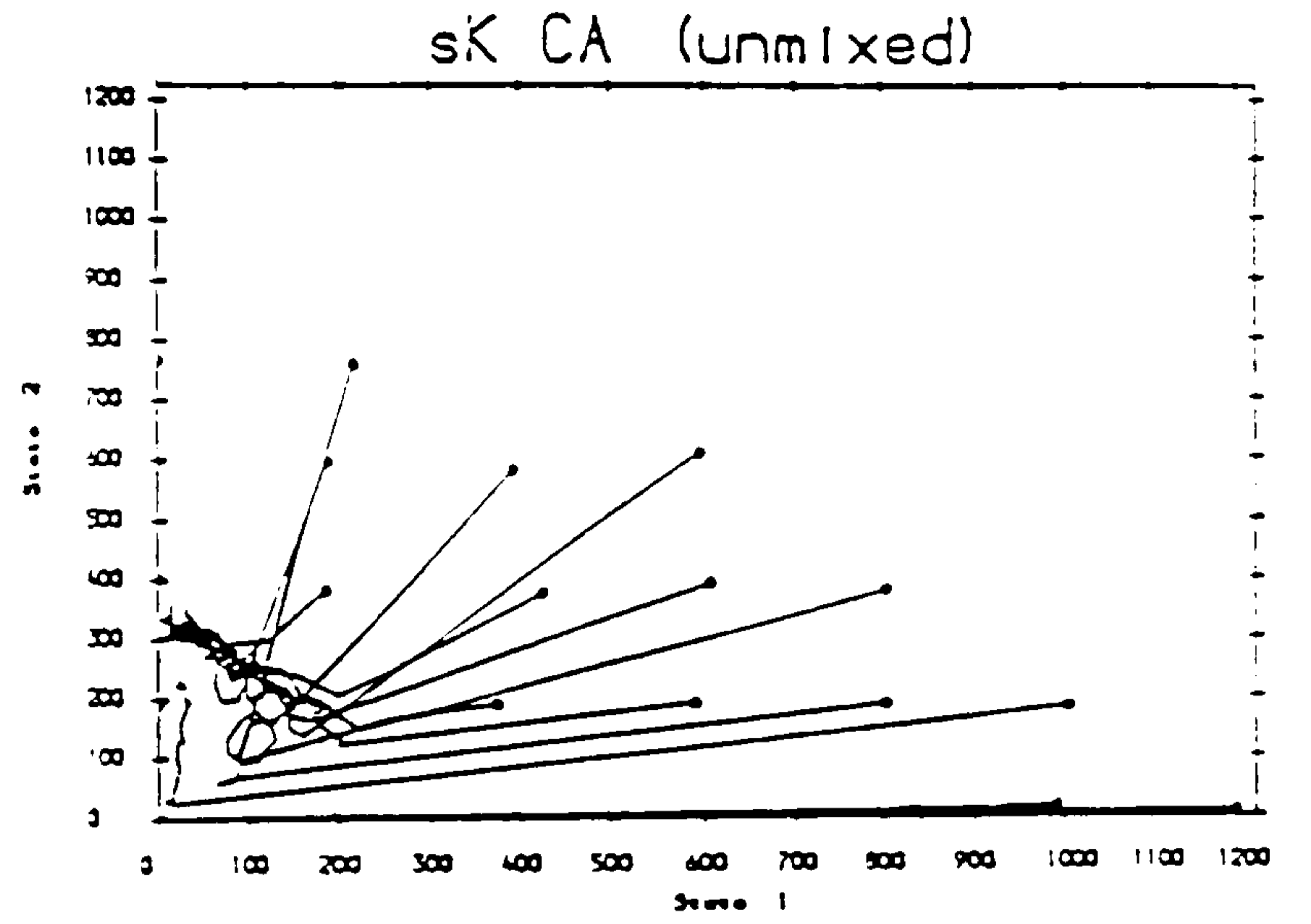
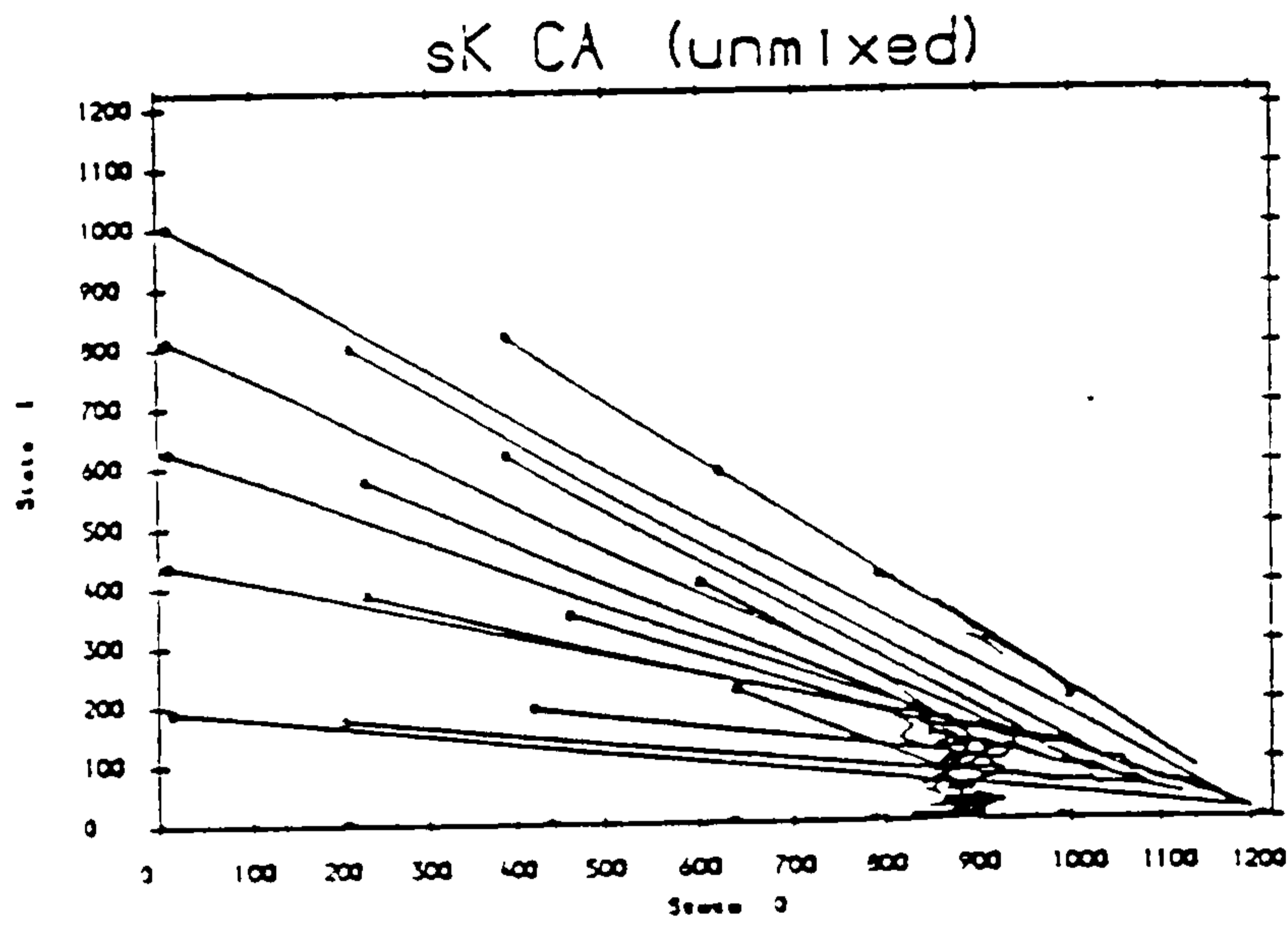




Figure 4.20: Class 3 phase diagrams.  
(b) System E: Deterministic.

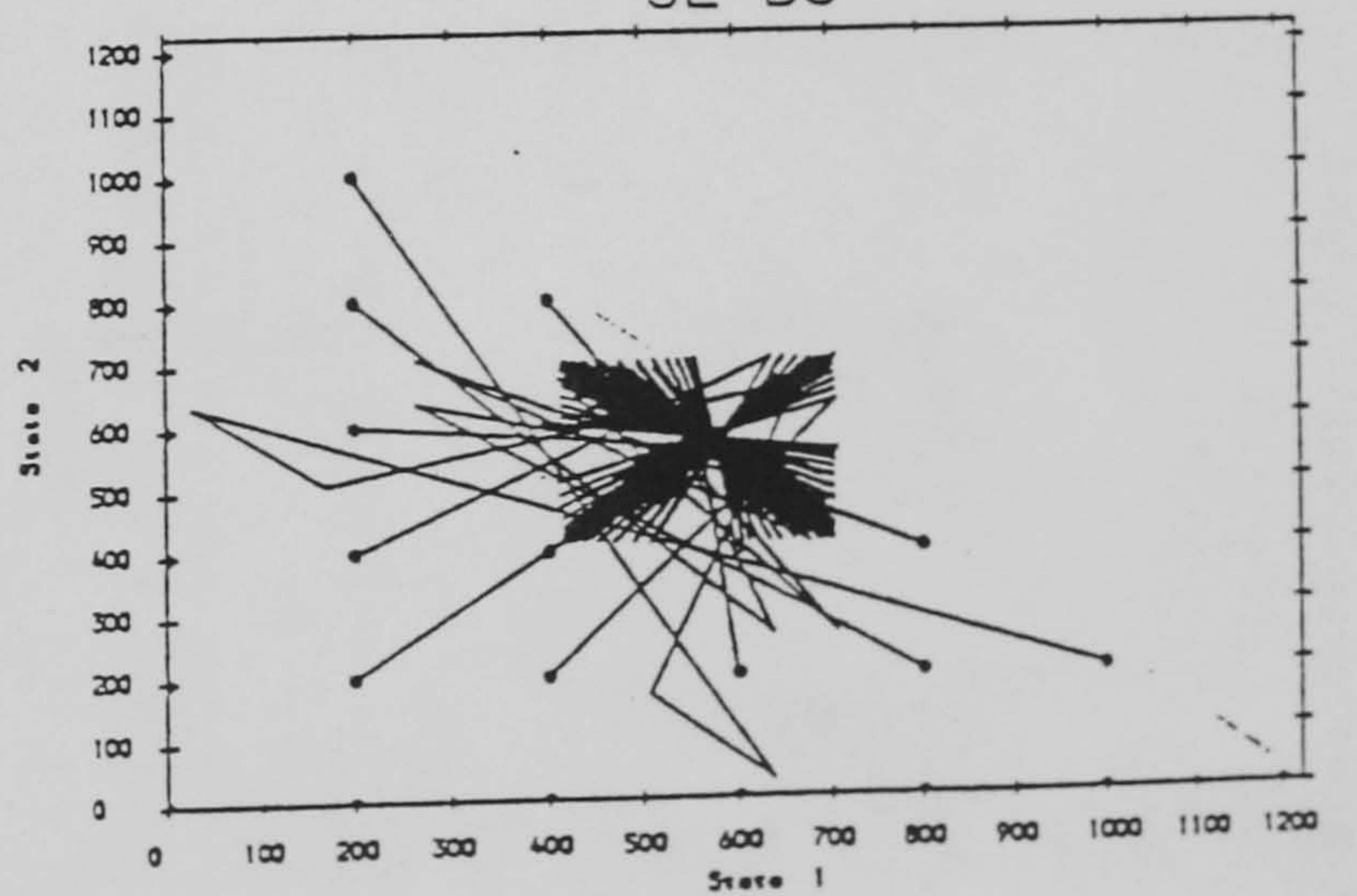
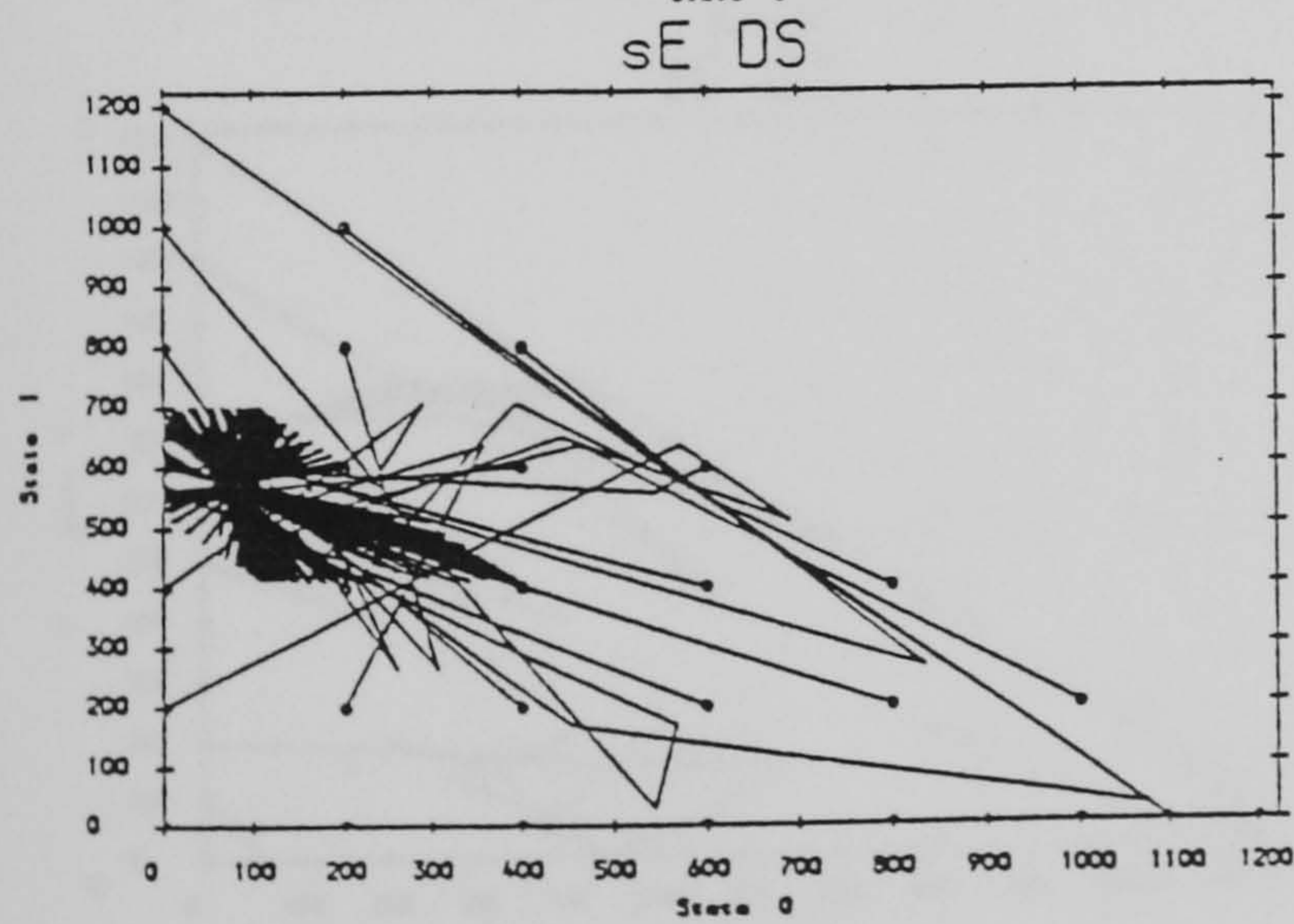
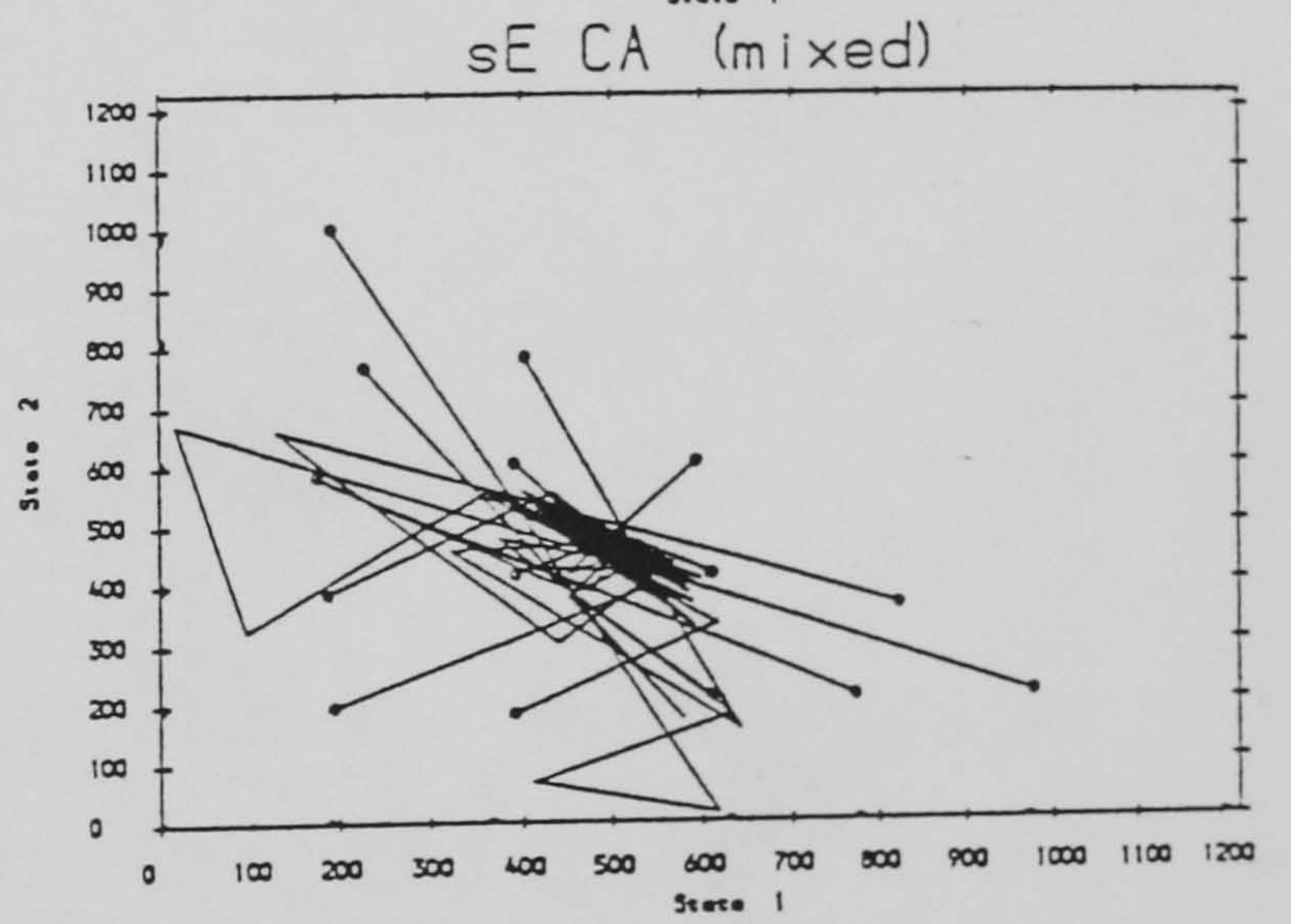
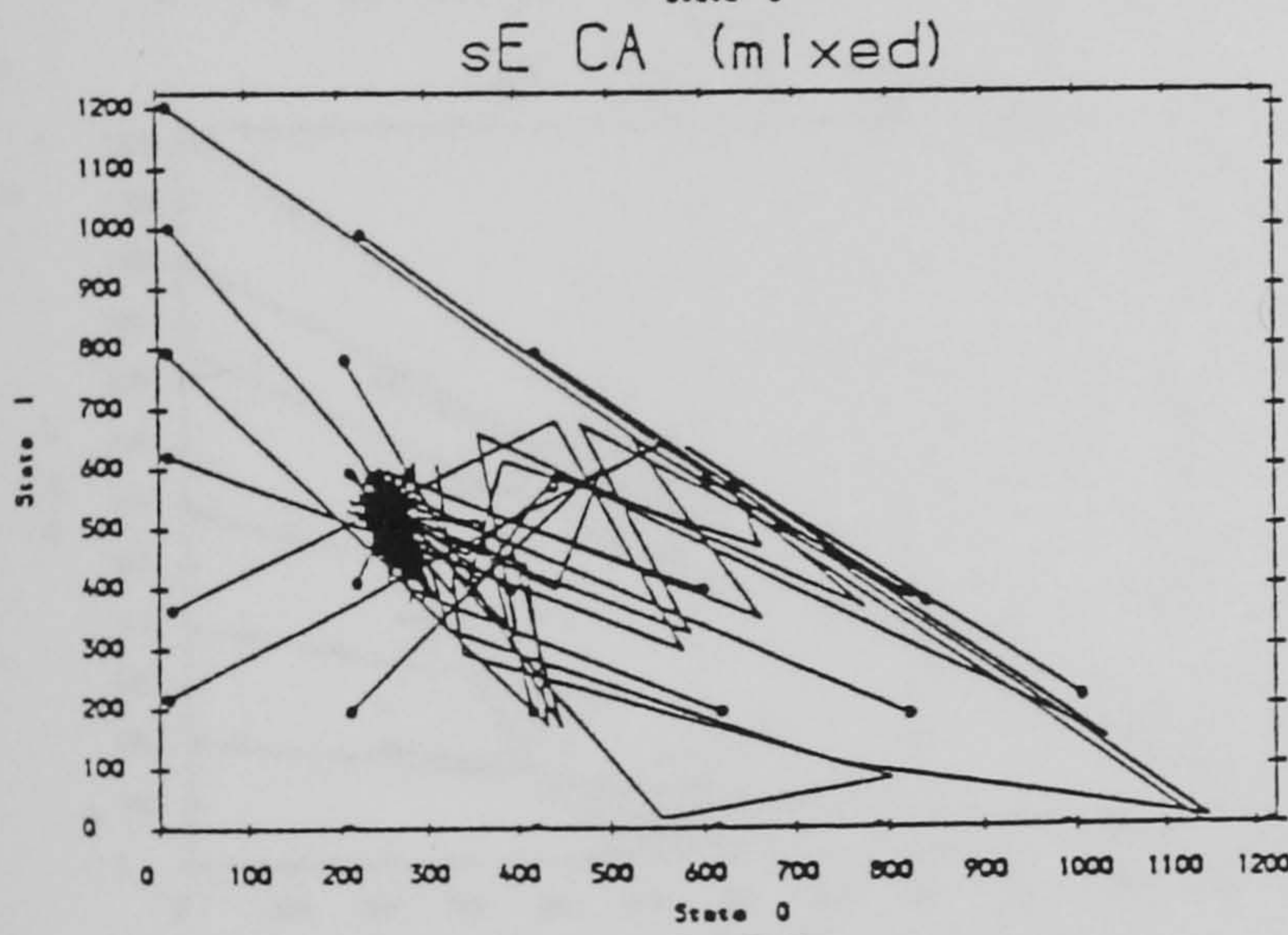
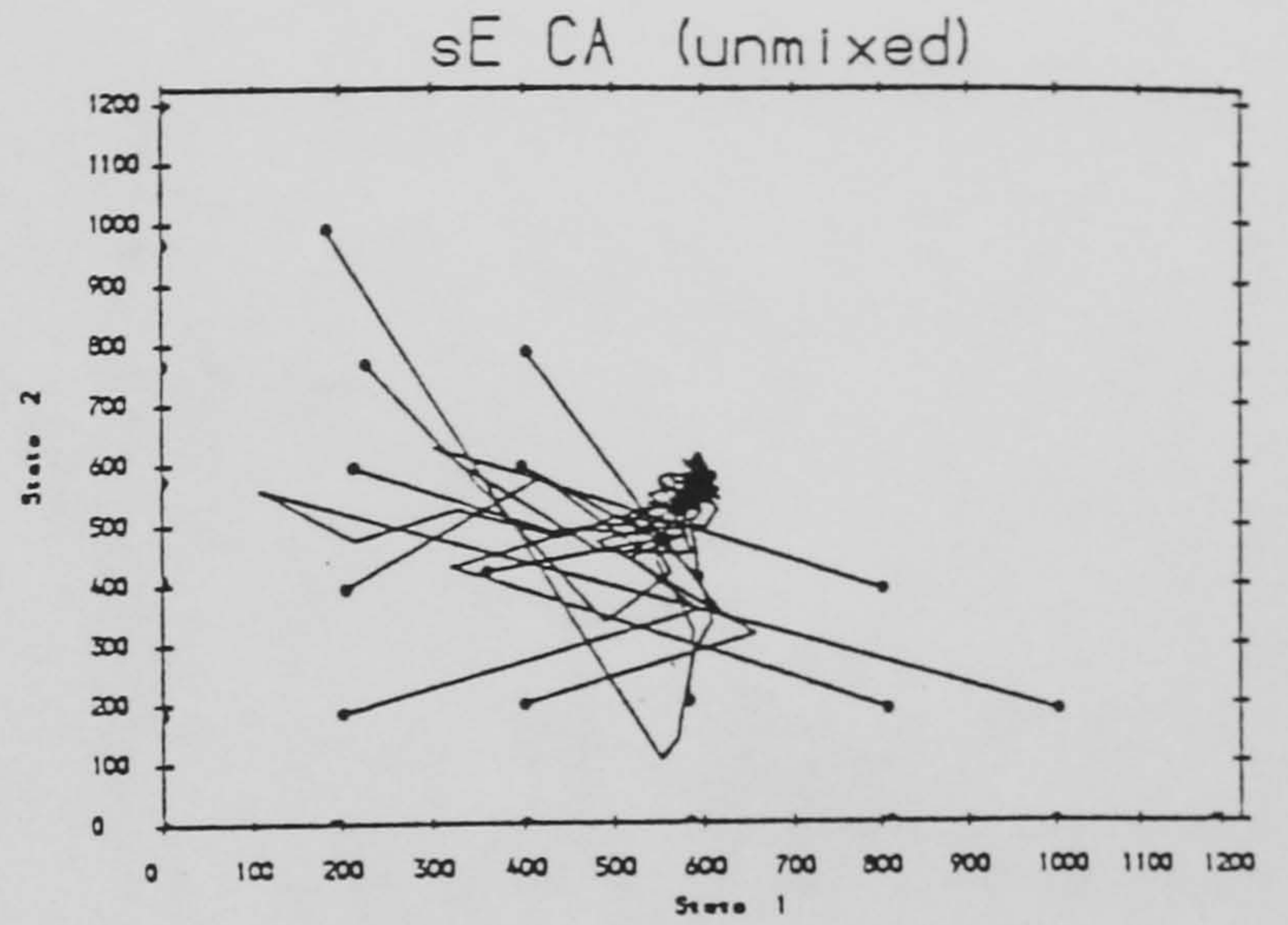
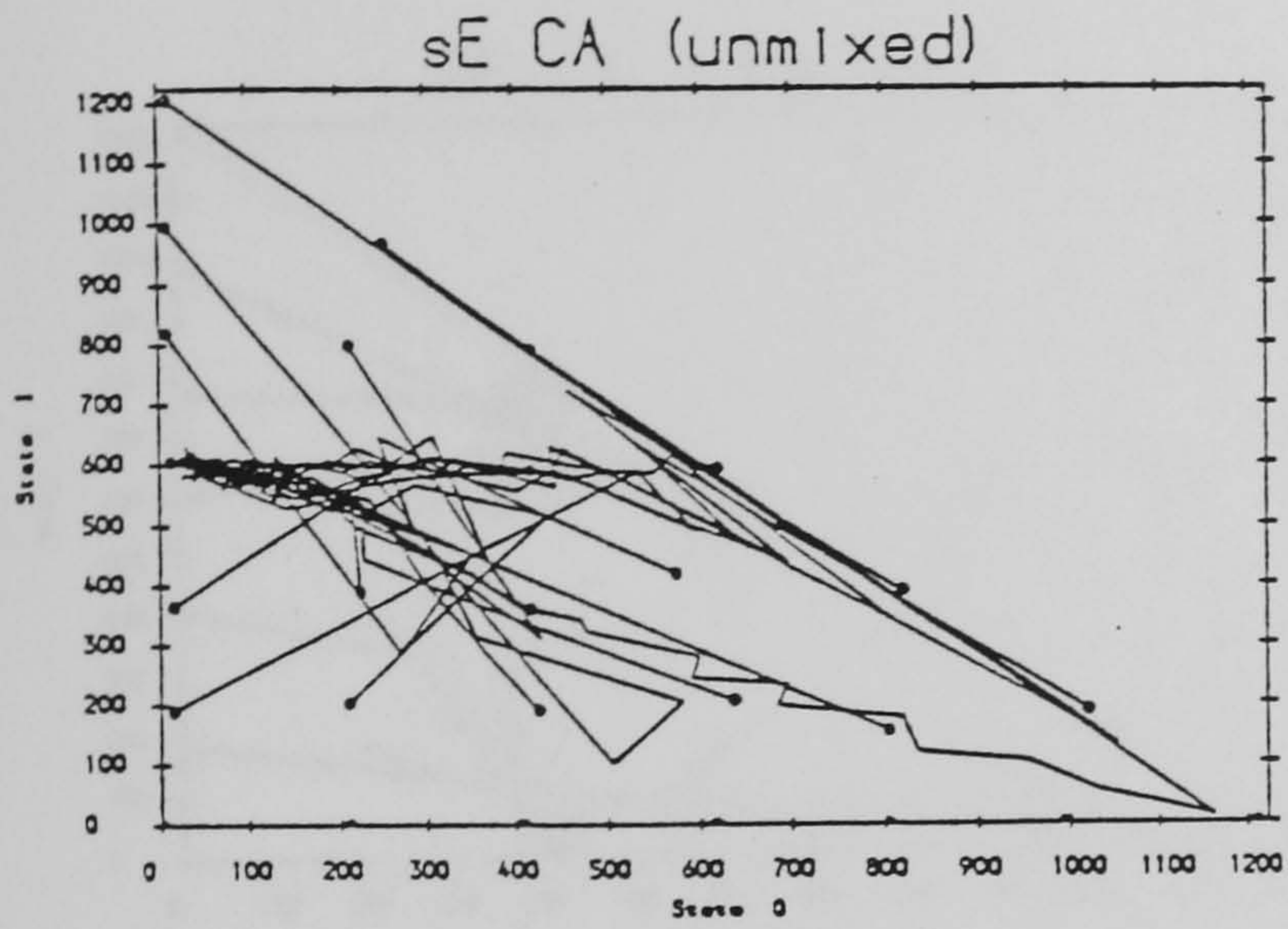




Figure 4.20: Class 3 phase diagrams.  
(c) System F: Deterministic.

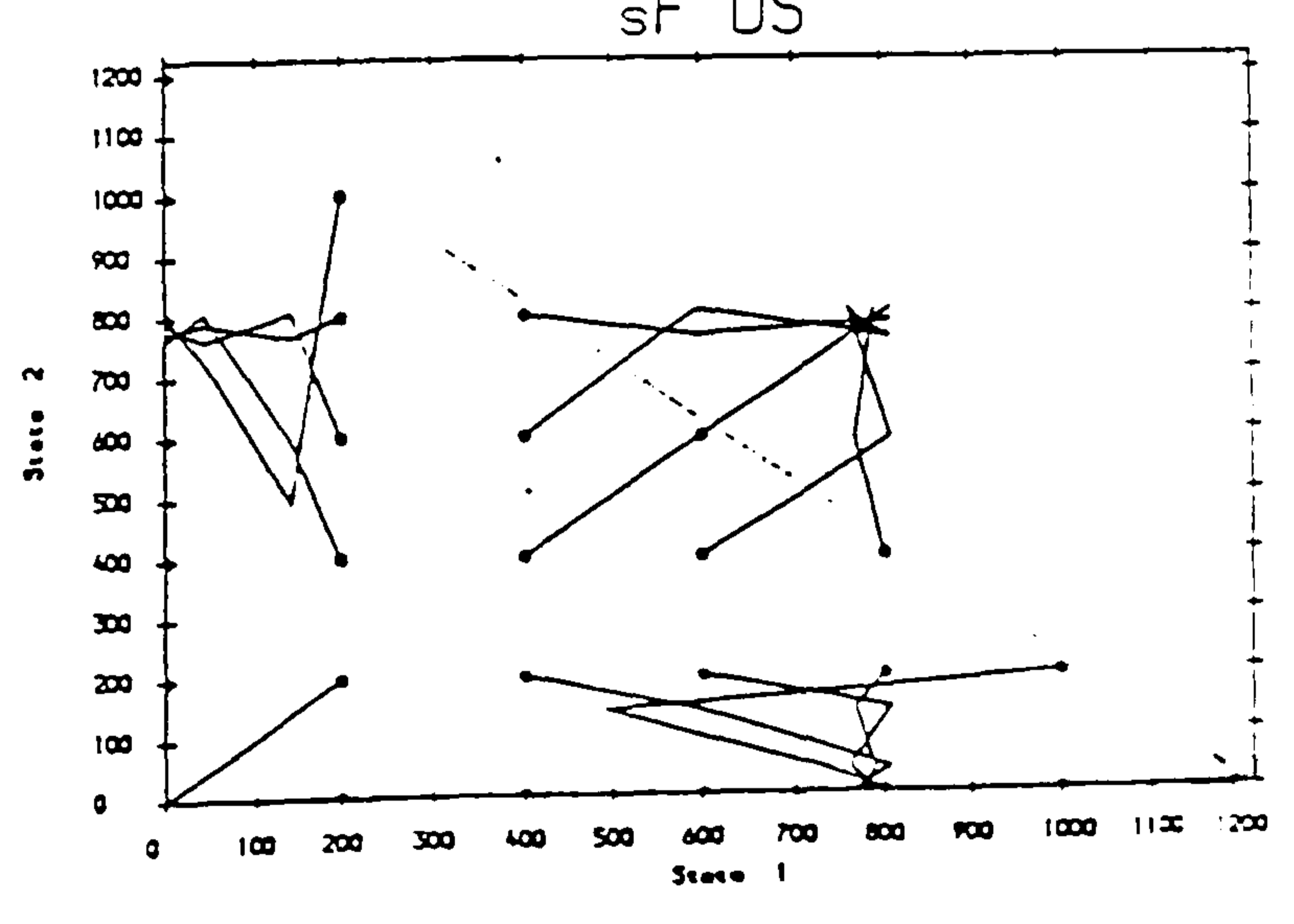
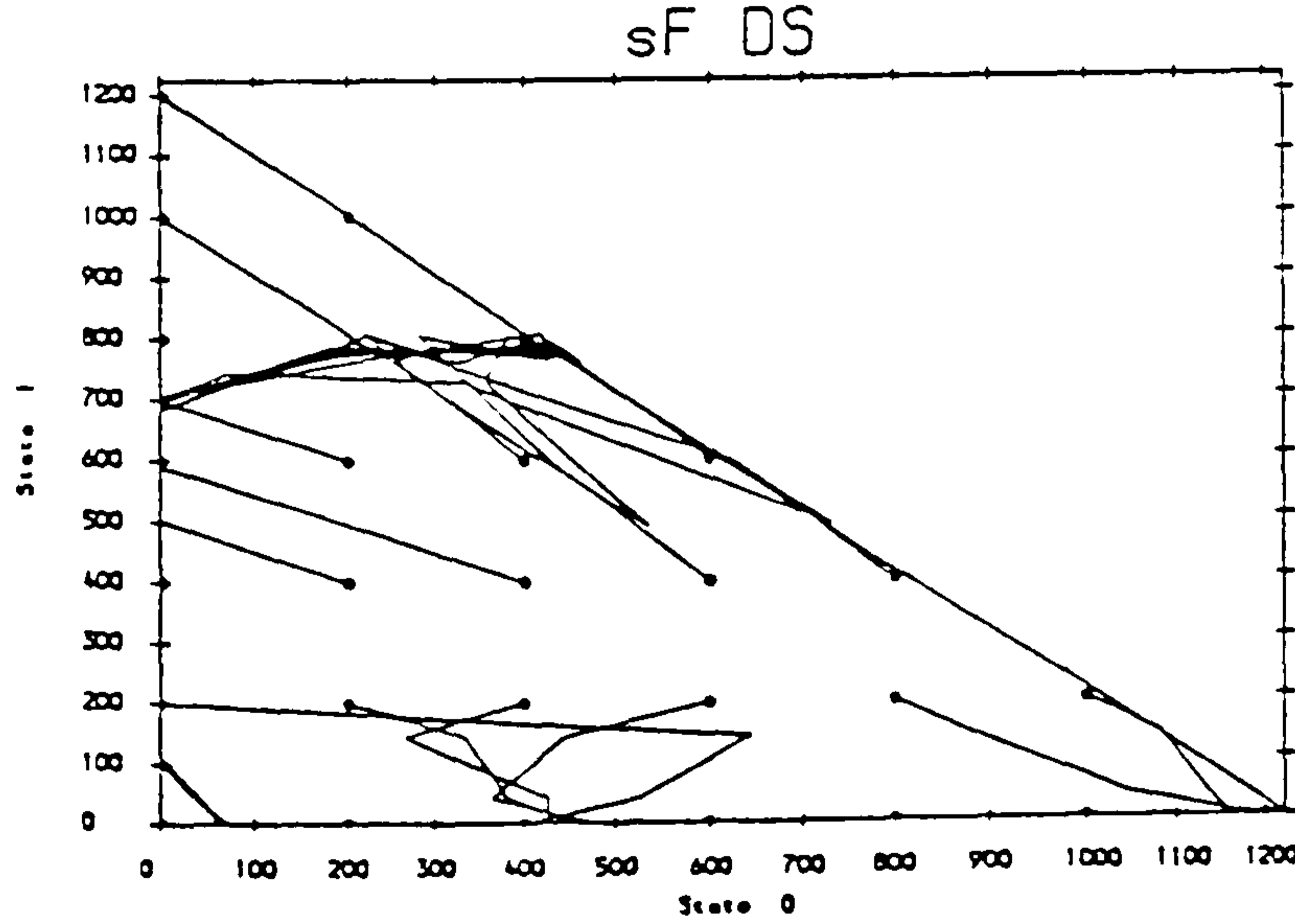
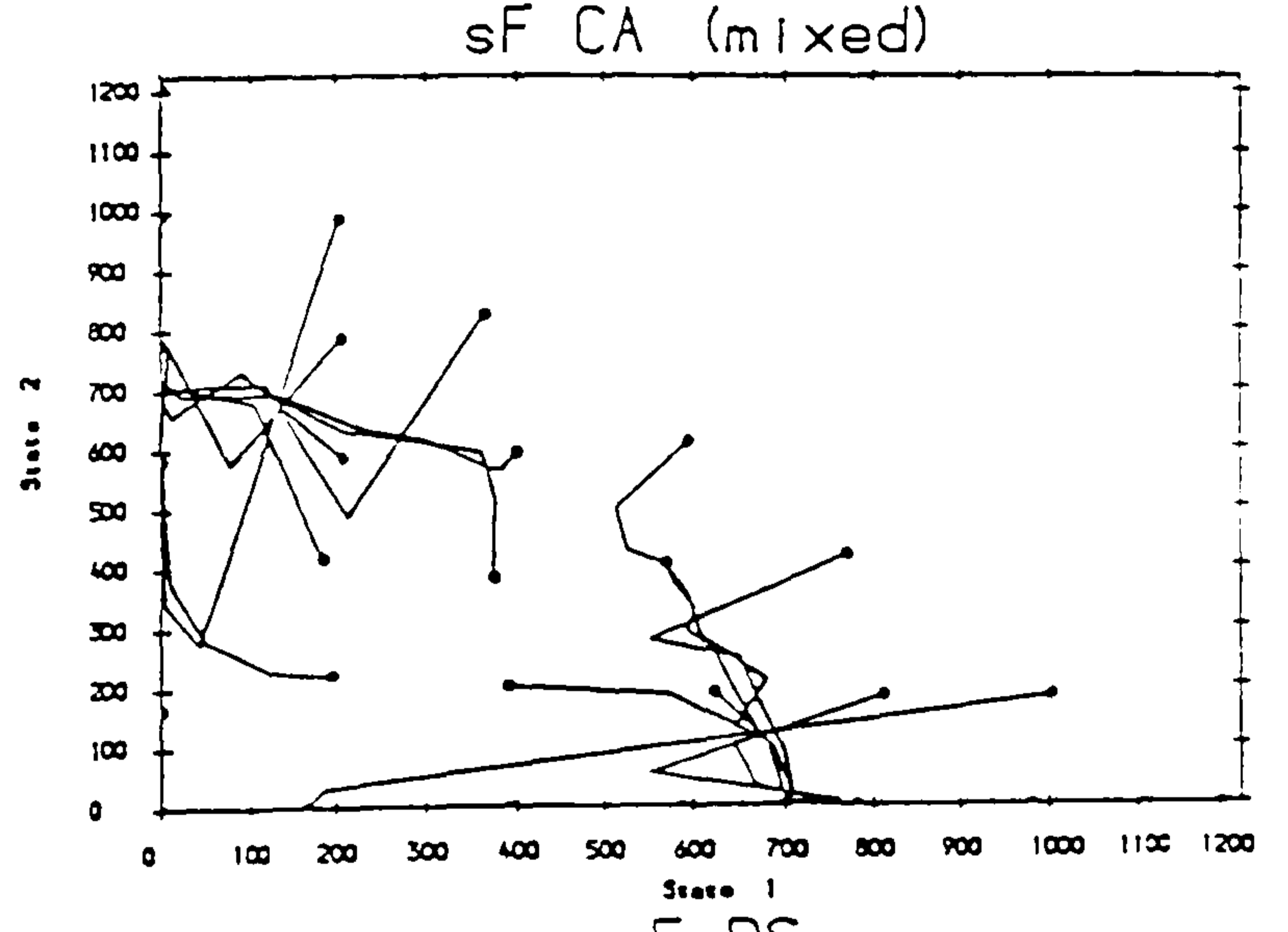
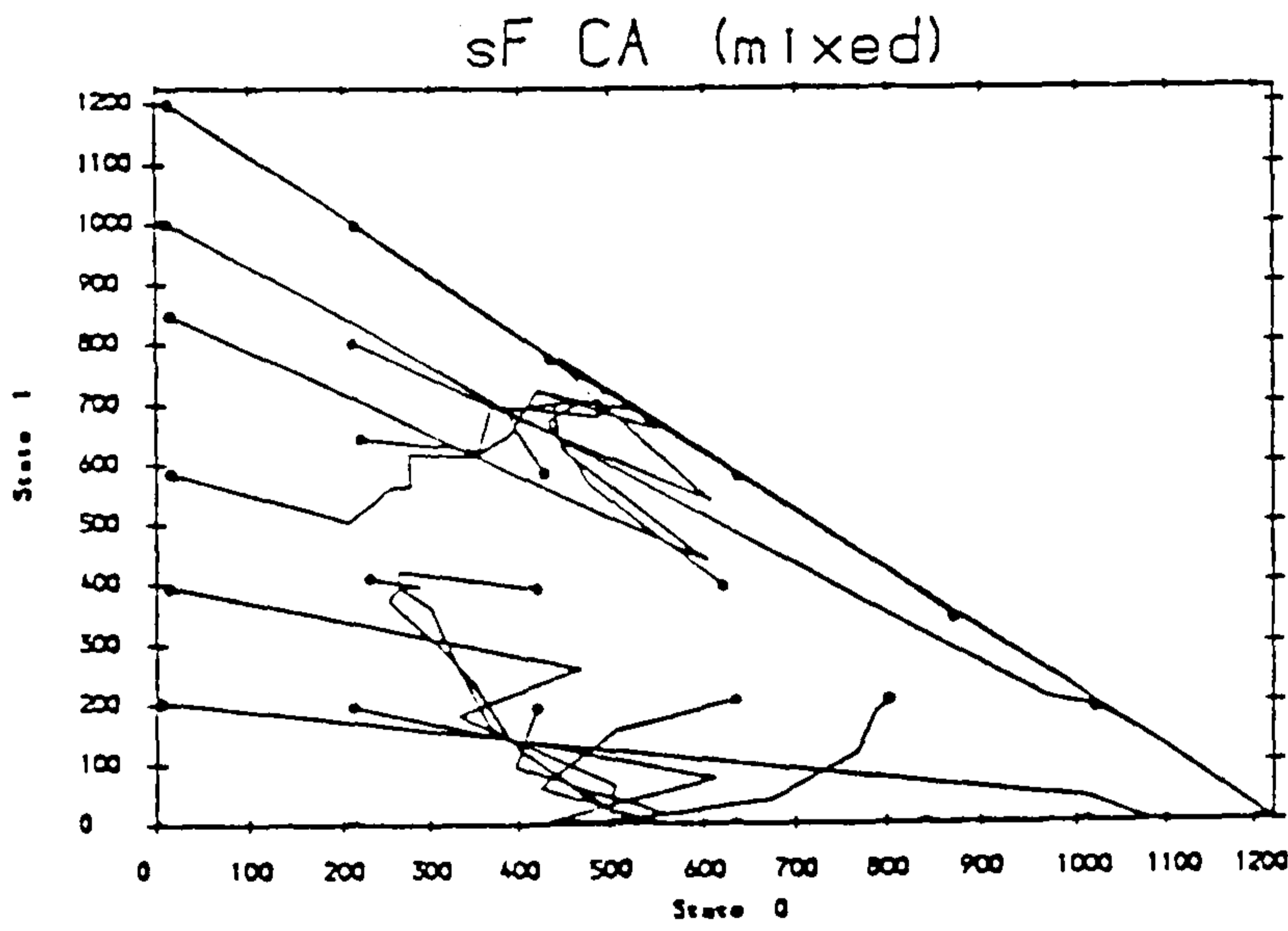
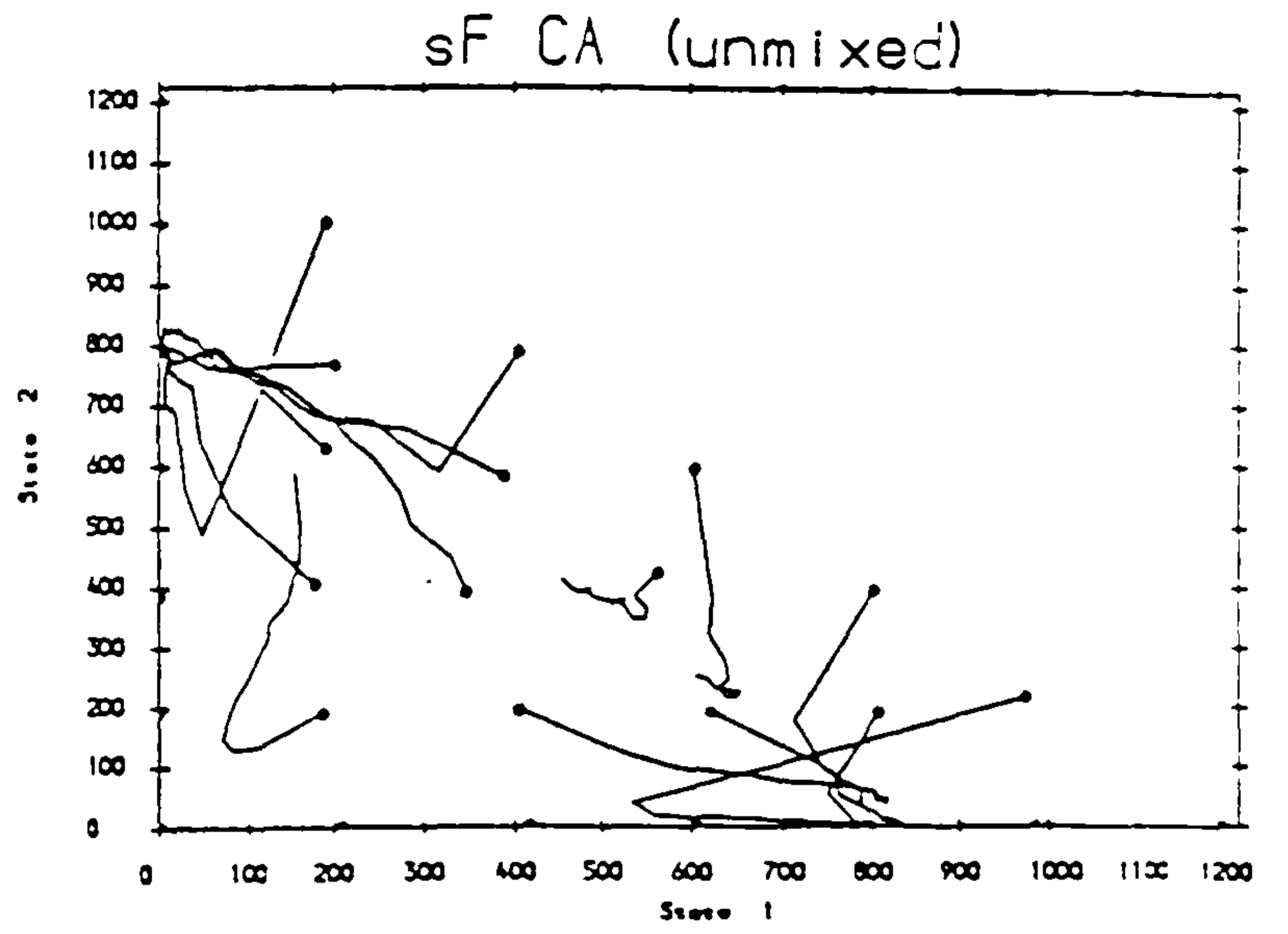
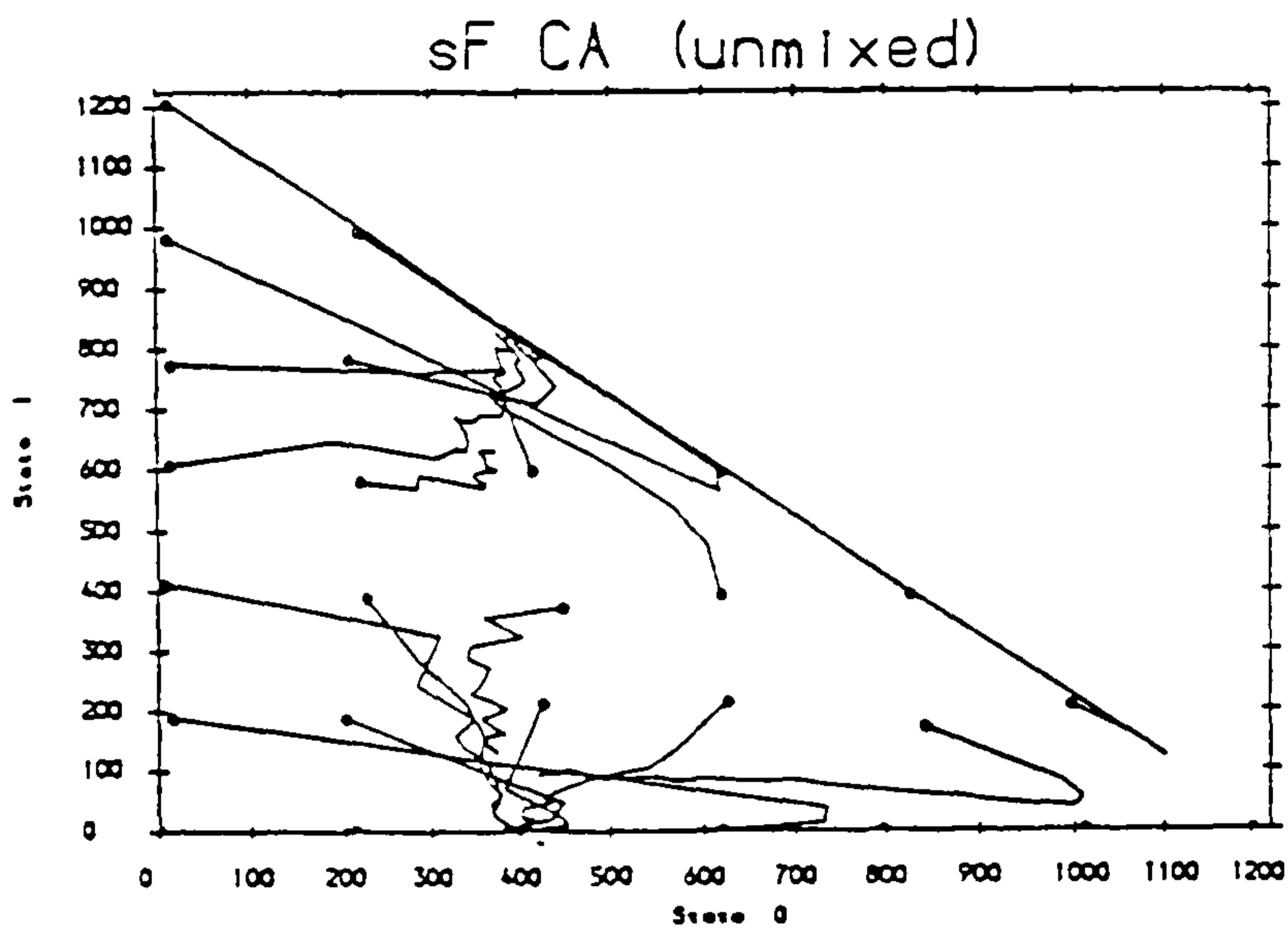


Figure 4.20: Class 3 phase diagrams.  
(d) System G: Deterministic.

