The Pennsylvania State University
The Graduate School
Eberly College of Science

MODELING BIAS IN DECISION-MAKING ATTRACTOR NETWORKS

A Dissertation in
Mathematics
by
Syed Safaan Sadiq

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2025

The dissertation of Syed Safaan Sadiq was reviewed and approved by the following:

Leonid Berlyand
Professor of Mathematics
Dissertation Advisor
Co-Chair of Committee

Carina Curto
Professor of Applied Mathematics and Brain Science, Brown University
Dissertation Advisor
Co-Chair of Committee

Vladimir Itskov
Associate Professor of Mathematics

Jessica Conway
Associate Professor of Mathematics and Biology

Reka Albert
Distinguished Professor of Physics and Biology

Abstract

Attractor neural network models of cortical decision-making circuits represent them as dynamical systems in the state space of neural firing rates with the attractors of the network encoding possible decisions. While the attractors of these models are well studied, far less attention is paid to the basins of attraction even though their sizes can be said to encode the biases towards the corresponding decisions. The parameters of an attractor network control both the attractors and the basins of attraction. However, findings in behavioral economics suggest that the framing of a decision-making task can affect preferences even when the same choices are being offered. This suggests that the circuit encodes both choices and biases separately, that preferences can be changed without disrupting the encoding of the choices themselves. In the context of attractor networks, this would mean that the parameters can be adjusted to reshape the basins of attraction without changing the attractors themselves. How can this be realized and how do the parameters shape decision-making biases?

We study this question in the context of threshold linear networks (TLNs), a common model with recurrent dynamics. In Chapter 2, we rigorously prove in the case of two competing neural populations how the parameters of the network shape the basins of attraction and, consequently, bias. In Chapter 3, we raise the problem into larger networks in a class of TLNs called CTLNs, where network parameters are derived from a directed graph structure. We explore and find the challenges of computer-assisted approaches to the problem. Starting in Chapter 4, we focus on CTLNs derived from directed acyclic graphs. We prove how the dynamics of the network can be derived from the combinatorics of the directed acyclic graph. While falling short of determining the full basins of attraction, in Chapter 5, we demonstrate numerically how connectivity shapes the basins' encoding of bias under assumptions of low dimensional dynamics. In Chapter 6, we consider the existence of trajectories beginning in a state of excitatory/inhibitory balance and rigorously prove how their existence can be inferred from the directed graph structure. Finally, in Chapter 7, we generalize results from Chapters 4 and 6 to a class of TLNs we call heterogeneous CTLNs (hCTLNs).

Table of Contents

List of Figures

Chapter 1
Introduction

The course of our lives is charted by how we navigate a gauntlet of decisions. At each stage, our brain must select from a menu of options our next action. How does it do this? How does the brain select an option off the menu? Naively, we might assign each item on the menu a "value" and then claim that the highest value choice will be the decision. Such an approach is how decision-making is conceived by standard economics [1]. However, studies in behavioral economics have demonstrated convincingly that this cannot be the whole story and that actual decision-making is often *menu-variant*.

One such example is the Decoy Effect [2]. Consider two choices, call them X and Y, which when compared with each other have various strengths and weaknesses that would make it so that either one would be chosen with equal likelihood when offered together on a menu. If offered to a room of people each choice will be selected at a roughly $50\% - 50\%$ split. Now, let there then be a third, irrelevant, choice Z added to the menu which is clearly inferior to one of the two original choices, say X. It has been found that the mere presence of this third choice on the menu skews preferences toward its superior option, in this case X. As there is never a reason to pick Z, this new menu of three choices still has the same two ultimate choices, X and Y, but now more people in the room will be drawn to X at perhaps a $60\% - 40\%$ split. Figure 1.1 illustrates how the presence of an irrelevant choice can shift the decision-making boundary between choices. Under the standard economics paradigm, the framing of the decision should not bias our choices, but the Decoy Effect clearly demonstrates that this is not the case.

These phenomenological observations also find biological support in the activity of neurons in the lateral intraparietal area (LIP), a region of the cerebral cortex in primate brains, which does appear to be menu-variant [3]. The LIP is of substantial interest in the study of decision-making and these findings lend credence to the idea that decision-making is a complex process and that the biases that shape it are nontrivial