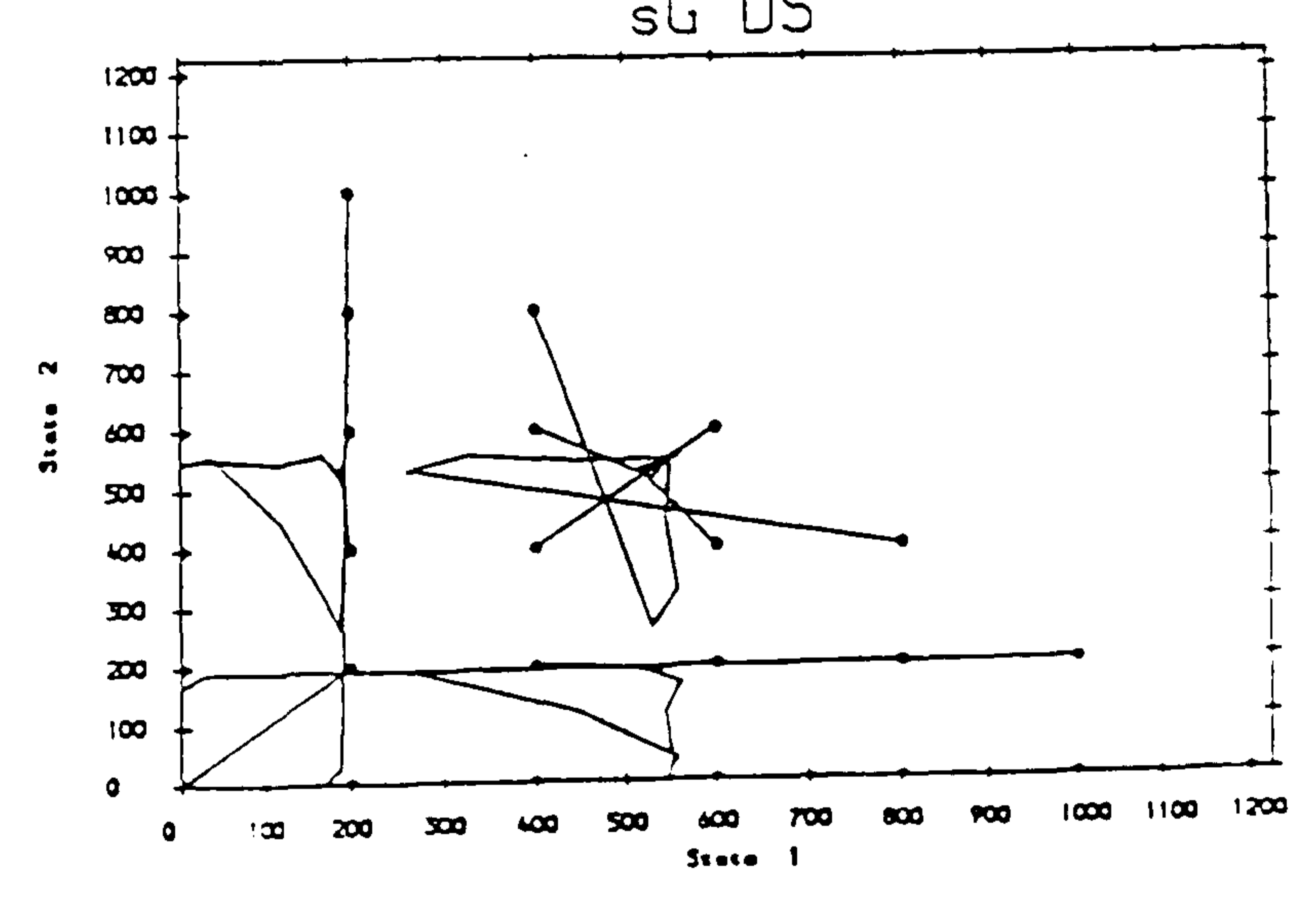
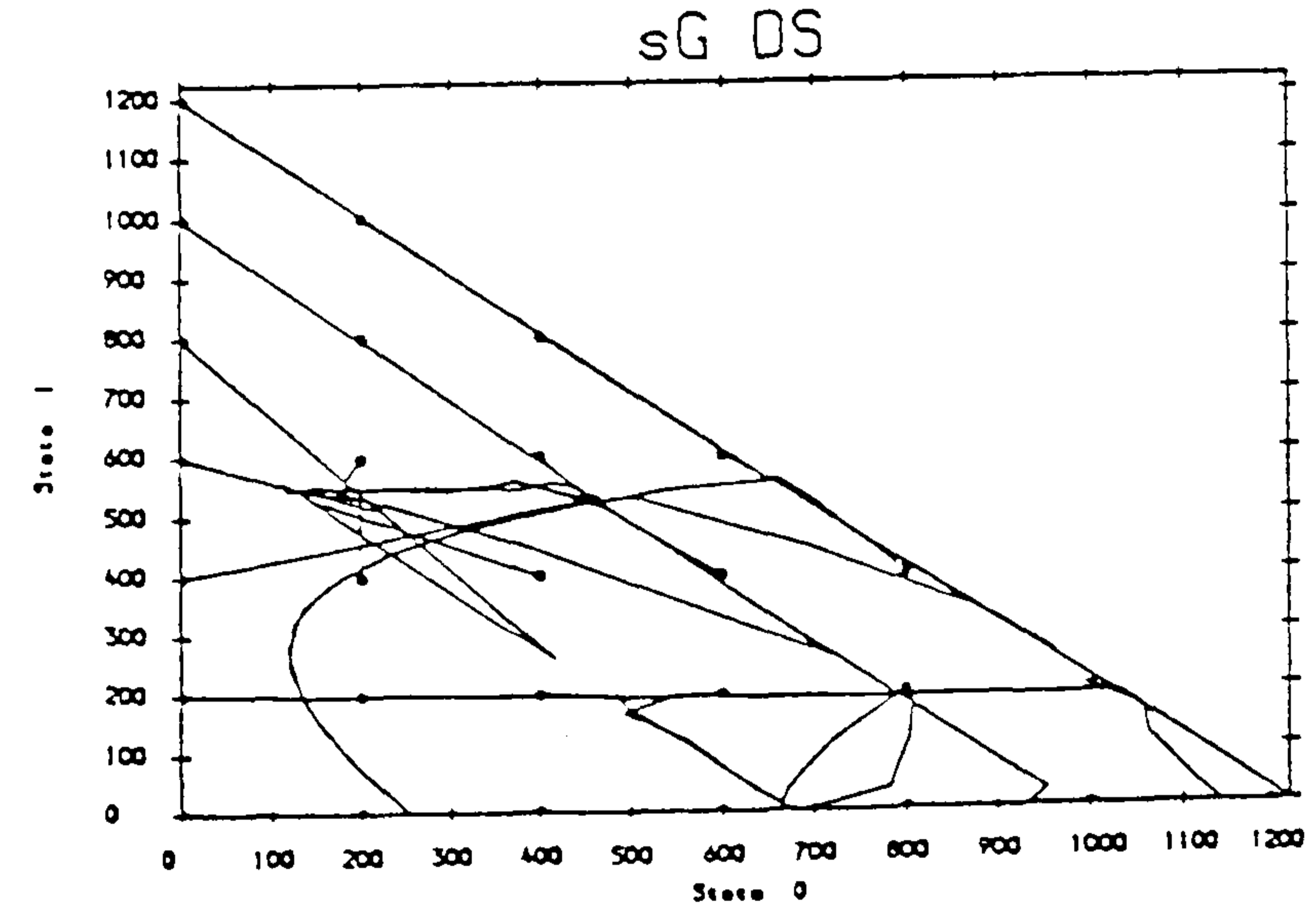
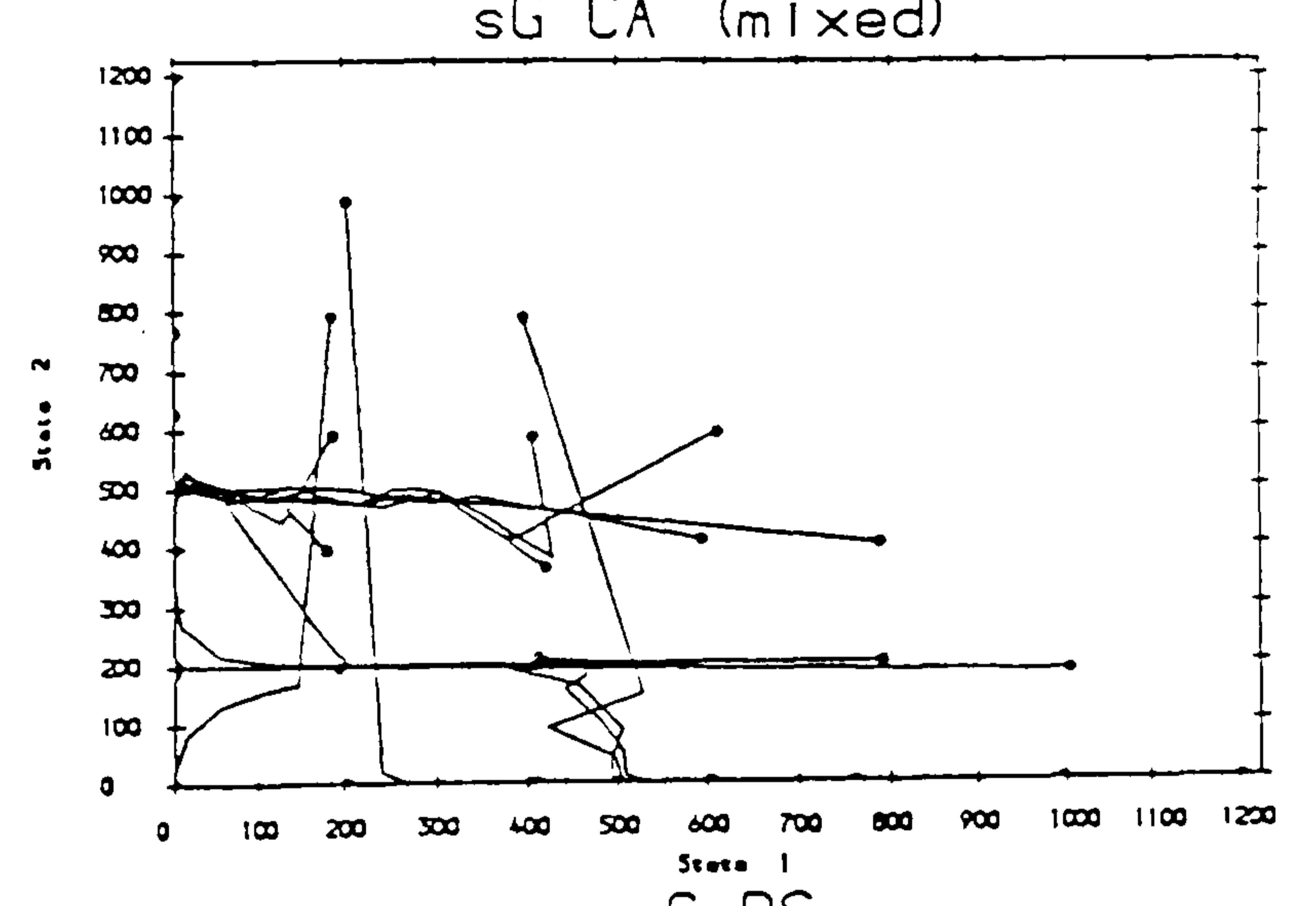
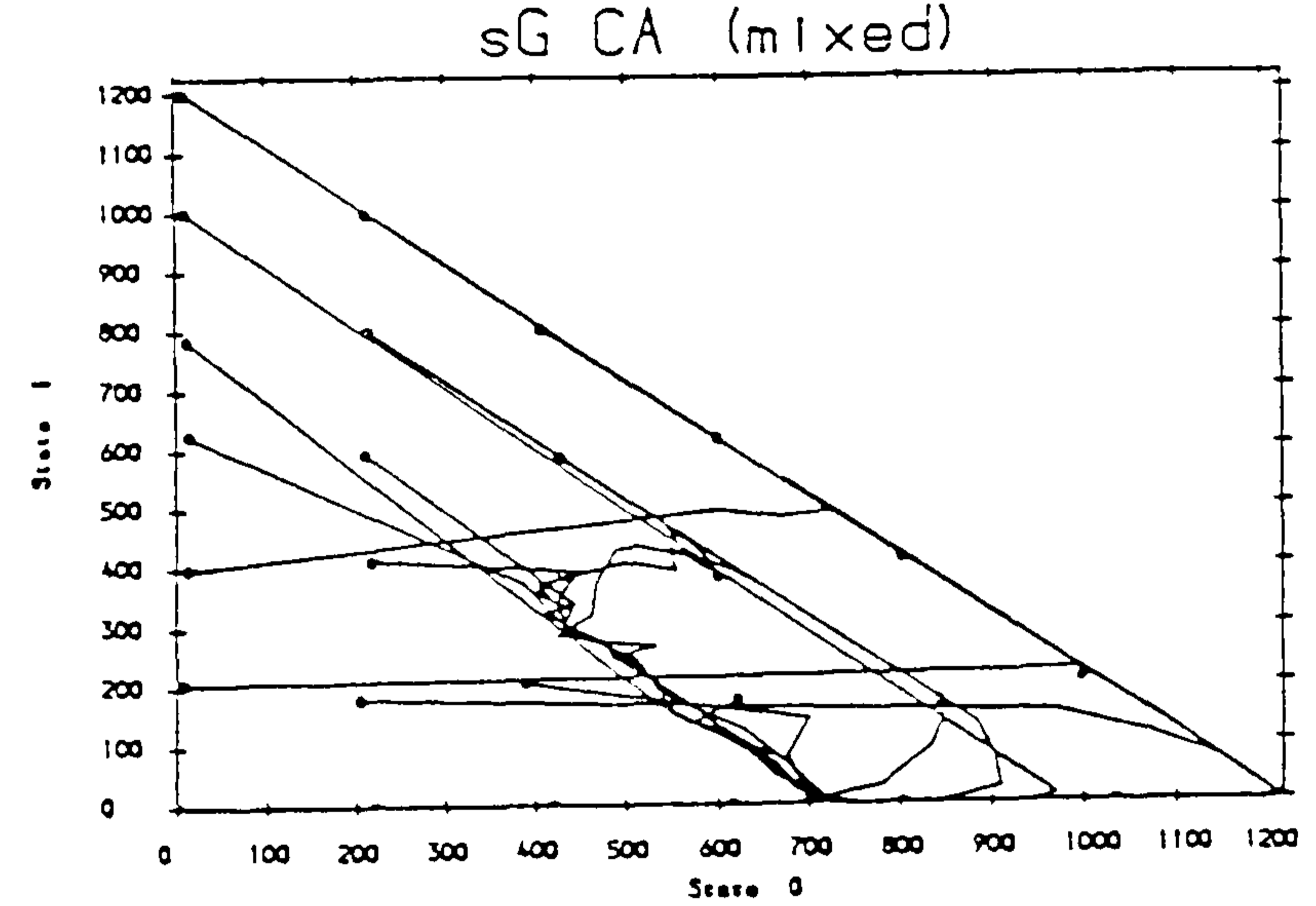
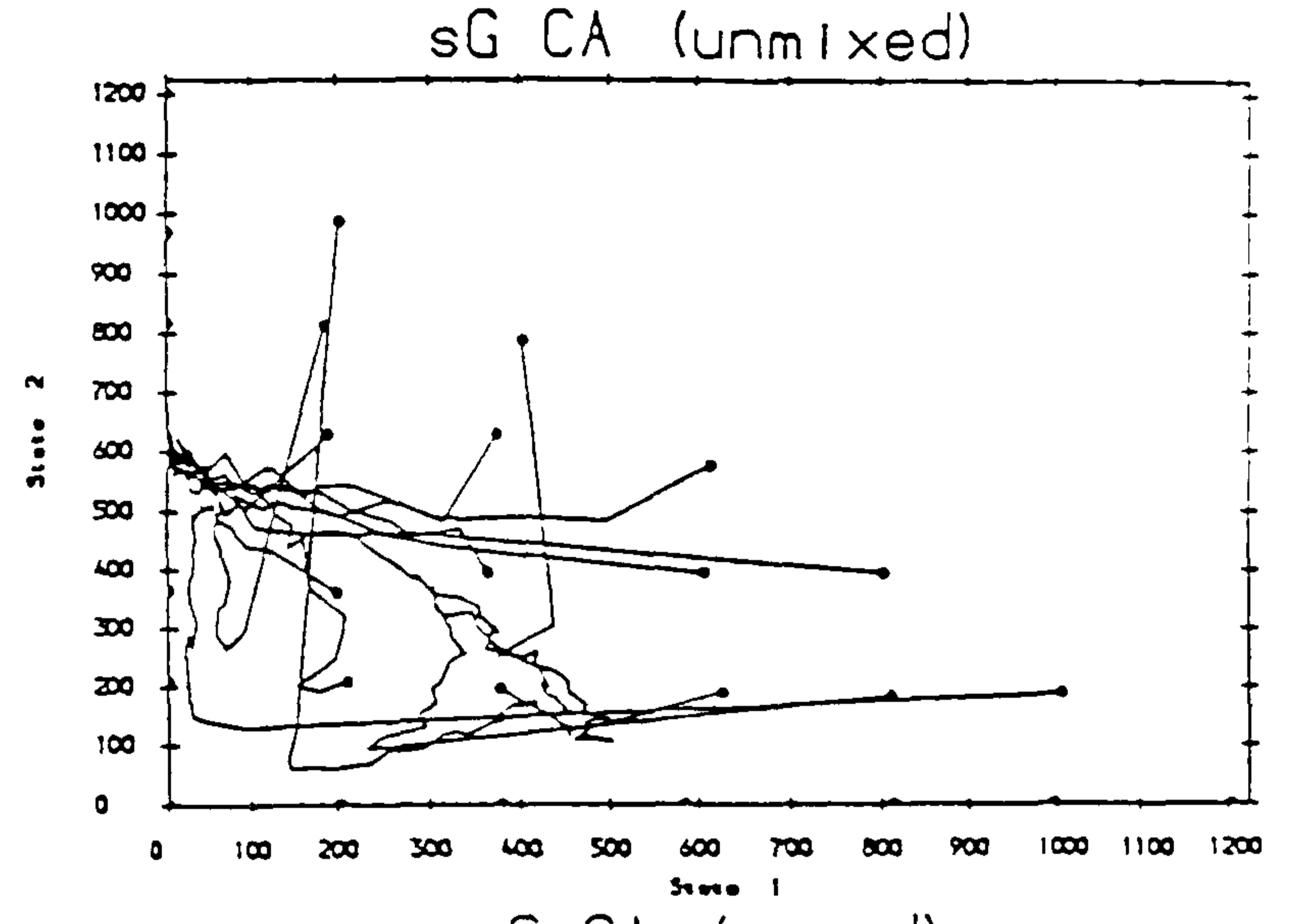
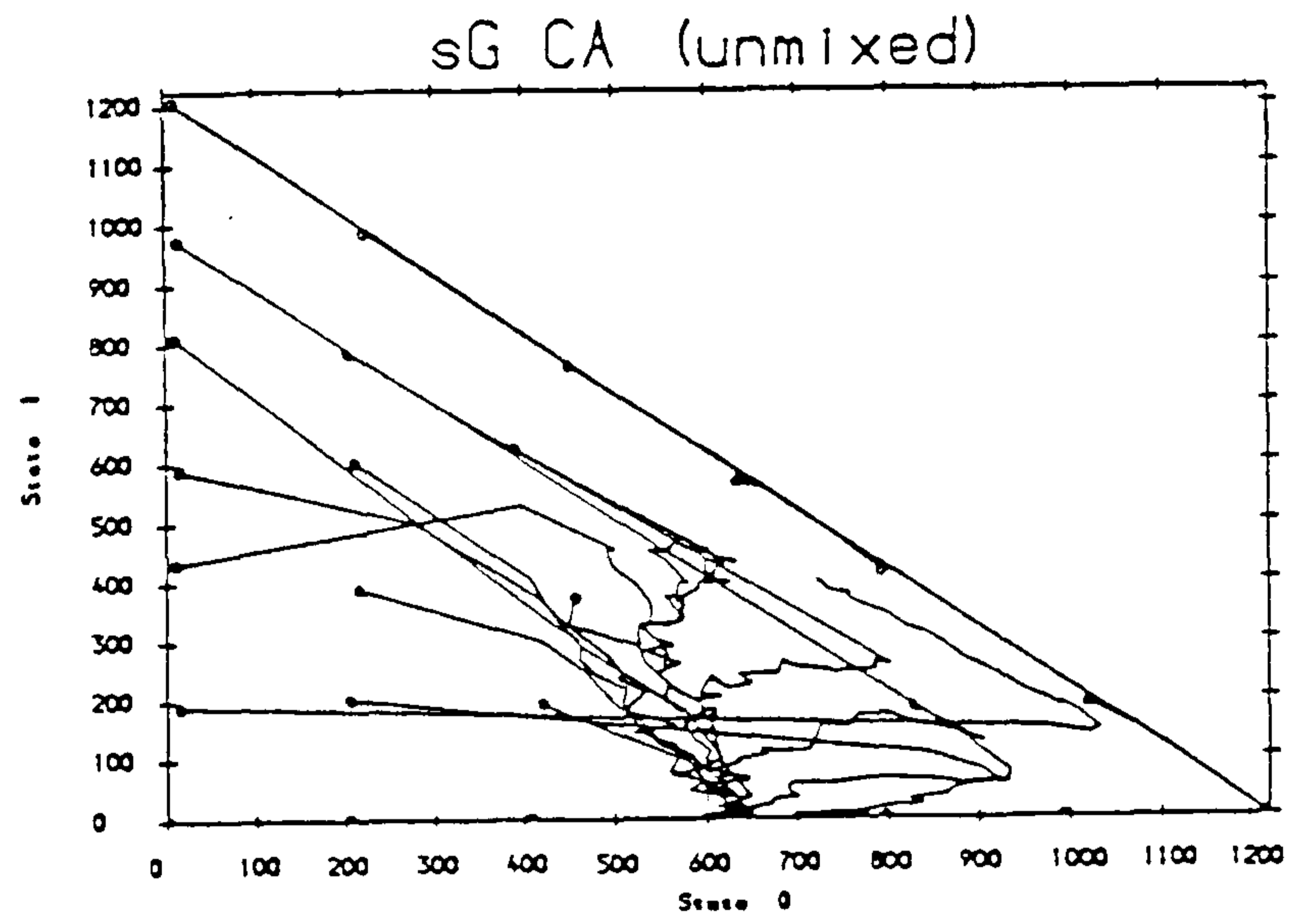


Figure 4.20: Class 3 phase diagrams.  
 (a) System K: Probabilistic.

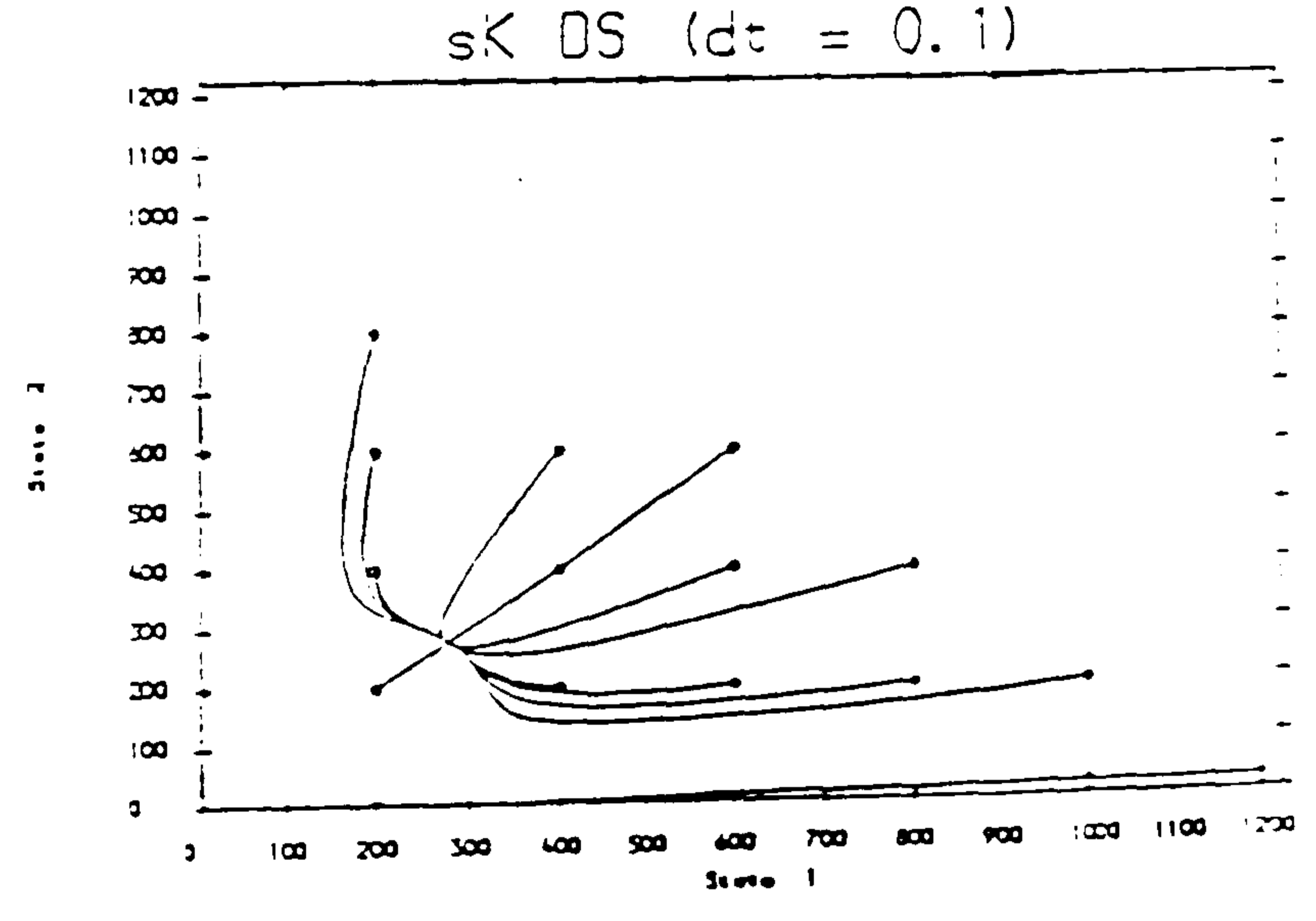
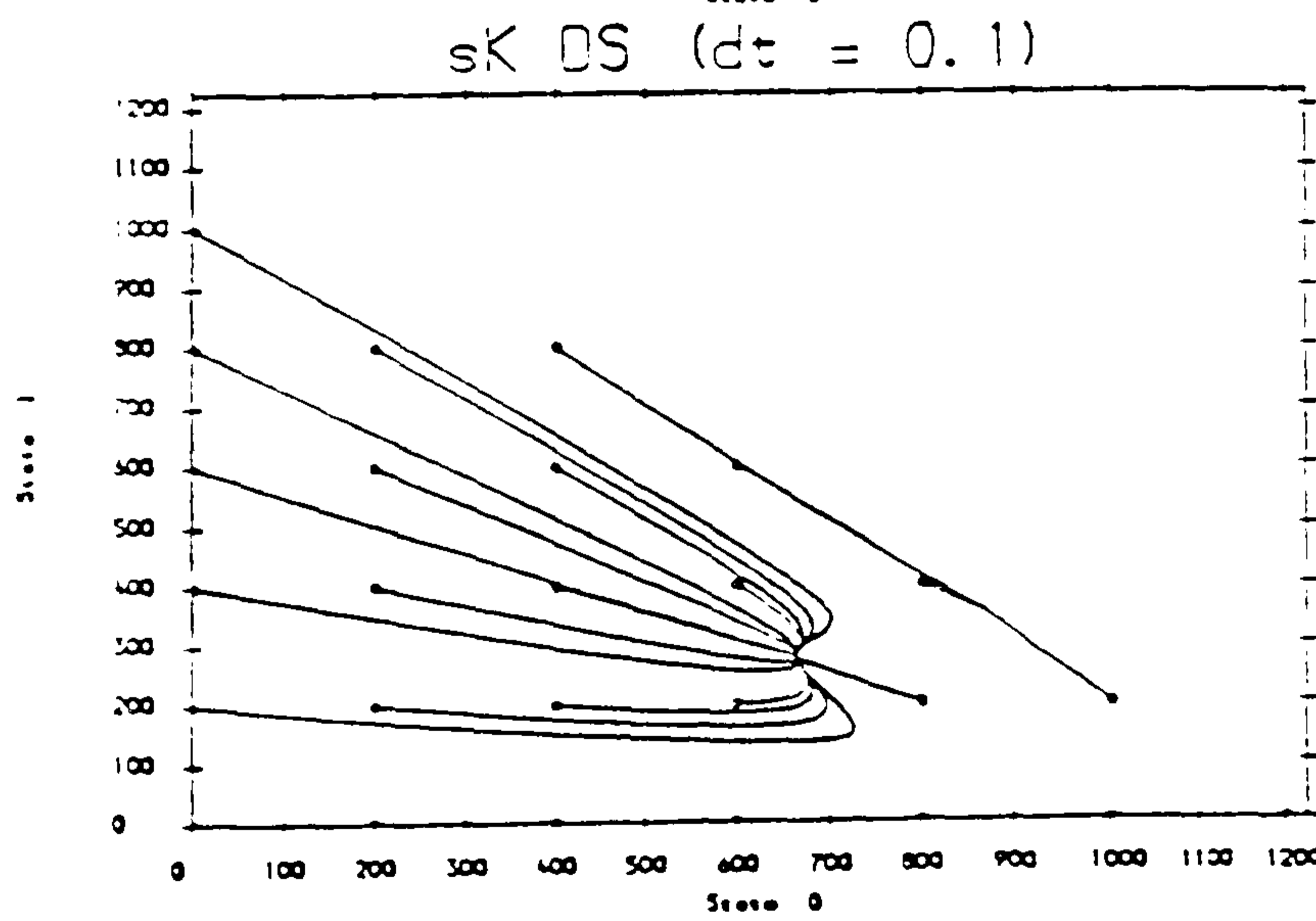
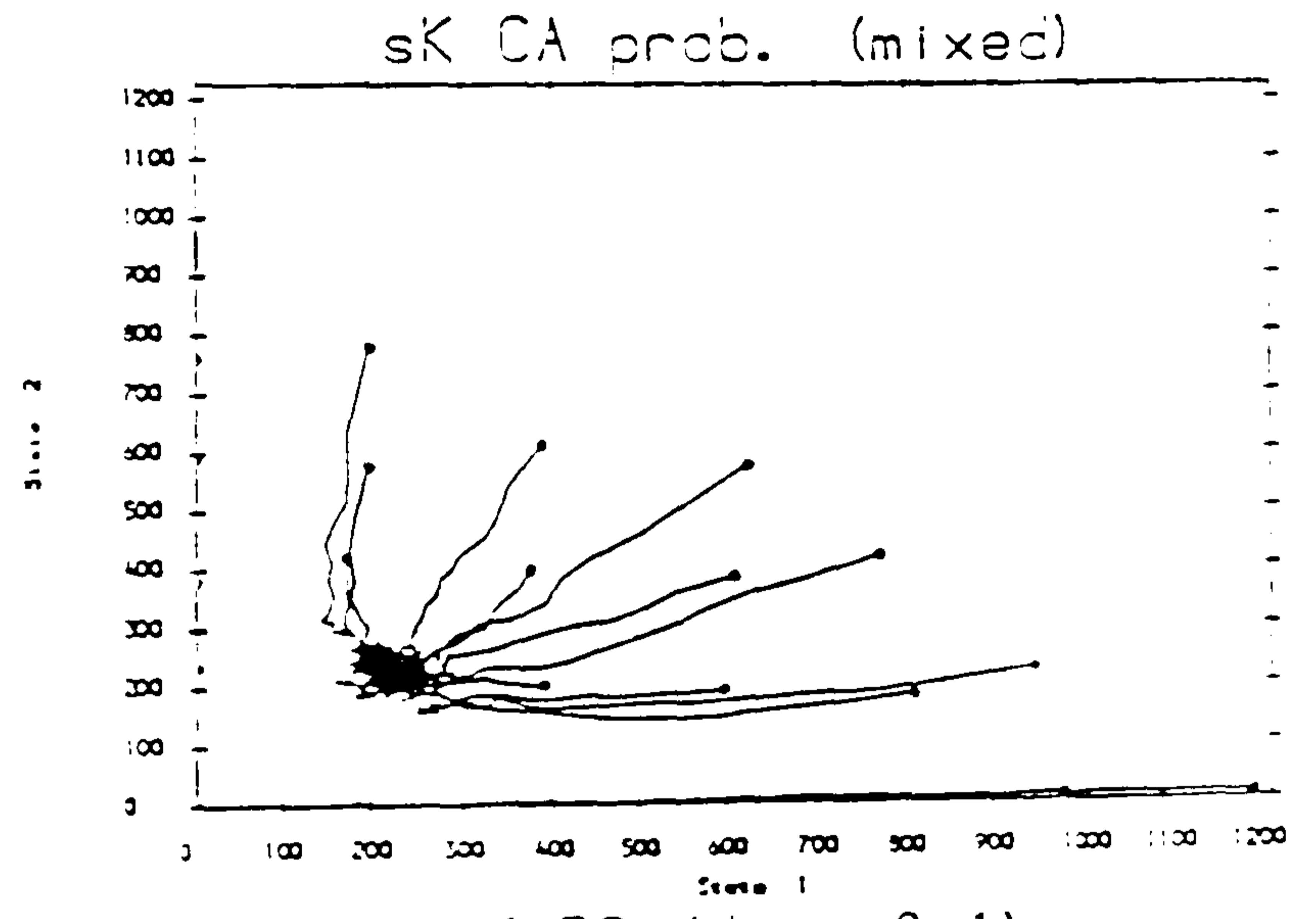
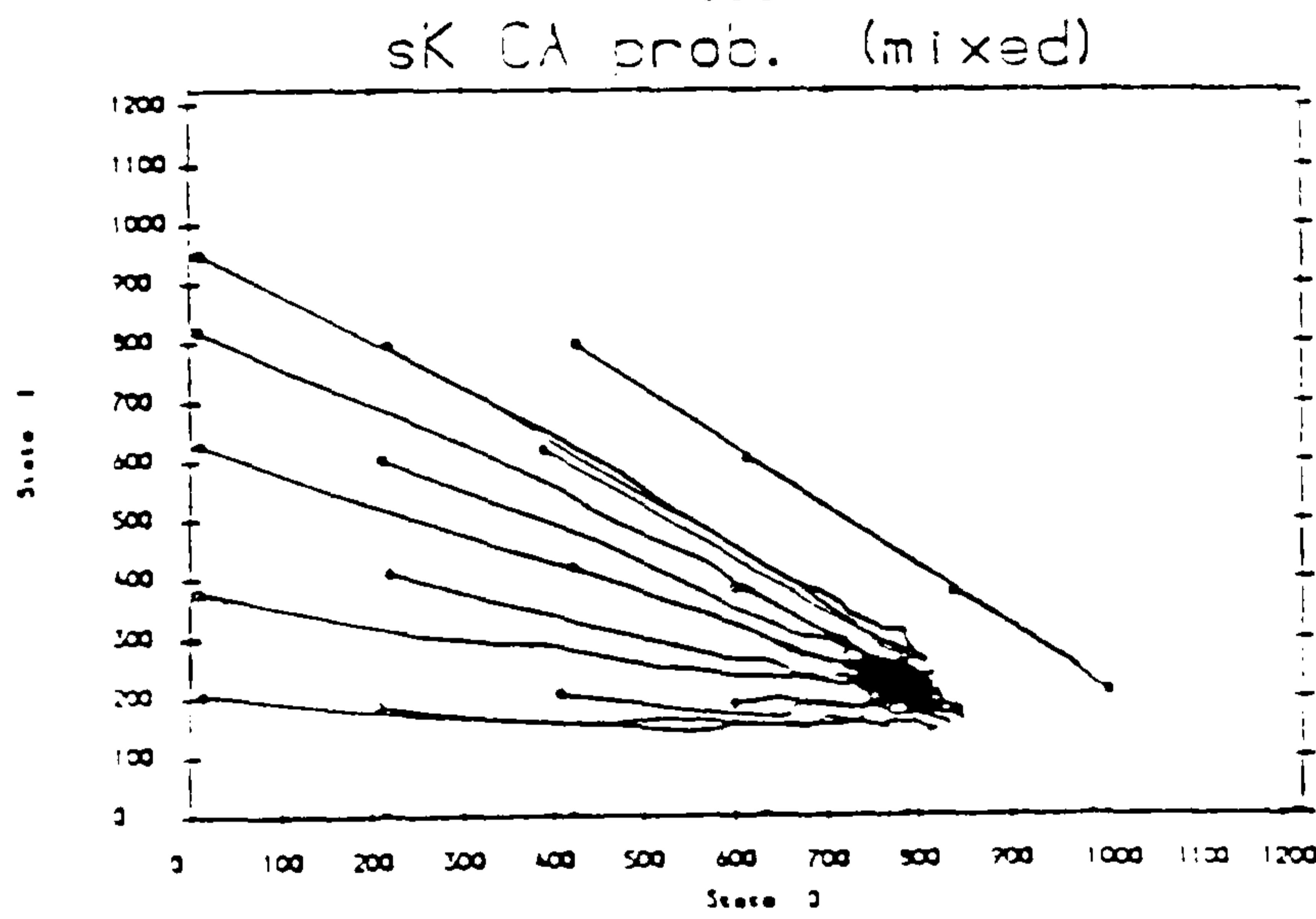
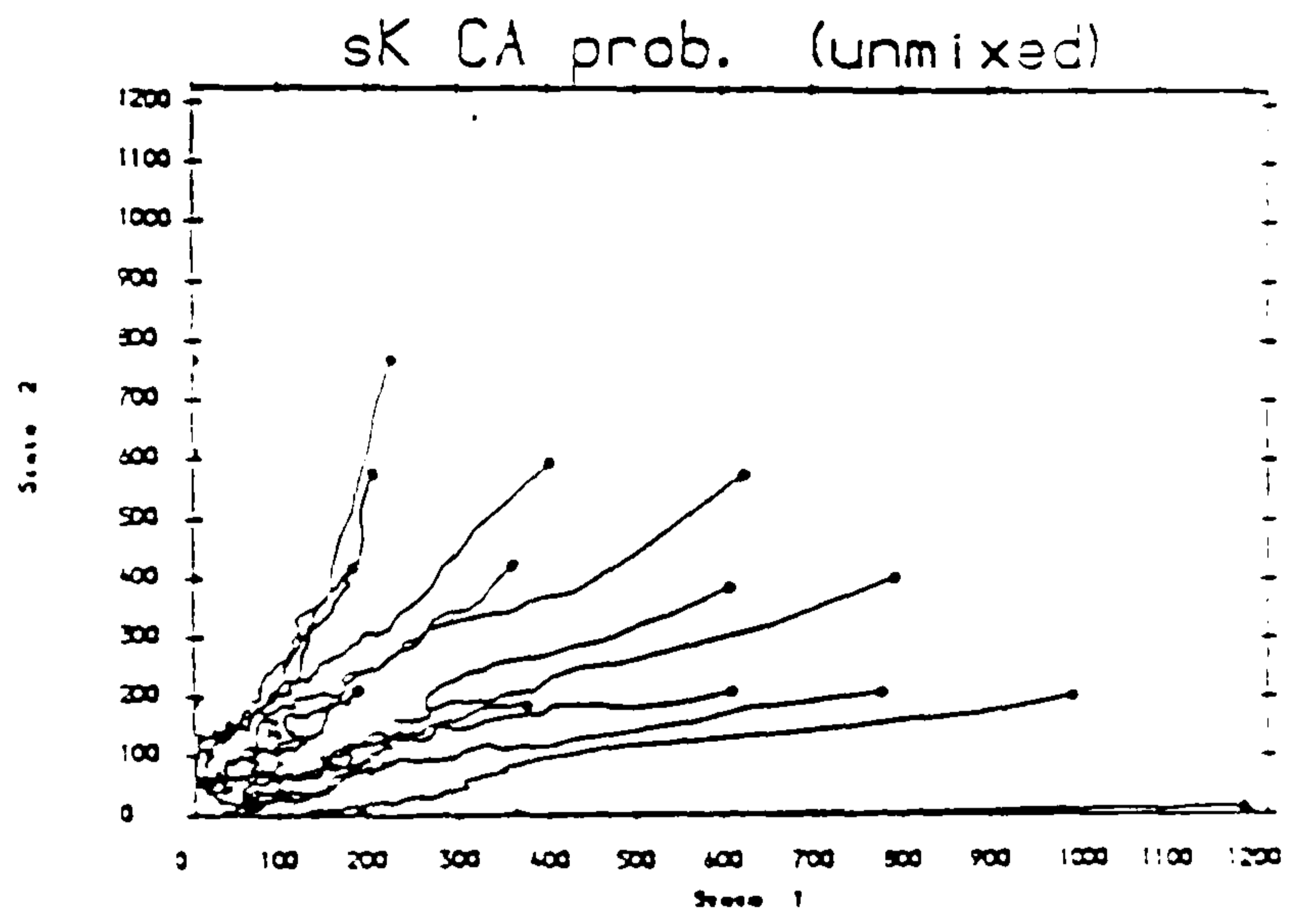
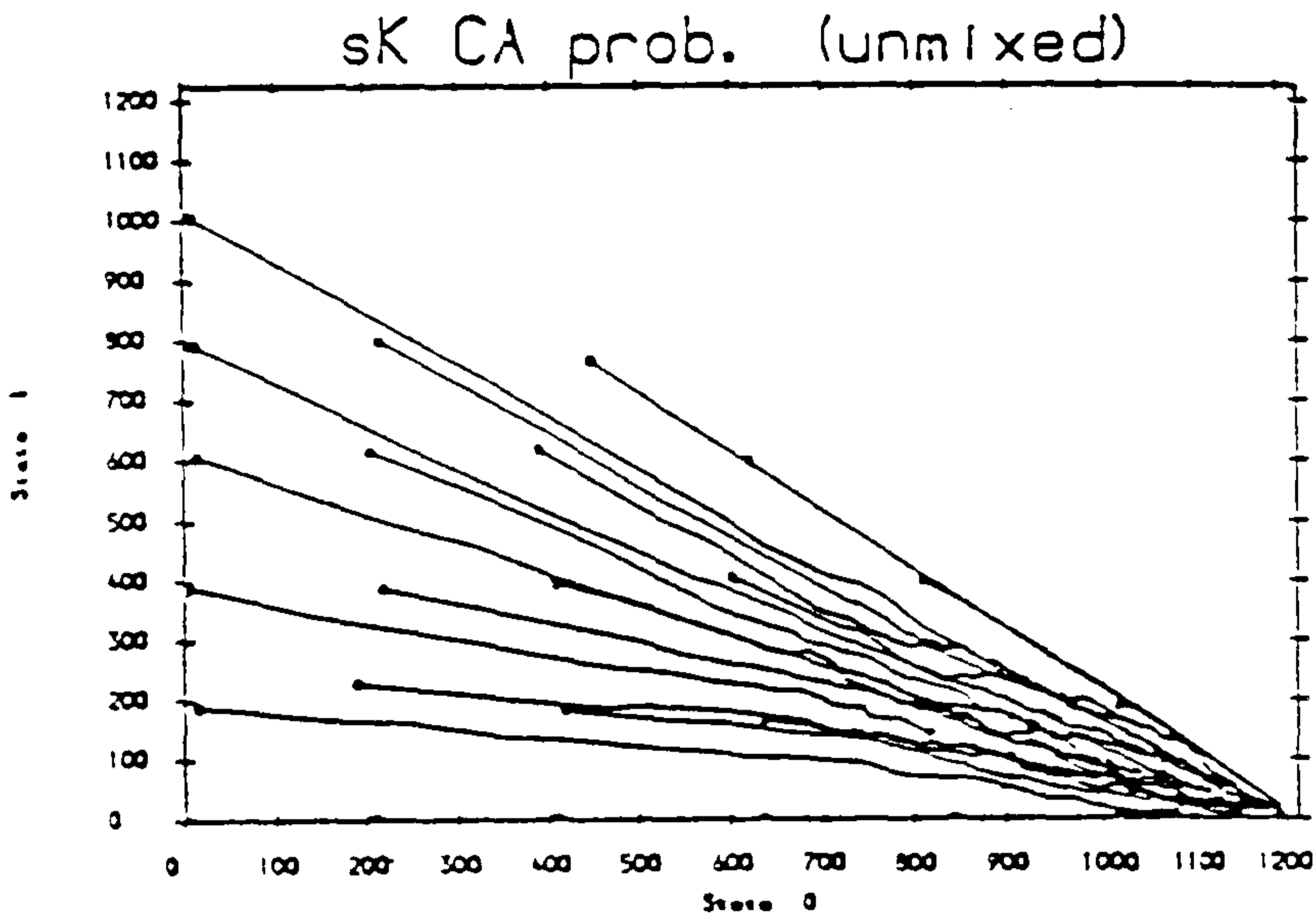




Figure 4.20: Class 3 phase diagrams.  
 (b) System E: Probabilistic.

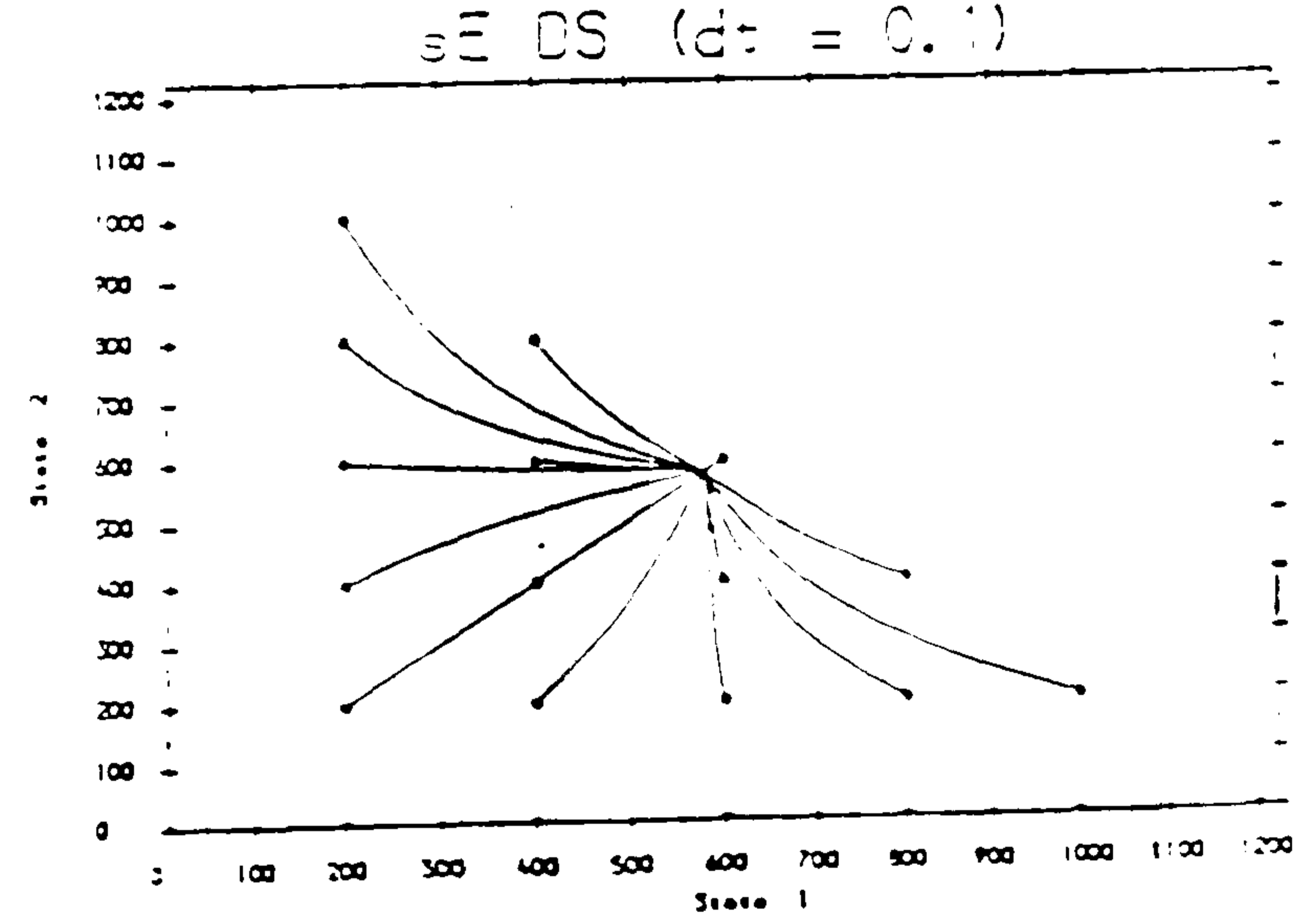
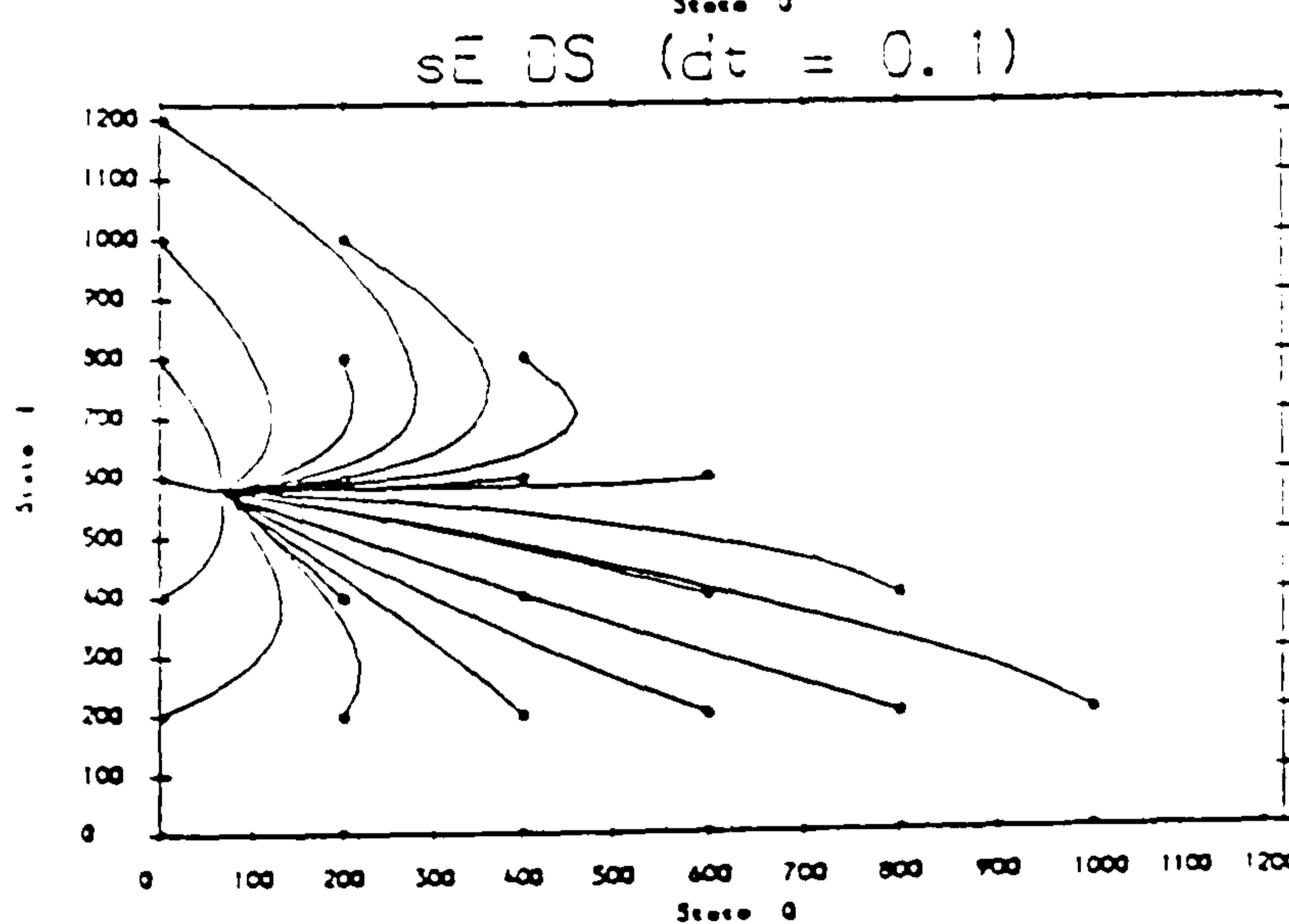
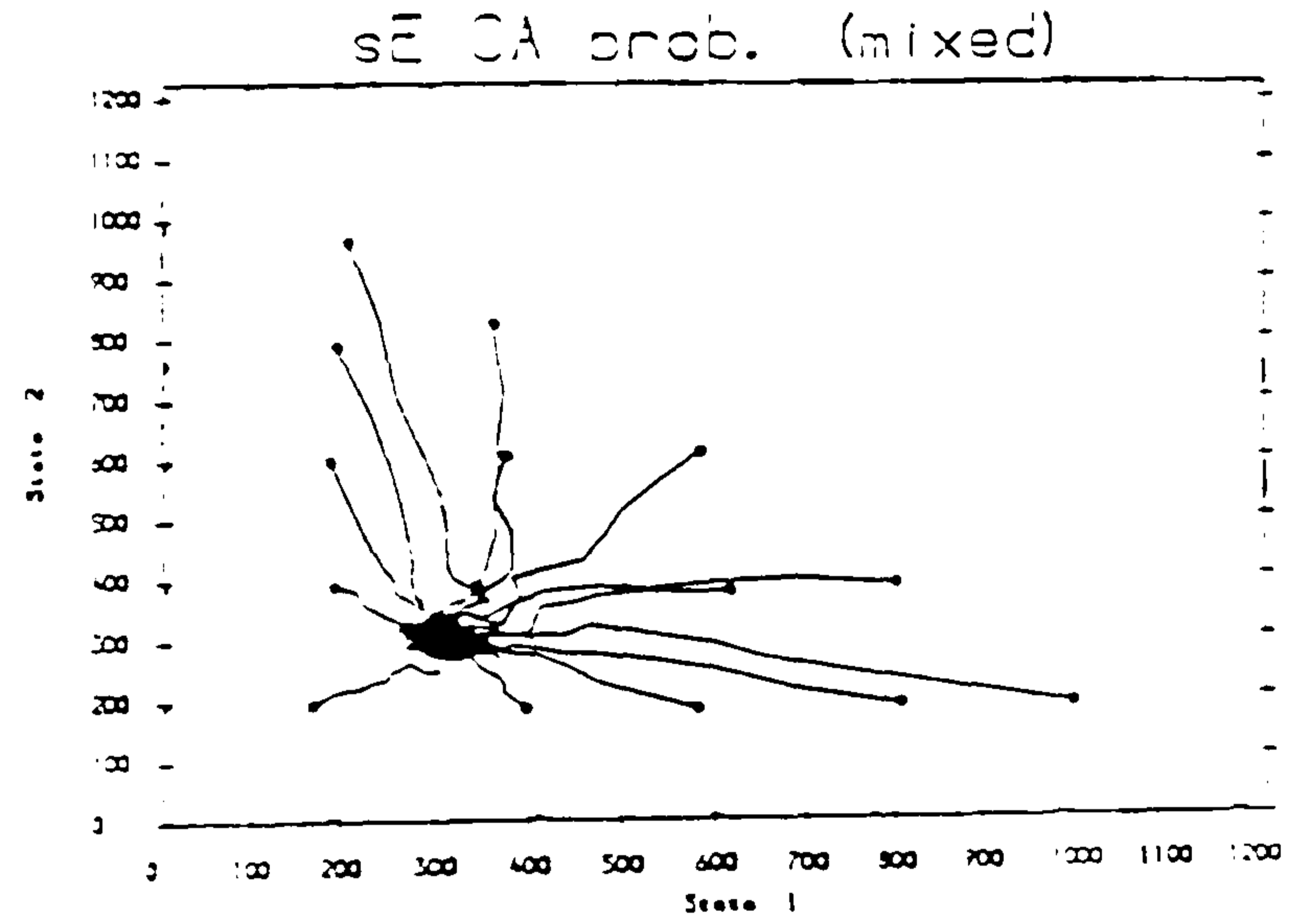
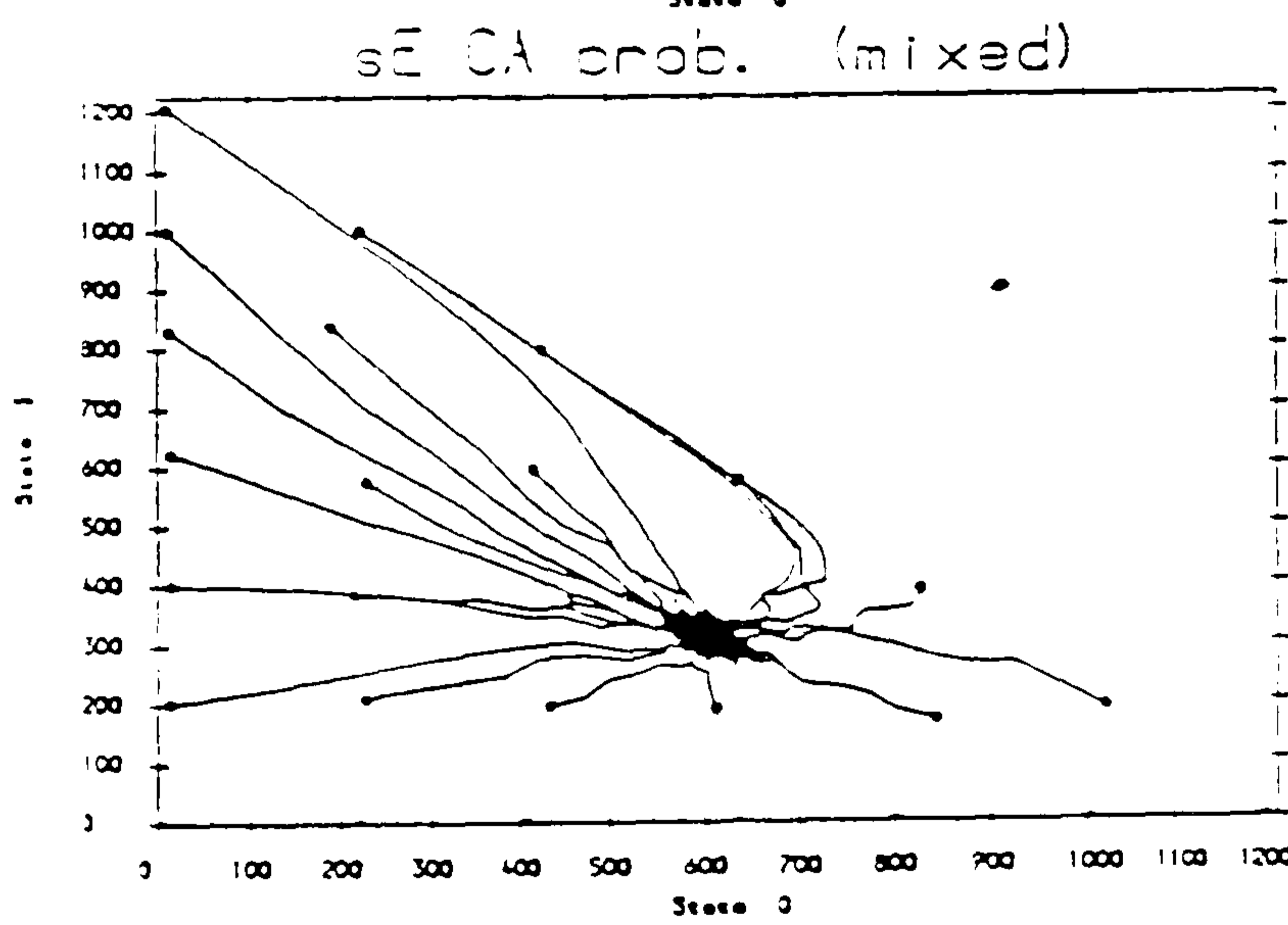
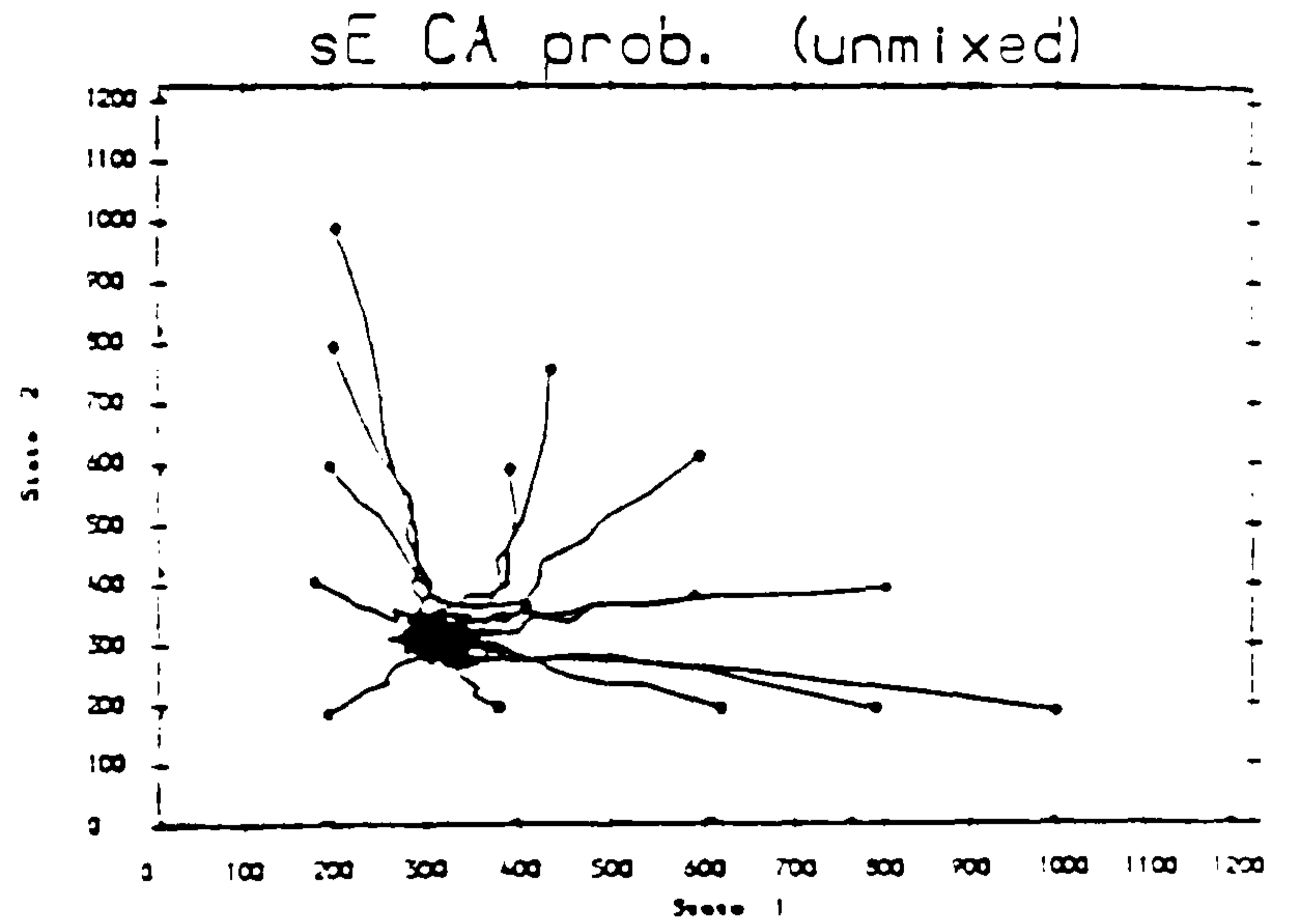
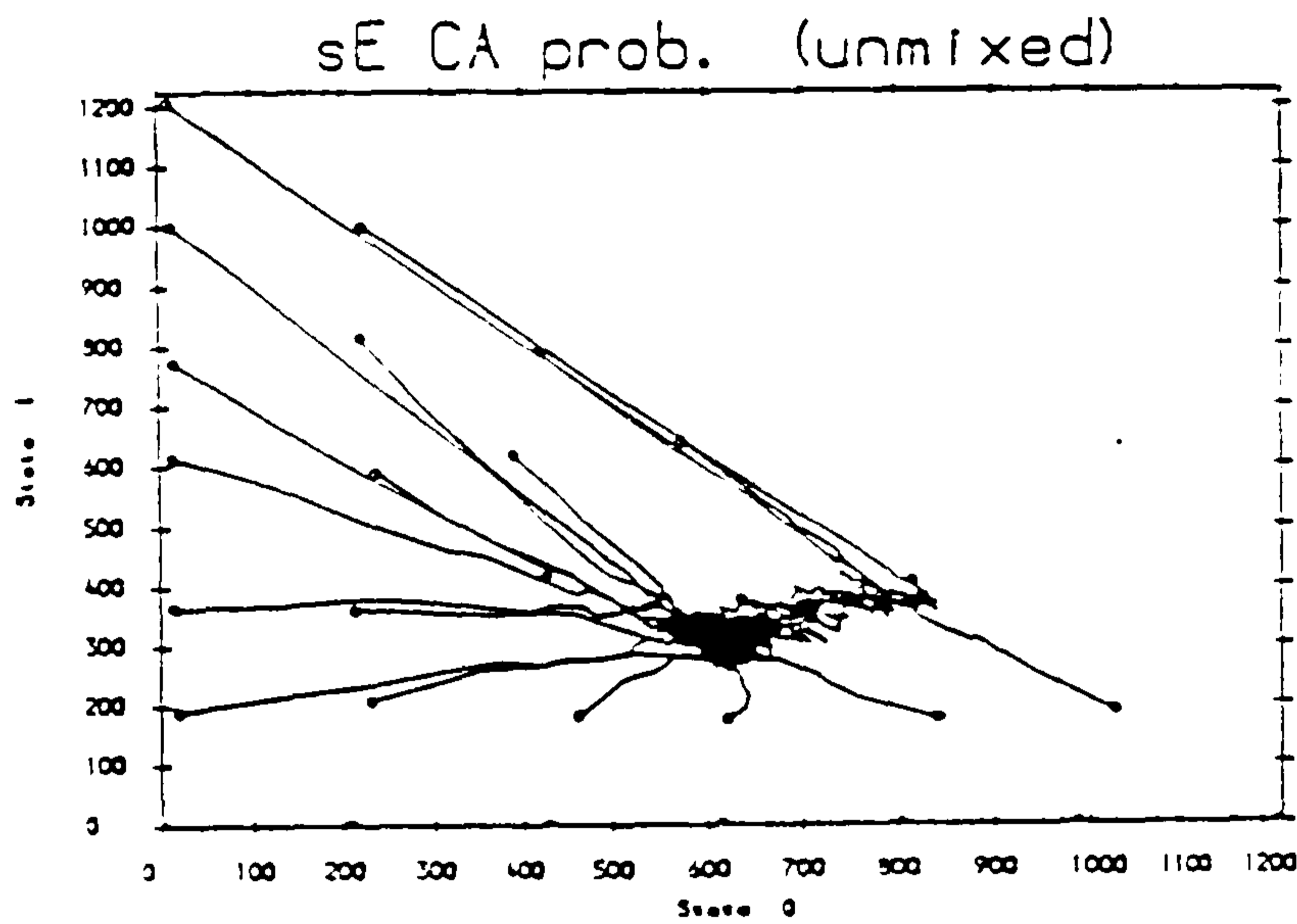


Figure 4.20: Class 3 phase diagrams.  
(c) System F: Probabilistic.

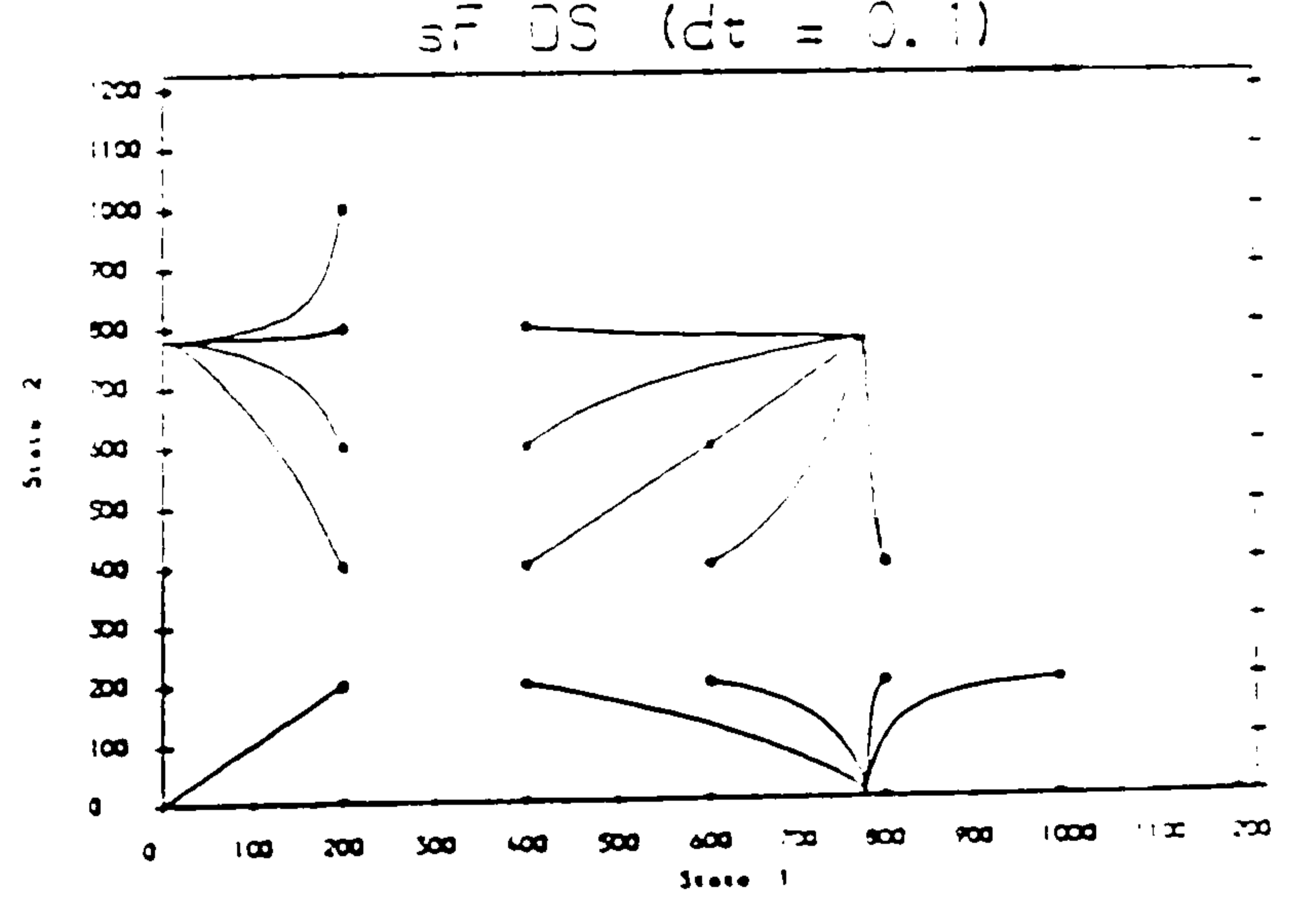
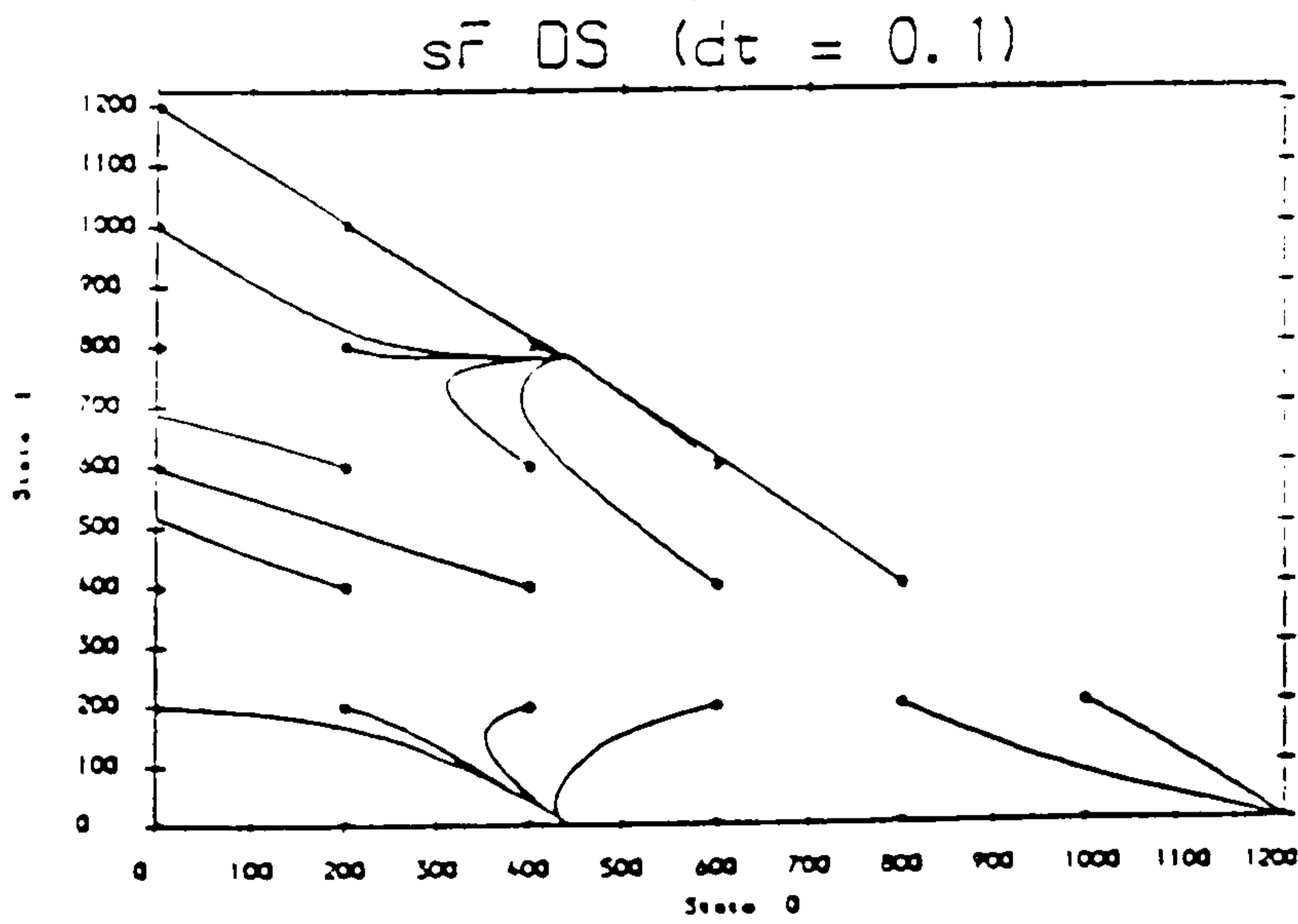
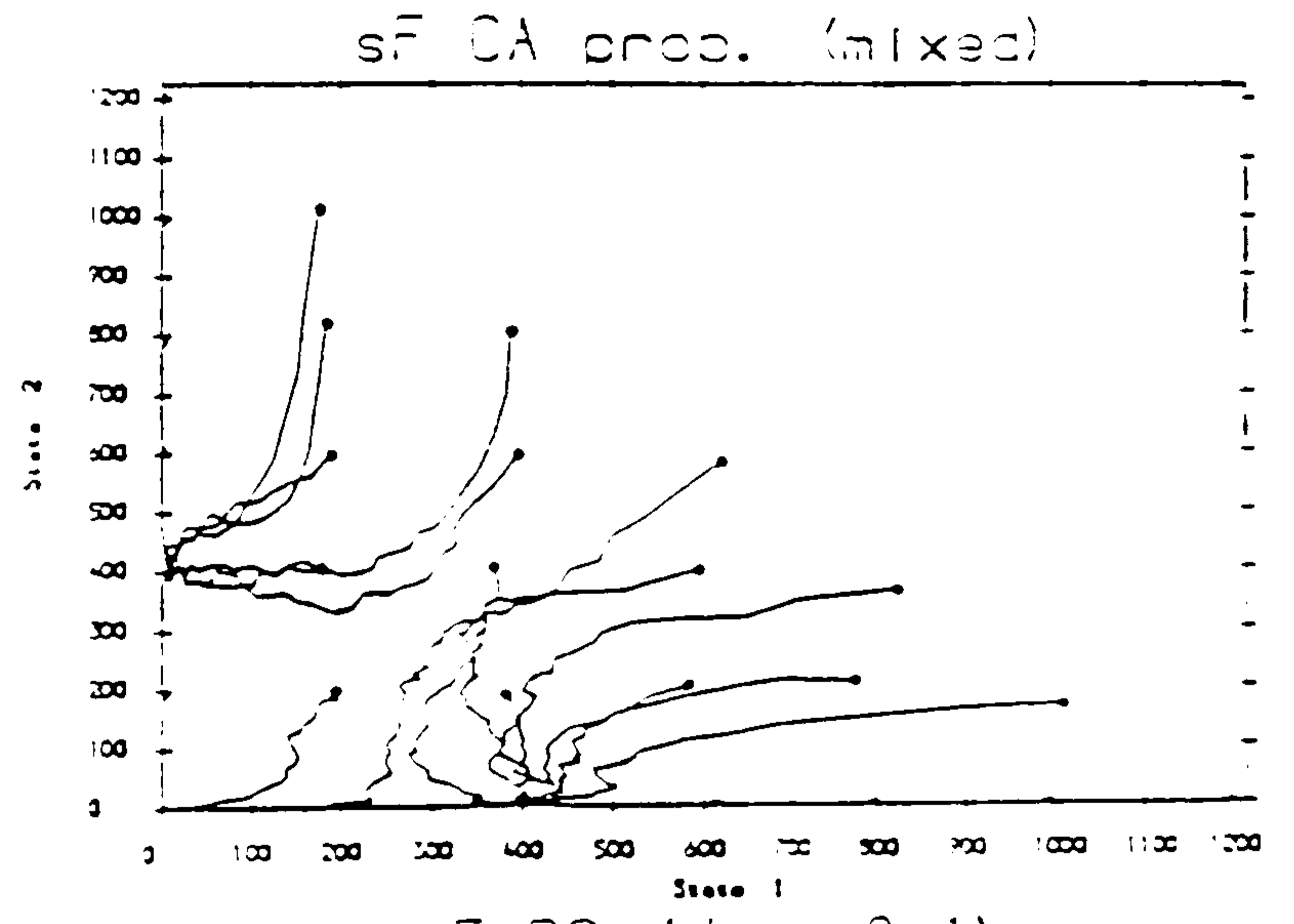
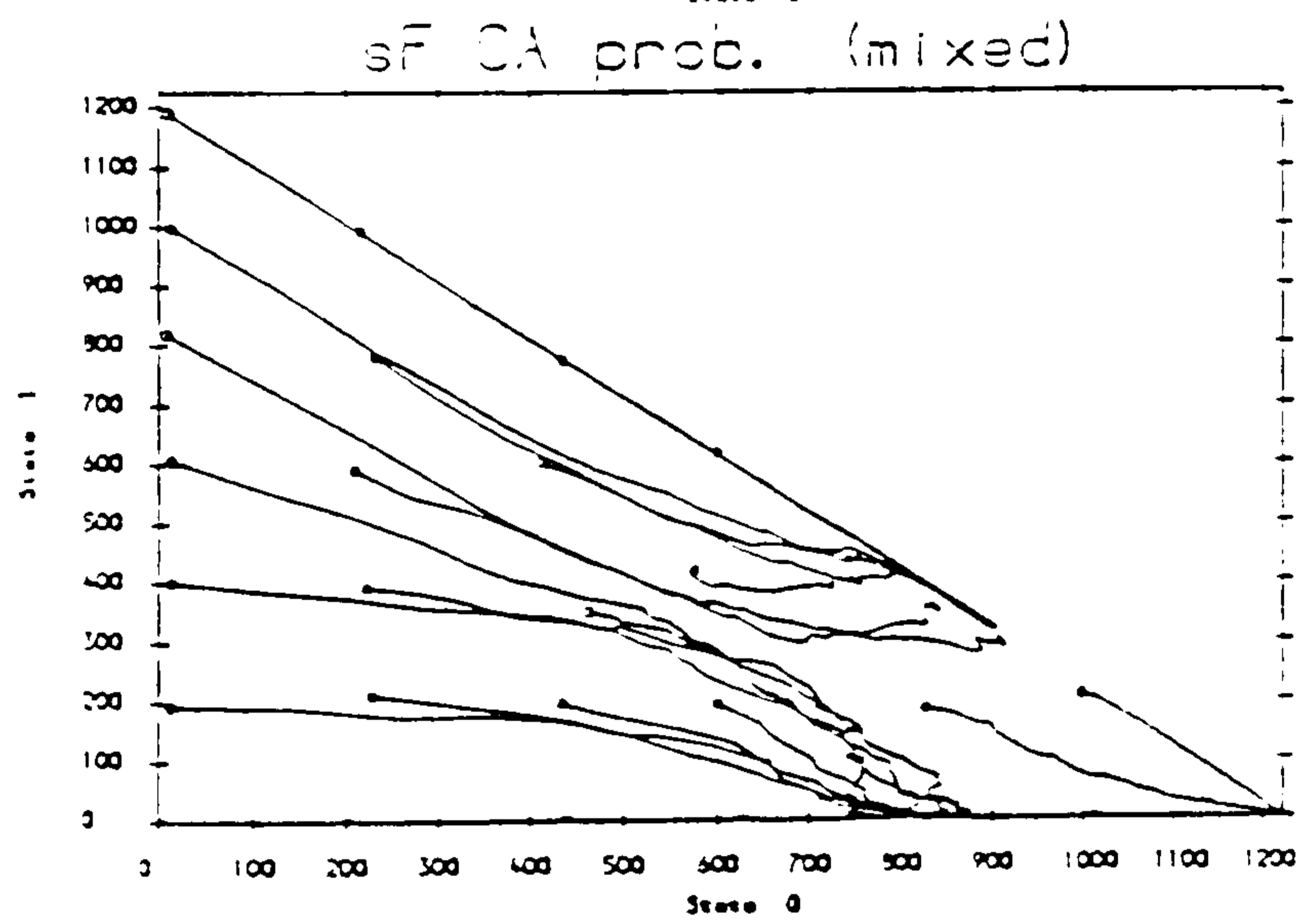
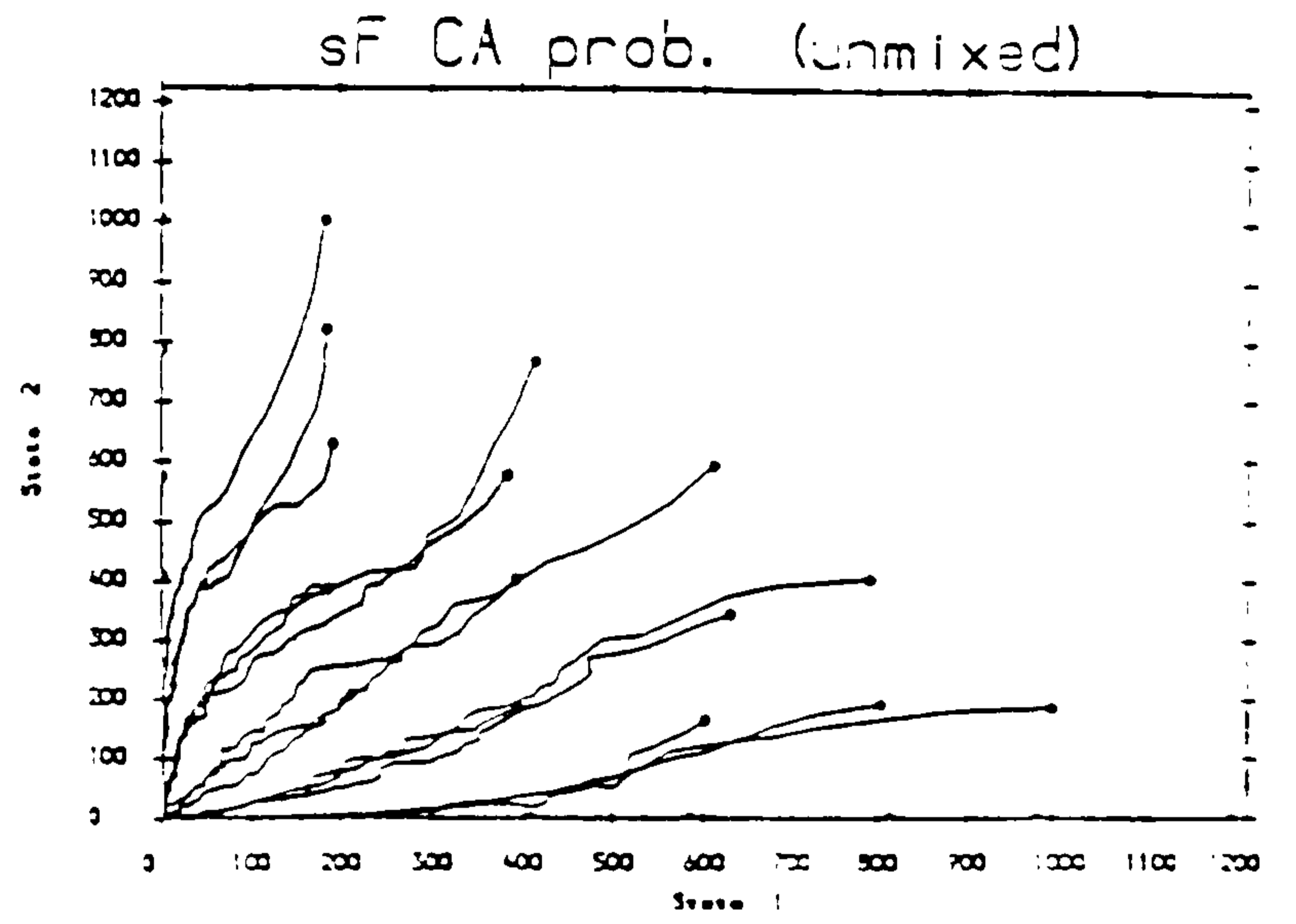
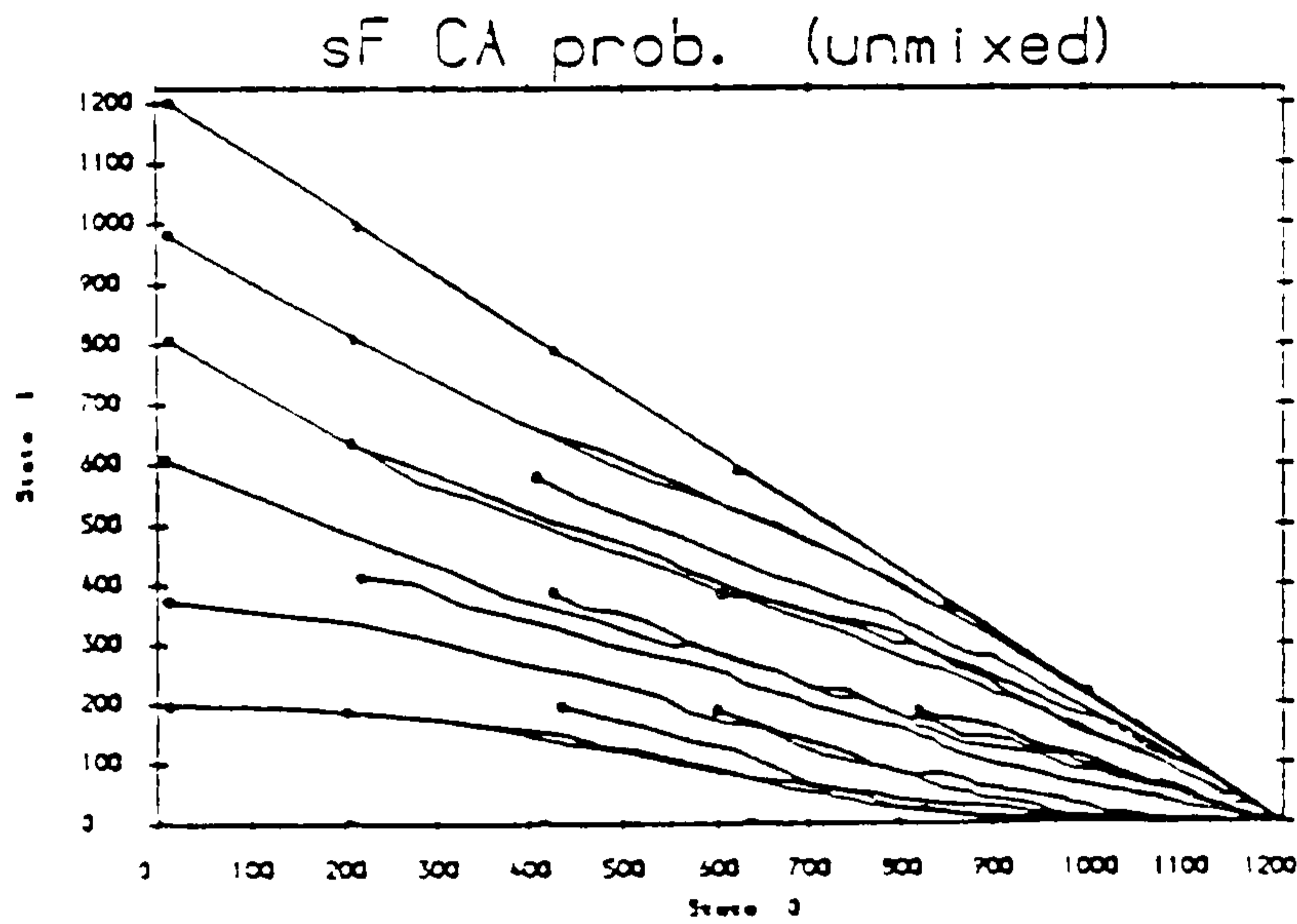
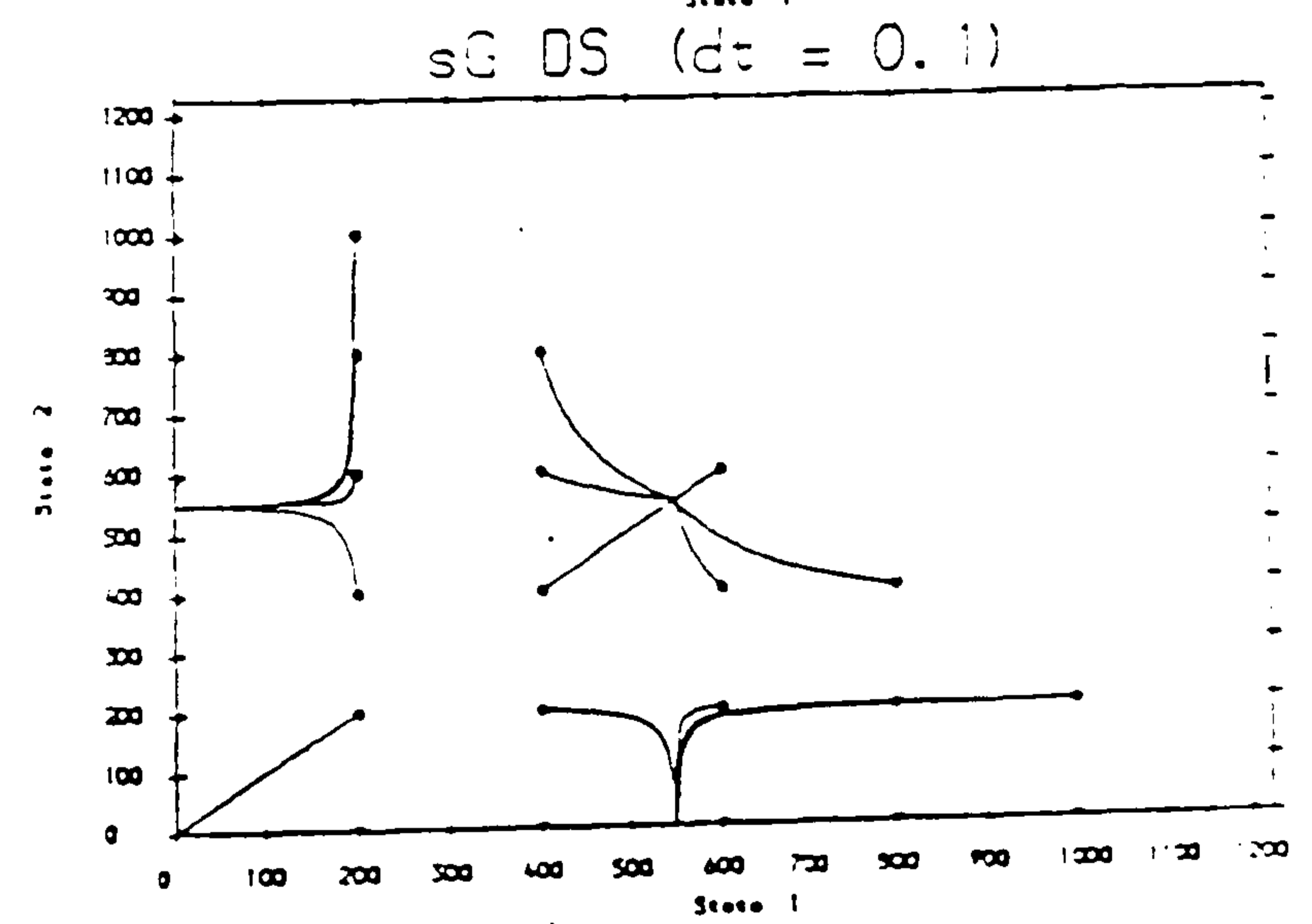
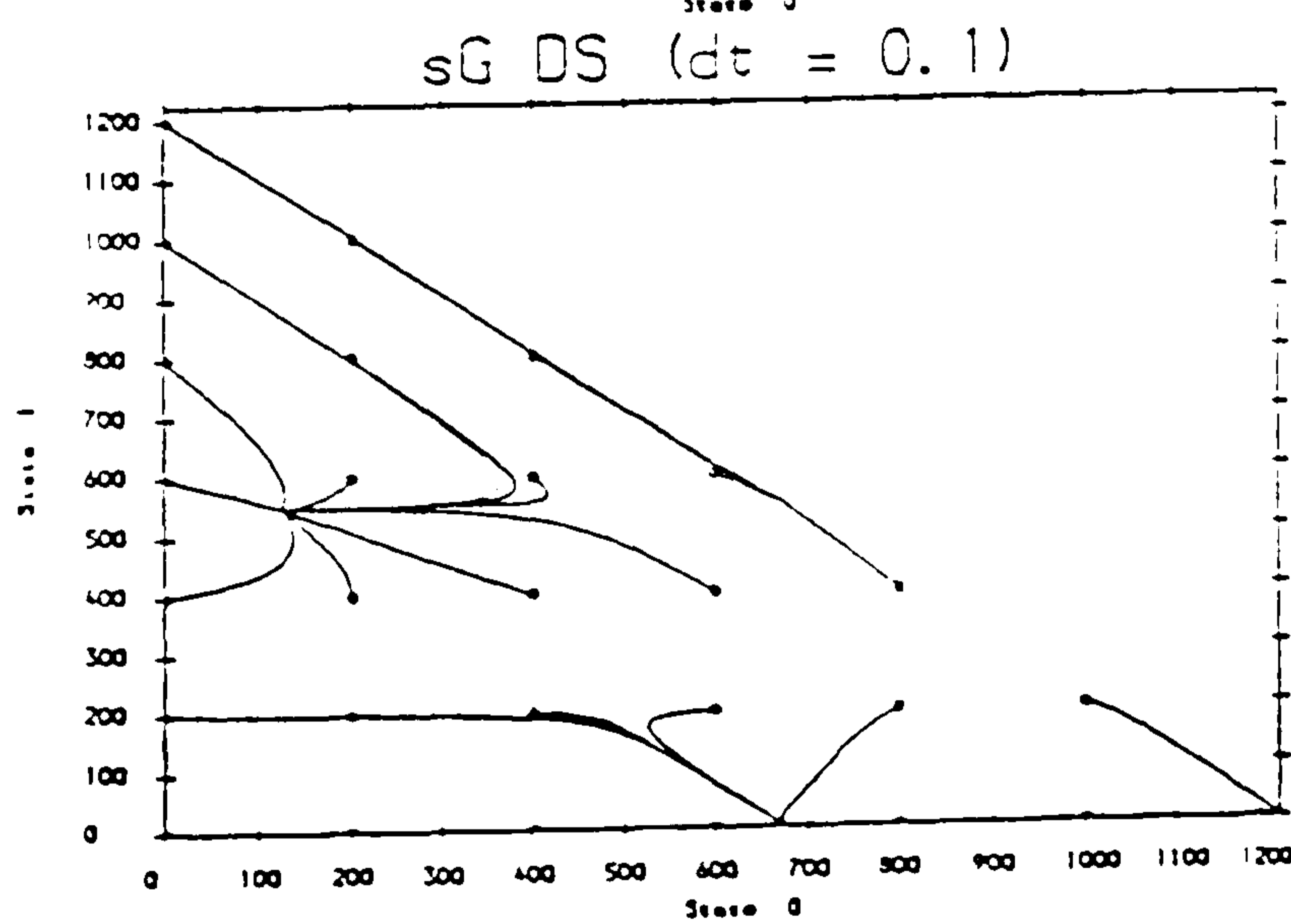
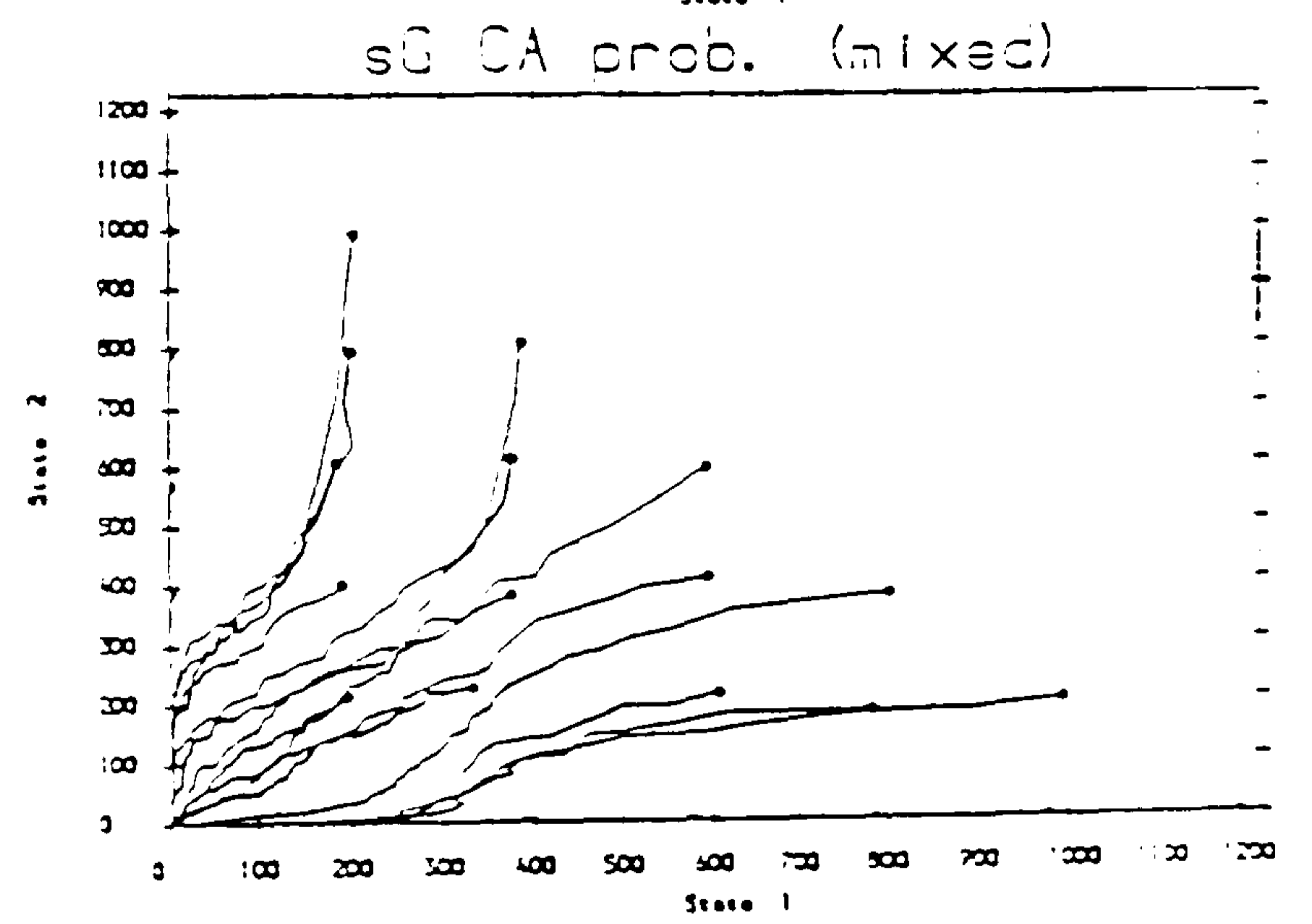
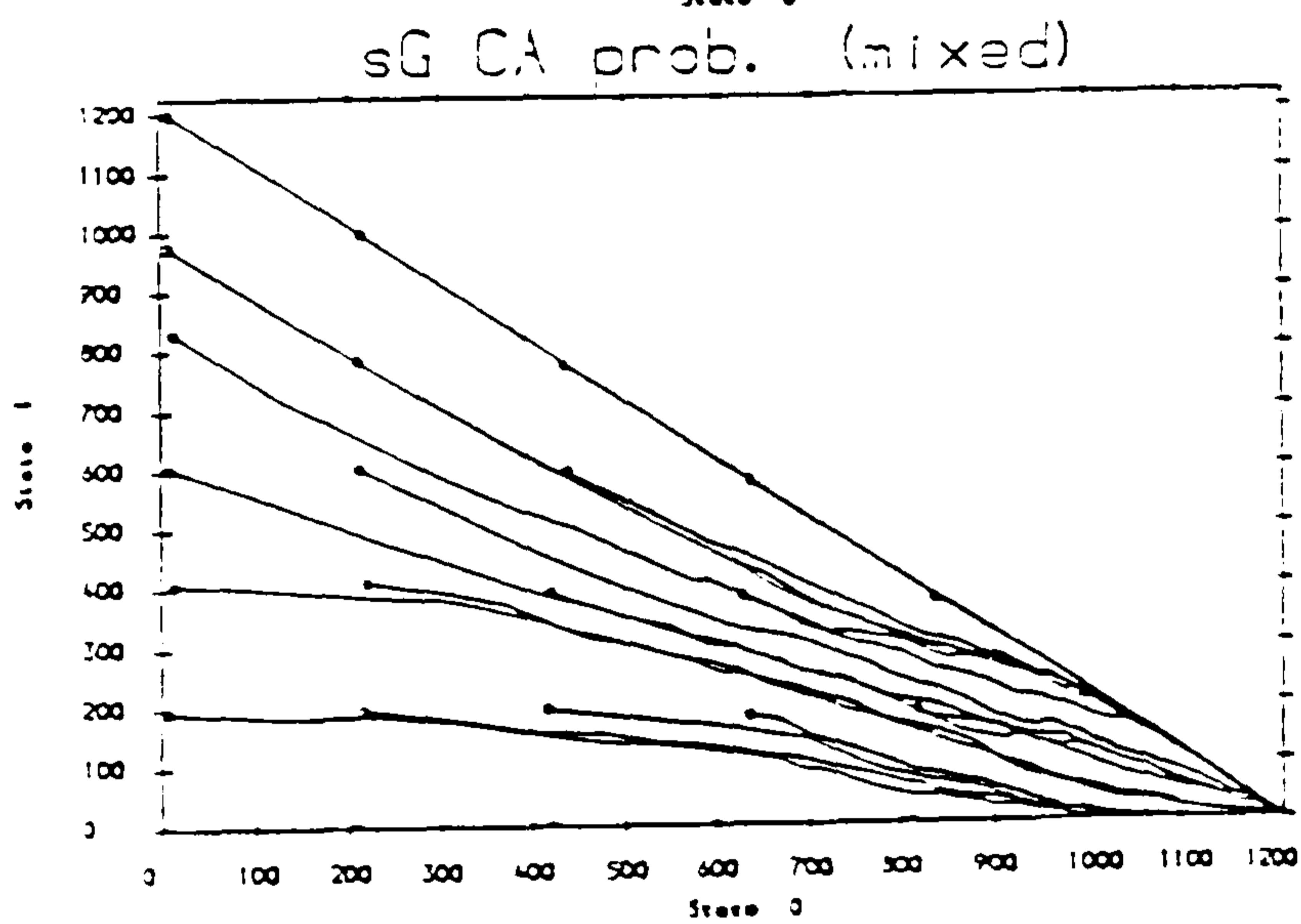
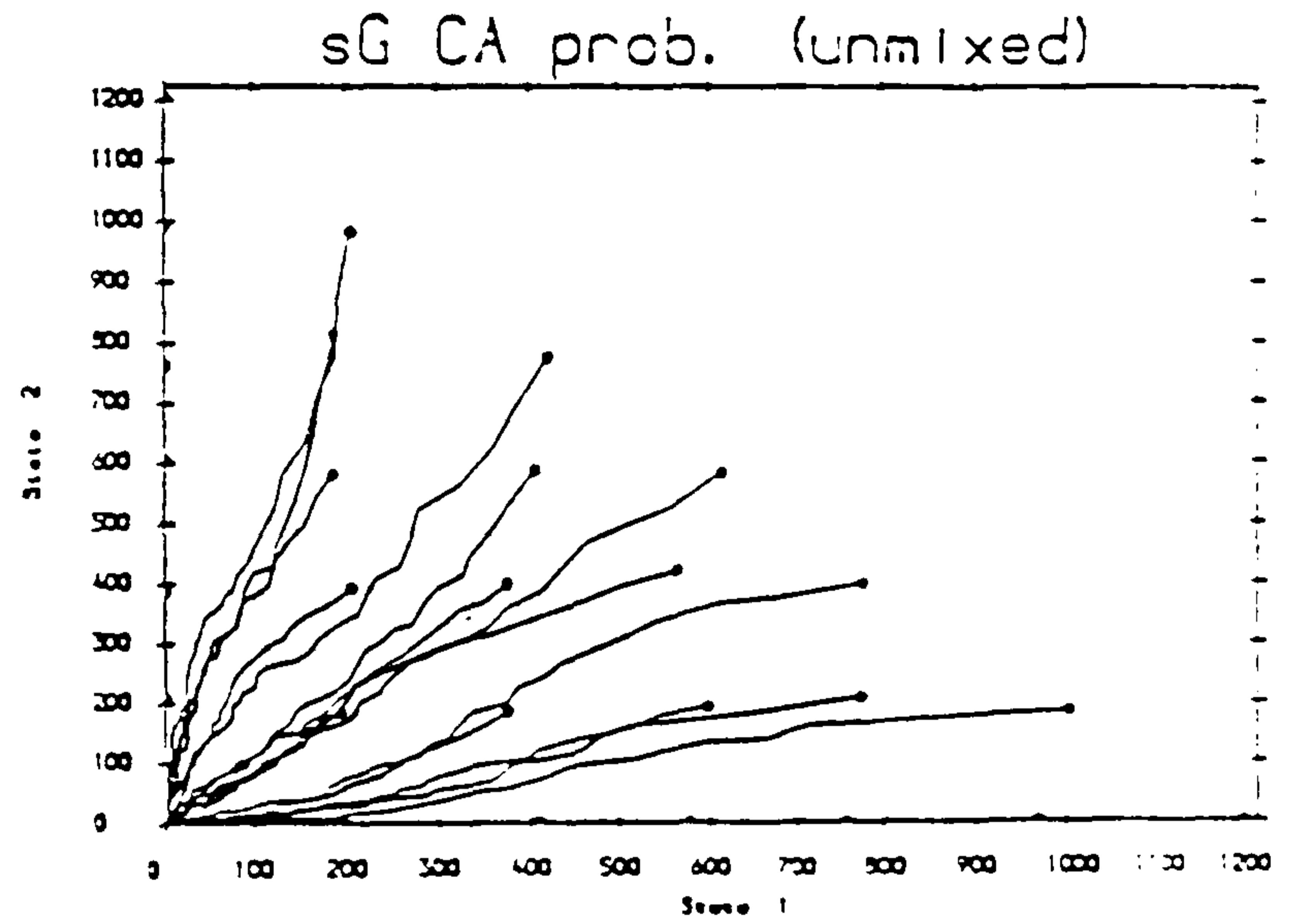
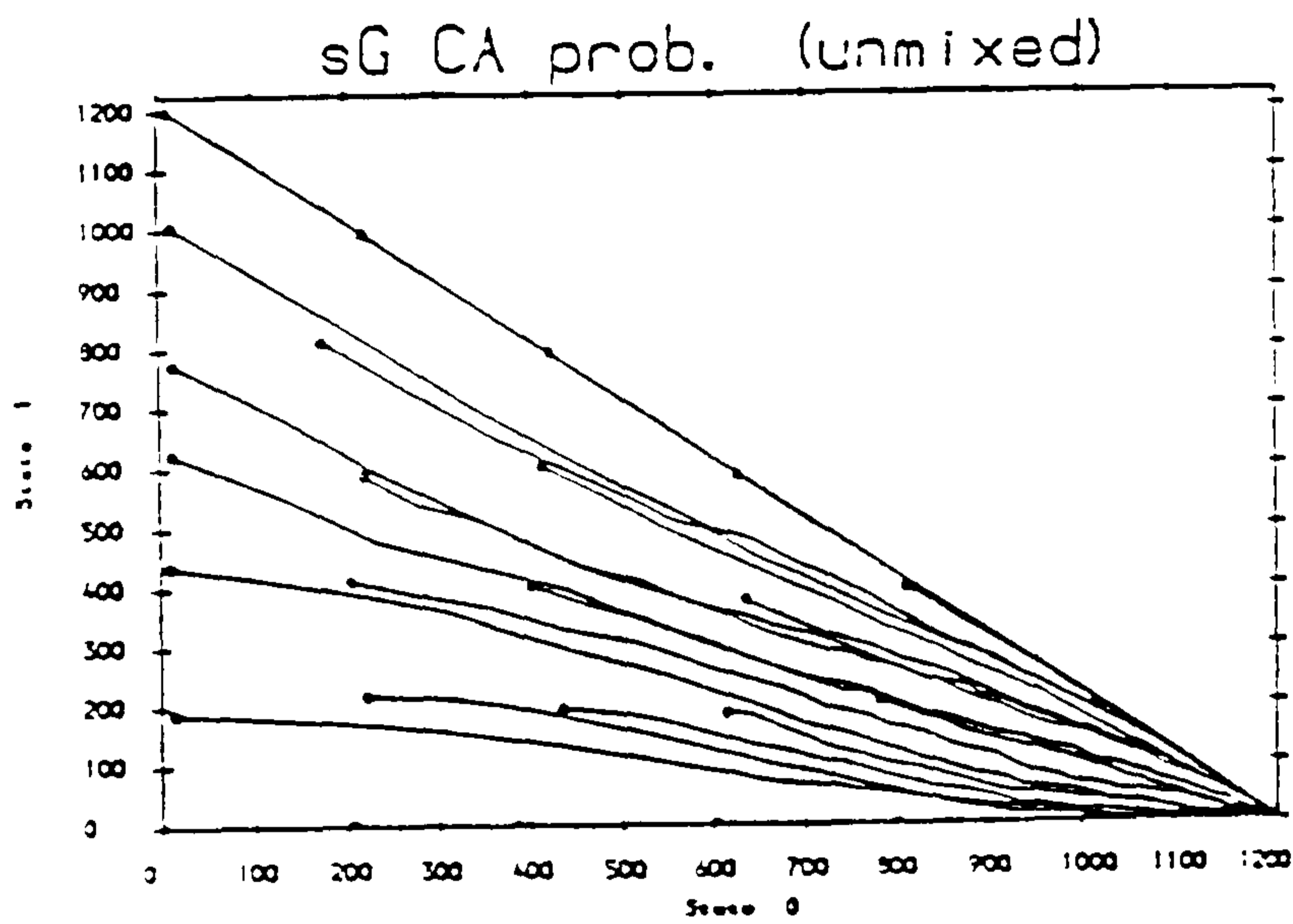