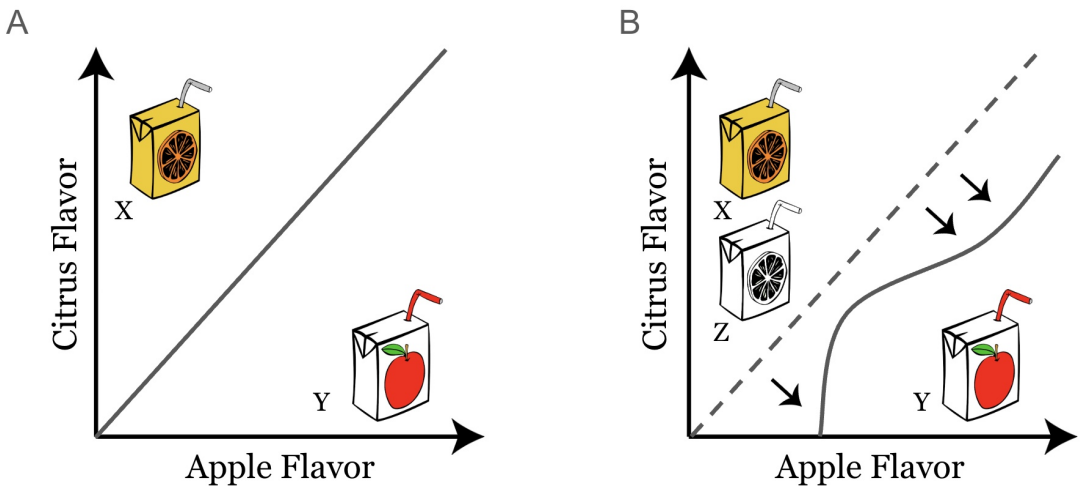


Figure 1.1. The Decoy Effect. (A) Orange juice "X" and apple juice "Y" have distinct flavors and depending upon preferences toward one or the other, either may be chosen roughly equivalently. The diagonal decision boundary reflects the $50\% - 50\%$ split. (B) The presence of a poorer quality orange juice "Z" does not add a true choice, but it increases the number of situations where "X" is the preferred choice, shifting the decision boundary.

to understand. Moreover, these findings suggest a natural question which has been the foundation of my doctoral work.

Question: How is the brain able to encode biases separately from the decisions themselves?

We will study this mathematically through a dynamical systems approach, applying the theory of attractor neural networks to study how a network model of a neural circuit encodes decision-making bias.

1.1 Decision-Making as an Attractor Network

Attractor networks are a popular computational framework for studying neural circuits. In the vein of the Hopfield model [4] for associative memory, we conceive of a circuit of neurons (Fig 1.2A) as a dynamical system in the state space of neural activity. For a given input (an initial condition) the activity converges towards an attractor. The attractors of the network represent the various outcomes of neural computation for the circuit and generally correspond to some phenomena of interest. In the case of the Hopfield model the attractors are stable fixed points representing the memories which are being retrieved. Other models make use of continuous attractors. Examples include ring-shaped

attractors which describe the maintenance of heading directional information [5] and line attractors which keep track of eye position [6]. Attractor networks have also been used in the modeling of decision-making circuits with the attractors representing the decisions themselves [7].

Much of the literature surrounding attractor networks is focused on how many attractors can be encoded into a network and the types of attractors [8]. Far less attention is paid to the basins of attraction. As the basins of attraction are the sets of initial conditions converging to respective attractors, they are representative of the bias towards those corresponding decisions as depicted in Fig 1.2B. Shaping the basins then shapes the bias. So then, we can frame our question in the context of attractor network models.

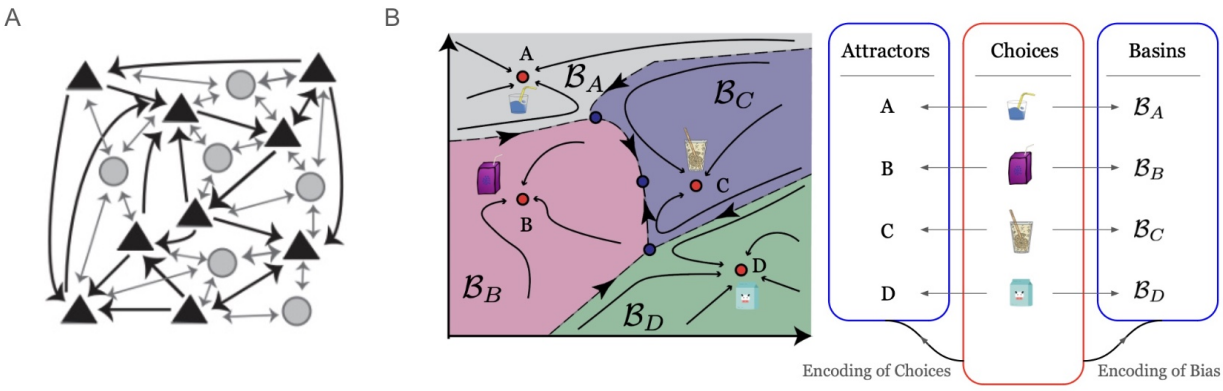


Figure 1.2. Attractor networks and decision-making. (A) Diagram of a cortical circuit with triangles representing excitatory neurons and circles representing inhibitory interneurons. (B) This schematic illustrates the attractor network interpretation of a decision-making circuit. The point attractors in the state space correspond to the choices of the decision-making task, in this case drink selection, and the sizes of the basins of attraction correspond to the bias towards those choices. There are additionally unstable fixed points which lie on the decision boundaries.

Question: How can the parameters of a network be used shape the basins of attraction (the bias) without changing the attractors (the decisions) themselves?

Specifically, we will investigate this question in the setting of firing-rate models of network activity.