Figure 4.20: Class 3 phase diagrams.  
 (d) System G: Probabilistic.





which was resolved in favour of state  $s_2$  resulting in a highly biased phase diagram.

System E shows very strong attraction to a state where there are equal numbers of states  $s_1$  and  $s_2$ . This is because the states  $s_1$  and  $s_2$  are stable for population values from zero to  $N/2$  as the state transition diagram shows. For greater populations than  $N/2$  the population will gradually convert to  $s_0$  which will then convert to the other active population, thus equalising the populations. However the system does not settle to a limit point but continues to oscillate in a chaotic manner. The reason for the chaotic oscillations is that the rule set is complex enough to contain possible lattice conflicts. Thus for certain initial configurations there is no state of the lattice which the CA can reach which is stable under the actions of the rule set. With a relatively small, finite lattice the number of conflicts is small and the size of the chaotic region is consequently small. However, as the lattice size is increased the possible conflicts and the length of time to resolve these will become greater. The mixed grid has a larger chaotic region because the mixing interferes with the stabilisation of the lattice configuration. It also prevents states  $s_1$  and  $s_2$  from eliminating  $s_0$  completely. The DS exhibits oscillations also but in this case it is due to the rules leading to very high rates of change of the state populations. However, there is a connection between the behaviour of the DS and the CA, because the conditions for stability of the CA are so narrow this corresponds to a very small region of the phase space where the rate of change of the populations is zero. Outside this region the rates of change are very high and consequently with a  $\Delta t=1$  the system can never settle to the limiting region but continually overshoots and so oscillates about it. The DS also exhibits another property which the CA cannot, it occupies points on the phase diagram where one of the populations (in this case  $s_0$ ) is negative. This is because the DS is only constrained to have the three populations summing to  $N$ , the CA however cannot express negative populations and so does not show this behaviour. With the probabilistically

evolved CA we see a large shift in the position of the limiting region corresponding to much lower populations of  $s_1$  and  $s_2$ . This is due to the probabilistic evolution impering the ability of states  $s_1$  and  $s_2$  to survive against the transition to  $s_0$ . If the limiting region is a function of a set of narrow conditions, that is the probability function of the conditions is very narrow, then an unconditional transition such as  $s_x \rightarrow s_0$  will act more often as the rate of evolution of the system is reduced. This is because the system will spend more time outside the limiting region where the narrow conditions do not apply and so the default transition will occur instead giving this a greater effect than for the normal system. The DS does not exhibit this and simply evolves smoothly to the limiting region which has now become a limit point. This is because the rate of change of the populations will be ten times less and thus the oscillations will be ten times less.

System F has very different behaviour with the active states competing for domination of the lattice. If we examine the state transition diagram we can see that this is because they survive best when their populations are greater than  $N/2$  so any slight imbalance favours the larger population over the smaller and the latter eventually swamps the lattice. The mixed CA and DS are very similar to the unmixed CA except that the DS exhibits a limit point outside the phase diagram bounds of the CA as for system E. Once again we see that the probabilistic evolution has destroyed the ability of the active states to stabilise at non-zero population values on the unmixed lattice. However, although it reduces the stable population values on the mixed grid, the active states do survive at non-zero population values. We can understand this as follows, with pure probabilistic evolution we are fundamentally altering the type of organised behaviour which the CA can exhibit. Furthermore because of the complex distribution of ages of cells we obtain, it becomes very difficult for the CA to exhibit any organised behaviour and so we usually obtain a completely quiescent limit state. The probabilistic DS is once again simply a smoother



version of the original.

Finally system G, as noted earlier contains a rule conflict similar to system K and so we see that state  $s_2$  survives preferentially. Without this bias we would expect similar behaviour to system F. The active states survive optimally for populations of  $N/2$  down to  $N/4$  and so if one of the states has a population below this limit the other will grow at a greater rate and eventually swamp the system. The mixed system is similar except that the bias is totally effective, and the DS is again symmetrical, showing no bias. It does however allow a limit point where the active populations are equal. The probabilistic evolution as we expect destroys the survival properties of the active states, both for mixed and unmixed lattices in this case. The DS is again simply a smooth version of the original.

### Summary

We have looked at the application of our statistical mechanical approach to the various classes of CA. We have seen that the state population equations do indeed model the actual behaviour of the CA state populations for both the deterministic and probabilistic cases. As we expected the dynamical equations agree best with the mixed CA since the self-organising behaviour of the unmixed lattice causes deviations from a homogeneous distribution of states. We can therefore apply the DS theory approaches used for the threshold CA to all the systems in order to obtain information on the limiting behaviour. However, we have also seen that deviations between the behaviour of the CA and the DS can occur, in particular limiting points/regions may be shifted and entirely new limiting regions can appear in the DS phase space. We believe that in the limit of  $\Delta t \rightarrow 0$  the behaviour of the two systems will converge. For class 3 systems substantial differences



occur in certain cases, and these need further investigation. However, we feel that these deviations may be connected with the very high rates of change of the state populations for the systems we have studied. This may lead to the DS behaving unstably, but the distributed nature of the CA may “damp” the instability.

We should note that for the CA phase diagrams the separation of the various domains of attraction of the limit points will not be a clear one. Since there will be many global configurations which will correspond to the same point on the phase diagram and the limit point which they evolve towards will in general depend on which one of these configurations is actually present on the lattice as the initial global state.

## 5. Simulating Dynamical Systems with Cellular Automata

It has already been noted in the previous Chapters that the behaviour of CA can be considered as the competition and/or cooperation of the various state populations for space on the lattice. This suggests that there is a connection between the field of population genetics and CA. We shall therefore consider some standard population genetic systems and attempt to cast these in terms of our general CA scheme.

One of the simplest population genetic models is the the Lotka-Volterra Predator-Prey system[90-92]. This system was originally developed independently by A.J.Lotka and V.Volterra as a simple model of the cyclic behaviour found in classical predator-prey systems such as the hare and lynx. The system can be defined as follows,

$$\dot{A} = k_1A - k_2AB \quad 5.1$$

$$\dot{B} = k_3AB - k_4B \quad 5.2$$

Where  $A$  is the prey population and  $B$  is the predator population. The prey population increases exponentially in the absence of predators at a rate determined by  $k_1$ . The predators have the effect of reducing this rate of growth by an amount determined by  $k_2$ . Predators multiply at a rate which depends on the number of prey,  $k_3A$ . Finally in the absence of prey the predators die out at an exponential rate  $k_4$ .

If we now consider what is happening physically, this will give us an insight into how to cast this system into a CA framework. The prey increase at a rate proportional to the number present, we could therefore model this by changing quiescent cells into the state representing prey with a probability proportional to the number of prey cells in its neighbourhood. If we assign state  $s_1$  to prey and  $s_2$  to predators we obtain the following set of rules,

$$s_0 \& \{1s_1\} \rightarrow s_1 \mid \text{Pr} = c_1 1/r \quad 5.3$$

$$s_0 \& \{2s_1\} \rightarrow s_1 \mid \text{Pr} = c_1 2/r \quad 5.4$$

...

$$s_0 \& \{cs_1\} \rightarrow s_1 \mid \text{Pr} = c_1 c/r \quad 5.5$$

...

$$s_0 \& \{rs_1\} \rightarrow s_1 \mid \text{Pr} = c_1 r/r \quad 5.6$$

The predators decrease at a rate proportional to the number present which will be equivalent to each predator cell having a given constant probability of changing to the quiescent state.

$$s_2 \rightarrow s_0 \mid \text{Pr} = c_4 \quad 5.7$$

If we now assume the constants  $k_2$  and  $k_3$  are equal then we can make the analogy that prey cells are changing to predator cells at a rate proportional to the number of predator cells in their neighbourhood.

$$s_1 \& \{1s_2\} \rightarrow s_2 \mid \text{Pr} = c_2 1/r \quad 5.8$$

$$s_1 \& \{2s_2\} \rightarrow s_2 \mid \text{Pr} = c_2 2/r \quad 5.9$$

...

$$s_1 \& \{cs_2\} \rightarrow s_2 \mid \text{Pr} = c_2 c/r \quad 5.10$$

...

$$s_1 \& \{rs_2\} \rightarrow s_2 \mid \text{Pr} = c_2 r/r \quad 5.11$$

The DS which we obtain from these rules is as follows,

$$\dot{n}_1 = \left[ \sum_{c=0}^r c_1 \frac{c}{r} P(c, n_1) \right] n_0 - \left[ \sum_{c=0}^r c_2 \frac{c}{r} P(c, n_2) \right] n_1 \quad 5.12$$

$$\dot{n}_2 = \left[ \sum_{c=0}^r c_2 \frac{c}{r} P(c, n_2) \right] n_1 - [c_4] n_2 \quad 5.13$$

We are not particularly concerned with the changes in the quiescent state population



since this just represents empty space. However we should note that the initial number of quiescent cells should be much greater than either the predator or prey cells so that space limitation does not have an effect on the evolution of the system. This is because space limitation is not taken into account in the Lotka-Volterra model. If we now compare the DS obtained from the CA with the actual model we can see that we require the expressions in square brackets to vary linearly with the population size on which the probabilities depend. The variation of this function is shown in Figure 5.1.

As can be seen the function does indeed vary precisely linearly with the population size. The small errors involved are shown in Figure 5.2. We now need to ask what the constant of proportionality is for this linear variation. To do this we must consider the two extremal values of the population size. If there are no cells in the state we are considering then the probability of finding any number of cells other than zero in this state will be zero and the probability of finding zero cells in this state will be one. The expression in square brackets then reduces to zero. If all cells on the lattice are in the state we are considering then the only non-zero count probability will be for  $c=r$  and this will be one. In this case the expression becomes  $c_i; i=1,2,4$ . We can therefore simplify the expressions to  $c_i n/N; i=1,2,4$  and if we compare the dynamical system obtained to the original Lotka-Volterra system we obtain the following equivalencies between the constants,

$$k_1 = c_1 n_0 / N \quad 5.14$$

$$k_2 = k_3 = c_2 / N \quad 5.15$$

$$k_4 = c_4 \quad 5.16$$

We can improve the CA model by not making the assumption that  $k_2=k_3$ . If we do this we can no longer simply convert prey cells to predator cells. Instead if we consider what is happening in the real situation, predators will be killing prey which is analogous to

Figure 5.1: Variation of CA population genetic function.

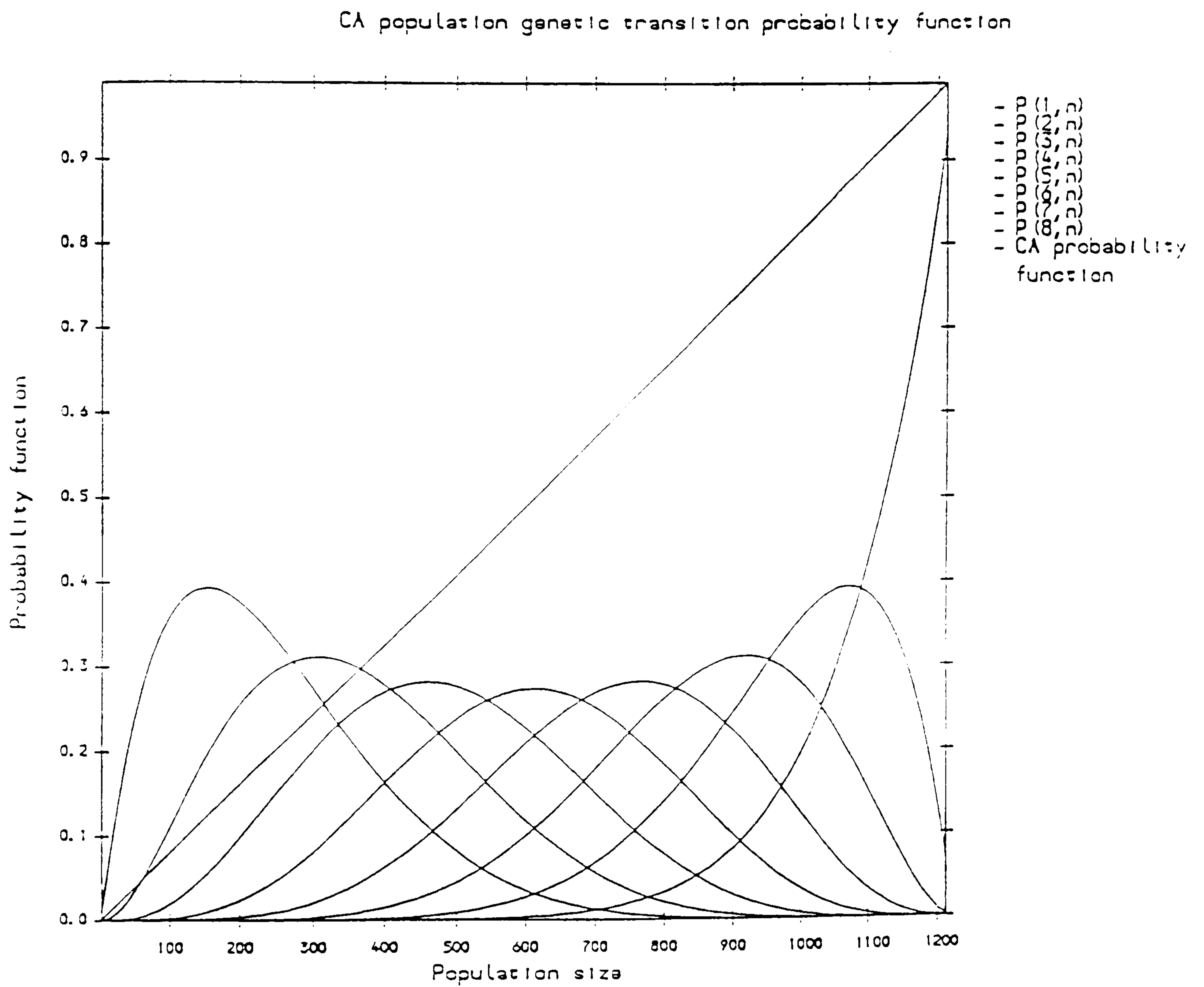
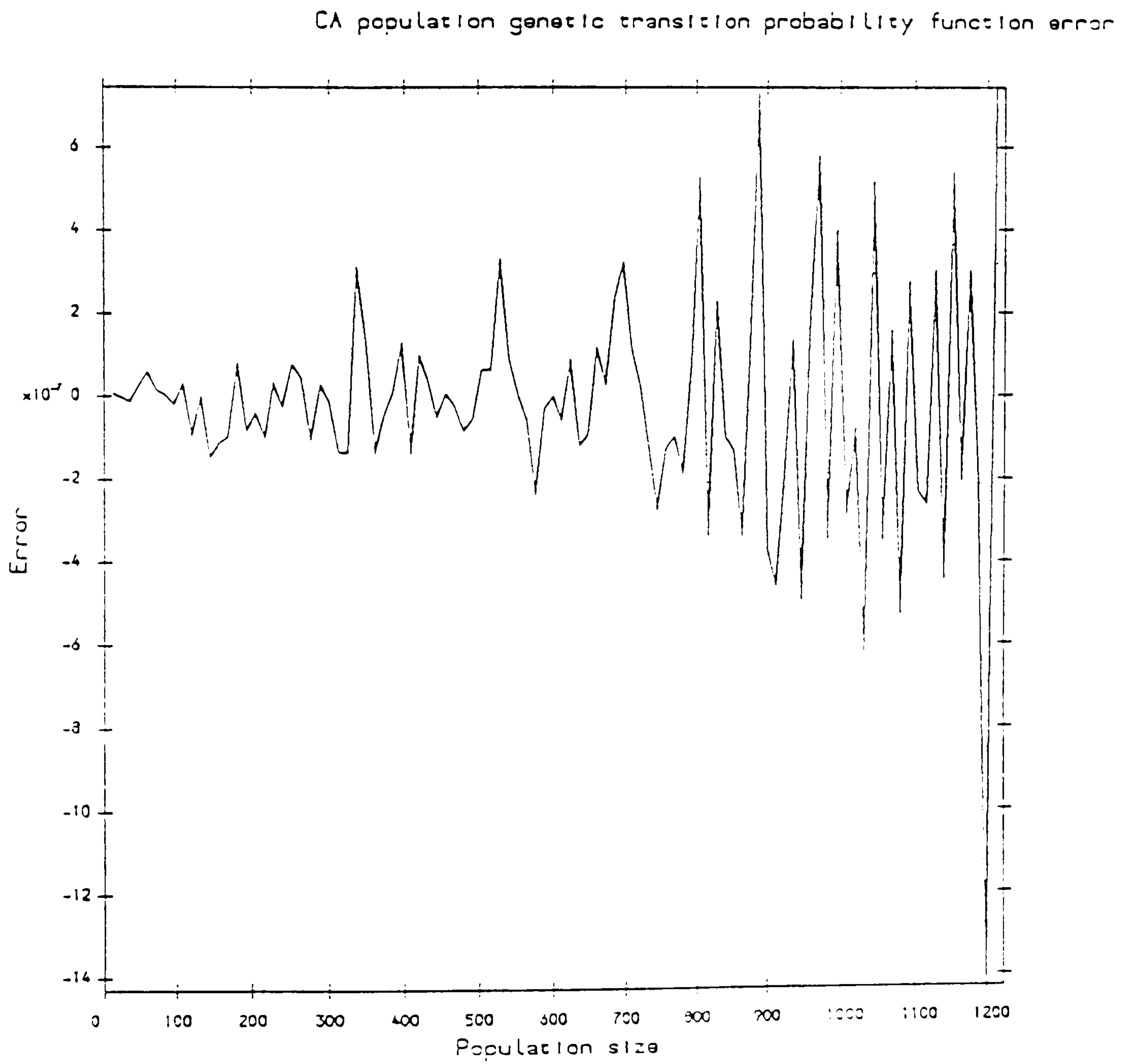


Figure 5.2: Error in CA population genetic function.





prey cells becoming empty or quiescent. Furthermore, the rate of increase of predators is related to the number of prey available, this suggests we should change quiescent cells to predator cells based on the number of prey in the neighbourhood. We can now write down an improved rule set,

$$s_0 \& \{c s_1\} \rightarrow s_1 \mid \text{Pr} = c_1 c / r \quad 5.17$$

$$s_1 \& \{c s_2\} \rightarrow s_0 \mid \text{Pr} = c_2 c / r \quad 5.18$$

$$s_0 \& \{c s_1\} \rightarrow s_2 \mid \text{Pr} = c_3 c / r \quad 5.19$$

$$s_2 \rightarrow s_0 \mid \text{Pr} = c_4 \quad 5.20$$

These give us the following DS,

$$\dot{n}_1 = \left[ \sum_{c=0}^r c_1 \frac{c}{r} P(c, n_1) \right] n_0 - \left[ \sum_{c=0}^r c_2 \frac{c}{r} P(c, n_2) \right] n_1 \quad 5.21$$

$$\dot{n}_2 = \left[ \sum_{c=0}^r c_3 \frac{c}{r} P(c, n_1) \right] n_0 - [c_4] n_2 \quad 5.22$$

Comparing equations 5.21 and 5.22 with 5.1 and 5.2 we see that the growth term of the predators for our CA does not have the correct functional dependence on the populations. This is because our transition rule simulation of predator growth depends only on the number of prey and the free space and not on the number of predators. In order to obtain the correct functional we can alter this transition rule so that it simulates the survival of predators as being proportional to the number of prey,

$$s_2 \& \{c s_1\} \rightarrow s_2 \mid \text{Pr} = c_3 c / r \quad 5.23$$

These give us the following DS,

$$\dot{n}_1 = \left[ \sum_{c=0}^r c_1 \frac{c}{r} P(c, n_1) \right] n_0 - \left[ \sum_{c=0}^r c_2 \frac{c}{r} P(c, n_2) \right] n_1 \quad 5.24$$

$$\dot{n}_2 = \left[ \sum_{c=0}^r c_2 \frac{c}{r} P(c, n_2) \right] n_1 + \left[ \sum_{c=0}^r c_3 \frac{c}{r} P(c, n_1) \right] n_2 - [c_4] n_2 \quad 5.25$$

Substituting the equivalent linear functions for the expressions in square brackets we obtain the following,

$$\dot{n}_1 = [c_1 n_0 / N] n_1 - [c_2 / N] n_1 n_2 \quad 5.26$$

$$\dot{n}_2 = [(c_2 + c_3) / N] n_1 n_2 - [c_4] n_2 \quad 5.27$$

These equations now have the same functional form as the original Lotka-Volterra system. An alternative method of obtaining the correct functional form would be alter the predator growth transition rule such that predators are created from quiescent cells with a condition which is dependent on both the number of prey and the number of predators. This however introduces complications which are discussed later in the Chapter when generalisation of this approach is considered. The example of the incorrect derivation of a CA model illustrates that care is needed in the conversion of dynamical systems to their equivalent CA representation.

The temporal evolution of the lattice for both the simplified and improved CA models are shown in Figures 5.3 and 5.4. The phase diagrams are shown in Figures 5.5 and 5.6.

The global behaviour of the CA and especially its equivalent DS are very similar to that for the true Lotka-Volterra DS. One important difference between the CA and both the CA DS and the true DS is the shape of the orbits. The CA orbits are flattened relative to those of the DS, this is because the equivalence equation between the CA prey growth rate constant  $c_1$  and the DS constant  $k_1$  involves the size of the quiescent population  $n_0$  which is of course not constant. In order to calculate a CA growth rate constant which is equivalent to that for the DS one must estimate the average quiescent population. The typical orbit is in fact approximately the correct shape, it is the orbits which begin at more extreme initial points which have the distorted shape which is to be expected. Note that trajectories which approach very close to the axes such that the population levels are



Figure 5.3: Lattice evolution for simplified Lotka-Volterra CA.

State 1    State 2

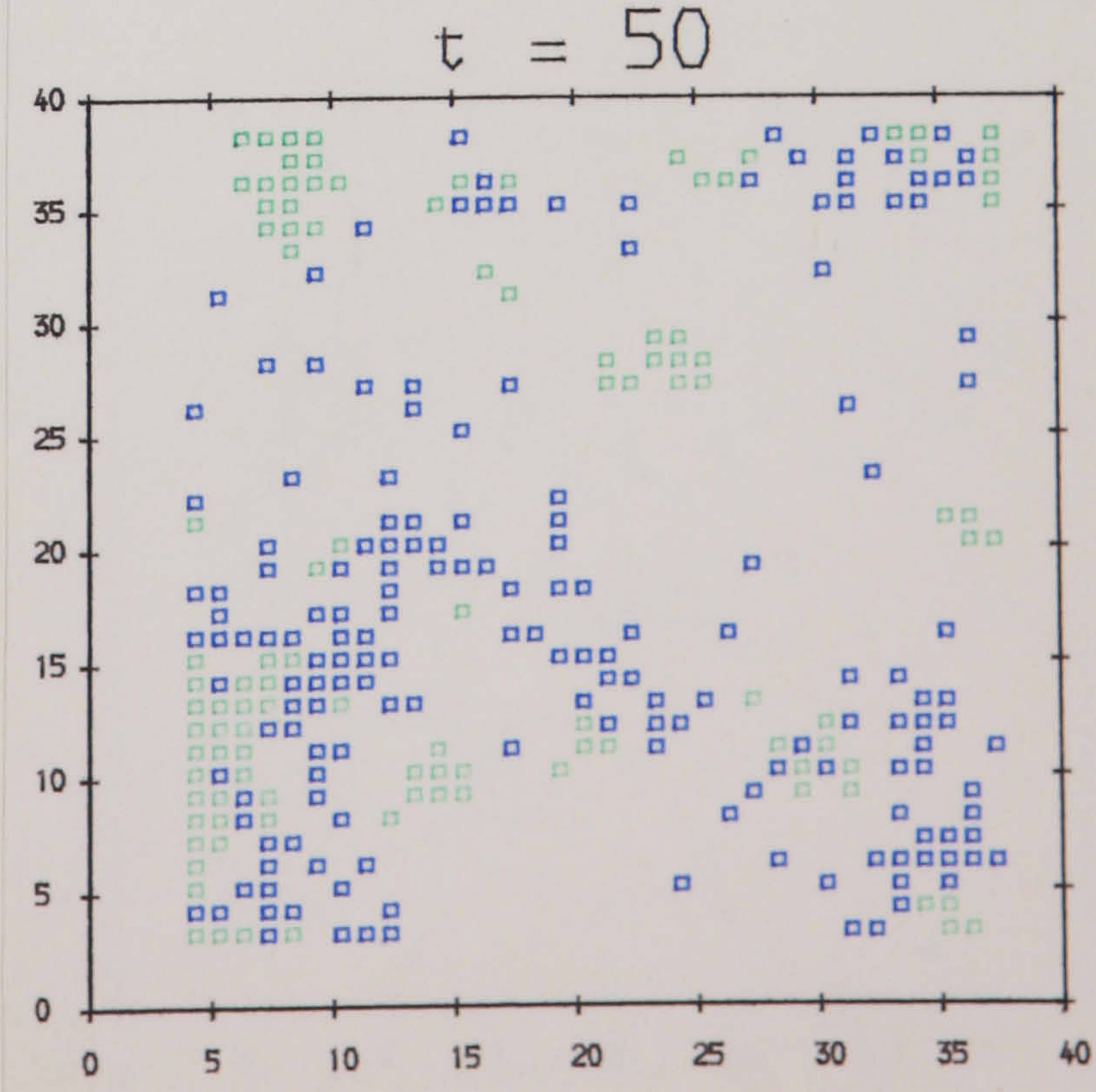
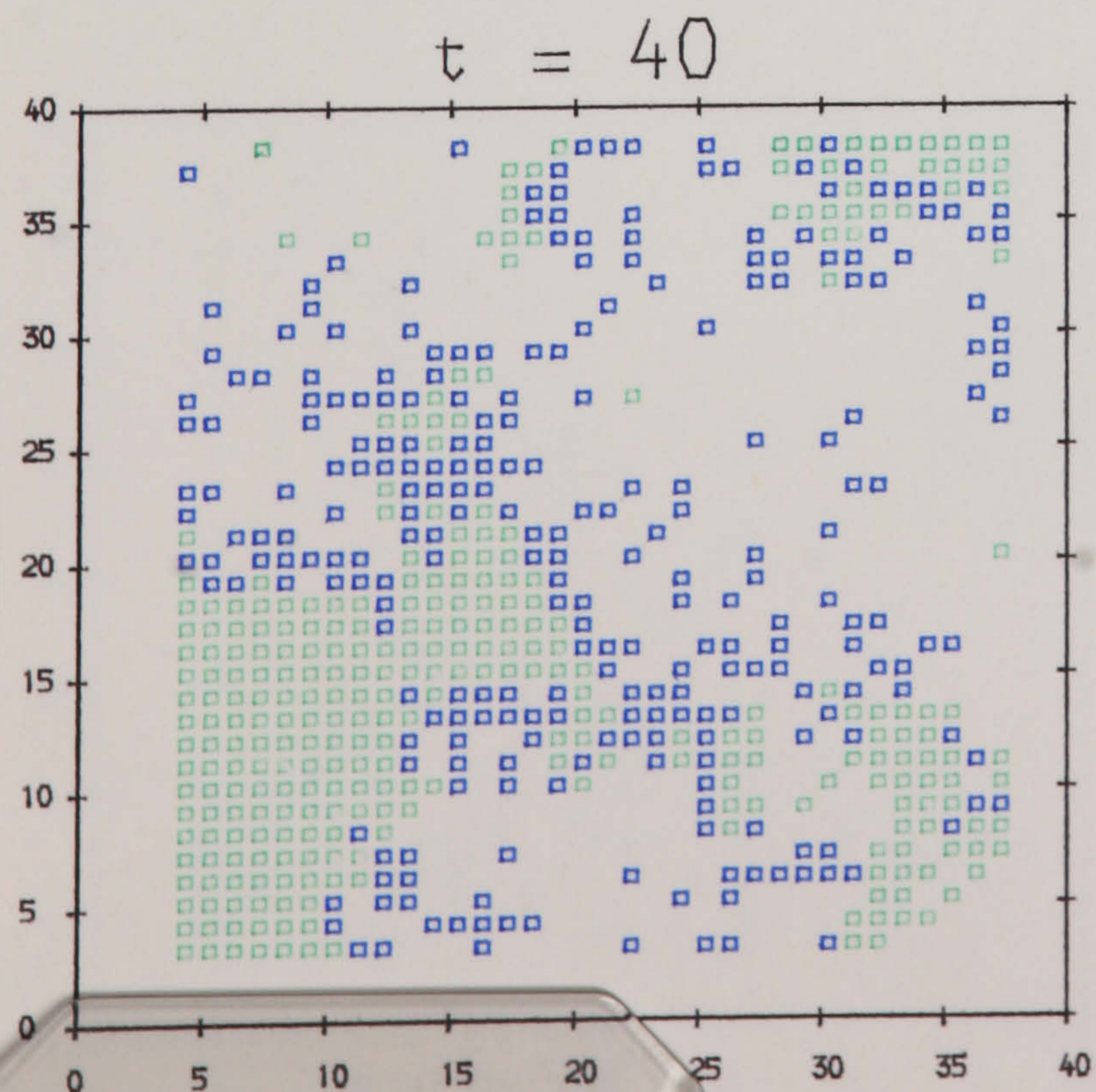
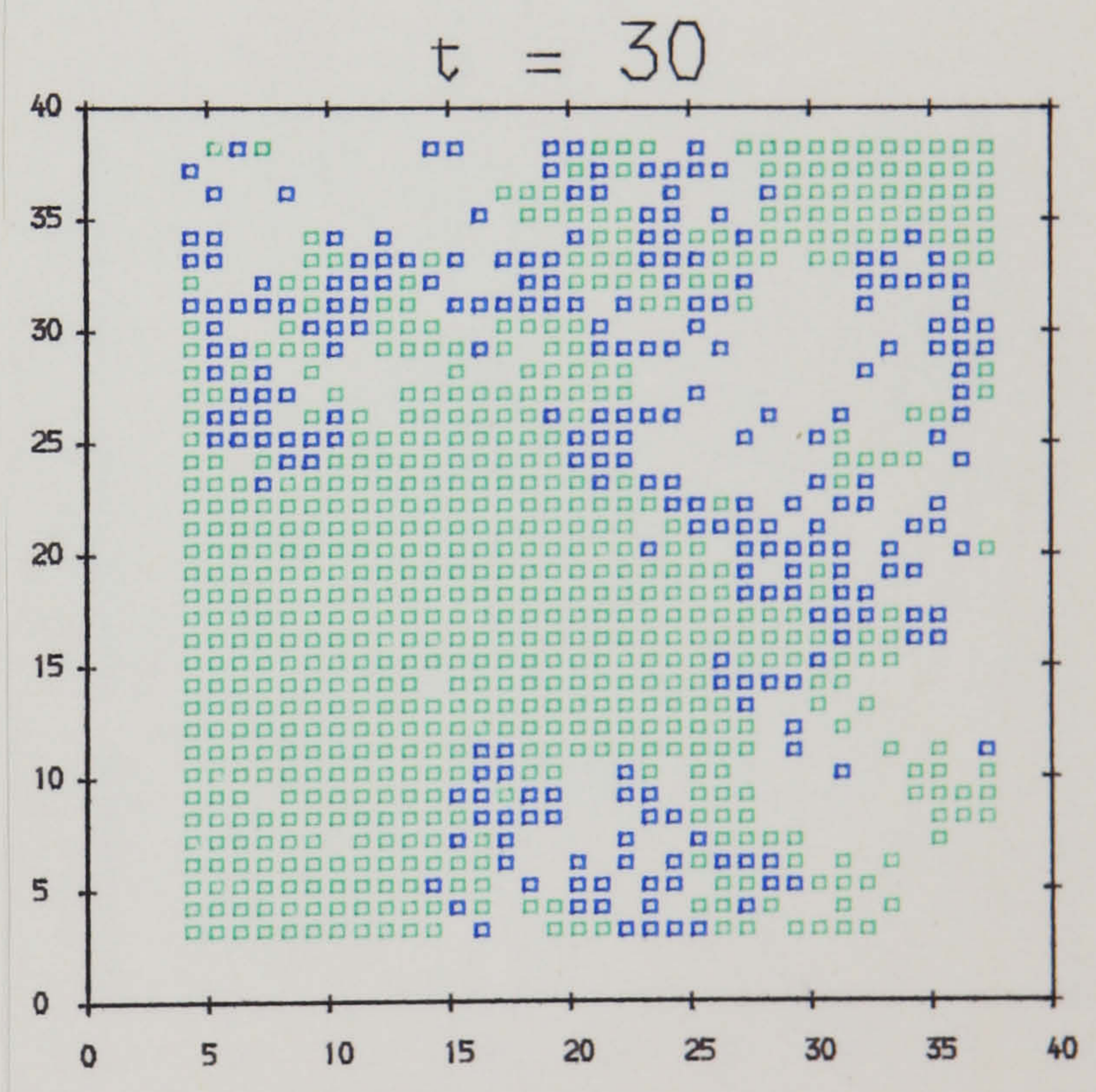
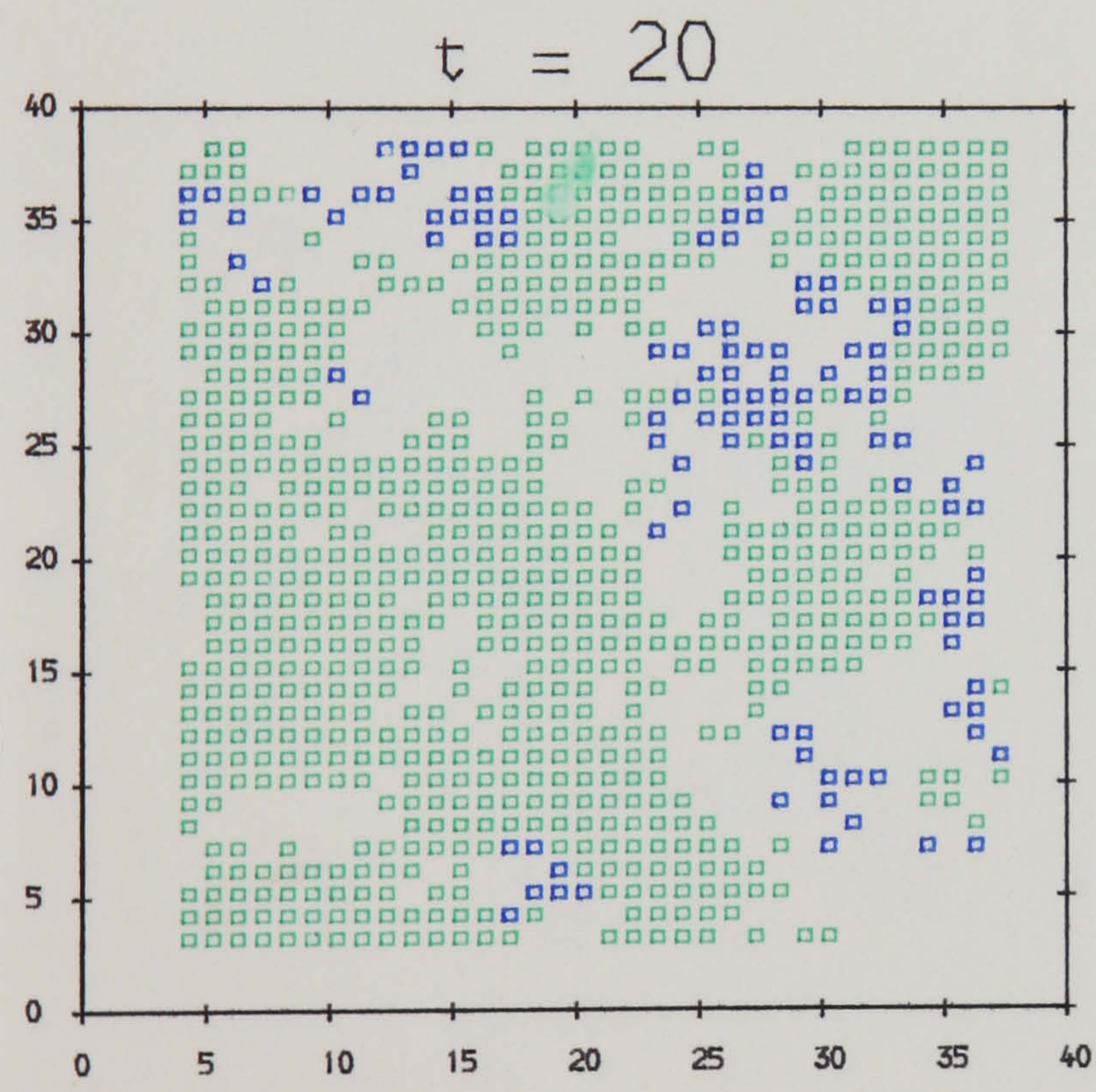
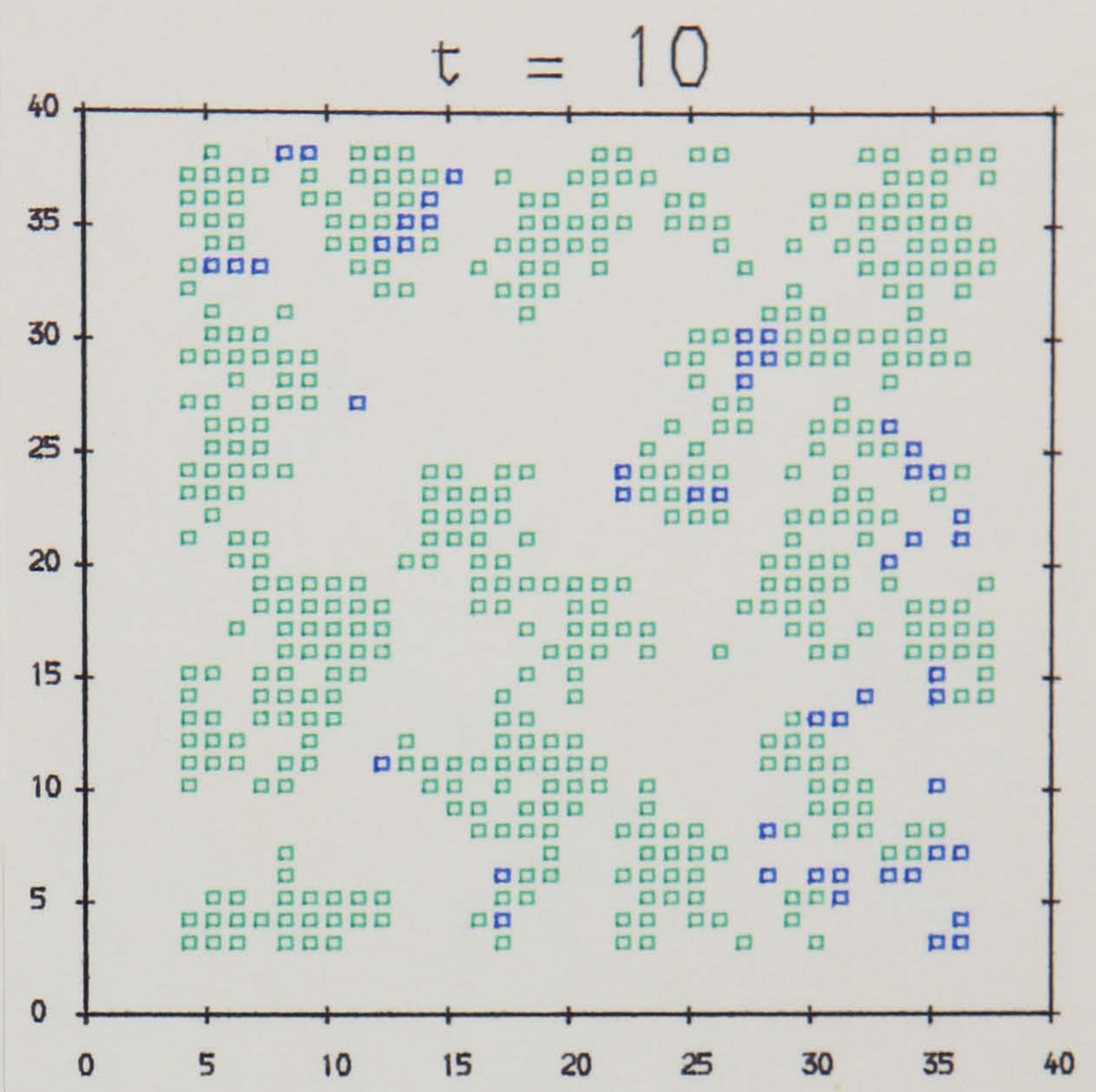
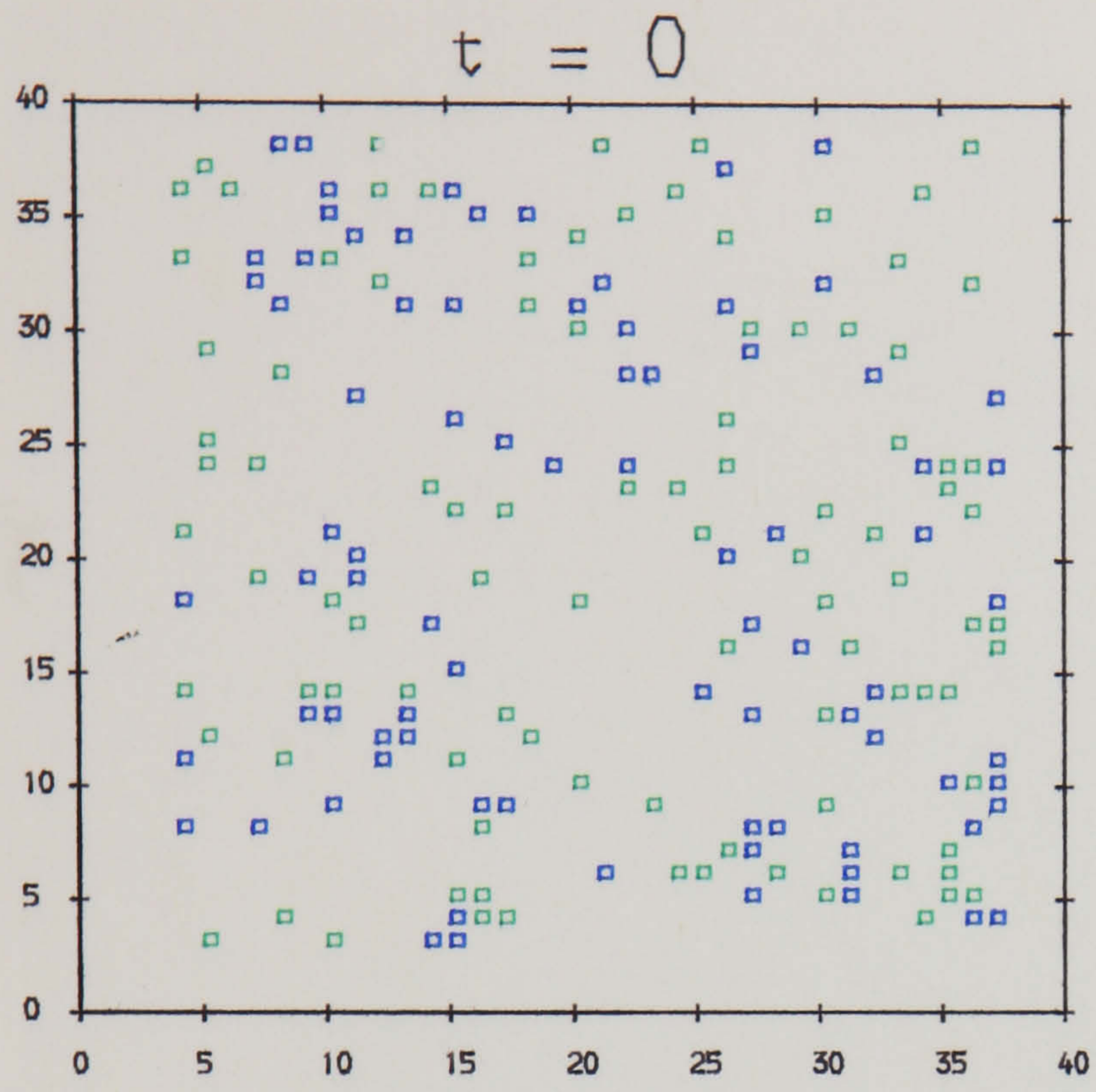




Figure 5.4: Lattice evolution for improved Lotka-Volterra CA.

State 1    State 2

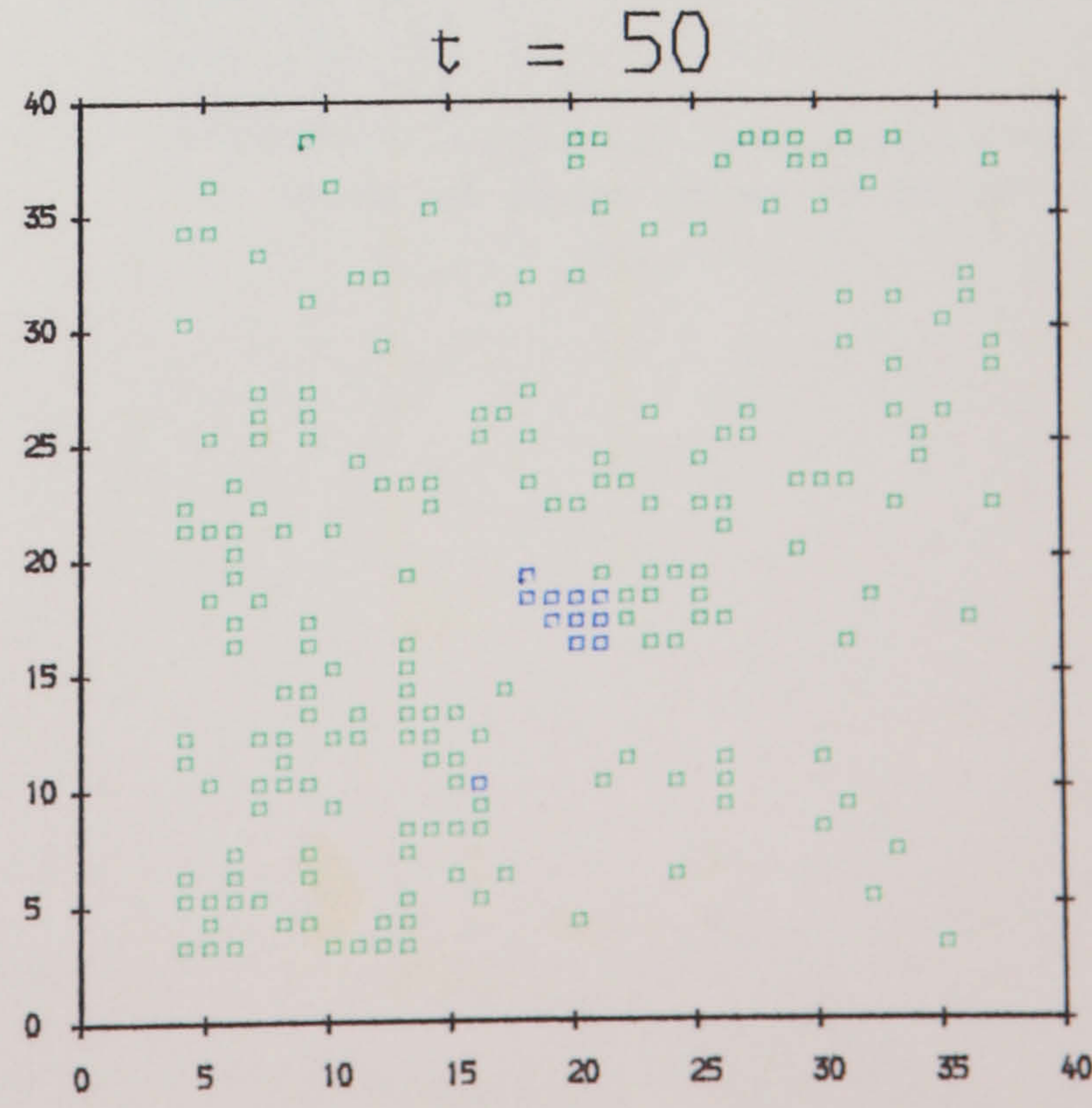
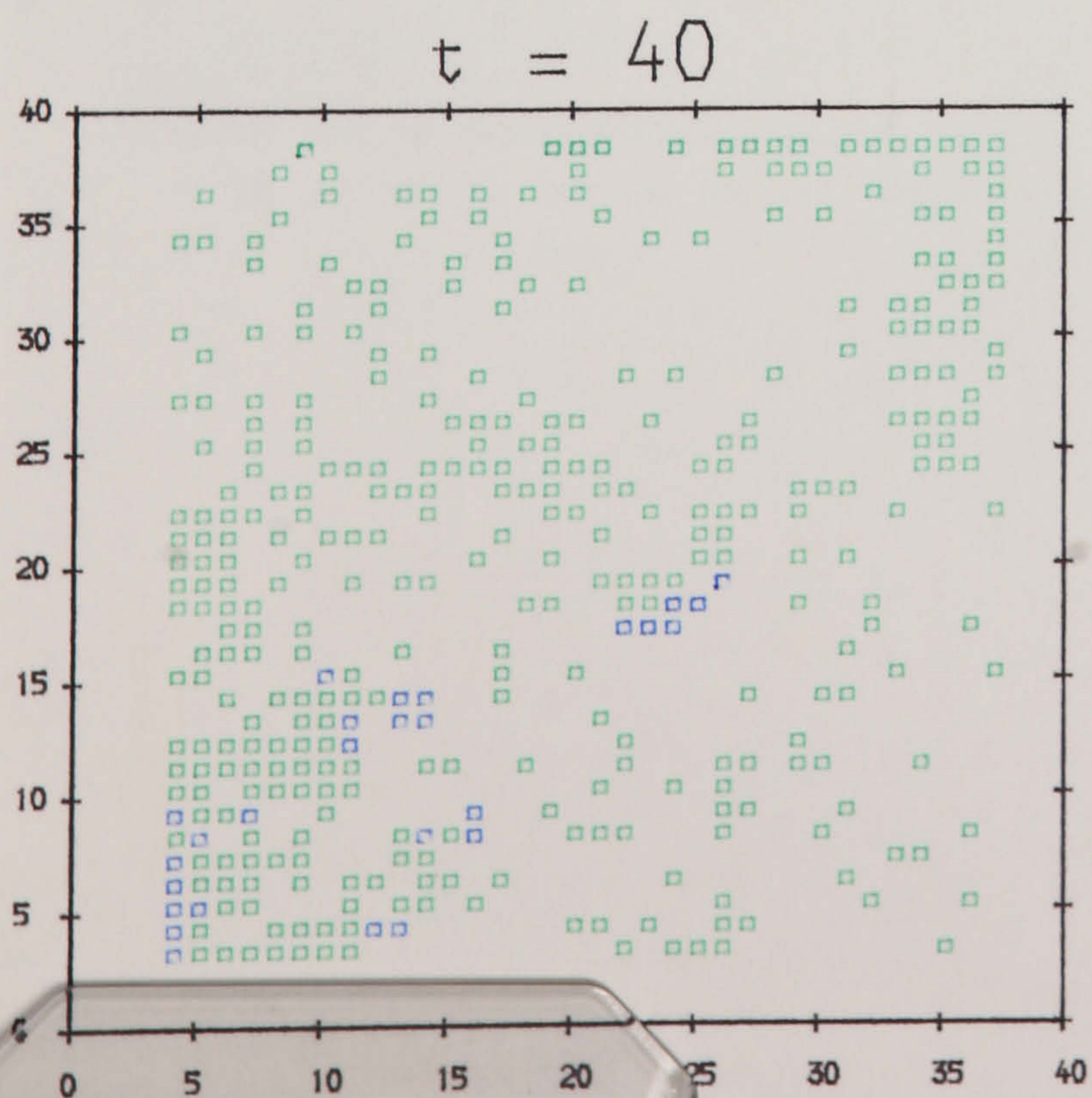
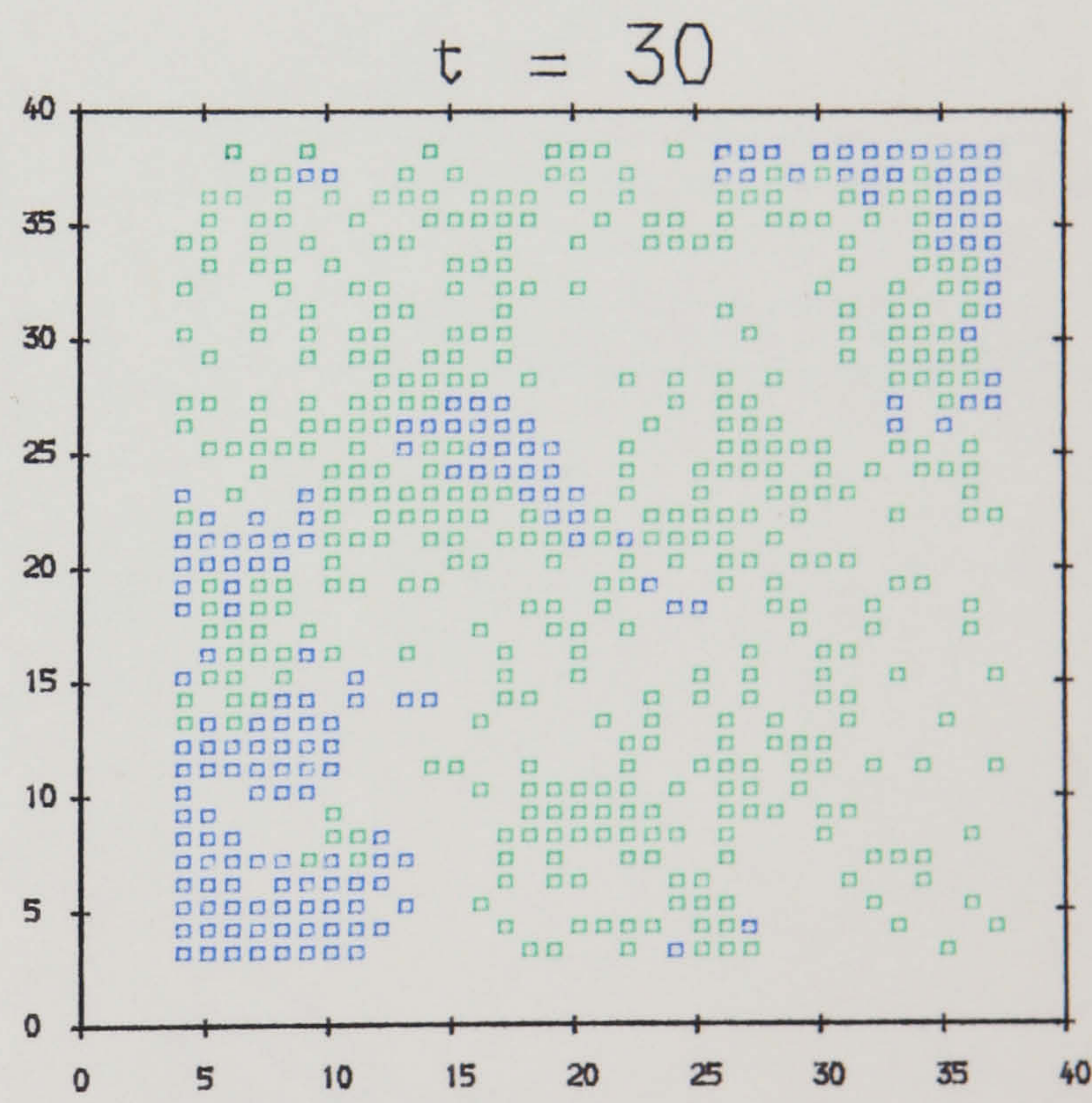
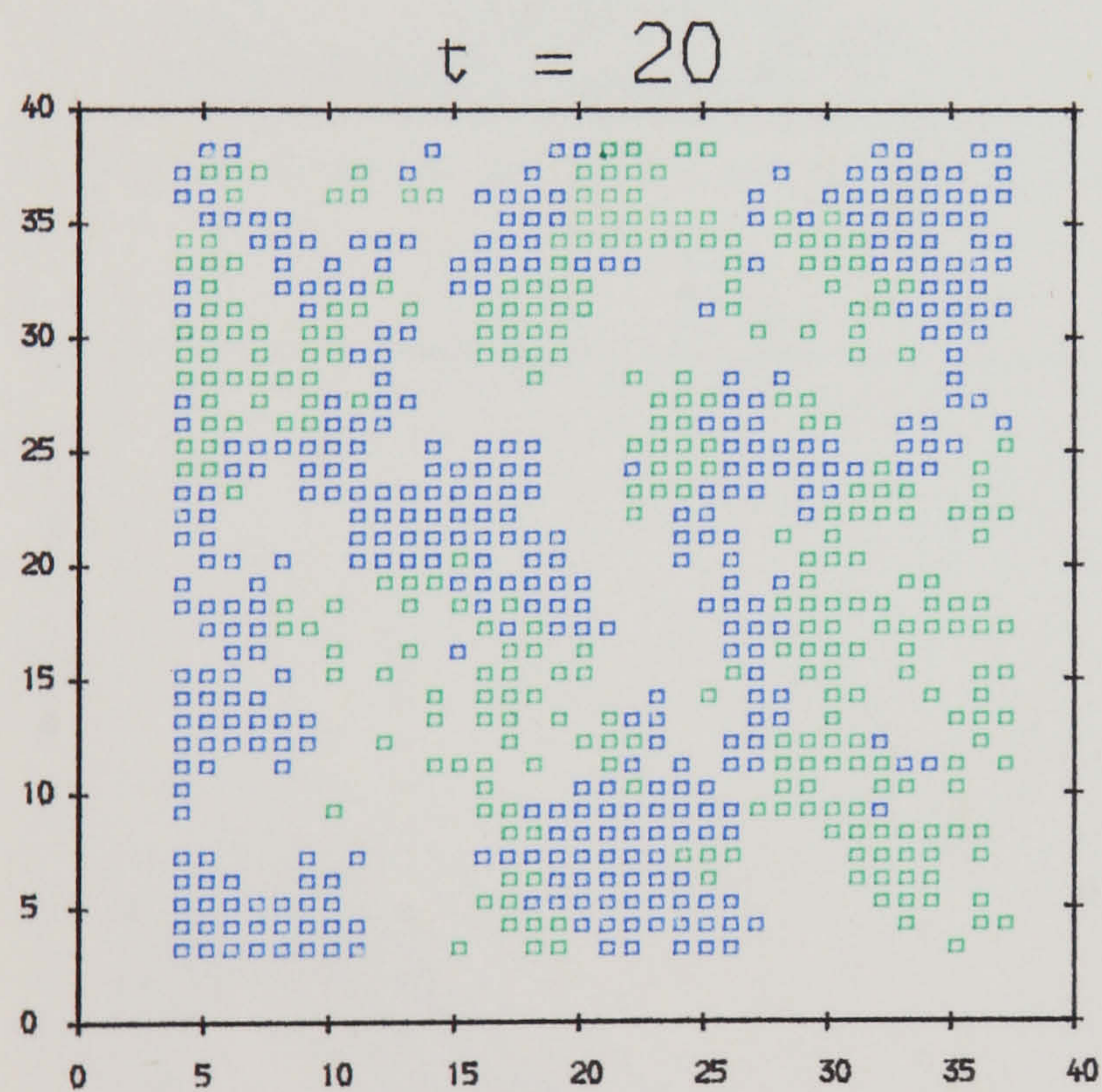
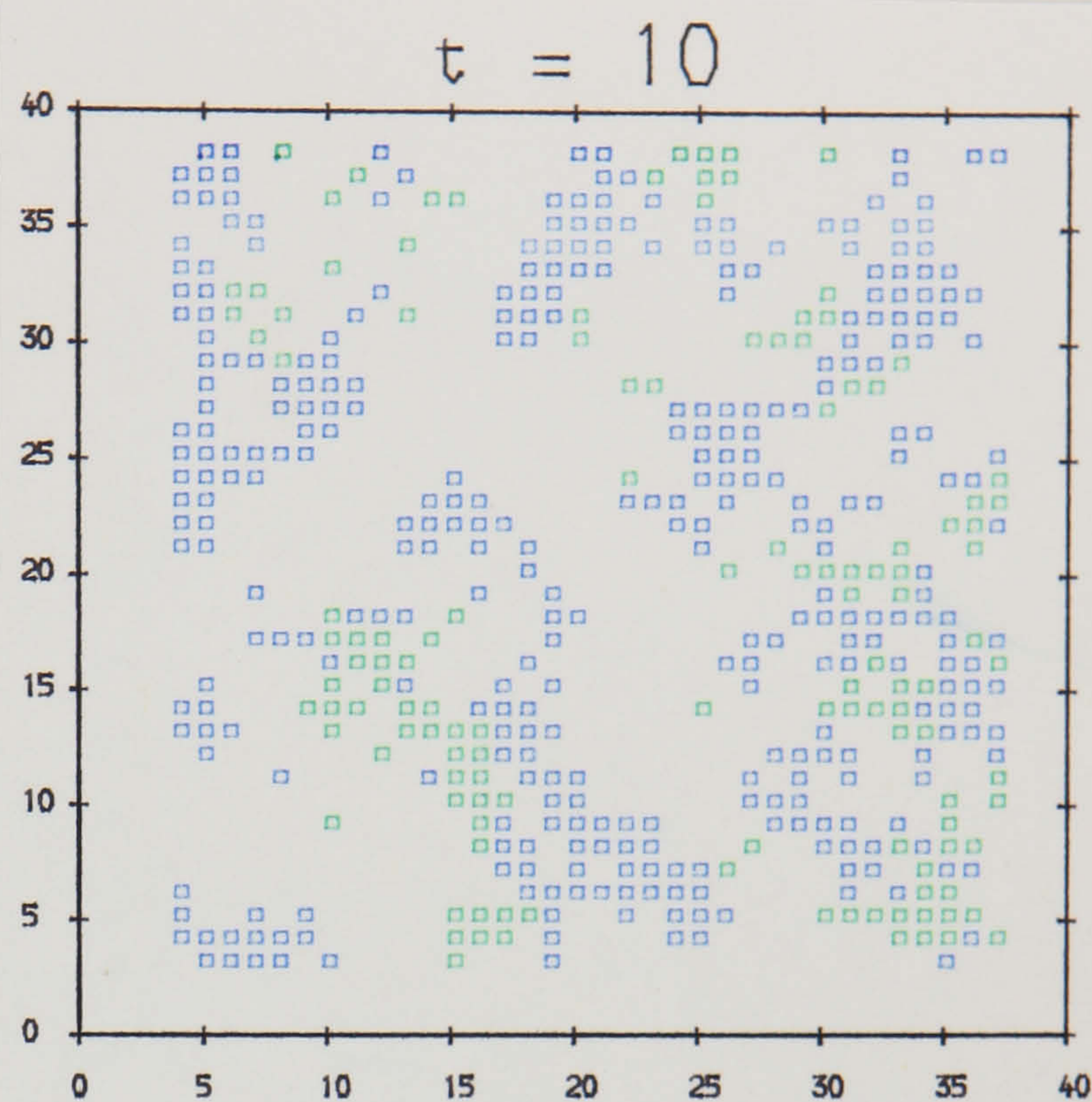
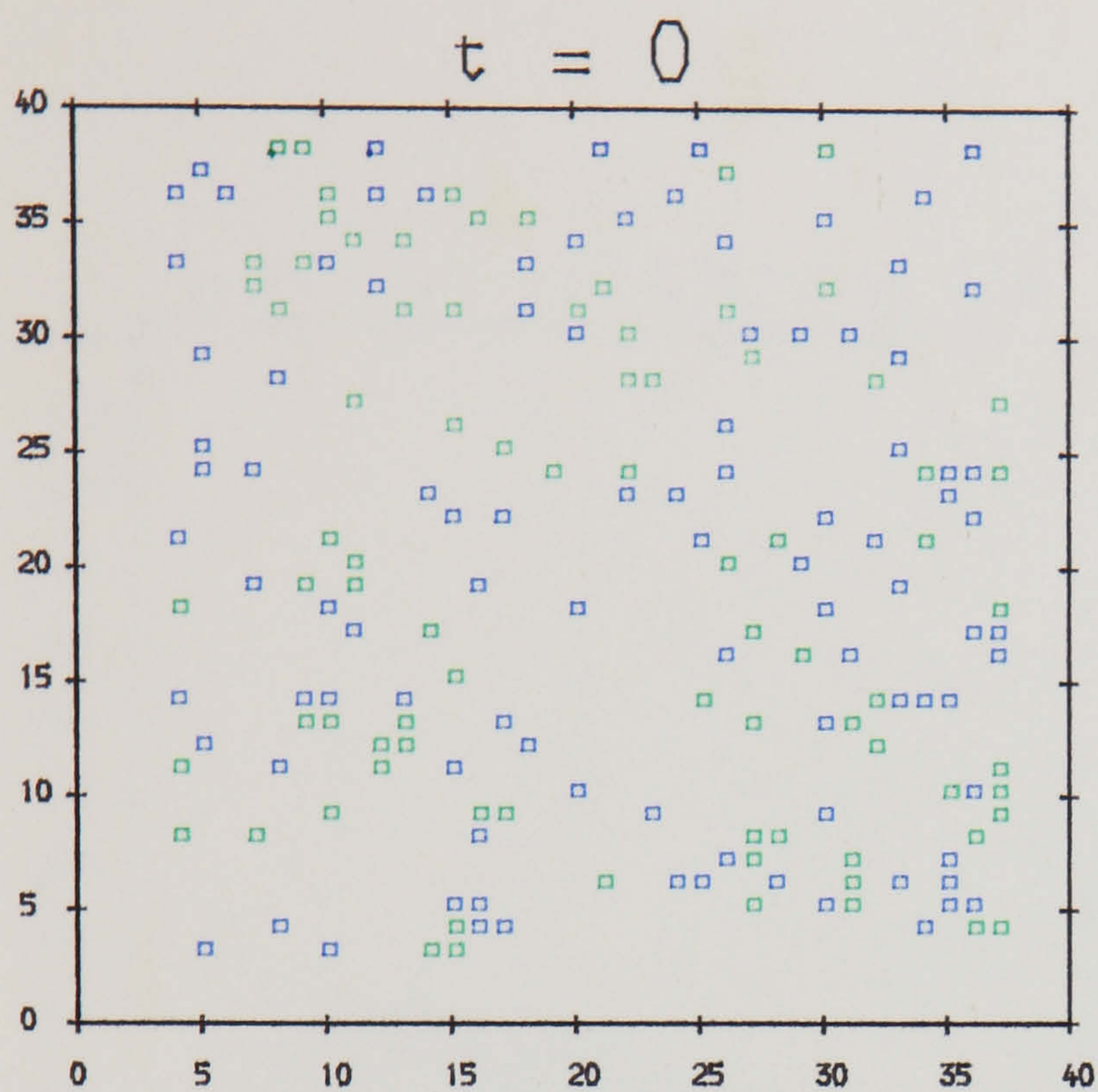




Figure 5.5: Phase diagram for simplified Lotka-Volterra model.

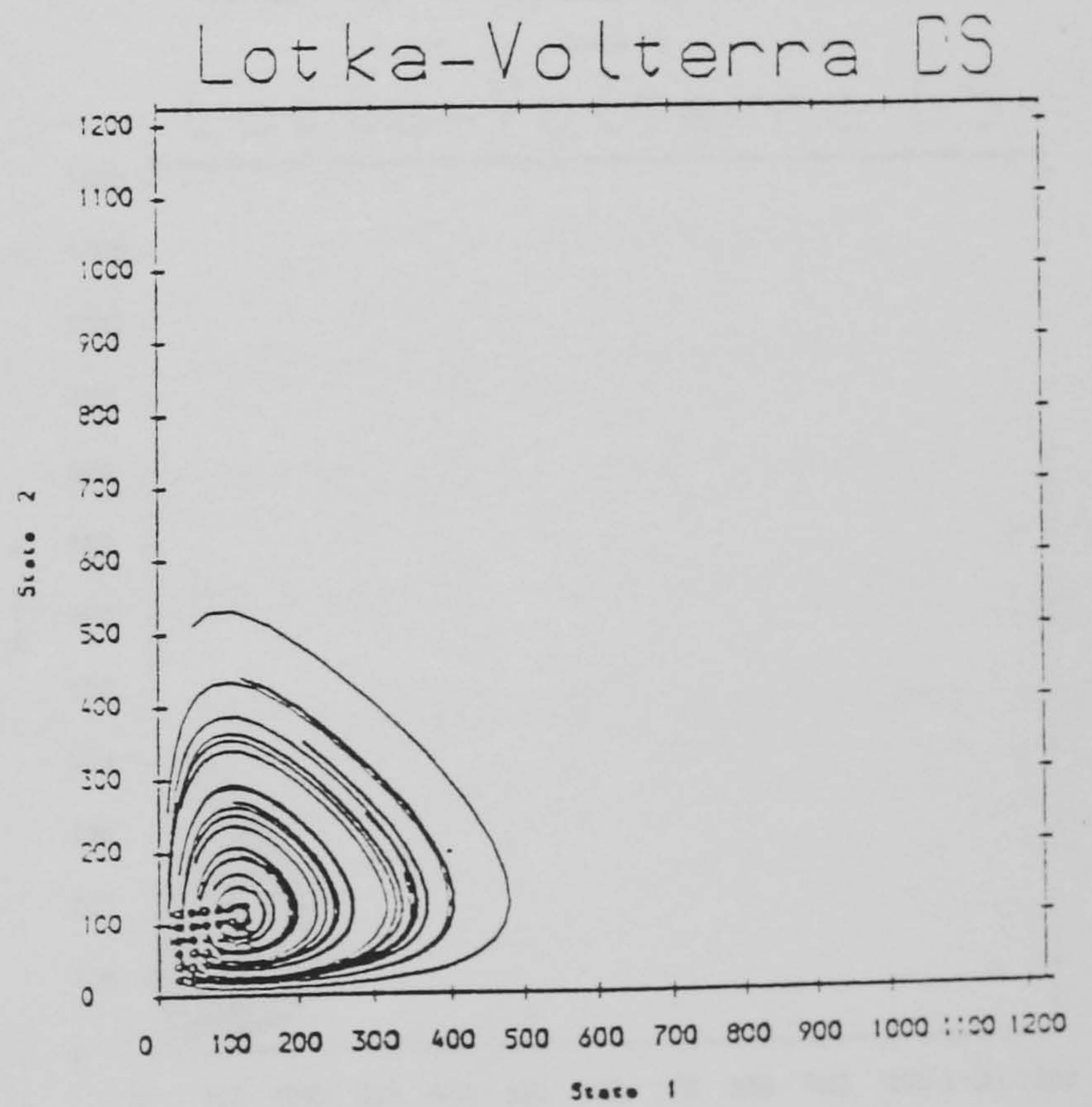
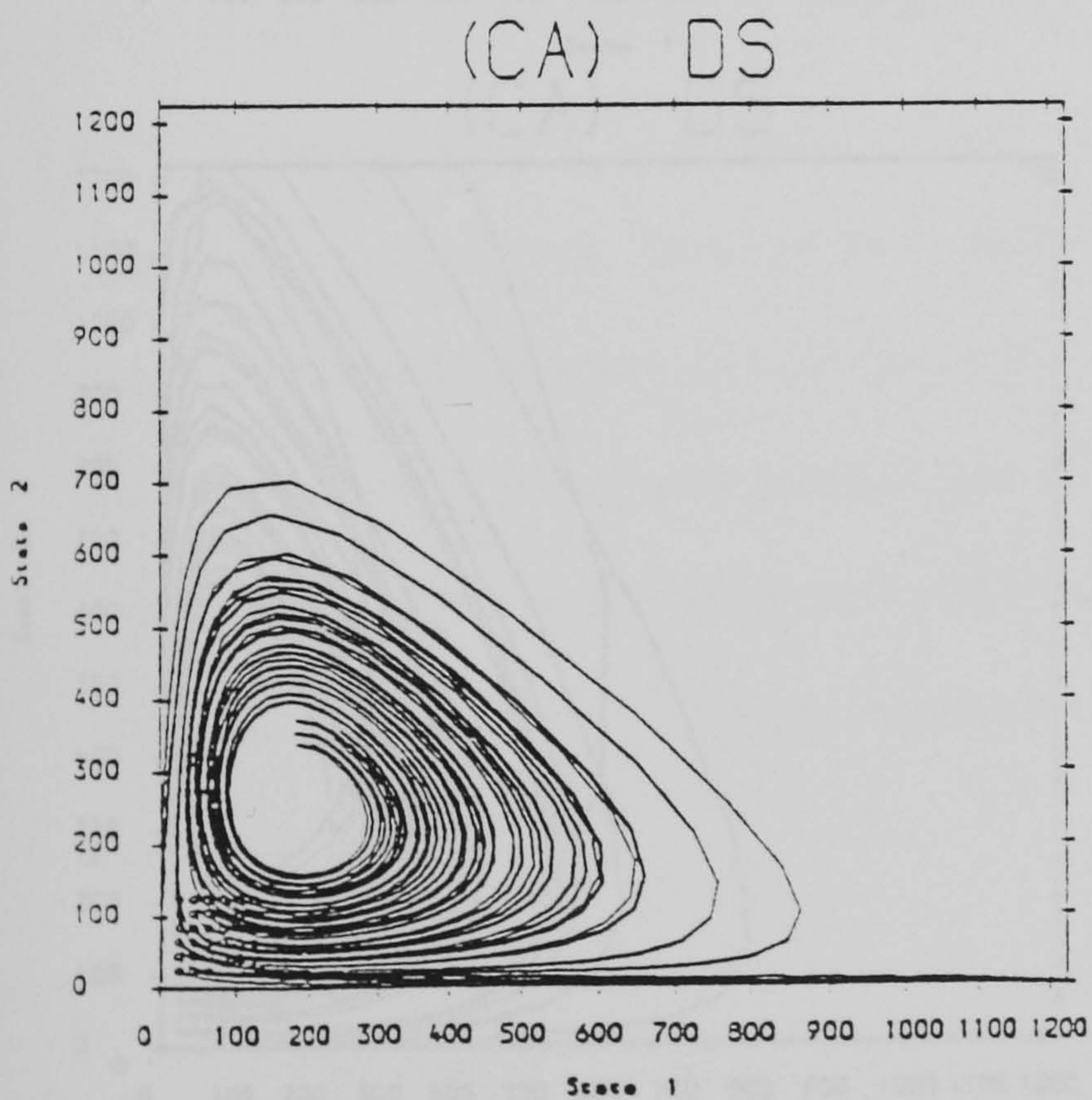
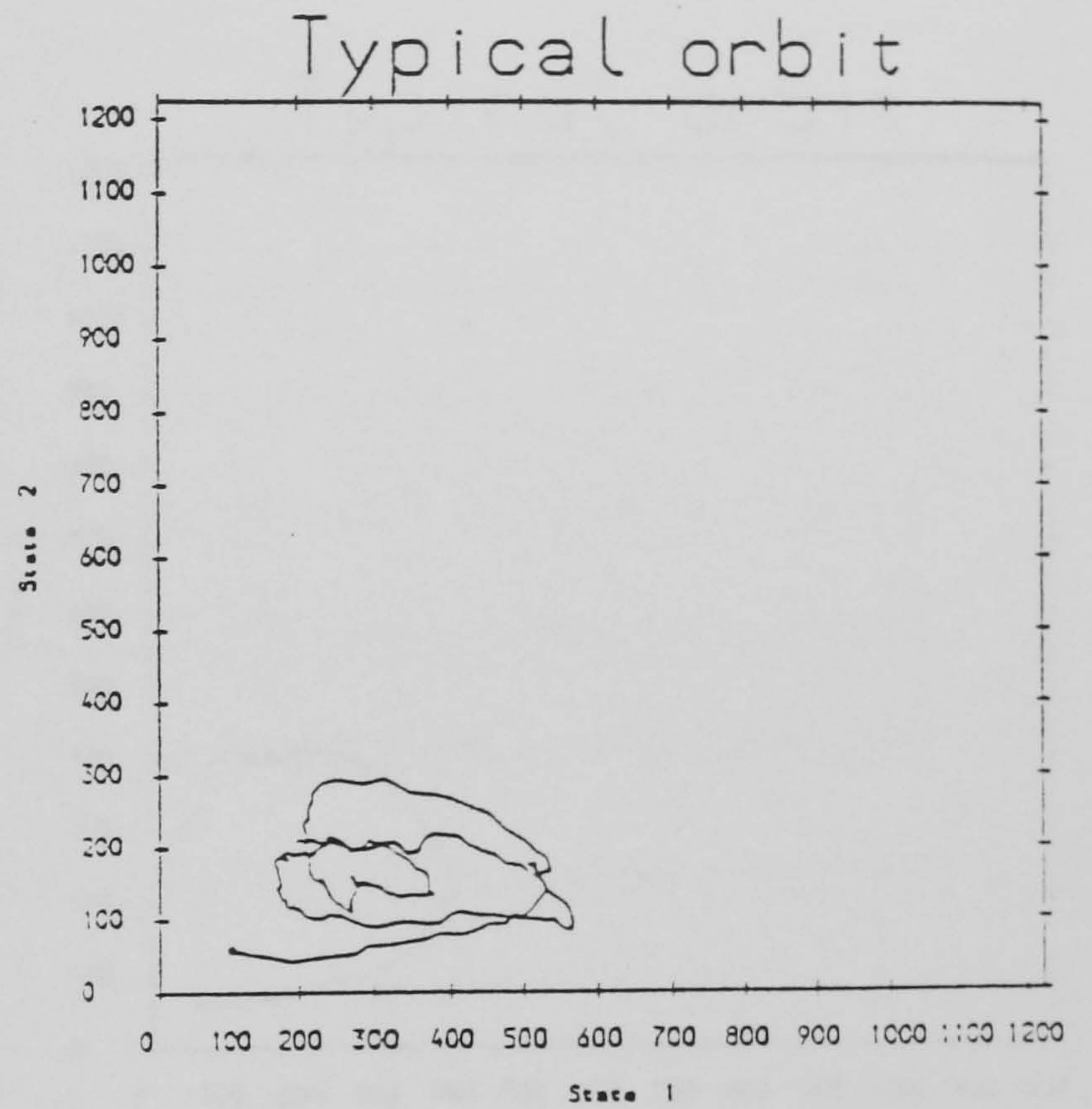
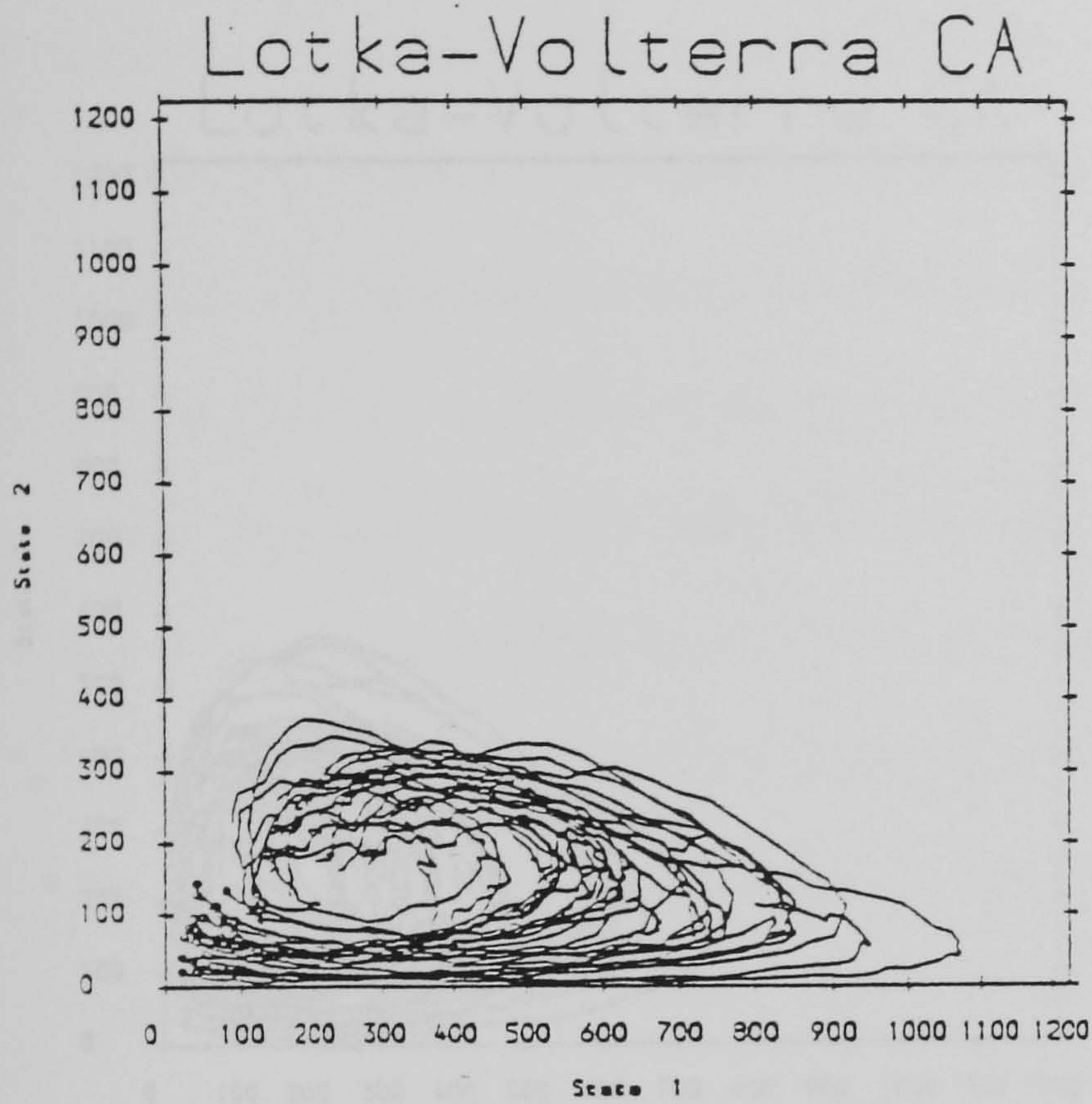
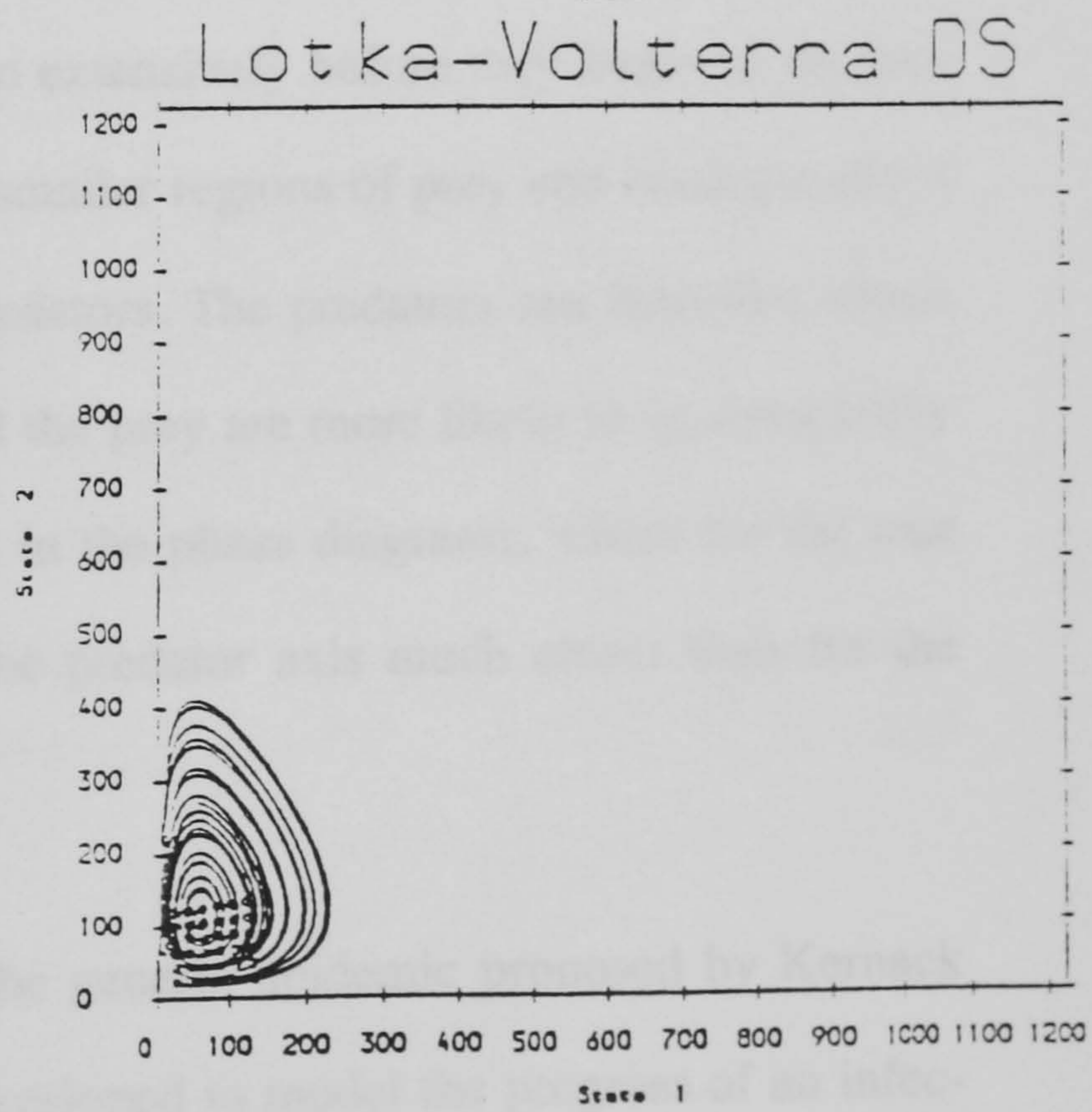
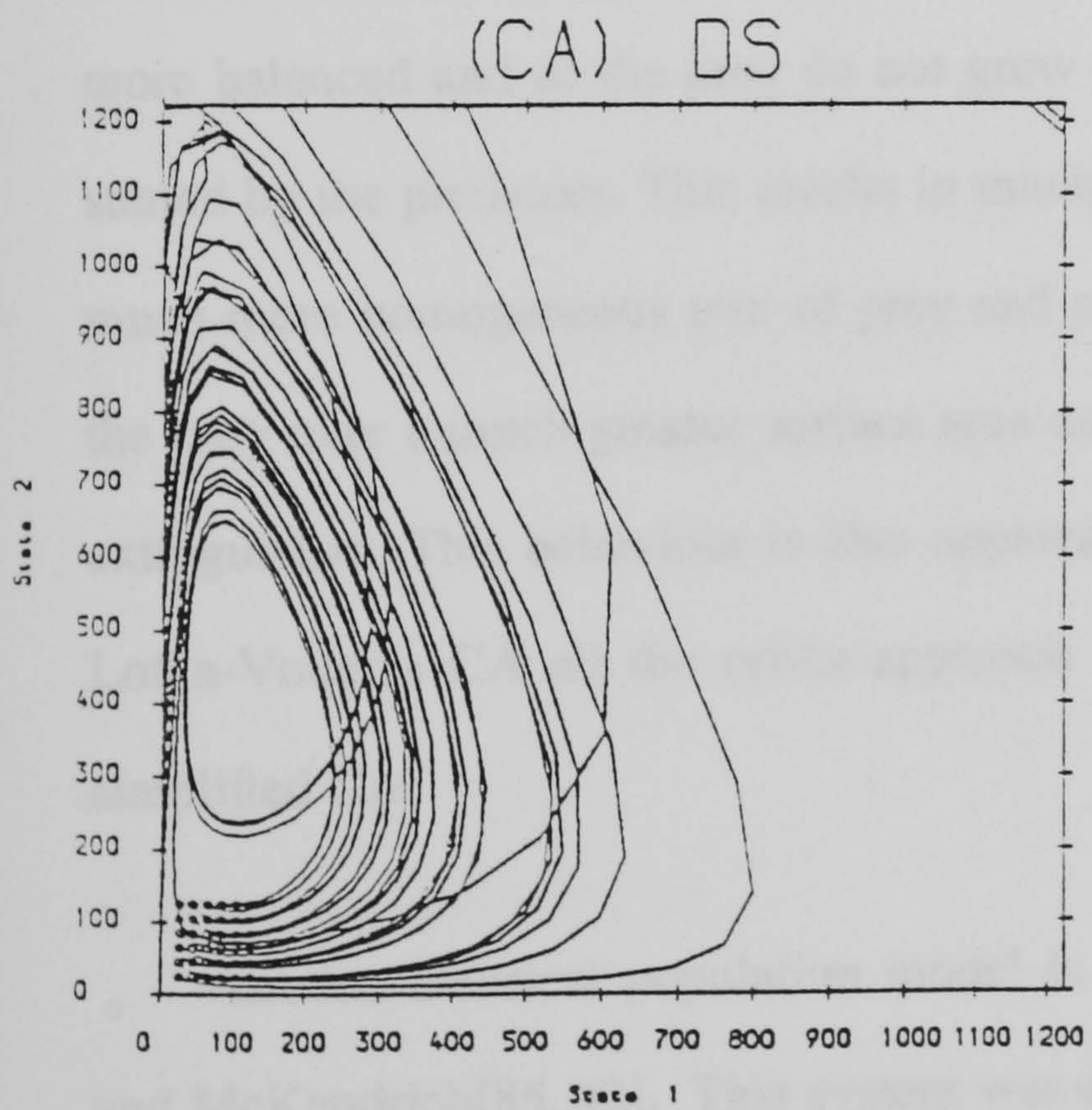
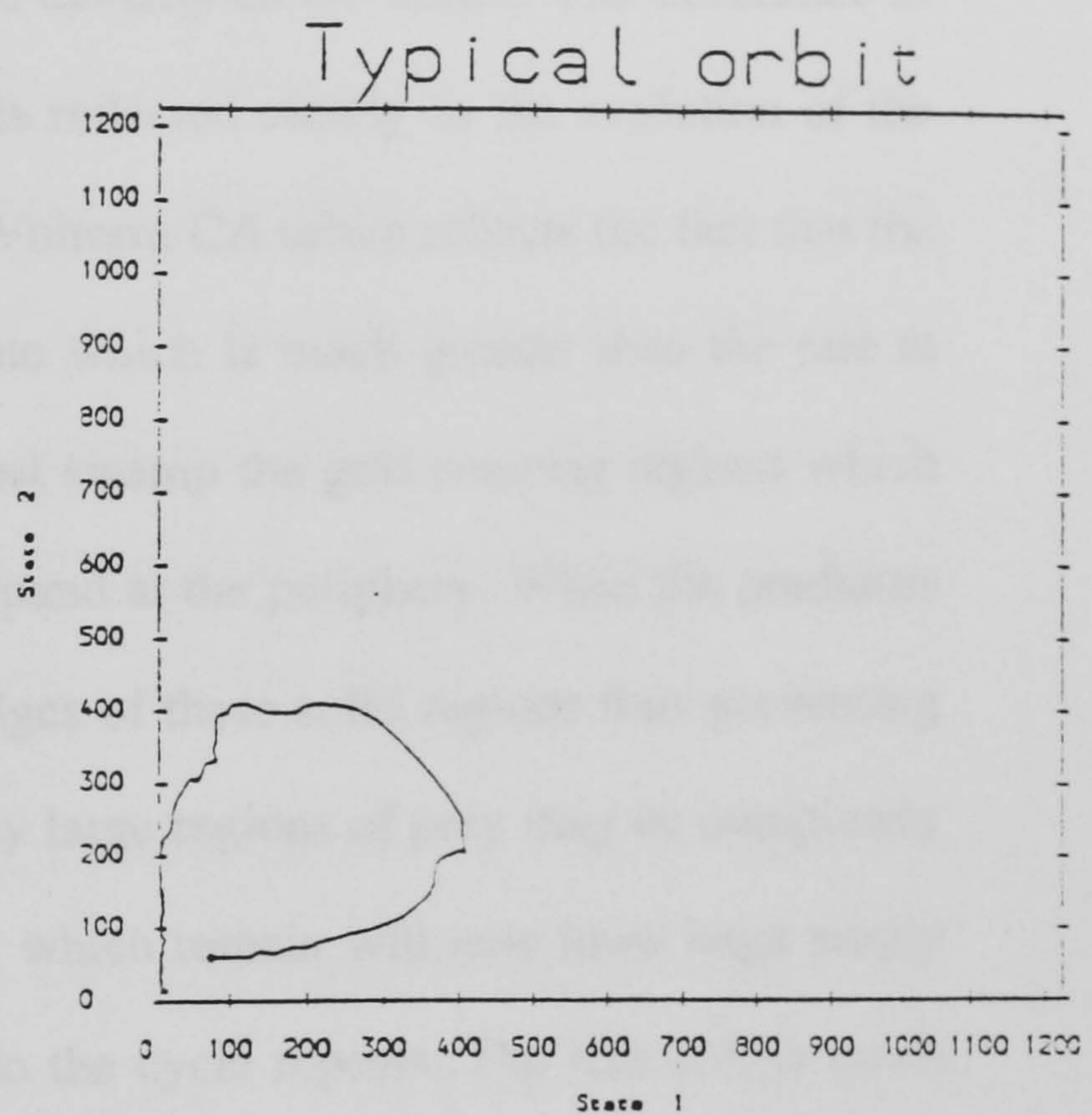
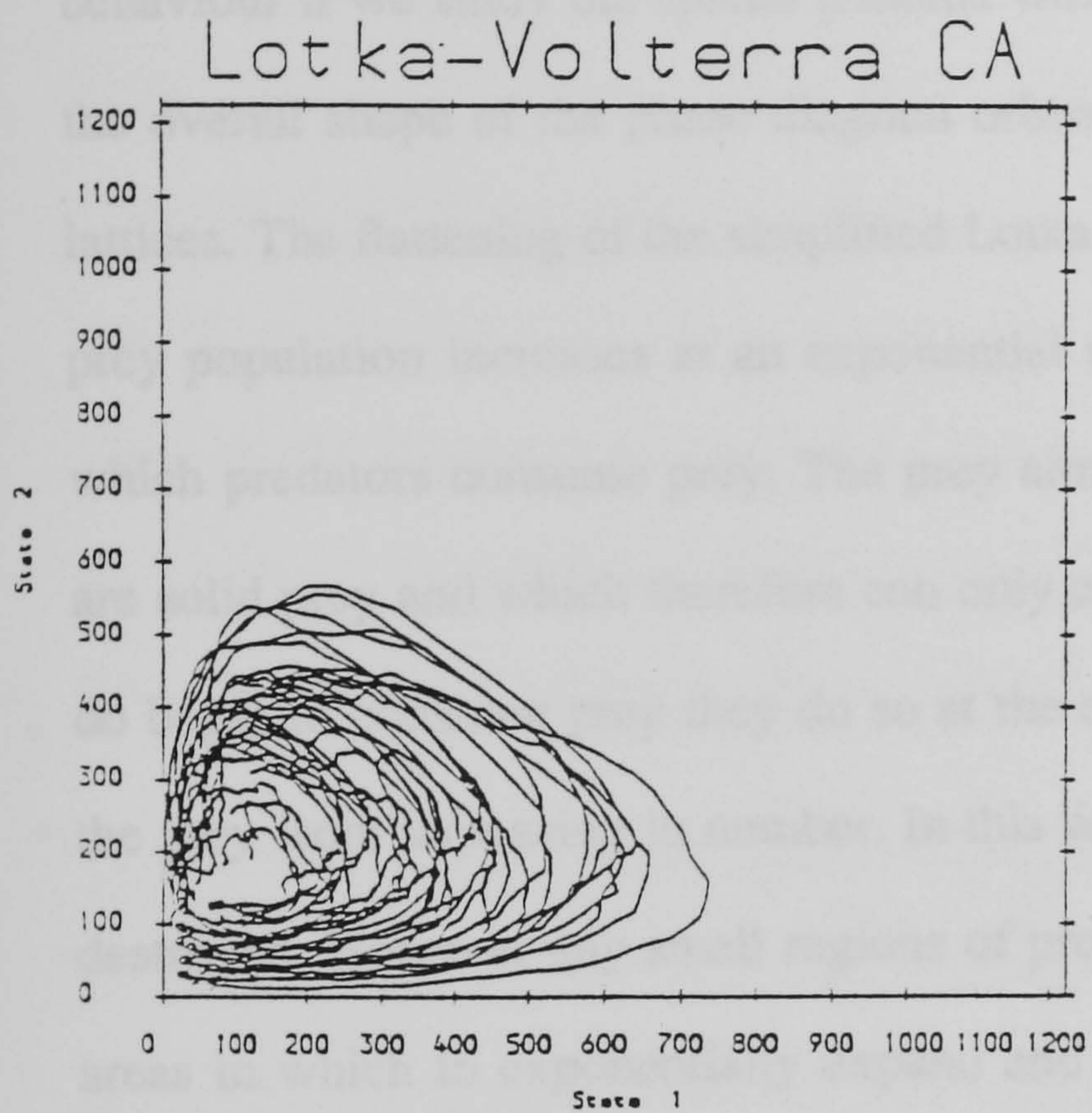