1.2 Firing Rate Models and TLNs

The rate coding hypothesis suggests that neurons communicate through their firing rates rather than by the particulars of individual action potentials [9]. Firing rate models are dynamical systems operating in the state space of neural firing rates where each state variable corresponds to the firing rate of a neuron or neural population. To construct such a model, the first thing that must be considered is the relationship between the firing rate of the neurons and the synaptic input current it receives. The following differential equation captures this [10]:

$$\tau \frac{dx}{dt} = -x + \Phi(I_{total})$$

In this equation, x is the firing rate of the neuron and I_{total} is the total synaptic current being received by the neuron. The function $\Phi(z)$ is a firing rate function which describes the relationship between I_{total} and x. Finally, the parameter τ is a time constant.

The input current received by a neuron i in a recurrent neural network can be decomposed as the sum of some external input current, $\theta_i(t)$, plus a weighted sum, $\sum_{j=1}^{n} W_{ij}x_j$, of the firing rates of the neurons in the network. The weights W_{ij} form the connectivity matrix W and represent the strength of each synapse. The firing rate differential equation can then be rewritten as:

$$\tau \frac{dx_i}{dt} = -x_i + \Phi\left(\sum_{j=1}^{n} W_{ij}x_j + \theta_i(t)\right),\ i \in [n]$$

where $x_i(t)$ is the firing rate of the i-th neuron in a network of n neurons. As a point of notation, we use $[n]$ to denote the set of indices (i.e. $[n] = \{1, 2, \ldots, n\}$). The choice of firing rate function $\Phi(z)$ makes a significant difference in the mathematical tractability of a problem. While the linear firing rate function $\Phi(z) = z$ remains popular in neuroscience modeling, it would be of no help in our study of decision-making because a linear dynamical system cannot demonstrate multistability. If we want to study bias in decision-making circuits, we need to encode at least two decisions (attractors) in the circuit, so we need a nonlinear firing rate function.

We will use threshold linear networks (TLNs) [11], a recurrent neural network model governed by the following differential equations:

$$\frac{dx_i}{dt} = -x_i + \left[\sum_{j=1}^{n} W_{ij}x_j + \theta_i\right]_+, i \in [n]$$

.

where $[\cdot]_+ = \max\{0, \cdot\}$ is the ReLU activation function. Often, the actual neural activity at fixed points is of less interest than which neurons are firing, which is called the *support* of a fixed point. The set of fixed point supports for a TLN with connectivity matrix W and external input current vector $\vec{\theta}$ is written as:

$$\mathrm{FP}(W, \vec{\theta}) := \{\sigma \subseteq [n] \mid \sigma \text{ supports a fixed point } x^* \text{ of the TLN with parameters } W \text{ and } \vec{\theta}\}.$$

TLNs are a rich class of firing rate models which display a wealth of dynamical properties periodic attractors and, crucially for our purposes, multistability [12].

1.3 Neuroscience of decision-making

Neuroscience experiments in decision-making generally involve a subject being presented with a task where, upon some stimulus, the subject must decide which among various alternatives is in accordance with the stimulus. One of the canonical decision-making tasks is simian motion discrimination [13]. Monkeys are fixed in head position and presented with a visual field of moving dots. They must then make a saccadic eye movement in the net direction of the motion to receive rewards. A benefit of this kind of highly constrained task, where the decision is as specific as an eye movement, is to allow for concentrating on a smaller region of the brain. Saccadic eye movement in particular is associated with the LIP. A more general decision-making task may engage more parts of the brain and require recording of various regions to get a meaningful picture of the dynamics.

One of the common problems within decision-making studies is replicability across labs [14]. A 2021 paper by the International Brain Laboratory, a consortium of labs, attempts to resolve this by presenting an assay of decision-making tasks involving mice which produced consistent results across the member labs. The core task involves a head-fixed mouse being presented a screen with a grating on either the left or the right. The mouse must turn a steering wheel to the left or right to bring the grating toward the center of the screen to receive the reward of sweetened water. The assay consists of variations on this core task which allow for investigating different aspects of the decision-making process, such as modifying the probability with which the grating appears on the right or left side and the contrast of the grating to see how prior experience affects the mouse's choices [14].

An important distinction arises in the two tasks that we have discussed. The grating task merely requires a decision between right or left, a forced set of two choices. On the other hand, a motion discrimination task could be decided in any direction, theoretically an infinite set of choices. The reason this distinction is significant for our purposes is that the second might cause is to think we need a continuous attractor to describe the results. However, in practice, a continuous choice is often approximated with a discrete set of choices. For example, instead of treating each possible direction of saccadic eye motion as different, they may grouped into upper right, lower right, upper left, and lower left motions, reducing a continuous set of choices into just four [13].

The key point here is that tasks with continuous choices are often approximated to have a forced set of choices. For this reason, our analysis will primarily focus on TLNs with discrete point attractors. While TLNs are know to have continuous, and even dynamic, attractors, there fortunately exists a class of TLNs whose attractors are known to be only discrete point attractors.