Figure 5.6: Phase diagram for improved Lotka-Volterra model.





of the order of ten cells still follow the same evolutionary path indicating that the CA model is robust to low population numbers even though the statistical arguments relating it to the DS will not hold. The reason for this is the CA simulation is very closely related to the actual physical processes. We can begin to understand the origin of the cyclic behaviour if we study the spatial patterns which develop on the lattice. The difference in the overall shape of the phase diagram orbits is reflected clearly in the evolution of the lattices. The flattening of the simplified Lotka-Volterra CA orbits reflects the fact that the prey population increases at an exponential rate which is much greater than the rate at which predators consume prey. The prey almost swamp the grid creating regions which are solid prey and which therefore can only expand at the periphery. When the predators do begin to consume prey they do so at the edges of these solid regions thus preventing the prey from increasing in number. In this way large regions of prey may be completely destroyed. However any small regions of prey which remain will now have large empty areas in which to exponentially expand and so the cycle repeats. The true CA is much more balanced and so the prey do not grow so extensively before they begin to be consumed by the predators. This results in much smaller regions of prey and consequently a much more homogeneous mix of prey and predators. The predators can therefore attack the prey over a much greater surface area and the prey are more likely to be completely extinguished. This behaviour is also apparent in the phase diagrams, where for the true Lotka-Volterra CA all the orbits approach the predator axis much closer than for the simplified CA.

Another classical population model is the general epidemic proposed by Kernack and McKendrick[85, 93]. This system was developed to model the progress of an infectious disease in a population of individuals which are in close contact. The system can be defined as follows,



$$\dot{S} = -\beta SI \quad 5.28$$

$$\dot{I} = \beta SI - \gamma I \quad 5.29$$

$$\dot{R} = \gamma I \quad 5.30$$

Here  $S$  represents susceptibles, that is individuals who are uninfected at present but liable to infection.  $I$  represents infected individuals who can transmit the disease onto uninfected individuals. Finally  $R$  represents recovered individuals who are no longer infectious and are immune to the disease. The system of equations simulates the rate of infection of susceptible individuals as proportional to the number of infected individuals. These newly infected individuals are therefore added to the number of infected individuals which are also modelled as recovering at a constant rate. The newly recovered individuals are thus added to the current number of recovered individuals.

This system is in fact more closely related to a CA than the previous system since here we have a single population of individuals who are simply changing state. Thus susceptible cells must change to infected cells with a probability proportional to the number of infected cells in their neighbourhood. Infected cells must change to recovered cells with a constant probability. We should note that this system has a very clear overall behaviour, which is that susceptibles cells will gradually be converted to recovered cells via the intermediate state of infected cells. This results in the following rule set,

$$s_1 \& \{c s_2\} \rightarrow s_2 \mid \text{Pr} = c_1 c / r \quad 5.31$$

$$s_2 \rightarrow s_3 \mid \text{Pr} = c_2 \quad 5.32$$

Giving us the following DS,

$$\dot{n}_1 = - \left[ \sum_{c=0}^r c_1 \frac{c}{r} P(c, n_2) \right] n_1 \quad 5.33$$

$$\dot{n}_2 = \left[ \sum_{c=0}^r c_1 \frac{c}{r} P(c, n_2) \right] n_1 - [c_2] n_2 \quad 5.34$$

$$\dot{n}_3 = [c_2] n_2 \quad 5.35$$

Note that in this system there is no explicit requirement for a quiescent state. Indeed if we include cells in the quiescent state on the lattice we may find individual susceptible or infected cells which are not in contact with any other cells apart from quiescent cells. These will then remain in their initial state since there is no movement of the various populations on the lattice. Furthermore, depending on the initial distribution both in terms of relative numbers and spatially, we may find different patterns of spread of the disease.

The temporal evolution of the lattice for the CA model is shown in Figure 5.7. The phase diagram is shown in Figure 5.8.

We can see that the dynamical behaviour of this system is much simpler than that of the Lotka-Volterra system and consequently the agreement between the phase diagrams is very good. The initial lattice shown in Figure 5.7 contains 5% infected cells in a background consisting entirely of susceptible cells. With the rates of infection and recovery used in this example we see that the number of infected cells remains approximately constant at the original 5%. The recovered cell therefore increase at a constant rate. We would expect this type of behaviour from the phase diagram, since the trajectories from initial points with low numbers of infected and recovered cells are approximately straight lines. We should also note that after  $t=40$  there are regions of recovered cells with no associated infected cells corresponding to outbreaks of the infection which have spontaneously halted.

This model can be extended to the recurrent epidemic[85]. In this system the recovered individuals become susceptible again at a constant rate  $\mu$ . The system can



Figure 5.7: Lattice evolution for general epidemic CA.

State 1    State 2

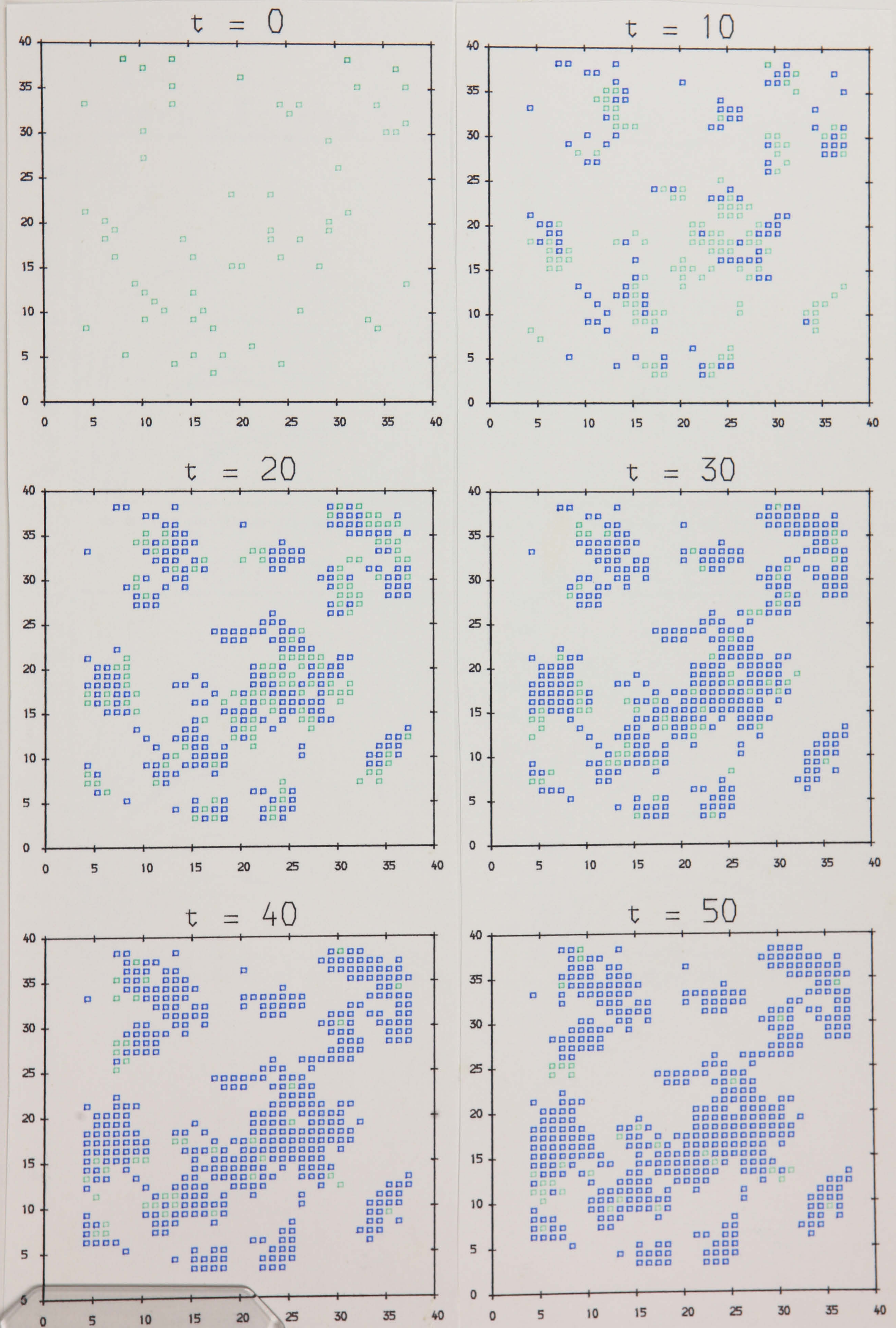




Figure 5.8: Phase diagram for general epidemic model (a).

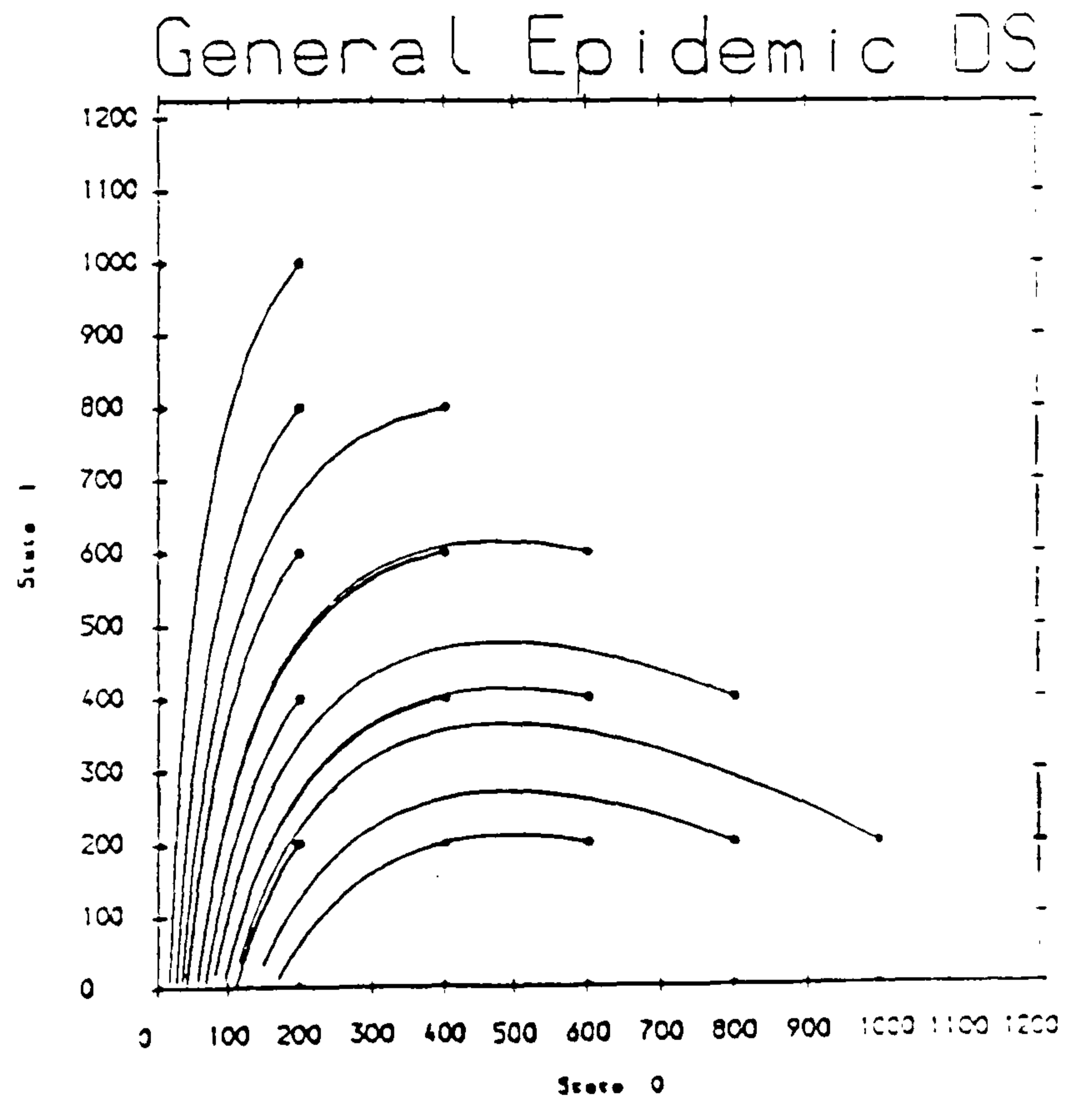
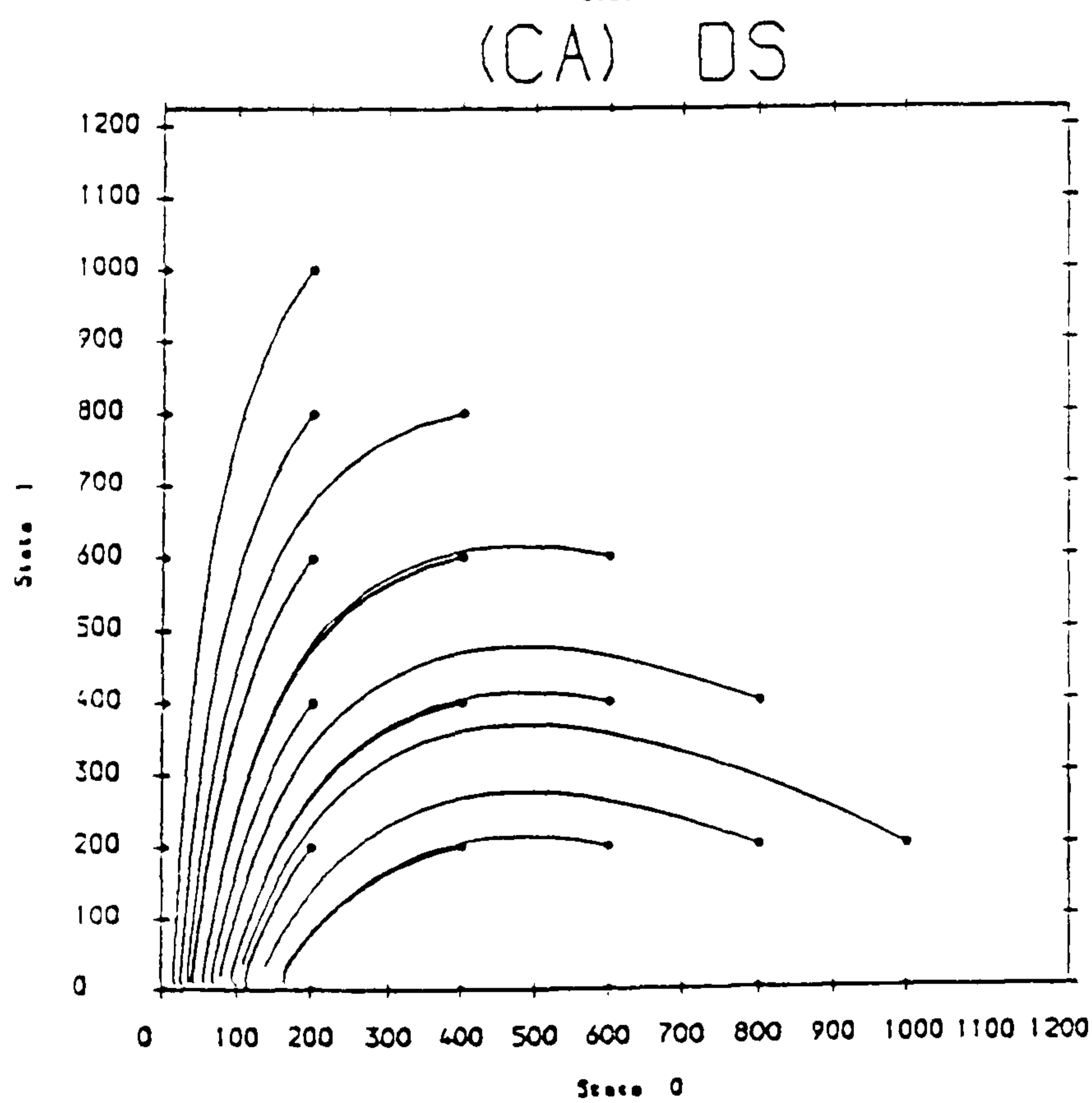
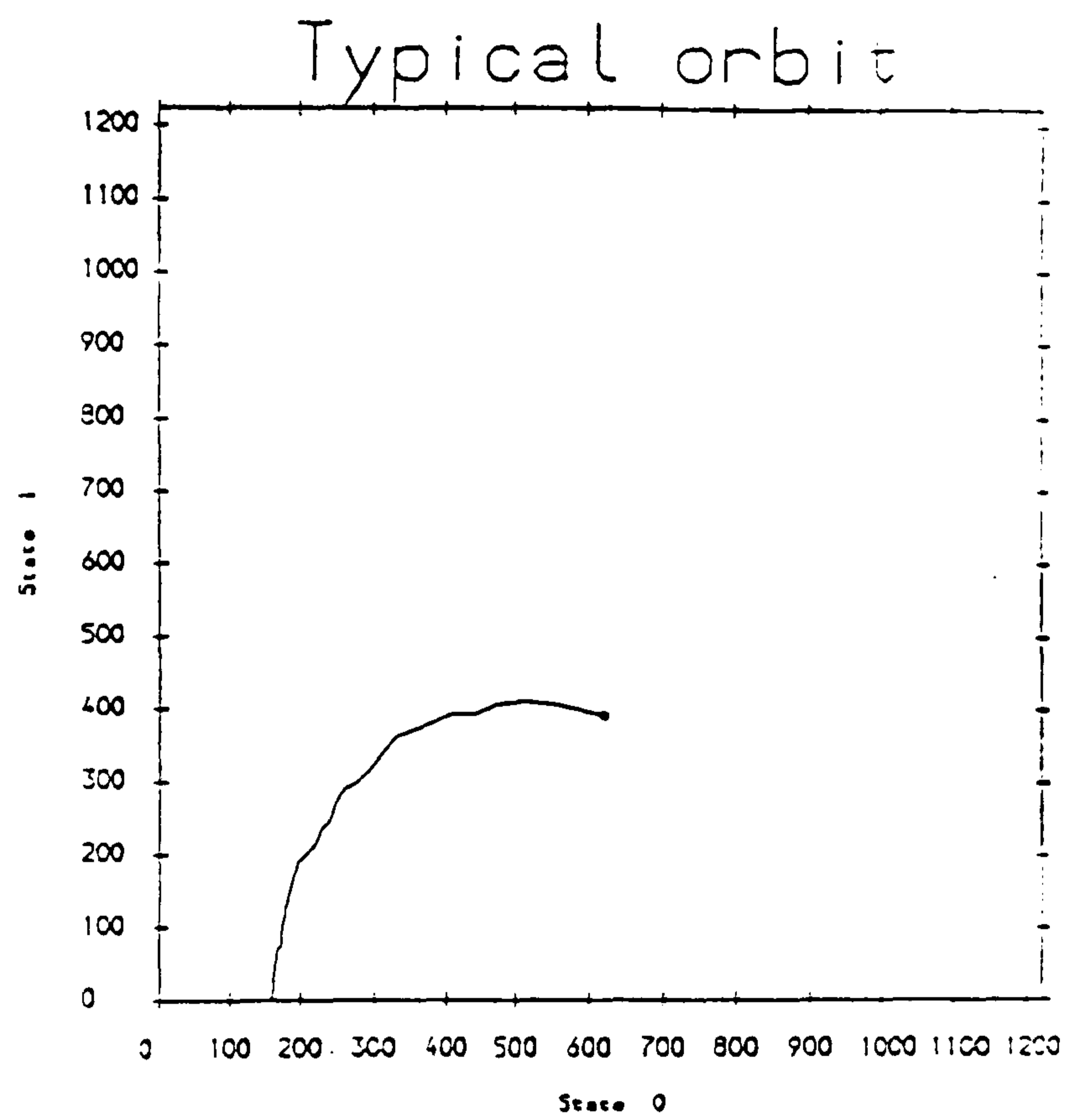
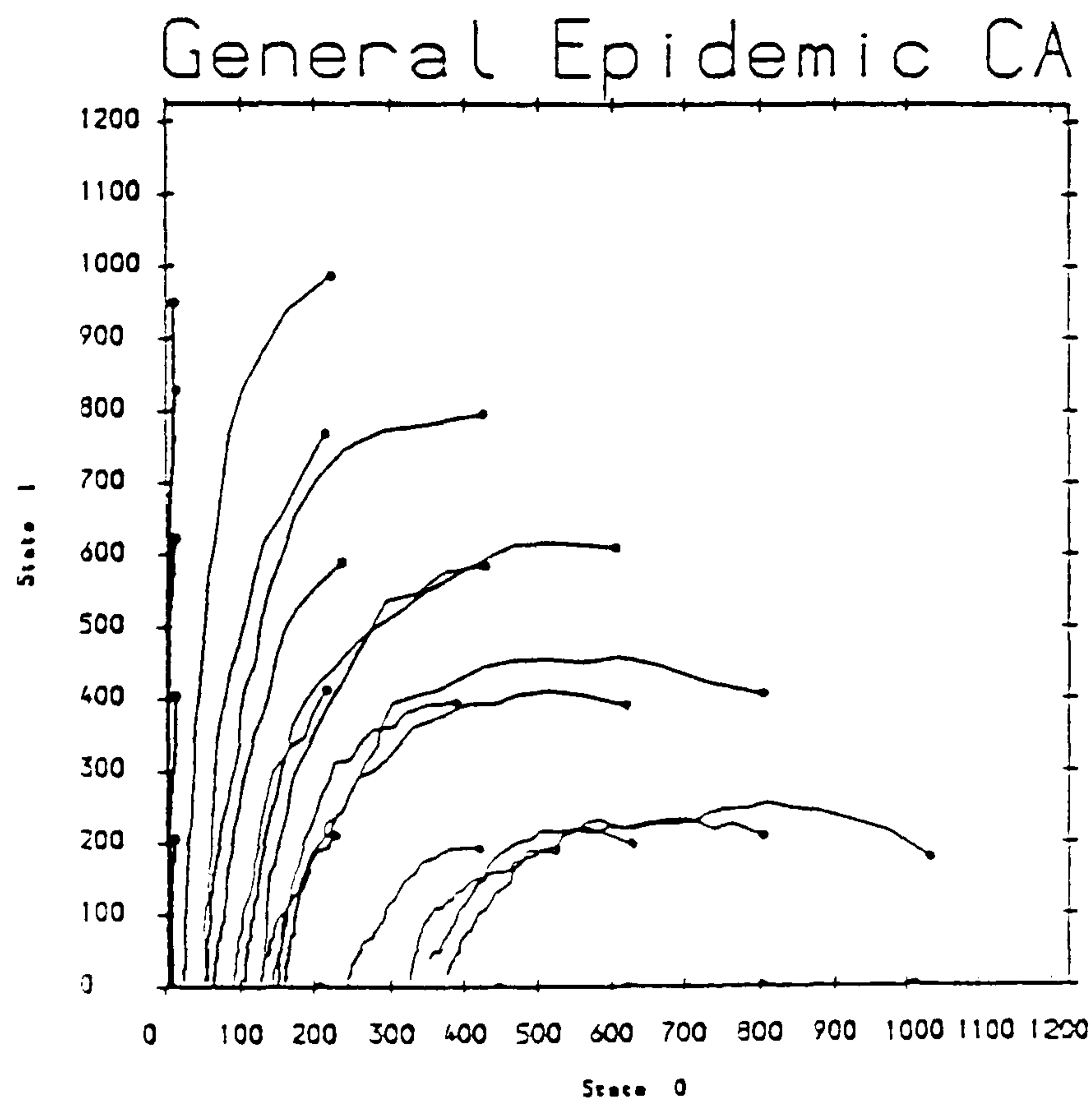
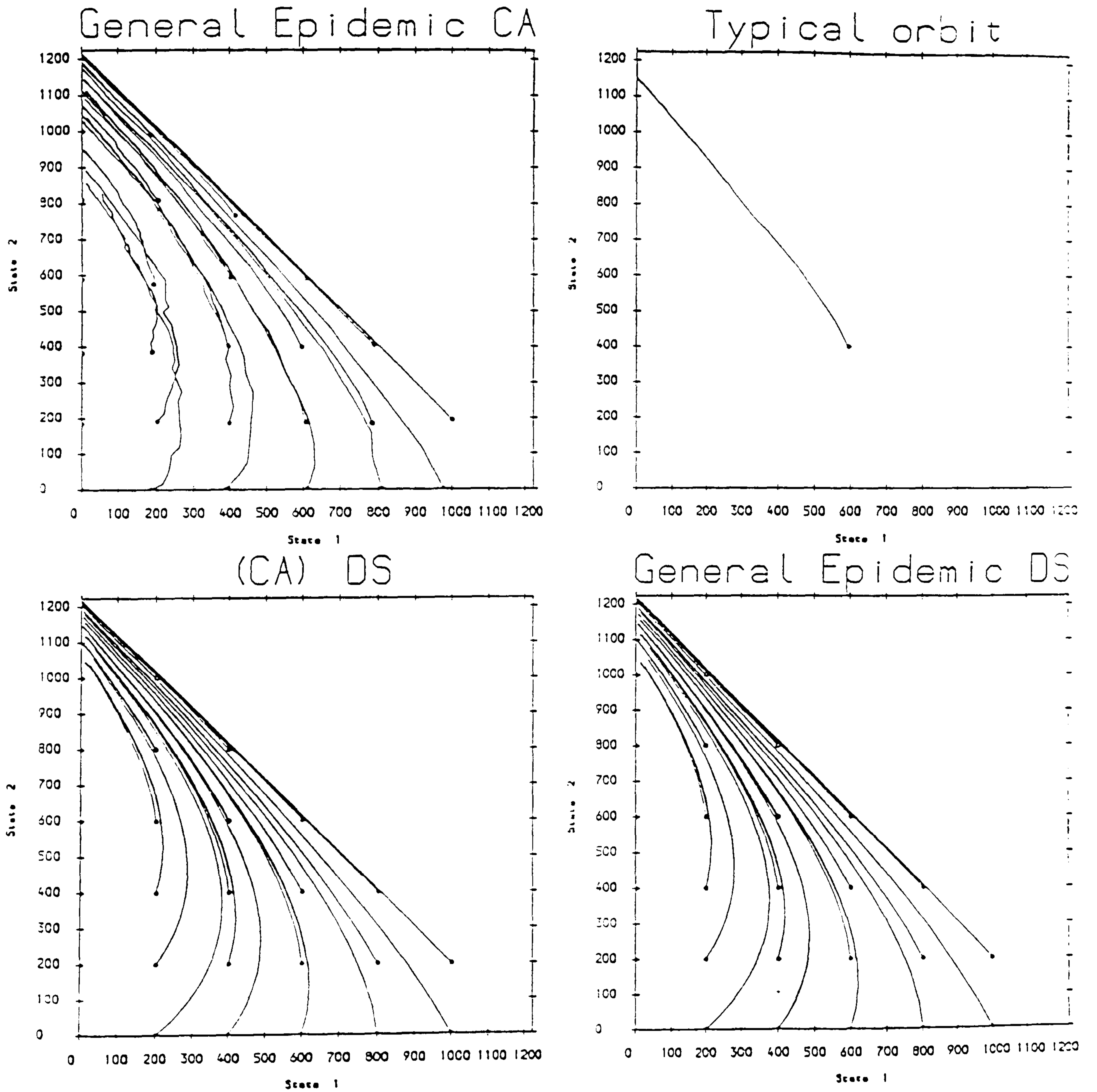




Figure 5.8: Phase diagram for general epidemic model (b).



therefore be defined as follows,

$$\dot{S} = \mu R - \beta SI \quad 5.36$$

$$\dot{I} = \beta SI - \gamma I \quad 5.37$$

$$\dot{R} = \gamma I - \mu R \quad 5.38$$

This system now has the form of a closed loop very similar to systems 5, 6 and 7 in Chapter 3. We would expect therefore either saturated oscillations or limit point behaviour, in fact the transition probabilities normally encountered in epidemics are such that the limit point behaviour is found. The rule set for the equivalent CA is just that for the general epidemic with the addition of the transition of recovered cells to susceptible cells,

$$s_1 \& \{c s_2\} \rightarrow s_2 \mid \text{Pr} = c_1 c / r \quad 5.39$$

$$s_2 \rightarrow s_3 \mid \text{Pr} = c_2 \quad 5.40$$

$$s_3 \rightarrow s_1 \mid \text{Pr} = c_3 \quad 5.41$$

Which gives us the following DS,

$$\dot{n}_1 = [c_3] n_3 - \left[ \sum_{c=0}^r c_1 \frac{c}{r} P(c, n_2) \right] n_1 \quad 5.33$$

$$\dot{n}_2 = \left[ \sum_{c=0}^r c_1 \frac{c}{r} P(c, n_2) \right] n_1 - [c_2] n_2 \quad 5.34$$

$$\dot{n}_3 = [c_2] n_2 - [c_3] n_3 \quad 5.35$$

The temporal evolution of the lattice for the CA model is shown in Figure 5.9. The phase diagram is shown in Figure 5.10.

The dynamical behaviour of this system is more complicated than for the general epidemic, however the agreement between the phase diagrams is still very good. The effect of recovered cells becoming susceptible again moved the limiting region to a point where



Figure 5.9: Lattice evolution for recurrent epidemic CA.