1.4 DAG CTLN Models for Decision-Making Circuits

The role of connectivity structure in shaping basins of attraction can be emphasized using a special subclass of TLNs called combinatorial TLNs (CTLNs). In CTLNs, the weight matrix is derived from a directed graph (e.g. Fig 1.3) and the external input currents are made uniform i.e. $\vec{\theta} = \theta \mathbb{1}$ where $\mathbb{1}$ is the all ones vector. We take a directed graph (with no self-edges) and derive weights according to the following rule [12]:

$$W_{ij} = \begin{cases} 0 & \text{if } i = j \\ -1-\delta & \text{if } j \not\to i \\ -1+\varepsilon & \text{if } j \to i \end{cases}$$
$$\text{with } \varepsilon, \delta > 0 \text{ and } 0 < \varepsilon < \tfrac{\delta}{1+\delta}.$$

.

We effectively create two kinds of synaptic weights. A neuron i is said to be *strongly inhibited* by another neuron j if $j \not\to i$ and *weakly inhibited* if $j \to i$. An excitatory connection corresponding to an edge is reducing the inhibition [12]. This class takes as a modeling assumption that the excitatory neurons are effectively inhibiting one another indirectly through inhibitory interneurons with the excitatory connections merely reducing the inhibition as depicted in Fig 1.3.

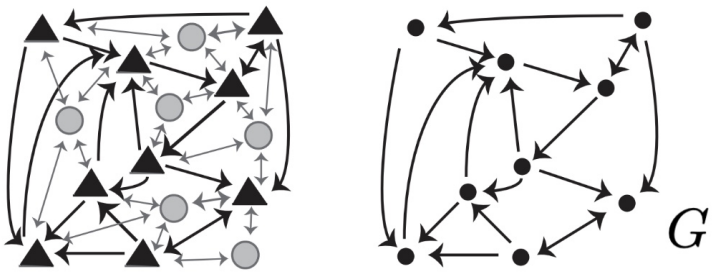


Figure 1.3. Deriving a CTLN directed graph from a neural circuit. An example of a directed graph representation of a neural circuit. The inhibitory interneurons are removed, so we are left with a network of just the excitatory neurons. The model incorporates their role in making the excitatory neurons effectively inhibitory toward one another. For a neuron i, $j \not\to i$ indicates *strong inhibition* as there are no excitatory connections to reduce the effective inhibition. Alternatively, $j \to i$ indicates *weak inhibition*, with the excitatory connection reducing the effective inhibition.

A remarkable property of CTLNs is that there exist correspondences between the combinatorial properties of the graph and the fixed points of the dynamical system [12]. This enables us to talk about the attractors of a CTLN in terms of the directed graph G. In many cases, the attractor supports are controlled entirely by the directed graph structure and the fixed point support set can be defined as a function of the graph. We then use the notation:

$$\mathrm{FP}(G) := \{\sigma \subseteq [n] \mid \sigma \text{ is the support for some fixed point } x^* \text{ of a CTLN derived from } G\}$$

In particular, CTLNs derived from directed acyclic graphs (DAGs) have very predictable stable fixed point attractors, one for each sink, with unions of sinks being the unstable fixed points [12]. Moreover, DAG CTLNs have no dynamic attractors [15], making it very easy to generate a variety of CTLNs with the same attractors by fixing the sinks and altering the graph structure of the non-sink neurons (Fig 1.4). Doing so will not change the attractors, but it will change the basins of attraction!

We can construct a DAG which mirrors the aforementioned Decoy Effect. We begin with two sinks corresponding to the main choices. By randomly sampling initial conditions and tracking the attractor to which they converge, we can numerically reconstruct the basins of attraction for the CTLN (Fig 1.5A) and see that they are equivalent. We can then compare this to a case where we add the asymmetrically dominated, irrelevant,

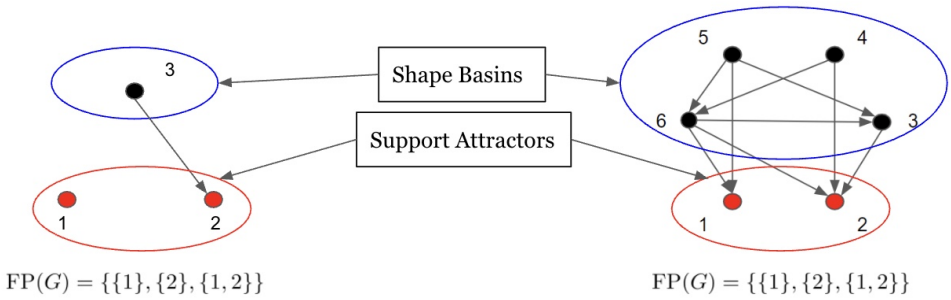


Figure 1.4. CTLNs of directed acyclic graphs (DAGs). Directed acyclic graphs have fixed points supported only on sinks and the unions of sinks. Only the fixed points supported on the sinks themselves yield attractors, one for each sink ($\{1\}$ and $\{2\}$). The union of sinks ($\{1, 2\}$) supports the saddle point which lies on the separatrix

choice as a non-sink neuron, neuron 3 (Fig 1.5B). Both of these circuits have the same attractors, but different basins of attraction. Notice that in the case of the Decoy Effect, the basin of attraction of the weakly inhibited neuron is larger, indicating that there are more initial conditions converging towards its attractor relative to the symmetric case. This captures how the bias has been skewed towards that choice. Notice that in both cases there is a fixed point supported on both sinks (marked $\{1, 2\}$) which has no trajectories converging to it. This is because this fixed point is a saddle point.

From this case of DAGs with two sinks, we can build an intuitive picture of binary choice decision-making dynamics. We have two basins of attraction separated by a decision boundary of codimension 1 and this decision boundary is the stable manifold associated with a saddle point supported on the union of the sinks. When we have a decision boundary [16], to see how the basins of attraction are shaped, we would need to understand the dynamics of the network relative to it. Consider Fig 1.6, where we have a stable manifold marked in green for a saddle point which separates two basins of attraction. That stable manifold represents the decision boundary.

1.5 Basins of Attraction and Decision-Making Bias: Choosing Initial Conditions

An assumption up until now has been that all of the initial conditions are equally relevant (Fig 1.6A). This certainly is implausible biologically, but then this begs the question of what initial conditions should we be focused on?

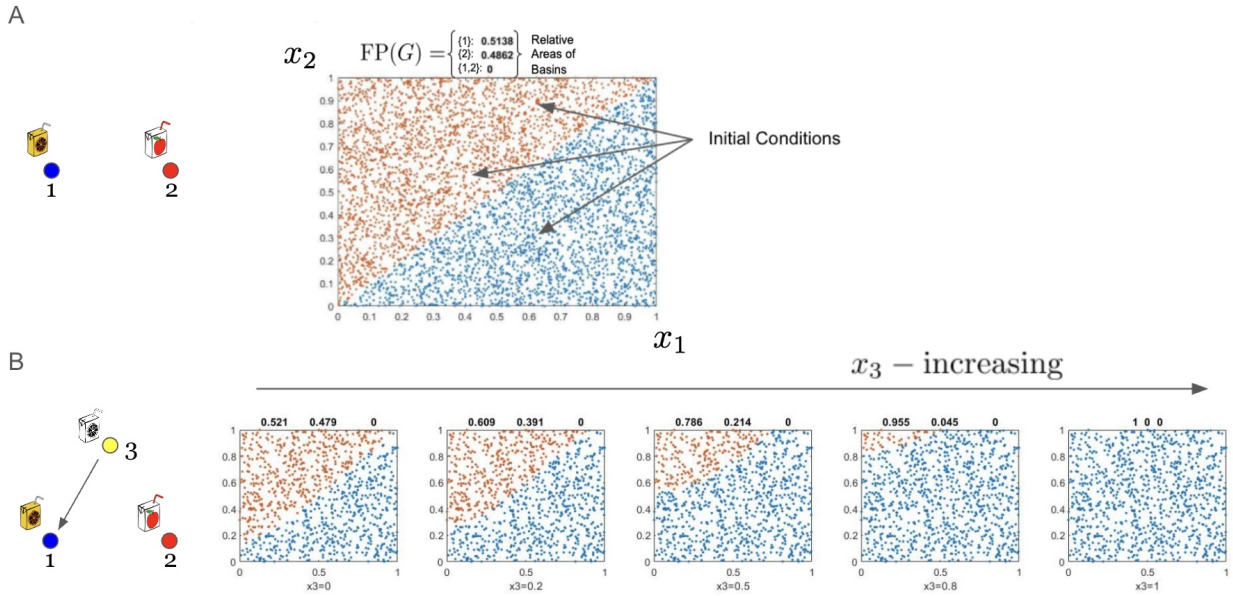


Figure 1.5. Non-sink neurons shape basins of attraction in DAG CTLNs. We show in this figure the results of Monte Carlo simulation to determine the basins of attraction. We randomly sample initial conditions $(x_1(0), x_2(0), \ldots, x_n(0))$ for a CTLN and we color code that point in state space according to the sink attractor to which it converges. (A) In this DAG, there exist only the two sinks. The figure color codes initial conditions in the x_1, x_2 plane to numerically capture the basins of attraction. We see that the basins of attraction are symmetric. (B) This DAG corresponds to the Decoy Effect scenario where the third neuron represent the irrelevant choice. It is never part of the support of an attractor, but nonetheless it shapes the dynamics. Here we have taken cross sections of the state space for different values of $x_3(0)$ in the initial condition. We see that the activity of the third neuron skews the basins of attraction making the one corresponding to x_1 larger. While the attractors themselves remain unchanged from A, the basins have been altered. Notably, the shifting of the decision boundary is even present in the $x_3(0) = 0$ cross section.

We could instead restrict ourselves to trajectories close to the decision boundary. Even if we don't quite know where the decision boundary is located, as DAG CTLNs are continuous dynamical systems, trajectories beginning sufficiently close to the stable manifold of the saddle point will eventually be near the fixed point before following the unstable manifold toward one of the attractors. We could then understand these trajectories by concentrating on initial conditions in the neighborhood of the saddle point, as in Fig 1.6B, and determining which basin is larger in this neighborhood.