State 1    State 2

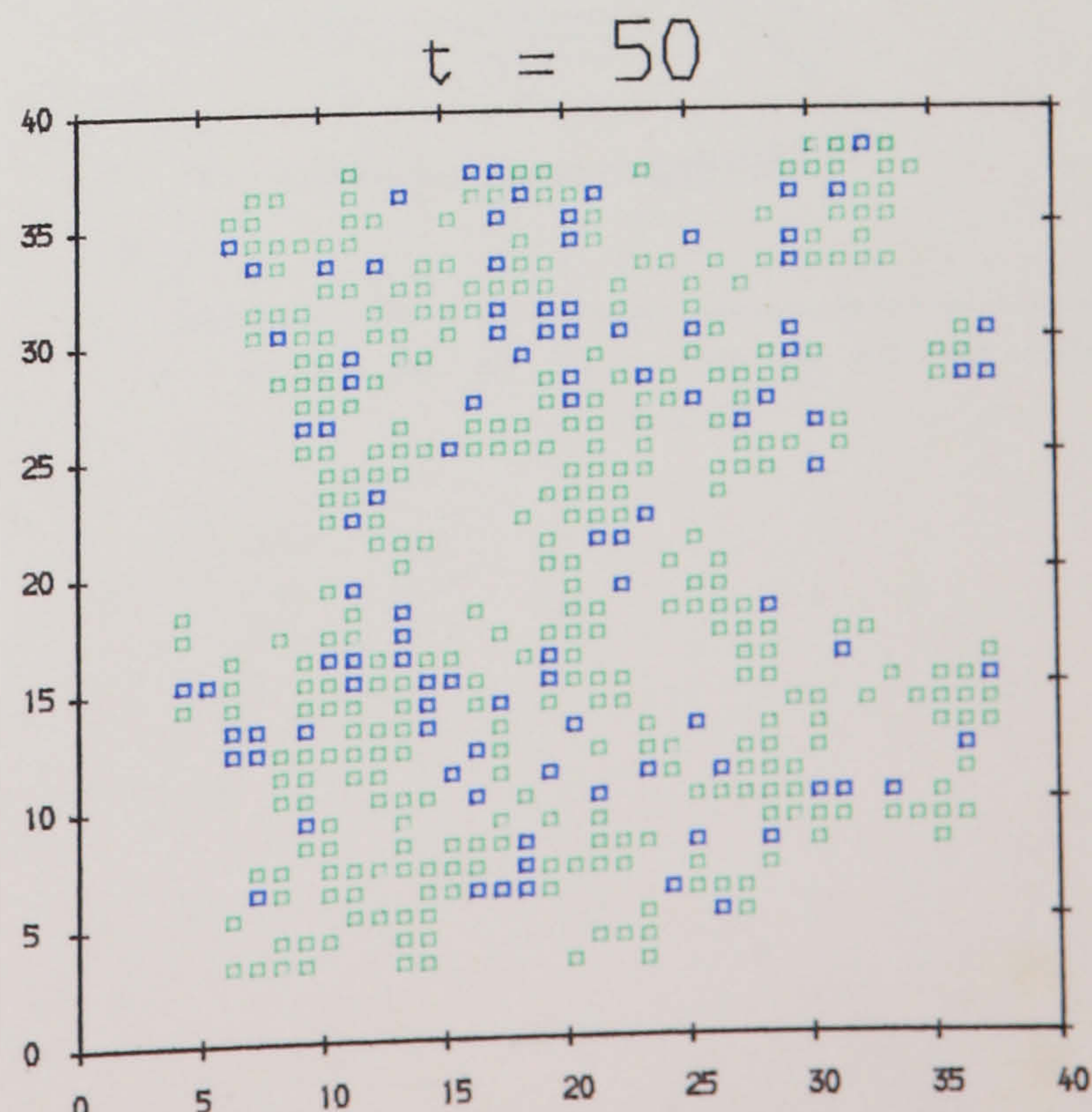
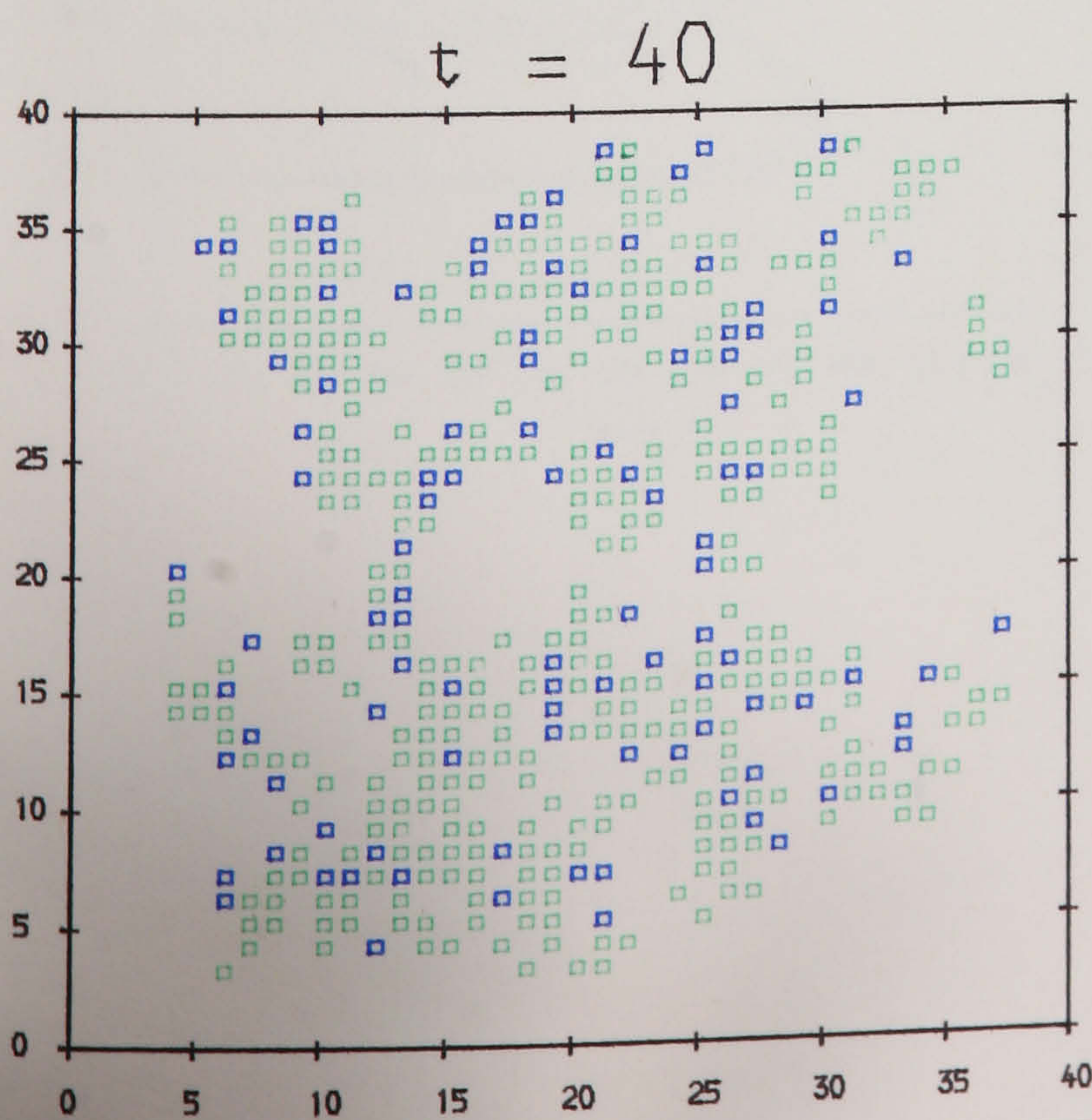
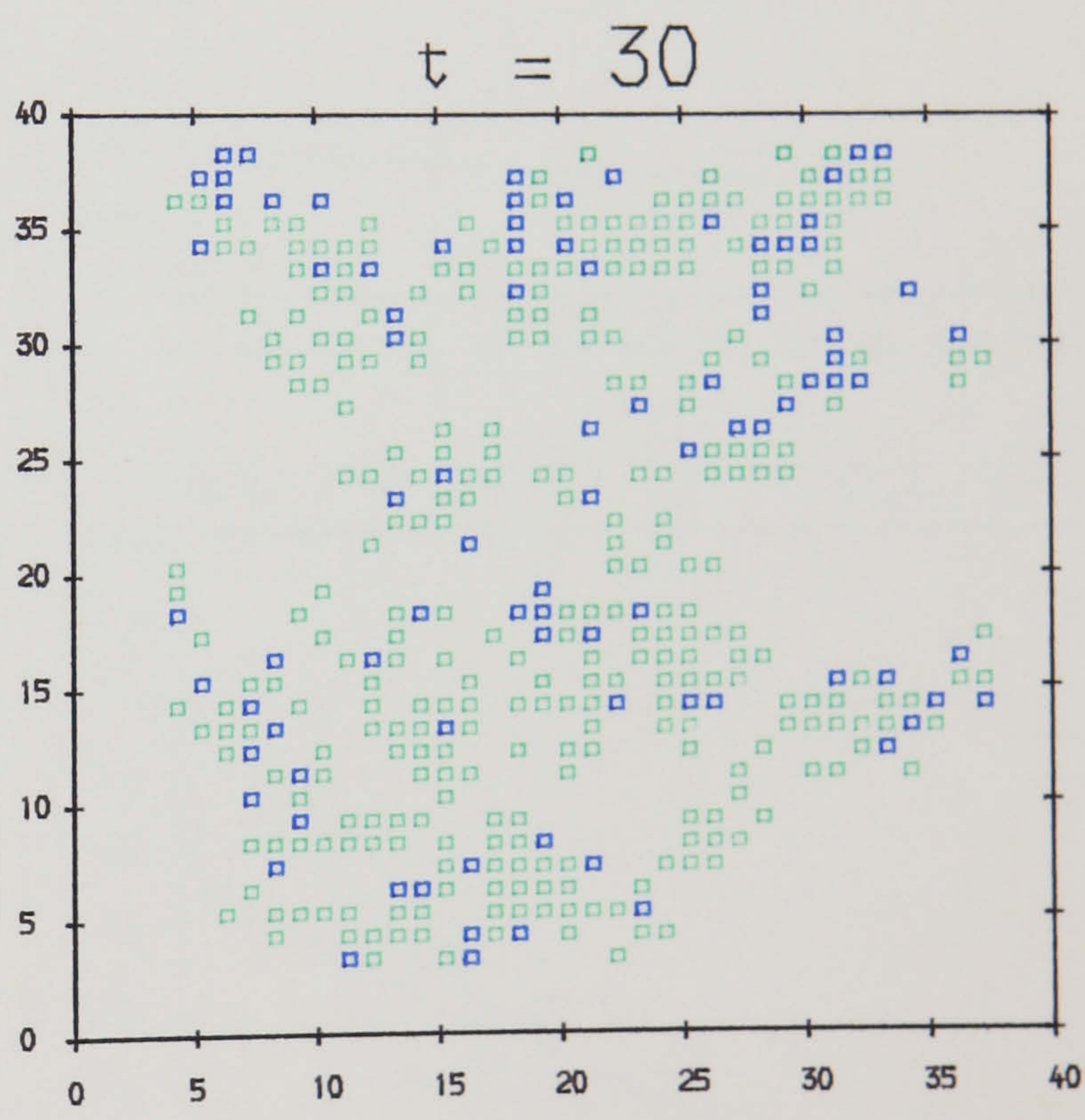
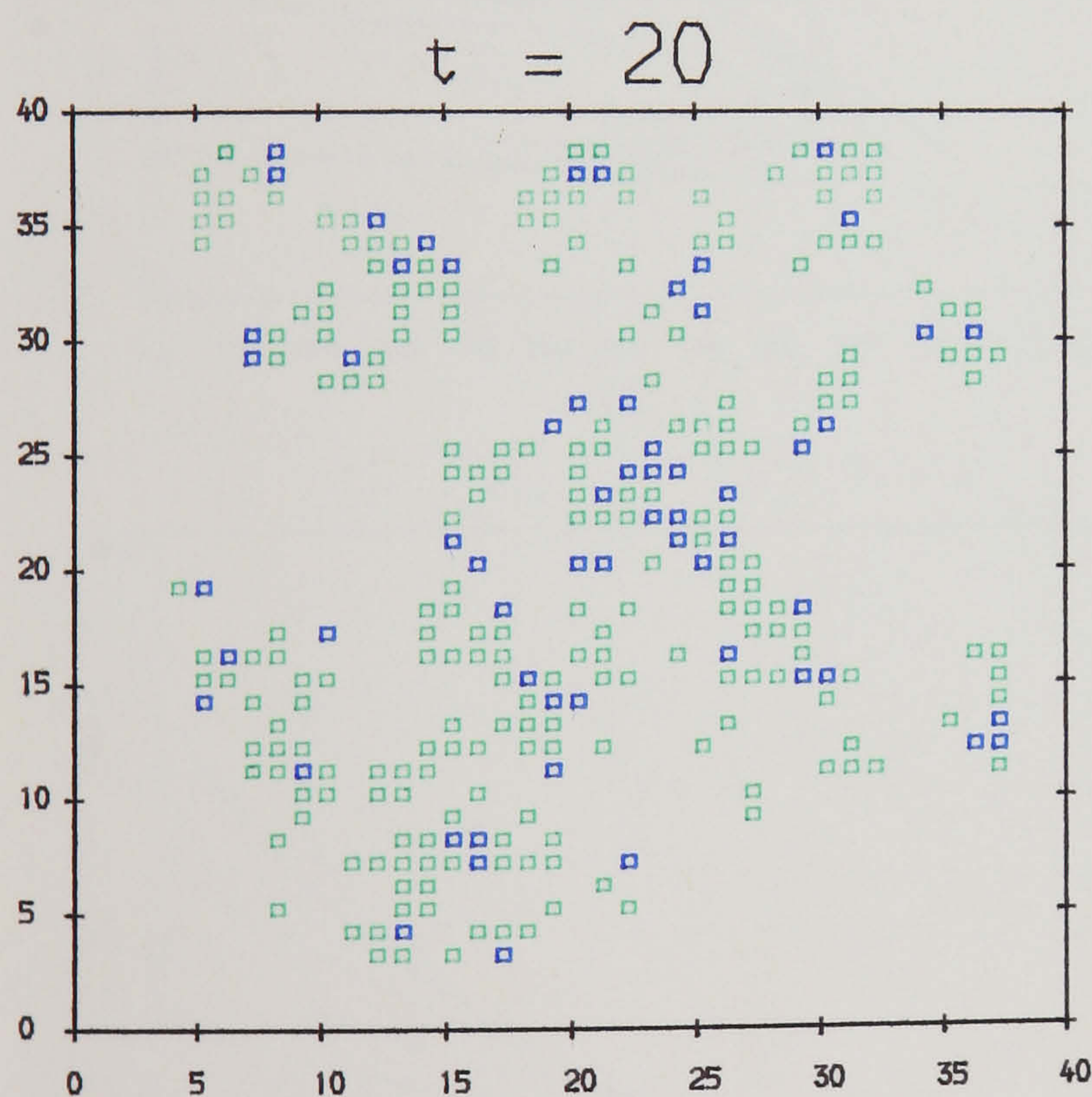
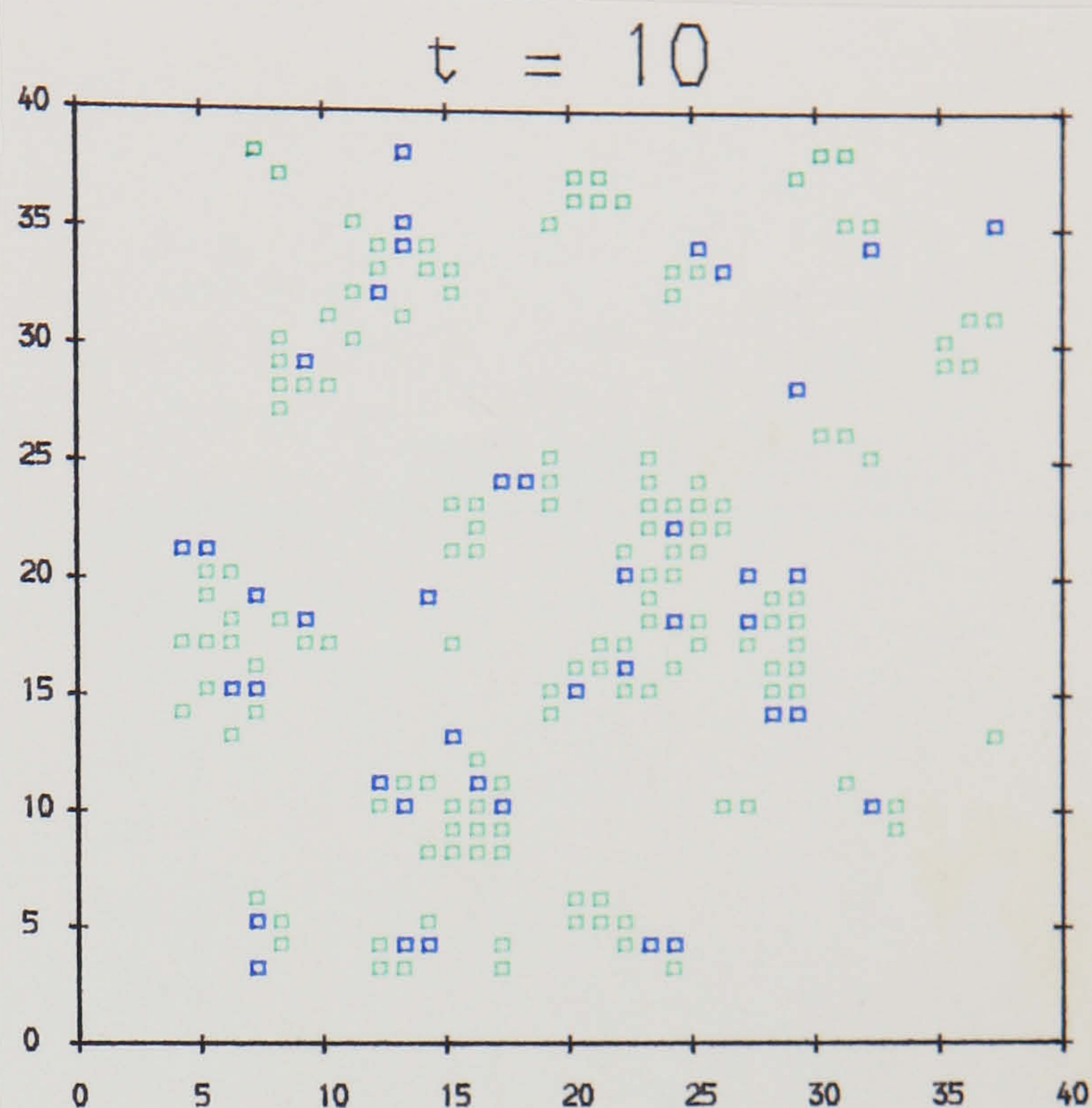
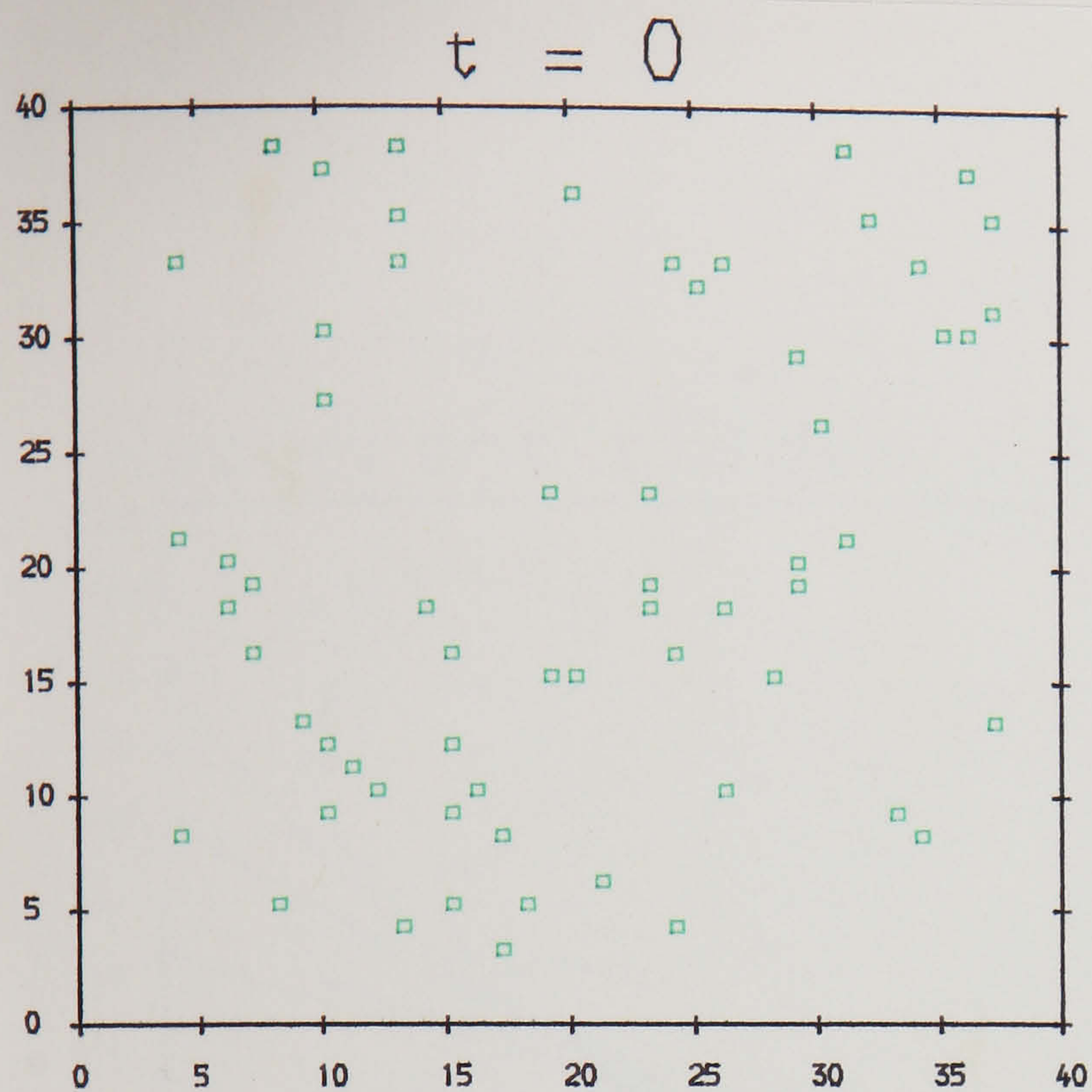




Figure 5.10: Phase diagram for recurrent epidemic model (a).

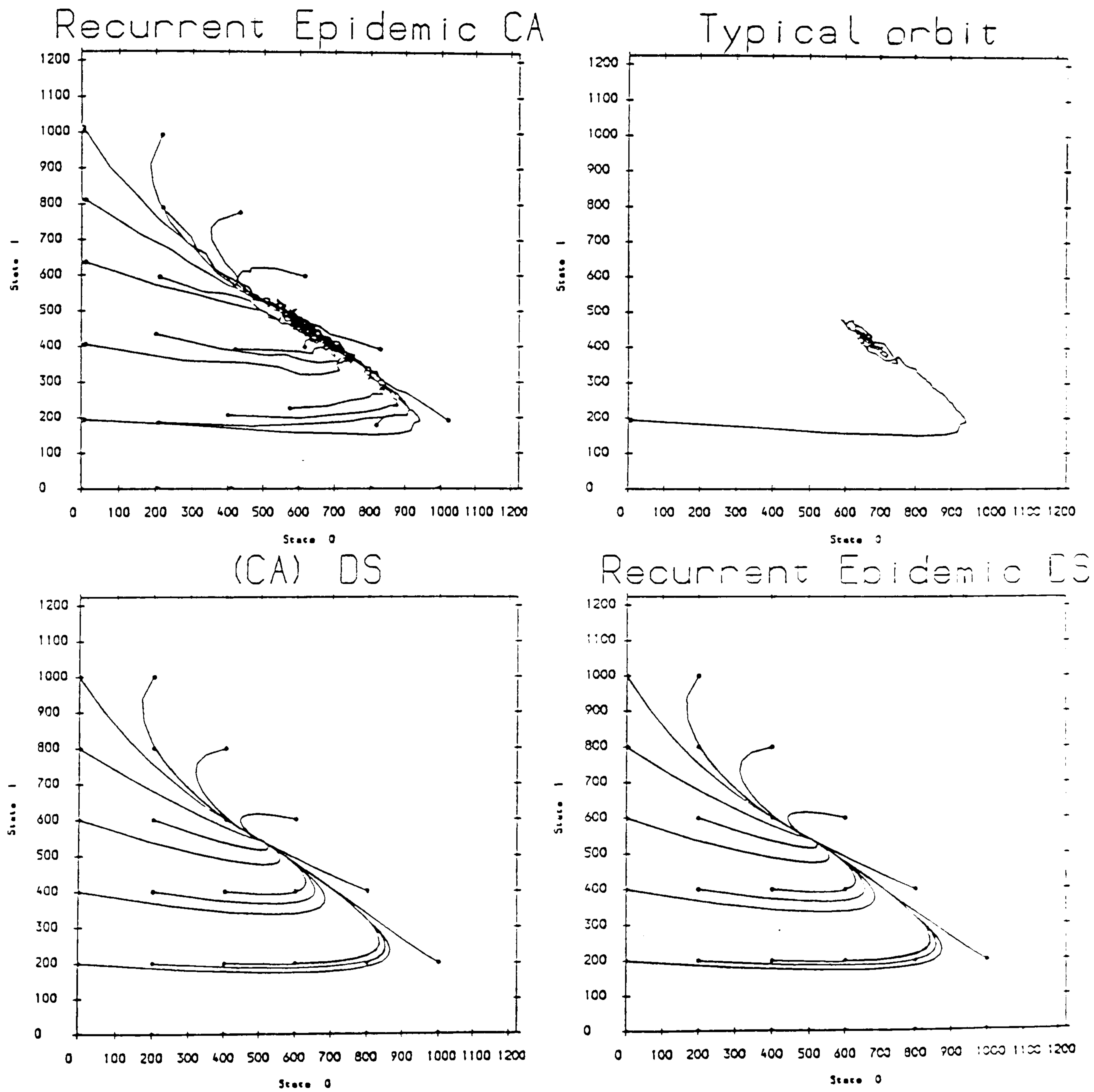
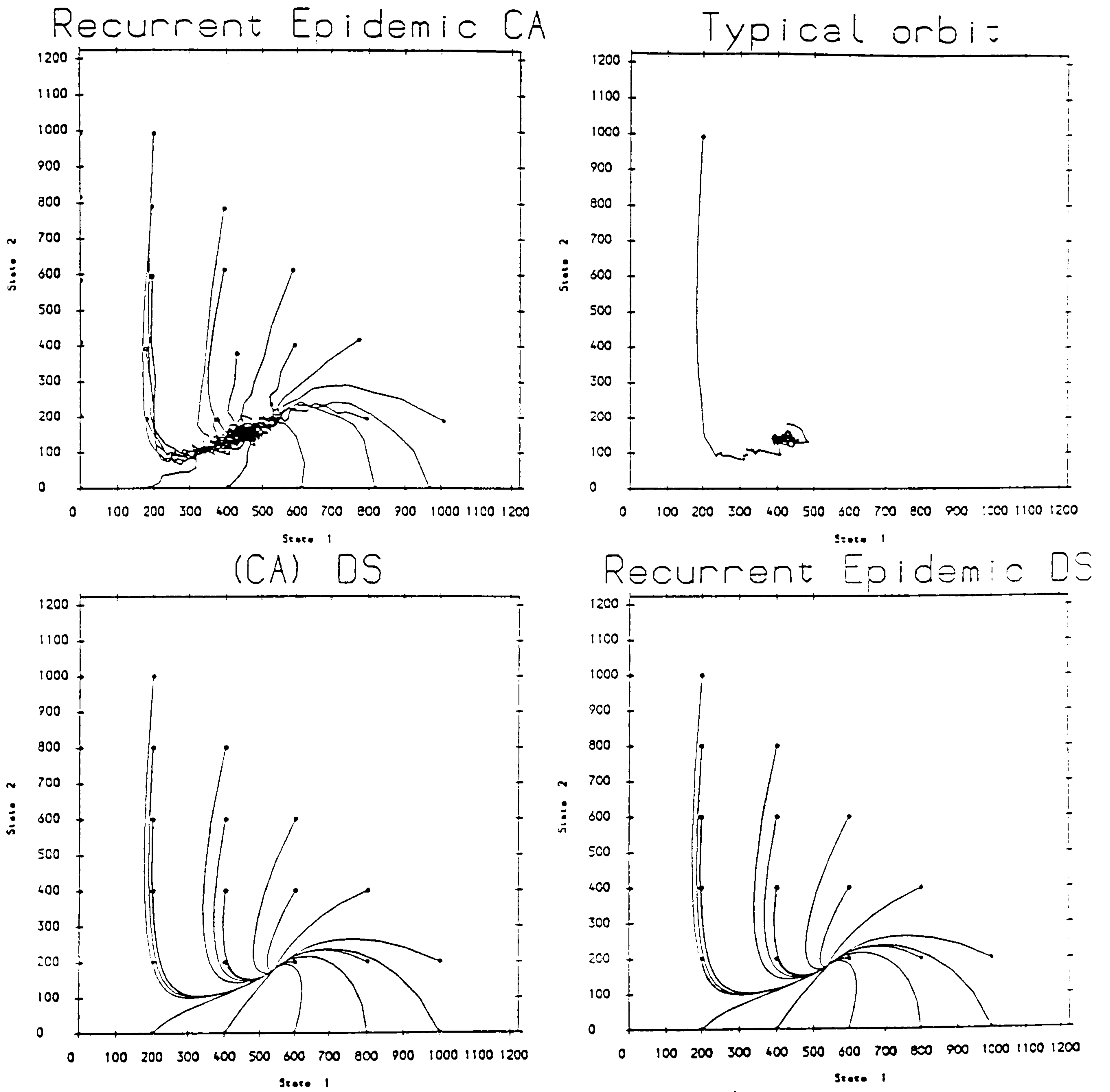




Figure 5.10: Phase diagram for recurrent epidemic model (b).



the number of cells changing from one state to another is exactly balanced by the counteracting change. The initial lattice shown in Figure 5.9 is identical to that for the general epidemic and in the early stages we see that the development of the pockets of infection is similar although the number of infected individuals is much higher. This property is maintained and the lattice develops a much more homogeneous distribution due to the continual exchange between the various states.

We may now consider the generalisation of this procedure to simulate any dynamical system in which changes in the variables can be identified with the transfer of some property. Consider a dynamical equation of the form,

$$\dot{x} = -\theta xyz \quad 5.45$$

Which implies that cells in state  $s_x$  are changing to some other state  $s_w$  with a probability dependent on the number of cells in states  $y$  and  $z$ . We may set up the CA rules to simulate this as follows,

$$s_x \& \{1s_y \& 1s_z\} \rightarrow s_w \mid \text{Pr} = N^2\theta 1/r^2 \quad 5.46$$

...

$$s_x \& \{rs_y \& 1s_z\} \rightarrow s_w \mid \text{Pr} = N^2\theta r/r^2 \quad 5.47$$

...

$$s_x \& \{1s_y \& rs_z\} \rightarrow s_w \mid \text{Pr} = N^2\theta r/r^2 \quad 5.48$$

...

$$s_x \& \{rs_y \& rs_z\} \rightarrow s_w \mid \text{Pr} = N^2\theta r^2/r^2 \quad 5.49$$

If the probability of a given number of cells being in a given state were independent of the number of cells in any of the other states then the equivalent DS could be written as follows,

$$\dot{n}_x = \left[ \begin{aligned} & - \sum_{c=0}^r 1c/r^2 P(1s_z) P(cs_y) - \sum_{c=0}^r 2c/r^2 P(2s_z) P(cs_y) \cdots \\ & - \sum_{c=0}^r rc/r^2 P(rs_z) P(cs_y) \end{aligned} \right] n_x \quad 5.50$$



The individual summations can be simplified by replacing them with their equivalent linear functions which leads to the following simplification,

$$\dot{n}_x = -\frac{n_y}{N} \frac{n_z}{N} n_x \quad 5.51$$

Which we see has the same functional dependence as equation 5.45. However the probability of a given number of cells being in a given state in a finite neighbourhood is not independent of the number of cells in the other states. The general expression for  $c_i$  cells being in state  $i$  in a neighbourhood size  $r$  where  $i=0..j$ ,  $j < k$  is given by,

$$P_c = \frac{\left[ \prod_{i=0}^j \binom{n_i}{c_i} \right] \binom{N - \sum_{i=0}^j n_i}{r - \sum_{i=0}^j c_i}}{\binom{N}{r}} \quad 5.52$$

The functional variation of this expression for two cell counts with the size of the relevant cell populations and the actual variation of the product of two population variables is shown in Figure 5.11. Both functions are monotonically increasing with the population size but the true probability function is much more strongly peaked. Figure 5.12 shows a comparison of the phase diagrams of the following CA and equivalent population genetic DS,

$$s_0 \& \{c_i s_1 \& c_j s_2\} \rightarrow s_1 \mid \text{Pr} = c_1 c_i c_j / (r^2) \quad 5.53$$

$$s_1 \rightarrow s_2 \mid \text{Pr} = c_2 \quad 5.54$$

$$s_2 \rightarrow s_0 \mid \text{Pr} = c_3 \quad 5.55$$

$$\dot{n}_0 = \left[ c_3 \right] n_2 - \left[ \frac{c_1}{N^2} \right] n_1 n_2 n_0 \quad 5.56$$



Figure 5.11: (a) Multi-count transition probability.

$c_1$  = Number of cells in state 1.  $c_2$  = Number of cells in state 2. Graphs show probability of finding counts of  $c_1, c_2$  in a neighbourhood as a function of the population sizes of states 1 and 2.

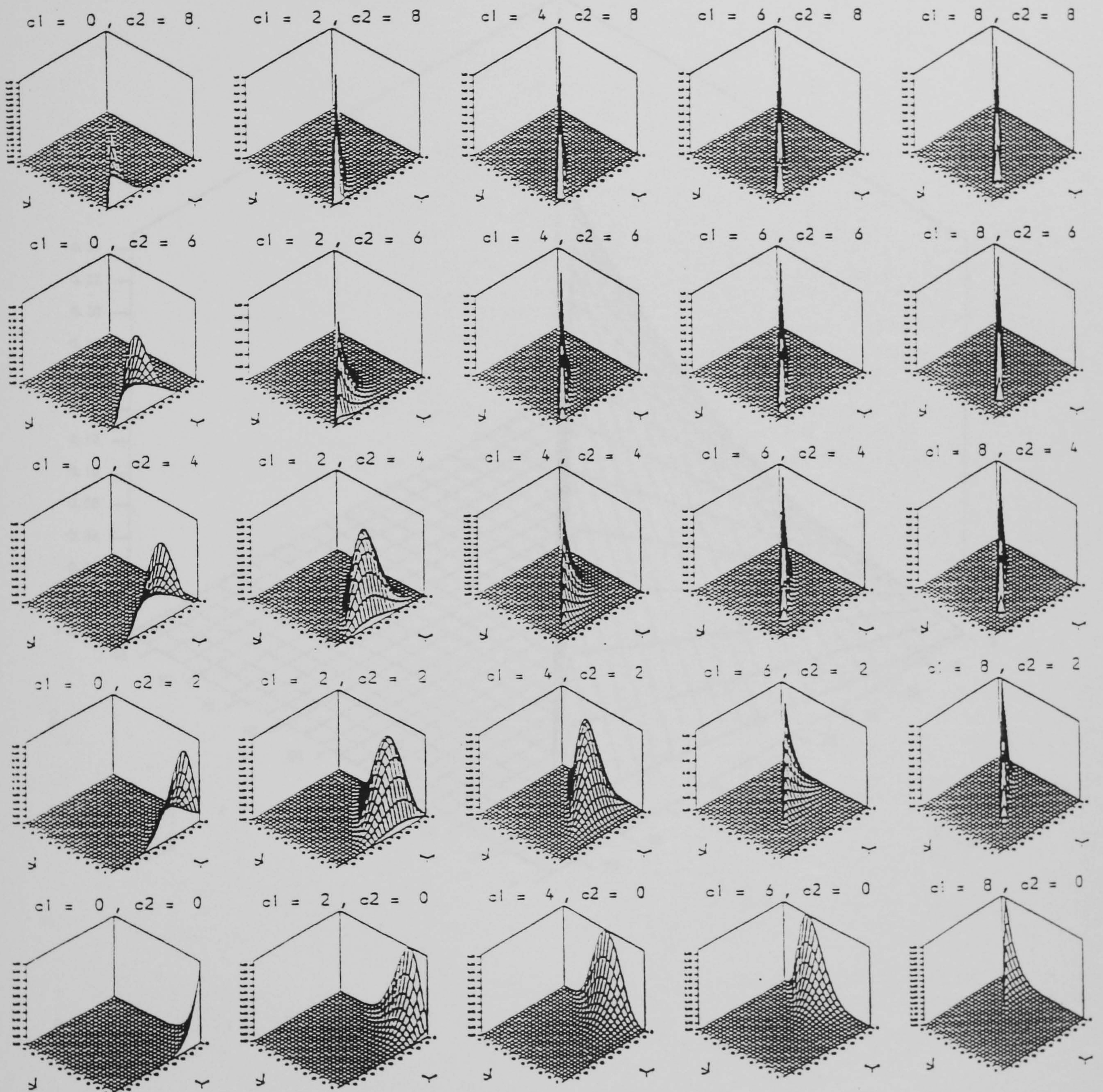




Figure 5.11: (b) Product of two variable populations.

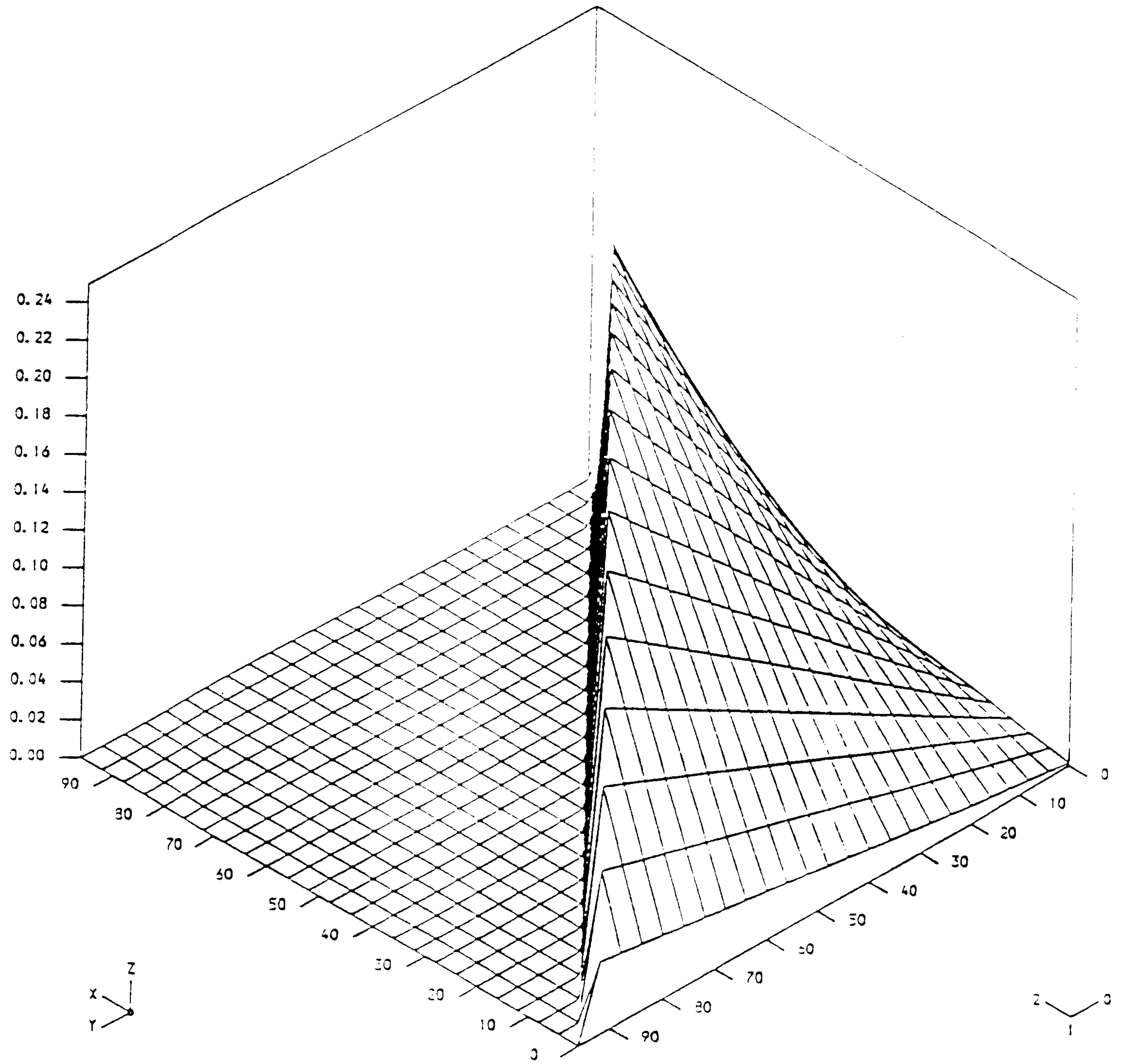
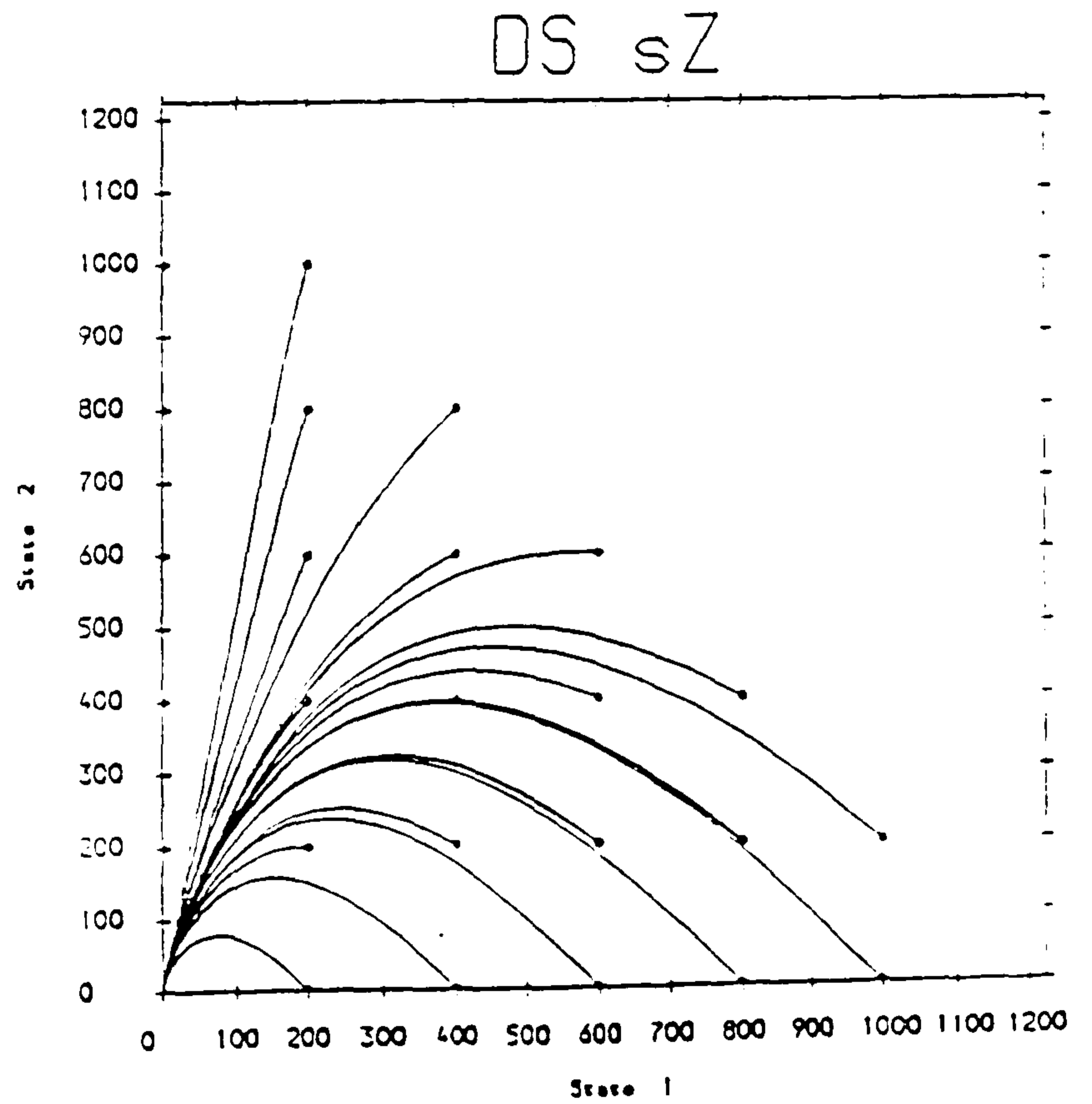
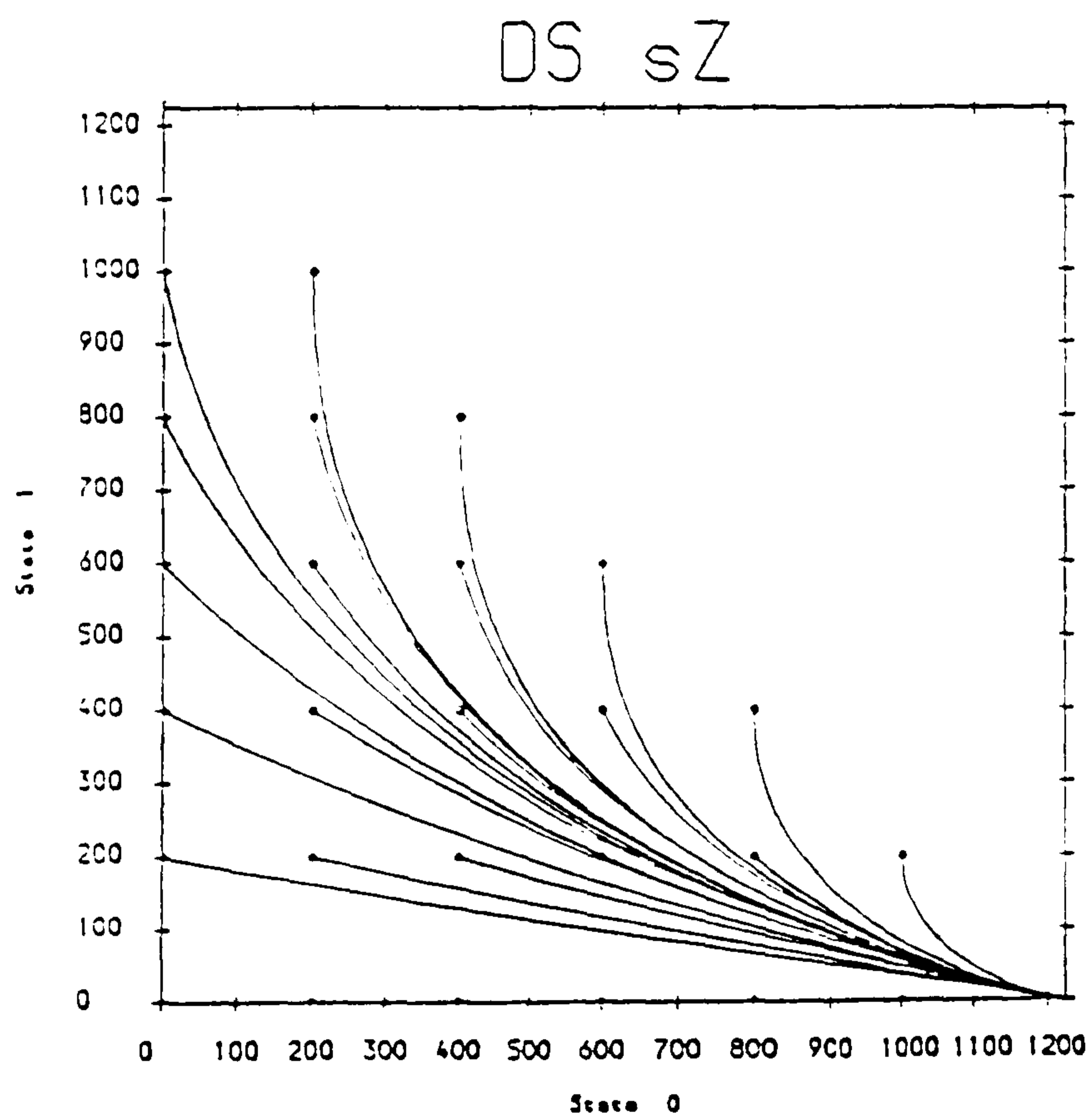
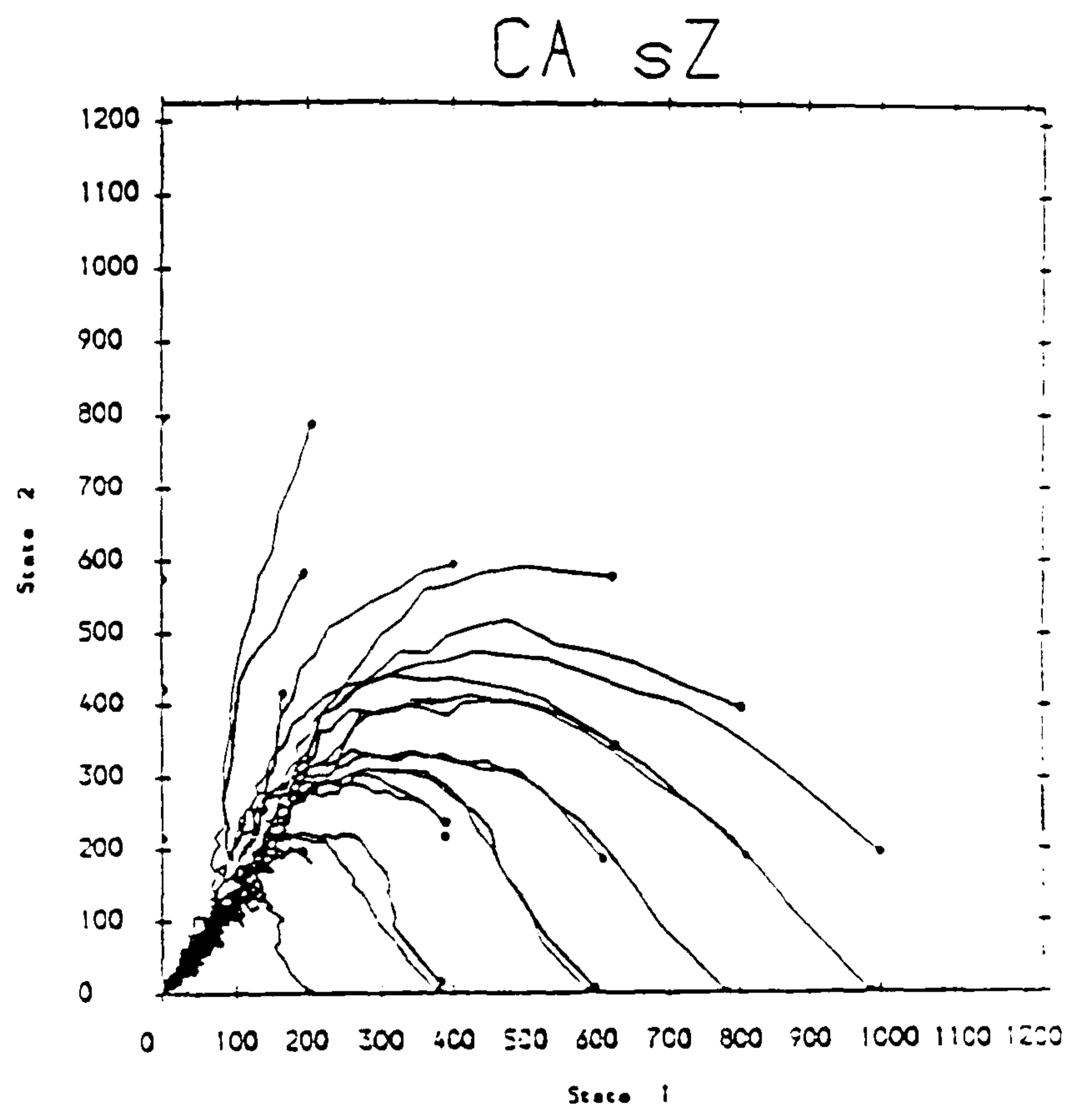
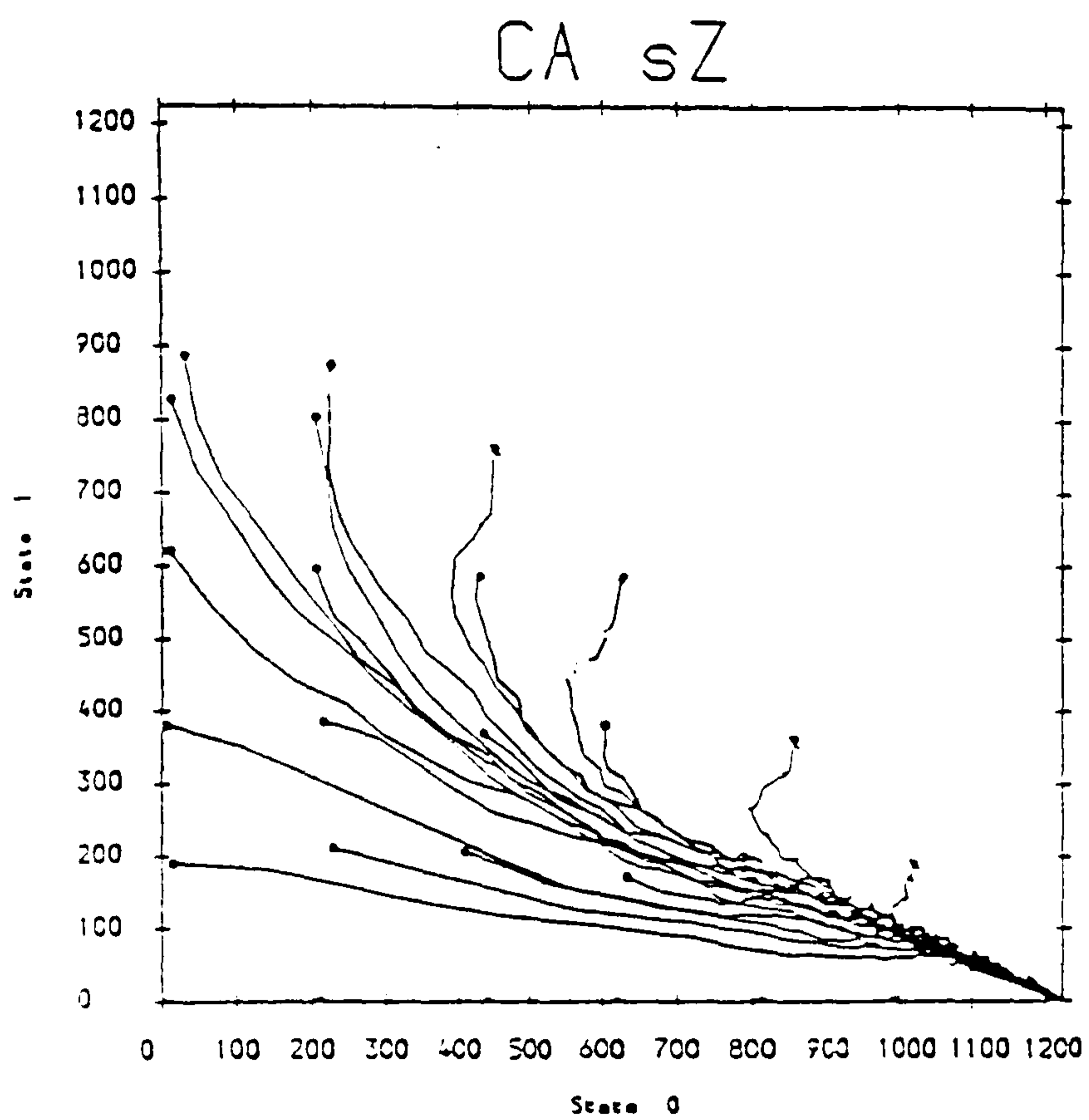


Figure 5.12: Complex population genetic CA and DS.





$$\dot{n}_1 = \left[ \frac{c_1}{N^2} \right] n_1 n_2 n_0 - [c_2] n_1 \quad 5.57$$

$$\dot{n}_2 = [c_2] n_1 - [c_3] n_2 \quad 5.58$$

We see that the agreement is still very good, although there is noticeable deviation for low population values where the probability functions are most different. The reason for this is that the deviation in the probability function from the product of the populations occurs when one of the populations is near saturation. This then limits the possible populations levels of the other states locally in a cell neighbourhood and so the probabilities of the populations levels are not independent.