Another initial condition which is of interest is informed by the biology of neural dynamics. There exists a broad literature arguing that neural circuits exist in a state of

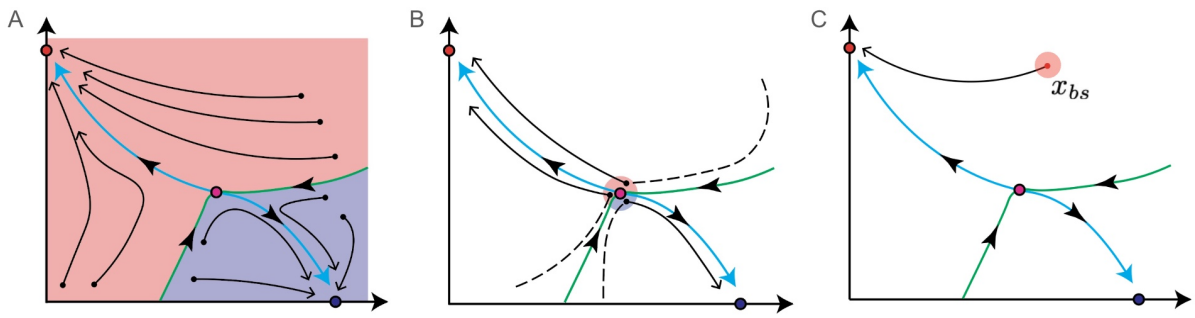


Figure 1.6. Bias, basins of attraction, and three hypotheses of neural dynamics A two dimensional schematic depicting three ways of relating basins of attraction to a decision-making circuit. (A) Determining bias as the relative sizes of the full basins of attraction, where all initial conditions matter equally, aligns with the high-dimensional reservoir dynamics hypothesis of neural dynamics. (B) Focusing on initial conditions near the saddle points emphasizes trajectories which approximate the branching stable (green) and unstable (cyan) manifolds of the saddle point, hewing along the decision boundary. This aligns with the hypothesis of low-dimensional subspace dynamics and associates bias with the relative sizes of the basins of attraction within the region around the saddle point. (C) Prioritizing the trajectory using the balanced state as the initial condition is in accordance with path-following dynamics and understands bias of the network as being towards the attractor to which the balanced state trajectory converges.

balance between inhibition and excitation [17–19]. This means that the input received by every neuron, both from within the network and from outside of it, should by zero for every neuron. The excitatory/inhibitory balance would then arise at the state $\vec{x}_{bs}$ such that $W\vec{x}_{bs} + \vec{\theta} = 0$. This *balanced state* typically lies in one of the basins of attraction and we could then see to which attractor it converges (Fig 1.6C).

A notable concern with balanced states is that their trajectories often do not make sense biologically, presenting a challenge to this approach. We call a CTLN where these issues do not arise *balanced*.

To summarize, we have suggested three paradigms. The first treats all initial conditions equally, the second focuses on initial conditions in the neighborhood of the saddle point as a means of studying trajectories near the decision boundary, and the third studies the singular initial condition of the balanced state.

Notably these three paradigms map on to three hypotheses of neural dynamics [20].

H1: High-dimensional reservoir dynamics

H2: Low-dimensional subspace structured dynamics

H3: Path-following dynamics

In the case of **H1**, where neural dynamics happens in the full state space of the network, we would need to consider initial conditions generally and this corresponds to the approach of looking at the size of the full basins of attraction relative to one another. If we were to take **H2**, where neural dynamics are believed to primarily operate on low dimensional manifolds, we would focus on initial conditions near the lower dimensional decision boundaries which relates to the paradigm of considering initial conditions near the saddle point. Finally, if we were to accept **H3**, where we are privileging a particular trajectory this would align with the idea of seeing to which attractor a trajectory beginning at the balanced state converges.

1.6 Summary of Results

The organization of this dissertation is as follows:

In Chapter 2 we discuss a two neuron TLN model for decision-making. We conduct a bifurcation analysis to prove the conditions required for them to be useful in the study of decision-making i.e. for there to be two stable attractors. We will then rigorously prove how the basins of attraction evolve through parameter space by determining bifurcations on the qualitative, coarse-grained dynamics of the network. This will show fully how the parameters can be manipulated to adjust the sizes of the basins of attraction while preserving the attractors. We conclude by explicitly calculating the basins and detailing the relationship between their sizes and the model parameters. As determining the basins of attraction *a fortiori* tells us about the basins of attraction near the saddle point and the attractor to which the balanced state converges, this constitutes a full solution to the problem in the two dimensional case.

As solving a general TLN in higher dimensions is impractical, in Chapter 3 we explore computer assisted approaches drawing on Conley Index Theory. We offer an algorithm to rigorously compute coarse-grained dynamics for TLNs while also showing the challenges it faces in determining the basins of attraction in higher dimensions.

In Chapter 4 we restrict our attention to CTLNs. Specifically, we focus on DAG CTLNs and give theoretical results demonstrating a deep association between the properties of these networks and generating function constructions on the DAG that we refer to as *localized path polynomials*. Using them, we will not only offer a novel proof for the relationship between the sinks of the DAG and the attractors of its CTLNs, but we will also use them to rigorously prove analytic solutions for the dynamics of a subclass of DAG CTLNs. After offering an analytic approach to resolving the associated initial

value problem however, we realize that the expressions end up being too cumbersome to allow us to resolve the full basins of attraction as we did in Chapter 2. Still, we press forward with studying the basins of attraction in the vicinity of the saddle point and of the balanced state.