One problem with this model of populations of organisms is that the units of which the populations consist are spatially fixed. Now if the units represent groups of non-migratory organisms which essentially remain in a spatially fixed territory the model will be adequate. However if we need to model groups or individuals which are spatially mobile then we need to include a process for moving cell states on the lattice. One way to do this would be to make quiescent cells take the state of one of their neighbourhood cells, but within the CA framework there is no natural way to set the cell from which the state has just moved to be quiescent. Another possibility would be the empty cells containing non-quiescent states and fill cells in the quiescent state probabilistically. This method has two severe drawbacks. Firstly the number of individuals would not be fixed but would perform a random walk about the initial population level (assuming no other growth or decay processes) in a similar way to the Diffuser CA in Chapter 2. Secondly the local population density would fluctuate in the same random fashion because we are actually performing a type of mixing of the lattice. This means that the spatial variations of the populations which we are trying to model would be destroyed.

The problem lies in the fact that in the CA framework the states are a property of the cells which are spatially fixed and it is the cells that compute the dynamics. The mobility we require is a property of the states. The solution therefore is to extend the CA framework so that an evolutionary step of a cell proceeds in two stages. Firstly the cell evolves normally using its standard rules, secondly non-quiescent cells “ask” the cell in the direction which its state wants to move (as defined by another rule) if it is quiescent. If it is it sends its state to this cell and changes itself to the quiescent state. The important facet of this extension to the CA framework is that it does not alter the basic simplicity of the inter-cellular connections and so the realisation of the lattice in hardware will not be complicated. Normally the “mobility rules” will be stochastic both in action and in direction to achieve a general “milling around” of the states. For synchronous CA there is obviously a potential conflict in that two or more non-quiescent cells may attempt to send their state to the same quiescent cell. However our population modelling CA are asynchronous because they are probabilistic, if we ensure that the state transition probabilities are low (or equivalently  $\Delta T \ll 1$ ), and so this problem will not occur.

### **Summary**

We have shown that CA can be constructed to model physical processes in which an analogy with spatial variables changing state can be made. We have shown that the dynamical behaviour of these CA in terms of their DS is identical to the classical dynamical equations describing the physical processes. This can be considered as a justification of these equations since our CA is closer to an exact simulation of the processes rather than a simplified model. The CA and its equivalent DS provide a link between the deterministic models of population genetics and the stochastic models which are important when population numbers are very low.



Our generalisation of this technique to DS with products of many variables indicates that if these are meant to represent local interactions or changes of state then the DS will diverge from the true behaviour of the system when high densities are present. This is a well known result, but here we have a theoretical explanation for it.

Our extension of the CA framework can be viewed in the following way. In a idealised physical system the particles move around at a constant velocity until they meet and interact. When they interact we can imagine them “computing” their new velocity or state. If we consider space to be divided into discrete cells, we can imagine this processes has the cells passing particles from one to another until an interaction is encountered whereupon the relevant cell computes the new dynamics. In terms of our CA the first process is performed by our extension to the CA framework, while the second is performed by the classical CA behaviour. So we now have a very close relationship between our CA framework and certain idealised physical systems.

## 6. Conclusions

In Chapter 2 we described a number of different mechanisms currently employed by workers to control the evolution of the CA lattice. These were synthesised into a CA framework which was flexible enough to allow us to explore the complete range of CA behaviours. The framework is closely related to other computational structures such as neural networks, threshold functions and logical operators. The rule space is however of a more manageable size than the classical CA, so that it would be feasible to explore a large fraction of the entire rule space in order to discover, for example, the detailed distribution of behaviours in the rule space. Finally we described six CA which were constructed utilising specific parts of the general framework to exhibit behaviour typical of CA with that particular property.

In Chapter 3 we re-introduced a statistical mechanical approach to CA which has appeared in various forms in the literature previously but has not been extensively developed. We showed that this leads to a system of dynamical equations which model the changes in the state populations as the CA evolves, and that the accuracy of this simulation can be quantified. We looked in detail at the calculation of the entropy of a CA lattice and various factors which effect it, the relationship between synchronous and asynchronous evolution and deterministic and probabilistic evolution. This led us to several variations on the standard SCA, mixed CA, random neighbourhoods, weighted noise and probabilistic evolution which all have the effect of maintaining a random distribution of states and preventing the self-organisation or characteristic patterns of the SCA from developing. In conclusion we proposed that a discrete, probabilistic, totalistic CA can be viewed as equivalent to a continuous, deterministic dynamical system.



In Chapter 4 we looked at the application of our statistical mechanical approach to the various classes of CA. We have seen that the state population equations do indeed model the actual behaviour of the CA state populations for both the deterministic and probabilistic cases. As we expected the dynamical equations agree best with the mixed CA since the self-organising behaviour of the unmixed lattice causes deviations from a homogeneous distribution of states. We can therefore apply the DS theory approaches used for the threshold CA to all the systems in order to obtain information on the limiting behaviour. However, we have also seen that deviations between the behaviour of the CA and the DS can occur, in particular limiting points/regions may be shifted and entirely new limiting regions can appear in the DS phase space. We believe that in the limit of  $\Delta t \rightarrow 0$  the behaviour of the two systems will converge, although this remains to be demonstrated empirically. For class 3 systems substantial differences occur in certain cases, and these need further investigation. However, we feel that these deviations may be connected with the very high rates of change of the state populations for the systems we have studied. This may lead to the DS behaving unstably when  $\Delta t$  is finite, but the distributed nature of the CA may “damp” the instability.

In Chapter 5 we developed a practical application of the statistical mechanical approach. We have shown that CA can be constructed to model physical processes in which an analogy with spatial variables changing state can be made. We have shown that the dynamical behaviour of these CA in terms of their DS is identical to the classical dynamical equations describing the physical processes. This can be considered as a justification of these equations since our CA is closer to an exact simulation of the processes rather than a simplified model. The CA and its equivalent DS provide a link between the deterministic models of population genetics and the stochastic models which are important when population numbers are very low.

Our generalisation of this technique to DS with products of many variables indicates that if these are meant to represent local interactions or changes of state then the DS will diverge from the true behaviour of the system when high densities are present. This is a well known result, but here we have a theoretical explanation for it.

We then proceeded to consider a problem of modelling spatially mobile state populations. In order to allow this type of behaviour within our general framework we introduced an extension to the CA framework which essentially resulted in a two-phase evolution, one in which the dynamics are computed and the other in which the static interactions are computed. We noted that this leads to a very close relationship between our CA framework and certain idealised physical systems.

Wolfram has suggested that his four qualitative classes,

- (1) Evolve to a homogeneous state.
- (2) Evolve to simple separated periodic structures.
- (3) Evolve to chaotic aperiodic patterns.
- (4) Evolve to complex patterns of localised structures.

can be identified with the limiting behaviour of continuous dynamical systems, limit points, limit cycles and chaotic or strange attractors. Class 4 having no direct analogue. There is however a slight inconsistency in this classification in that any limiting fixed state of the lattice is a limit point and homogeneous lattices are a subset of these. The homogeneous state is in fact equivalent to a trivial solution of a dynamical system in which all variables are zero. A further point to note is that limit points are actually a subset of the limit cycle behaviour with a period of one. This suggests that we may either define class 1 as systems with trivial limiting behaviour and class 2 as limit point/cycle



behaviour or class 1 as limit point behaviour and class 2 as limit cycle behaviour. We feel the former classification is preferable since the distinction between systems with a trivial zero solution and those with a non-zero solution is quite important whereas the distinction between limit point and limit cycle behaviour especially in the context of CA is not so important. Wolfram has also noted that certain sub-classes of behaviour can be distinguished in class 3 and 4[43]. This is to be expected since these two classes represent the more complex and with, increasing  $k$  and  $r$ , most abundant behaviour. Detailed investigation of the possible sub-classes may reveal clues to origin of the classes behaviour.

An important remaining problem is the relationship of class 4 CA to the other classes and its analogous behaviour in continuous dynamical systems. We have shown that the class 4 behaviour results in the phase diagram trajectories appearing to have a level of randomness superimposed on the trajectories of the equivalent continuous dynamical system. This is because the changes in the local numbers of cells in each state do not bear a simple relationship to the structured changes in typical class 4 configurations. The interesting point is that the self-organising behaviour does not perturb the global evolution of the state populations. Furthermore the global evolution of the state populations cannot be driven by the self-organising behaviour since the DS does not have this property. This suggests that the structured configurations act and interact only locally and there is limitation to the range over which correlations occur on a lattice evolving from an initially random state at least.

The above discussion suggests an interesting open question which is how do the behaviours of CA and their state populations equations compare when the lattice is evolving from a minimal seed of non-quiescent states (class 3) or a single special

configuration (class 4)? We suspect that in these rare instances in terms of the phase space the behaviour of the DS will be trivial but not that of the CA. This may provide a means of identifying the generic CA behaviour which is superimposed on that global behaviour of the DS.

A further question we may ask is how does the lattice evolve from an initially spatially non-uniform distribution of states and how does the behaviour of the state population dynamical systems compare with this. In terms of the dynamical systems we have a new level of complexity since our state populations will be a function of spatial position on the lattice as well as a function of time. In order to model the state populations in practice we would have to sub-divide the lattice into small groups of cells and model the number density of the states in each of the groups of cells. If we consider modelling the number density of cells at each cell in the lattice then we have the same number of state variables for the DS as for the CA. In fact we have essentially arrived back at the CA, since the number density in a region of size one cell can be represented by the states of the cell exactly.

An interesting result is that the behaviour of the DS of a CA is scale invariant which implies that the global behaviour of the CA remains the same regardless of the size of the lattice. This is only true, however, if the CA maintains the uniform distribution of states on the lattice. As Wolfram points out[43] for class 4 CA more and more complex structures may emerge as larger lattices are considered. For example structures which periodically produce propagating structures, if these structures are also immune to interference from other structures then the limit will be an infinite number of non-zero cells. Very large lattices may also allow self-reproducing structures which again may be immune to interference. These structures will however be extremely rare and so statisti-



cally we may be able to characterise the limiting behaviour more simply, however if one of these structures were to occur it would dominate the limiting behaviour of the CA leading to a very different limit from our statistical answer.

As we mentioned in Chapter 3, CA are fully discrete systems and as such their relationship with continuous systems is an important problem. The discreteness in space and time is comparable with a finite difference approximation to a continuous system. As long as “stable” discretisation is used the exact continuous results will be approximated more and more closely by the discrete system as the size of the space and time steps are decreased. We should note however that we normally obtain stability conditions on finite difference schemes which require the time step to be small compared with the space step. This gives some justification for investigating the comparative behaviour of the CA and DS as the time step or rule probability is decreased rather than the effect of larger neighbourhoods on larger lattices (which would be equivalent to higher space resolution). For SCA the type of approximation is Jacobi rather than Gauss-Seidel in finite difference terms since the past state of the system is used to calculate the new state rather than the new state of some of the cells. It is well known that this type of finite difference procedure often leads to stability problems and slow convergence. We may note that the stability problems are analogous to the “feedback catastrophe” problems in SCA noted in Chapter 3. As we argued there ACA correct this problem in that the calculation of the new state of a cell will use some new values since some of the cells in its neighbourhood will have evolved more times than the cell itself. CA have another form of discreteness which corresponds to an extreme form of round-off. Each cell can have only a few ( $k$ ) possible values rather than the  $2^{16}$  or  $2^{32}$  normal for computer representation of a variables value. Our results have shown that if the cells are used to represent spatially distributed variables in the form of a density distribution then this extreme form of round-off

does not cause a problem, essentially because we are representing the variables value in a distributed way. Wolfram notes[43] that specific examples of CA approximations to physically important partial differential equations would be valuable. We have shown that a certain class of partial differential equations have a direct relationship with CA and can be approximated very accurately.

The behaviour of deterministic CA can appear essentially random[61]. For example, if the initial state of the lattice is chosen randomly, a given spatial and/or temporal sequence of a set of cells will appear random under the normal test for randomness[62]. This is the same apparent randomness we would observe in the binary digits in the memory of a computer executing a finite difference approximation to a partial differential equation. We are looking at local dynamics of the system which bear such a complex relationship to the structured global behaviour that no pattern can be discerned and the behaviour is apparently random.

Many types of noise or imperfections which can be introduced into the classical CA have been proposed[43, 79, 94] for example,

- (a) Probabilistic application of the rule.
- (b) Multiple rules applied probabilistically.
- (c) Different sites obeying different rules.
- (d) Different neighbourhoods for different sites.
- (e) Different sites updated at different times.

However many of these are essentially the same or very closely related. Probabilistic rules are equivalent to asynchronous updating of the cells as we have shown in Chapter



3. Multiple rules applied probabilistically is a more general form of different cells obeying different rules which is closely related to random boolean networks or Kaufmann automata[77, 78]. Different neighbourhoods for different sites (which will have the same  $r$  otherwise difficulties are encountered in interpretation of the rules under the normal rule specification schemes) is a special case of choosing the cells of the neighbourhood randomly from the entire lattice. We have shown that this is equivalent to mixing the lattice at each time step. The noise which we have mainly considered, probabilistic rules, mixing, random neighbourhoods, leaves the state populations unaltered and only changes the local structure of the lattice. The position in the state population phase space therefore remains unaffected and we don't see any dramatic changes in behaviour. Wolfram notes[43] that there is often a critical magnitude of imperfection which results in a phase transition. We can understand this as a shift in the phase space from one domain of attraction to another. The important phase space will normally be that of the state populations unless the behaviour of the CA is driven by its self-organising behaviour in which case the global lattice state space will be more appropriate.

An important remaining problem is how to describe the behaviour of CA in terms of information processing operations. We hope that this work, in providing a description of the dynamical behaviour of CA, is a first step towards solving this important problem.

## References

1. J. von Neumann, in *Theory of self-reproducing automata*, ed. A. W. Burks, University of Illinois Press, 1966.
2. V. Aladyev, "Survey of research in the theory of homogeneous structures and their applications," *Mathematical Biosciences*, vol. 22, p. 121, 1974.
3. D. A. W. Thompson, *On growth and form*, Cambridge University Press, 1961.
4. E. F. Moore, "Machine models of self-reproduction," in *Essays on cellular automata*, ed. A. W. Burks, 1962.
5. G. T. Herman, "Computing ability of a developmental model for filamentous organisms," *Journal of Theoretical Biology*, vol. 25, p. 421, 1964.
6. M. A. Arbib, "Simple self-reproducing universal automata," *Information Control*, vol. 9, p. 177, 1966.
7. E. F. Codd, *Cellular Automata*, Academic, 1968.
8. A. Lindenmayer, "Mathematical models for cellular interactions in development," *Journal of Theoretical Biology*, vol. 18, p. 280, 1968.
9. S. Ulam, "Some ideas and prospects in biomathematics," *Annual Review of Biophysics and Biochemistry*, vol. 1, p. 277, 1969.
10. H. Nishio, "Real-time sorting of binary numbers by 1D cellular automata," Kyoto University Report, 1965.
11. P. C. Fischer, "Generation of primes by 1D iterative array," *Journal of the ACM*, vol. 12, p. 388, 1965.
12. A. J. Atrubin, "A 1D real-time iterative multiplier," *IEEE T Computer*, vol. 14, p. 394, 1965.



13. S. N. Cole, "Real-time computation by n-dimensional iterative arrays of finite state machines," *IEEE T Computer*, vol. 18, p. 349, 1969.
14. M. Gardener, "Mathematical Games," *Scientific American*, vol. 224, p. (Jan)104, (Feb)112, (Mar)106, (Apr)114, 1971.
15. R. T. Wainwright, *Lifeline*, vol. 1-11, 1971-1973.
16. M. Gardener, "Mathematical Games," *Scientific American*, vol. 226, p. (Jan)104, 1972.
17. R. T. Wainwright, "Life is universal," Proceedings of the Winter Sim Conf, p. 448, ACM, Washington DC, 1974.
18. E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal and triangular arrays," *Communications of the ACM*, vol. 15, p. 827, 1972.
19. J. R. Welberry and R. Galbraith, "A 2D model of crystal-growth disorder," *Journal of Applied Crystallography*, vol. 6, p. 87, 1973.
20. R. M. Baer and H. M. Martinez, "Automata and biology," *Annual Review of Biophysics and Biochemistry*, vol. 3, p. 255, 1974.
21. T. Kitagaum, "Cell space approaches in biomathematics," *Mathematical Biosciences*, vol. 19, p. 27, 1974.
22. J. R. Welberry and G. H. Miller, "A phase transition in a 3D growth-disorder model," *Acta Crystallographica A*, vol. 34, p. 120, 1978.
23. R. L. Dobrashin, V. I. Kryokov, and A. L. Toom (eds.), *Locally interacting systems and their applications in biology: Lecture Notes in Mathematics 652*, Springer-Verlag, 1978.
24. H. Gerola and P. E. Seiden, "Stochastic star formation and the spiral structure of galaxies," *Astrophysical Journal*, vol. 223, p. 129, 1978.

25. K. Preston, M. J. B. Duff, S. Levialdi, P. E. Norgren, and J. Toriwaki, "Basics of cellular logic with some applications in medical image processing," *Proceedings IEEE*, vol. 67, p. 826, 1979.
26. H. Haken, "Cooperative phenomena in systems far from thermal equilibrium and in none physical systems," *Reviews of Modern Physics*, vol. 47, p. 67, 1975.
27. G. Nicolis and I. Prigogine, *Self-organisation in non-equilibrium systems*, Wiley, 1977.
28. J. M. Greenberg, B. D. Hassard, and S. P. Hastings, "Pattern formation and periodic structures in systems modelled by reaction diffusion equations," *Bulletin of the American Mathematical Society*, vol. 84, p. 1296, 1978.
29. H. Haken, *Synergetics*, Springer-Verlag, 1978.
30. H. Haken, *Pattern formation by dynamic systems and pattern recognition*, Springer-Verlag, 1979.
31. G. Nicolis, G. Dewel, and J. W. Turner, "Order and fluctuation in equilibrium and non-equilibrium statistical mechanics," *Proceedings of the XVIIth International Solvay Conference on Physics*, Wiley, 1981.
32. H. Haken, *Chaos and order in nature*, Springer-Verlag, 1981.
33. L. S. Schulman and P. E. Seiden, "Statistical mechanics of a dynamical system based on Conway's Game of life," *Journal of Statistical Physics*, vol. 19, p. 293, 1978.
34. S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, vol. 55, p. 601, 1983.
35. T. Toffoli, "CA as an alternative to differential equations in modelling physics," *Physica D*, vol. 10, p. 117, 1984.



36. G. Y. Vichniac, "Simulating physics with cellular automata," *Physica D*, vol. 10, p. 96, 1984.
37. P. Clavin, "Cellular automata hydrodynamics with free boundaries in 2D and 3D," *Comptes Rendus de l'Academie des Sciences serie II*, vol. 303, p. 1169, 1986.
38. D. Montgomery and G. D. Doolen, "Magnetohydrodynamic cellular automata," *Physic Letters A*, vol. 120, p. 229, 1987.
39. S. Wolfram, "Cellular automata fluids 1. Basic theory," *Journal of Statistical Physics*, vol. 45, p. 471, 1987.
40. H. Chen, "A new cellular automata model for magnetohydrodynamics," *Physical Review Letters*, vol. 58, p. 1845, 1987.
41. A. Salomaa, *Theory of Automata*, Pergamon Press, 1969.
42. O. Martin, A. Odlyzko, and S. Wolfram, "Algebraic properties of cellular automata," *Communications in Mathematical Physics*, vol. 93, p. 219, 1984.
43. S. Wolfram, "20 problems in cellular automata," *Physica Scripta*, vol. T9, p. 170, 1985.
44. S. Wolfram, "Universality and complexity in cellular automata," *Physica D*, vol. 10, p. 1, 1984.
45. C. G. Langton, "Studying artificial life with cellular automata," *Physica D*, vol. 22, p. 120, 1987.
46. N. H. Packard and S. Wolfram, "Two dimensional cellular automata," *Journal of Statistical Physics*, vol. 38, p. 901, 1985.
47. W. Li, "Pretty pictures generated by Two-state Five-neighbor Cellular Automata," Technical Report CCSR-88-15, Center for Complex Systems Research, University of Illinois, Urbana-Champaign, November 15, 1988.

48. M. Woodroffe, *Probability with applications*, McGraw-Hill, 1975.
49. M. Gardener, "Mathematical Games," *Scientific American*, vol. 223, p. (Oct)120, (Nov)116, 1970.
50. E. Fredkin and T. Toffoli, "Conservative logic," *International Journal of Theoretical Physics*, vol. 21, p. 219, 1982.
51. A. M. Turing, "The chemical basis of morphogenesis," *Philosophical Transactions of the Royal Society of London*, vol. 237, p. 37, 1952.
52. I. Prigogine, *From being to becoming*, Freeman, 1980.
53. N. T. J. Bailey, *The elements of stochastic processes with applications to the natural sciences*, Wiley, 1964.
54. N. T. J. Bailey, *The mathematical approach to biology and medicine*, Wiley, 1967.
55. M. S. Bartlett, *Stochastic population models in ecology and epidemiology*, Methuen, 1960.
56. R. M. May, *Stability and complexity in model ecosystems*, Princeton University Press, 1974.
57. S. A. Levin (Ed.), *Studies in mathematical biology (I and II)*, Mathematical Association of America, 1978.
58. J. Vinals and J. D. Gunton, "Pattern formation in reversible cellular automata," *Journal of Physics A*, vol. 19, p. L933, 1986.
59. H. A. Ceccatto, "Effective discrete time dynamics in Monte-Carlo simulations," *Physical Review B*, vol. 33, p. 4734, 1986.
60. G. Grinstein, "Statistical mechanics of probabilistic cellular automata," *Physical Review Letters*, vol. 55, p. 2527, 1985.



61. S. Wolfram, "Random sequence generation by cellular automata," *Advances in Applied Mathematics*, vol. 7, pp. 123-169, 1986.
62. D. E. Knuth, *Seminumerical algorithms : The art of computer programming 2*, Addison-Wesley, 1969.
63. K. Kaneko, "Phase transitions in 2D stochastic cellular automata," *Journal of Physics A*, vol. 19, p. L69, 1986.
64. T. Toffoli, "On the large scale implementation of cellular spaces by means of integrated circuit arrays," *Instituto per le Applicazioni del Calcolo*, 1972.
65. E. F. Moore, "The firing squad synchronisation problem," in *Sequential Machines. Selected Papers*, ed. E. F. Moore, p. 213, Addison-Wesley, 1964.
66. A. Waksman, "An optimum solution to the firing squad synchronisation problem," *Information and Control*, vol. 9, p. 66, 1966.
67. R. Balzer, "An 8-state minimal time solution to the firing squad synchronisation problem," *Information and Control*, vol. 10, p. 22, 1967.
68. E. F. Moore and G. G. Langdon, "A general firing squad problem," *Information and Control*, vol. 12, p. 212, 1968.
69. V. I. Varshausky, "Synchronisation of a collection of automata with random pairwise interactions," *Automata and Remote Control*, vol. 29, p. 244, 1969.
70. V. I. Varshausky, V. B. Marakhovsky, and V. A. Peschansky, "Synchronisation of interacting automata," *Mathematical Systems Theory*, vol. 4, pp. 212-230, 1970.
71. G. T. Herman, "Models for cellular interactions in development with out polarity of individual cells," *International Journal of Systems Science*, vol. 3, pp. 149-175, 1972.

72. G. T. Herman, W. H. Liu, S. Rowland, and A. Walk, "Synchronisation of growing cellular arrays," *Information and Control*, vol. 25, pp. 103-122, 1974.
73. P. Rosenstiel, "Existence d'automates finis capables de s'accorder bien qu'arbitrairement correctes et nombreux," *International Computing Centre Bulletin*, vol. 5, pp. 215-244, 1966.
74. P. Rosenstiel, J. R. Fiskel, and A. Hollinger, "Intelligent graphs: networks of finite automata capable of solving graph problems," in *Graph Theory and Computing*, ed. R. C. Read, Academic Press, 1972.
75. A. Grasselli, "Synchronisation of cellular arrays: The firing squad problem in 2D," *Information and Control*, vol. 28, p. 113, 1975.
76. F. Romani, "Parallelism in speeding up cellular automata synchronisation," *Information and Control*, vol. 36, p. 245, 1978.
77. S. A. Kauffman, "The organisation of cellular genetic control systems," in *Lectures on mathematics in the life sciences vol.3*, ed. J. D. Cowan, p. 63, The American Mathematical Society, 1972.
78. S. A. Kauffman, "Emergent properties in random complex automata," *Physica D*, vol. 10, p. 145, 1984.
79. T. E. Ingerson and R. Buvel, "Structure in asynchronous cellular automata," *Physica D*, vol. 10, p. 59, 1984.
80. M. S. Bartlett, *Stochastic processes*, Cambridge University Press, 1962.
81. I. S. Gradshteyn and I. M. Ryzhik, *Table of integrals, series and products*, Academic Press, 1980.
82. M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions*, Dover Publications, 1965.



83. M. W. Hirsch and S. Smale, *Differential equations, dynamical systems and linear algebra*, Academic Press, 1974.
84. J. M. T. Thompson and H. B. Stewart, *Non-linear dynamics and chaos*, Wiley, 1986.
85. D. W. Jordan and P. Smith, *Non-linear ordinary differential equations*, Oxford University Press, 1983.
86. S. Grossberg and D. Levine, "Some developmental and attentional biases in the contrast enhancement and short term memory of recurrent neural networks," *Journal of theoretical biology*, vol. 53, p. 341, 1975.
87. Sun Microsystems Inc., *SunOS 4.0 Reference Manual*, 1988.
88. L. Cesari, *Asymptotic behaviour and stability problems in ordinary differential equations (3rd edition)*, Academic Press, 1971.
89. T. V. Upensky, *Theory of equations*, McGraw-Hill, 1948.
90. A. J. Lotka, *Elements of Physical Biology*, Williams and Wilkins, 1925. Reprinted as *Elements of mathematical biology*, Dover, 1956.
91. V. Volterra, "Variations and fluctuations of the number of individuals in animal species living together," *R. Comitato Talassografico Italiaio Memoria*, vol. 131, pp. 1-142, 1927.
92. V. Volterra, V. A. Kostitzin, A. J. Lotka, and A. N. Kolmogoroff, *The Golden Age of Theoretical Ecology, 1923-1940*, Springer-Verlag, 1978.
93. W. O. Kernack and A. G. McKendrich, "A contribution to the mathematical theory of epidemics," *Proceedings of the Royal Society of London A*, vol. 115, pp. 200-221, 1927.

94. E. Domany and W. Kinzel, "Equivalence of cellular automata to Ising models and directed percolation," *Physical Review Letters*, vol. 53, p. 311, 1984.



Figure 5.7: Lattice evolution for general epidemic CA.

State 1    State 2

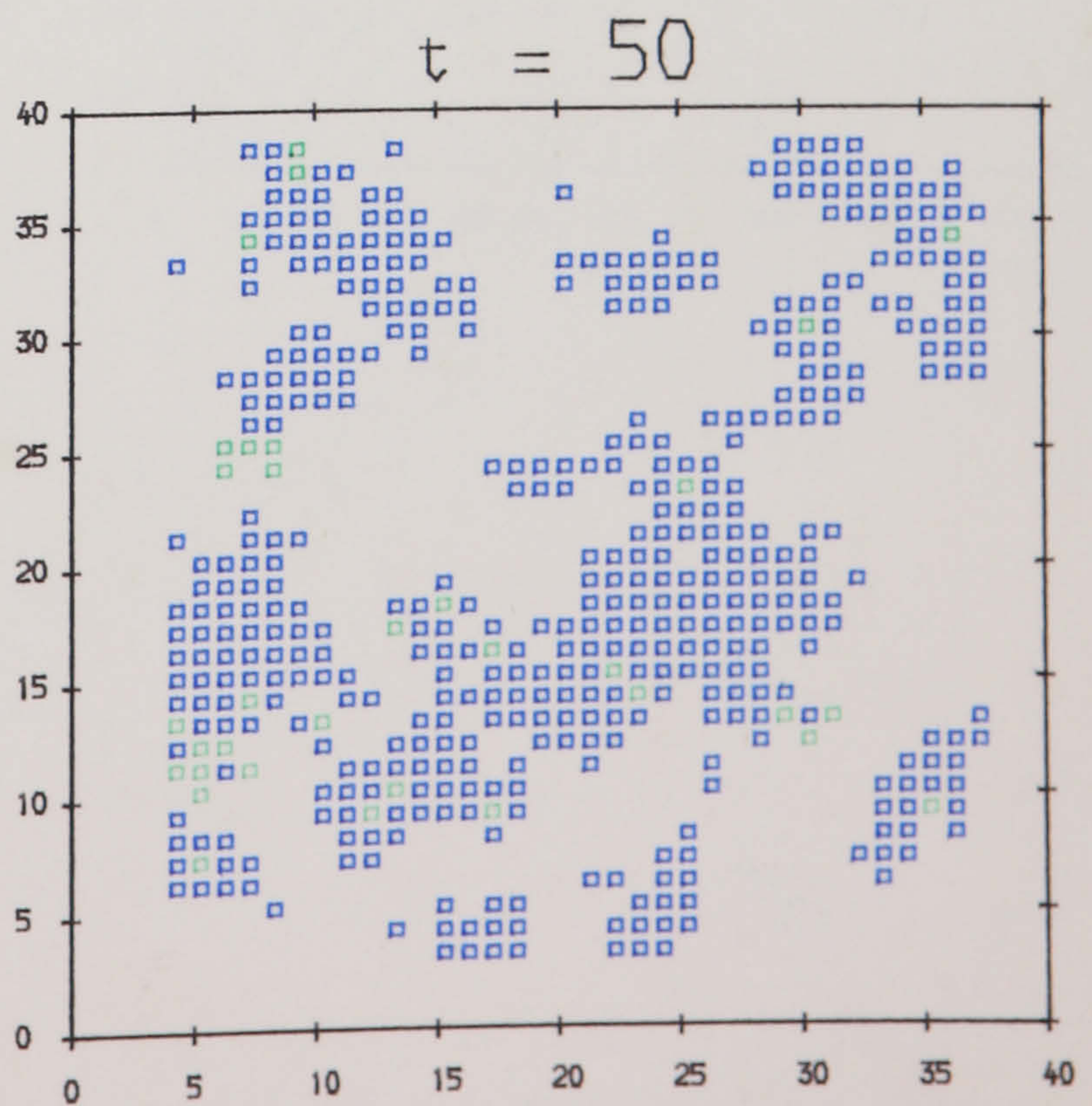
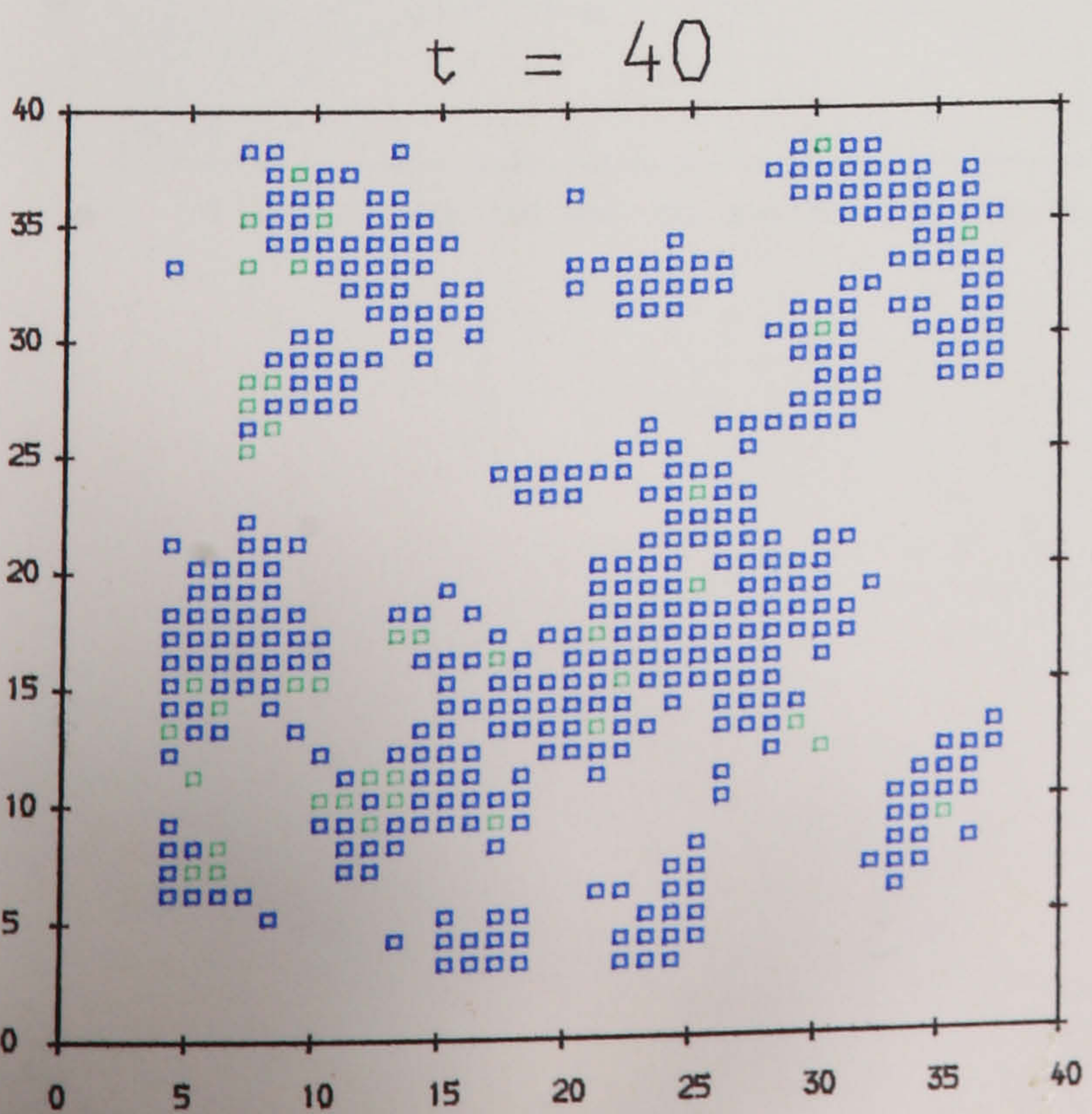
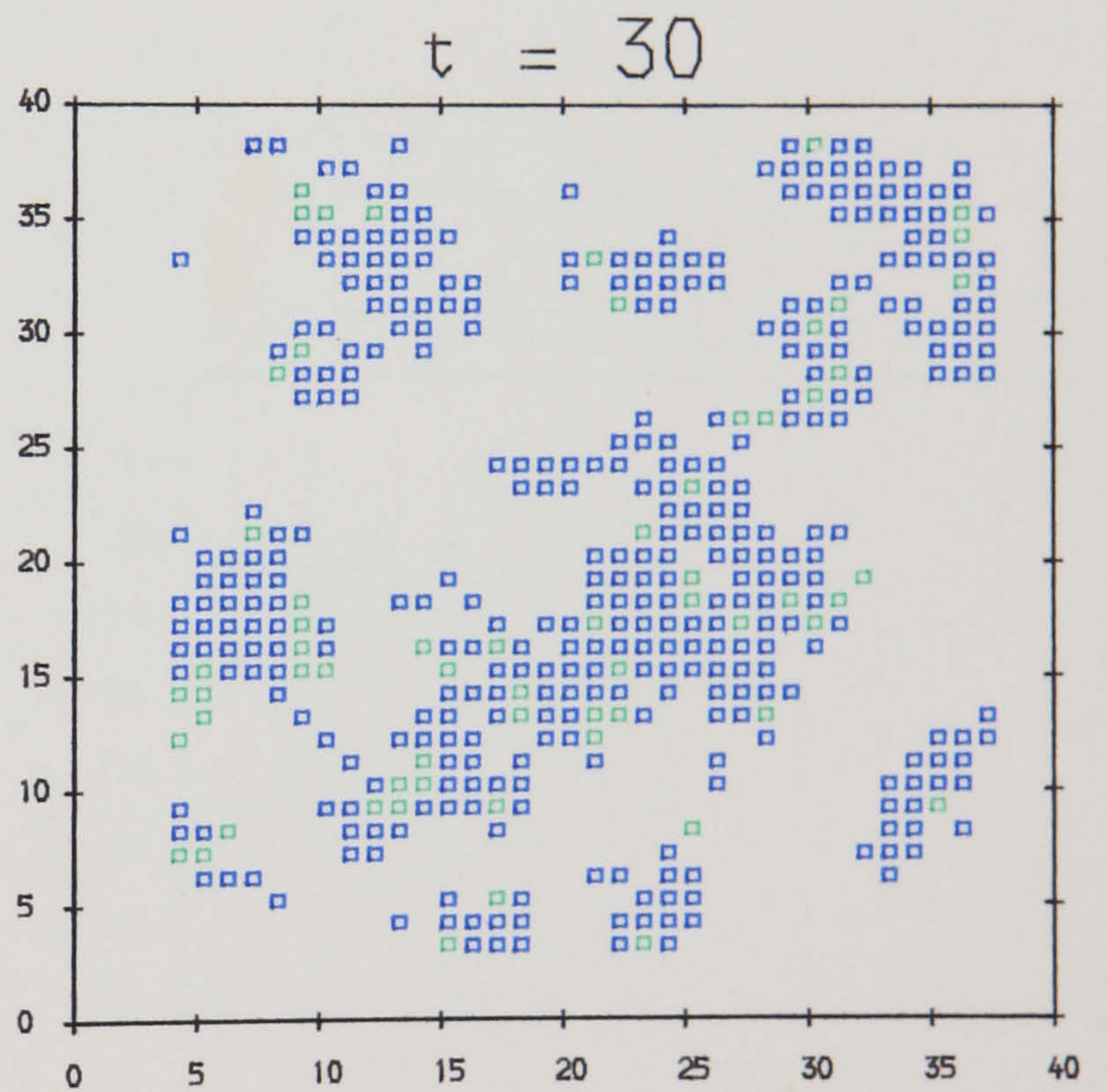
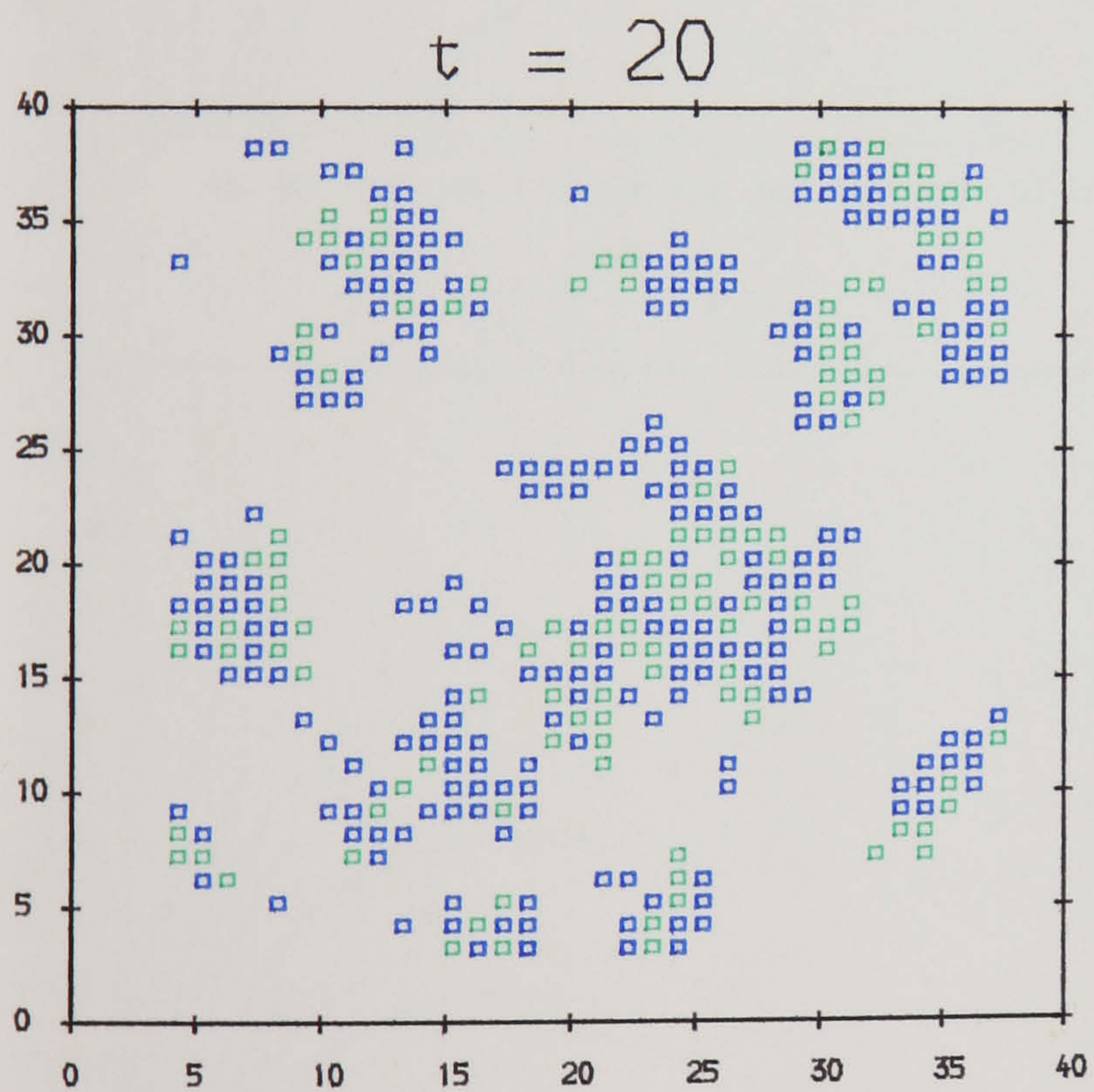
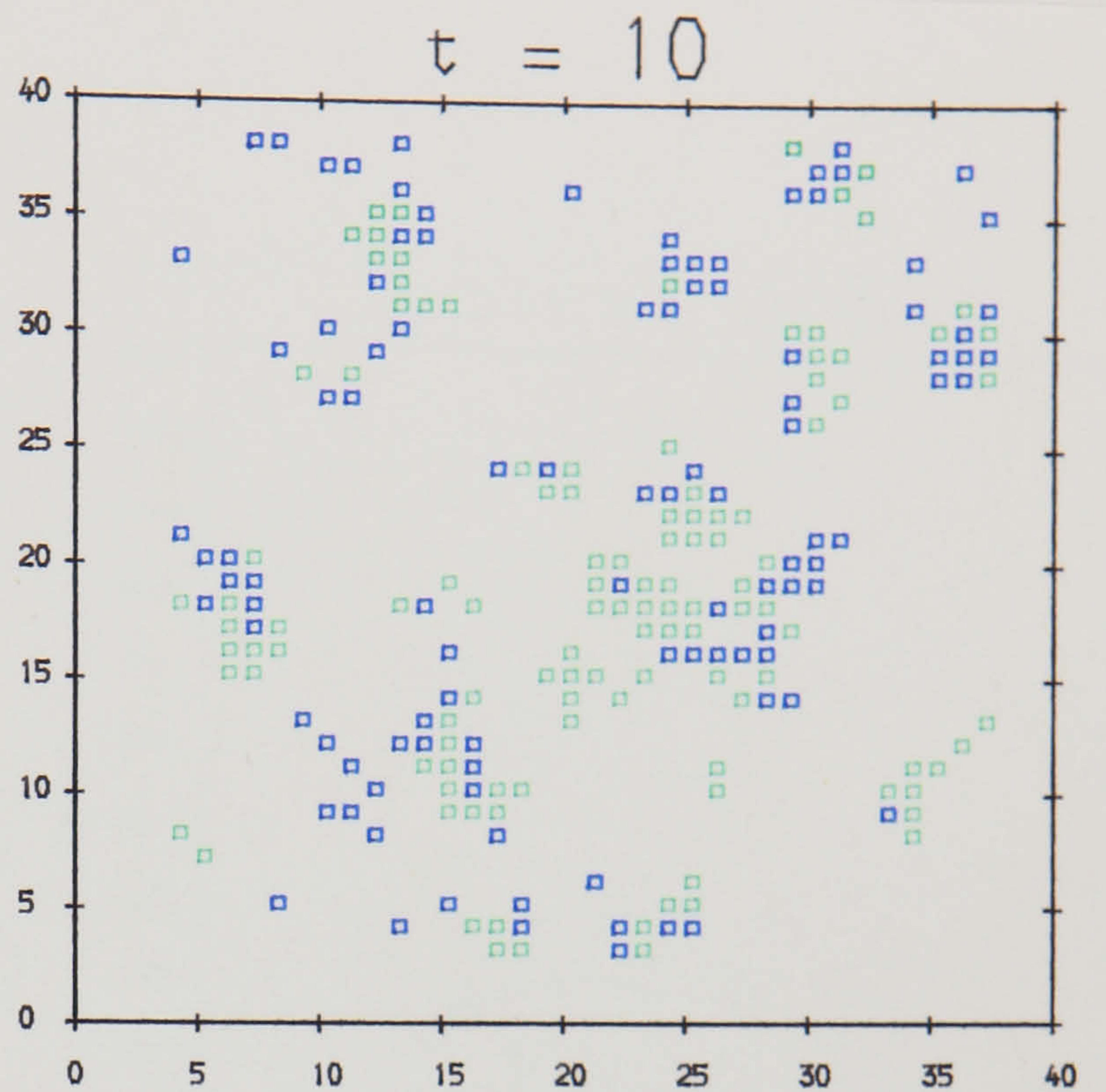
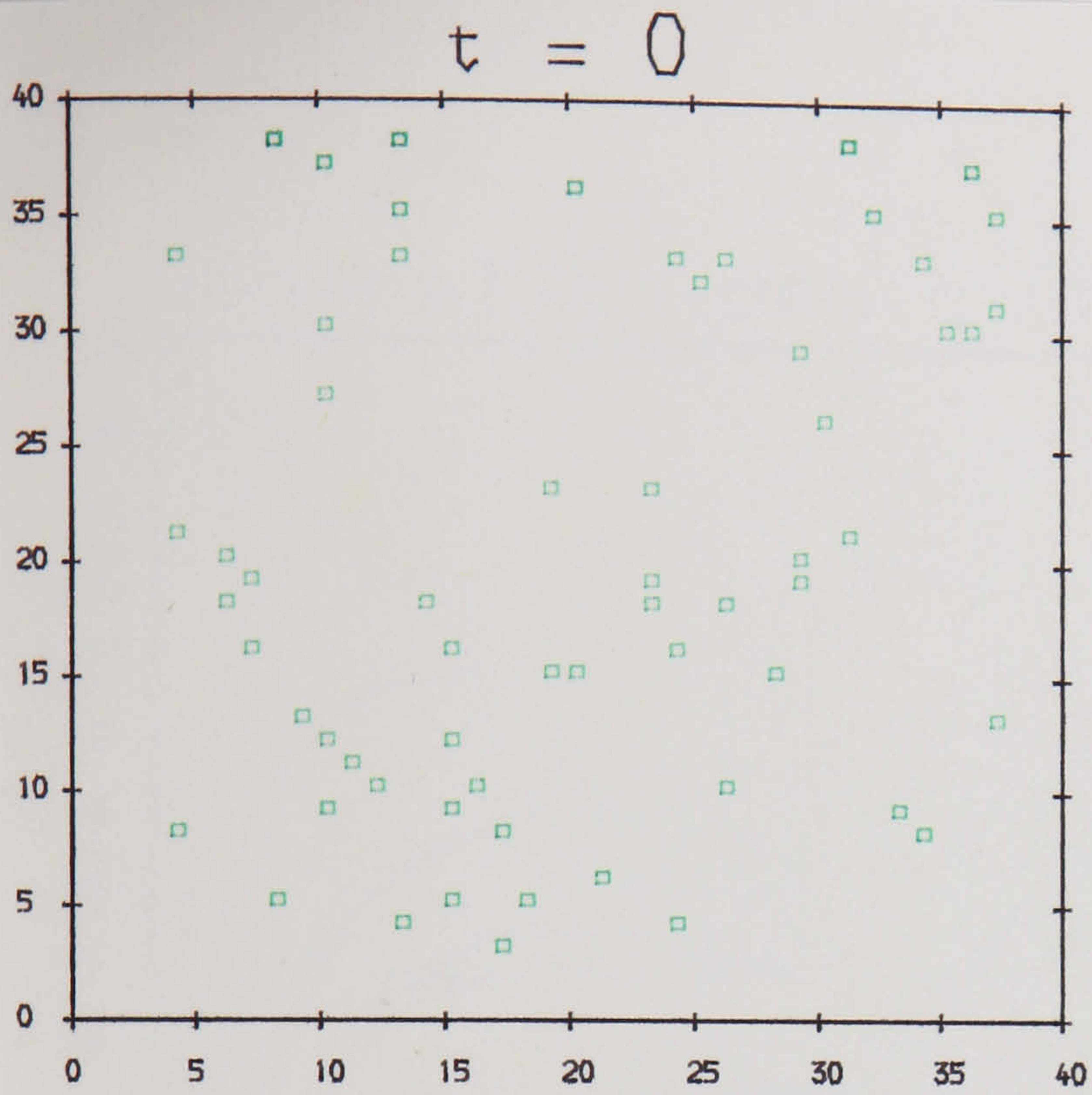








Figure 5.3: Lattice evolution for simplified Lotka-Volterra CA.

State 1    State 2