In Chapter 5 we show how the theoretical properties of DAG CTLNs suggest a relationship between sink in-degree and the sizes of basins of attraction near the saddle point. We conjecture a relationship between sink in-degree and the local basin size, and we find numerical evidence to support this relationship.

Chapter 6 is focused on the path-following, balanced state paradigm and we tackle head on the problem of determining when a CTLN is balanced. We rigorously prove results for general CTLNs and then demonstrate how incorporating the properties of DAG CTLNs and their localized path polynomials enables us to prove stronger results in that case. We will then discussed the notion of *balanced graphs*, graphs such that any CTLN derived from them is balanced, and how localized path polynomials can be used to build such graphs. We conclude by presenting an algorithm which has had partial success at predicting the attractor to which the balanced state converges and we provide numerical evidence demonstrating its effectiveness.

The last chapter will weaken the CTLN conditions by allowing the external input currents received by the neurons to vary, what will be referred to as an hCTLN (heterogeneous CTLN). We look at hCTLNs derived from DAGs and will generalize some of the theoretical results from DAG CTLNs to this setting.

Major Original Contributions

- The main theoretical contributions in Chapter 2 are Theorem 2 and Theorem 3. Together, these provide a comprehensive description of how the combinatorial dynamics and basins of attraction of competitive TLNs evolve with respect to each other in a four dimensional parameter space. Also, while Theorem 1 is primarily an illustrative exercise, it is nonetheless an original result.

- The novel contributions in Chapter 3 are Proposition 4, Proposition 5, and Algorithm 1. They allow the efficient generation of a state transition graph on any TLN subject to a convex polytope partition of the state space generated by a hyperplane arrangement and respecting the piecewise linearity of the TLN.

- Chapter 4 defines localized path polynomials and contains a number of original theoretical results related to DAG CTLNs, notably Corollary 6, Corollary 7, and

Proposition 9. Applied together in the context of a special class of DAG CTLNs, they produce Theorem 6 determining general solutions for solving the underlying linear systems composing DAGs. We also give a way of resolving the initial value problem for these systems.

- Chapter 5 is primarily computational, numerically detecting correlation between the sizes of the basins of attraction for DAG CTLNs in the neighborhood of saddle points and the indegrees of the sinks relative to one another. The analysis is motivated by Proposition 11 and Proposition 12.

- Chapter 6 again contains a number of theoretical results. With respect to determining when a CTLN is balanced, the notable results are Theorem 7 which gives a sufficient condition on parameters for any CTLN to be balanced and Theorem 8 which is an improved result specifically for DAGs. Additionally, Theorem 9 and Corollary 11 describes classes of balanced graphs. Lastly we contribute Algorithm 2 which has had partial success at predicting the attractor to which the balanced state trajectory converges.

- The key contributions of Chapter 7 are Lemma 12, Proposition 13, Proposition 14, and Proposition 15. Proposition 14 is used to obtain a stronger form of Theorem 6 whereas the remainder generalize results from Chapter 4 and Chapter 6 into the setting of hCTLNs.

Chapter 2
Combinatorial Dynamics of Two-Dimensional TLNs: The Binary Competition Model

One of the main benefits of working with TLN models is in the piecewise linearity of their differential equations. In this chapter we will show how this can be exploited in a simple two dimensional model of competing neural populations to demonstrate how the parameters can shape the basins of attraction while preserving the attractors.

Before we delve into the details of the model, we introduce some of the fundamental tools used to study TLN dynamics.

Recall that the TLN differential equations are of the form []:

$$\frac{dx_i}{dt} = -x_i + \left[\sum_{j=1}^{n} W_{ij}x_j + \theta_i\right]_+, i = 1, ..., n.$$

Note that the term inside the ReLU function is a linear function of x,

$$y_i(x) := \sum_{j=1}^{n} W_{ij}x_j + \theta_i.$$

Because of the threshold linearity of the ReLU function, $\frac{dx_i}{dt}$ will be linear on either side of the hyperplane:

$$H_i : y_i(x) = 0.$$

The hyperplanes $\{H_i\}_{i=1}^n$ can be used to divide the state space into chambers where the system is linear [12]. This shows that TLNs are a very special case of a continuous

Filippov system [21] and are patchworks of linear systems. Within each chamber of the hyperplane arrangement, the dynamics are governed by a linear dynamical system.

The chambers created by the partitioning of H_i (Fig 2.1) and their corresponding linear systems of ODEs will be referenced in the following manner:

Definition 1. $R_\sigma = \{x \in \mathbb{R}^n \mid y_i(x) > 0, \forall i \in \sigma \text{ and } y_k(x) \leq 0, \forall k \notin \sigma\}$

Remark 1. *We will occasionally use the symbol R_σ^+ to indicate the restriction of R_σ to the positive orthant i.e. $R_\sigma^+ = \{x \in R_\sigma \mid x \geq 0\}$.*

Definition 2. $L_\sigma = \left\{ \frac{dx_i}{dt} = -x_i + \sum_{j=1}^n W_{ij}x_j + b_i \mid i \in \sigma \right\} \bigcup \left\{ \frac{dx_k}{dt} = -x_k \mid k \notin \sigma \right\}$

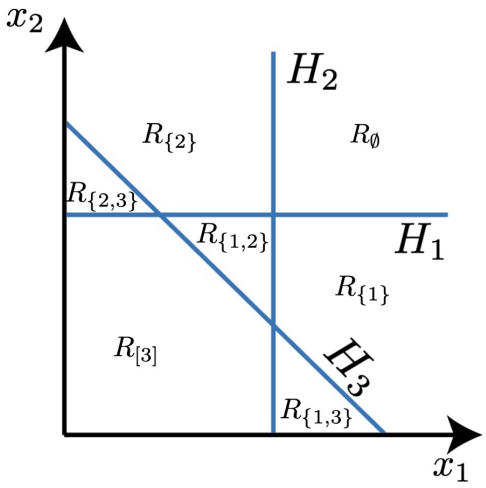


Figure 2.1. TLN state space partition using H_i hyperplane arrangement. An example cross section of the hyperplane arrangement for a three neuron TLN. Each neuron i produces a hyperplane H_i and the full hyperplane arrangement creates a partition of the state space into chambers R_σ governed by linear dynamics L_σ.

The local linearity of TLNs make analyses of the dynamics far more tractable than for most nonlinear dynamical systems. To illustrate this, we look now at an application of TLNs to the so-called "paradoxical effect" problem.

2.1 Example: Paradoxical Effect

The paradoxical effect is a term used to describe how increasing input current to an inhibitory neuron can, in certain contexts, reduce its steady state activity [22]. The

mathematical question that this introduces is under what conditions can a mathematical network model recreate this phenomena.

The phenomenon is most easily understood through using a *nullcline* analysis.

Definition 3. *Let* $\frac{dx}{dt} = f(x)$ *be an autonomous differential equation. The* ***nullcline*** *of this differential equation is the curve given by the relation* $\mathcal{N} : f(x) = 0$.

In the context of a system of an n-dimensional system of differential equations, each differential equation contributes a nullcline along which the derivative component in the direction of the associated state variable is zero. Taking the nullcline of each differential equation in the system we obtain the collection, $\{\mathcal{N}_i\}_{i=1}^n$.

Remark 2. *The intersection of the nullclines gives the steady states of the dynamical system. In the event that they intersect at isolated points, those will be the fixed points of the system.*

One of the great benefits of working with TLNs is that they have piecewise linear nullclines. For the differential equation:

$$\frac{dx_i}{dt} = -x_i + \left[\sum_{j=1}^{n} W_{ij}x_j + \theta_i\right]_+$$

the associated nullcline will be:

$$\mathcal{N}_i : x_i = \left[\sum_{j=1}^{n} W_{ij}x_j + \theta_i\right]_+$$

We now demonstrate the paradoxical effect in TLNs. Consider an E-I network, as depicted in Fig 2.2A, consisting of a single excitatory and a single inhibitory neuron with weight matrix:

$$W = \begin{bmatrix} w_e & w_i^e \\ w_e^i & w_i \end{bmatrix}$$

where w_e is the self-excitation of the excitatory neuron, w_i^e is the inhibition of the inhibitory neuron to the excitatory neuron, w_e^i is the excitation from the excitatory neuron to the inhibitory neuron, and w_i is the self-inhibition of the inhibitory neuron. The ODE system is then:

$$\frac{dx_e}{dt} = -x_e + [w_e x_e + w_i^e x_i + \theta_e]_+$$

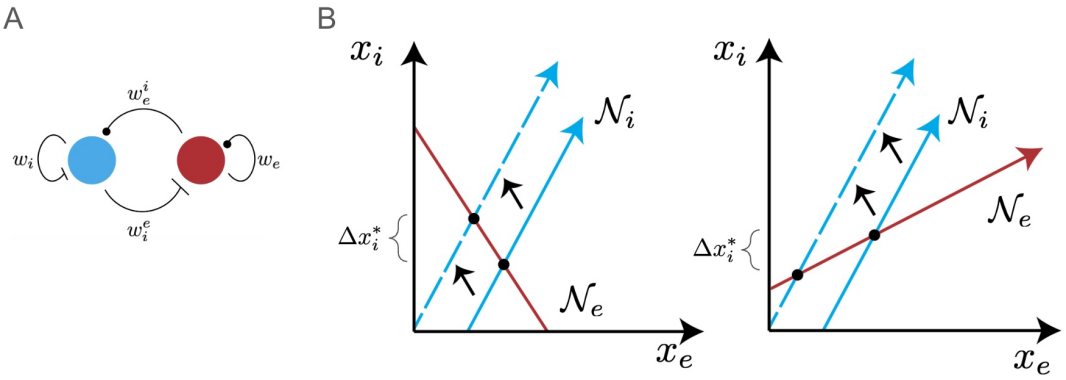


Figure 2.2. E-I Networks and the paradoxical effect. (A) An E-I network with two neurons. The parameters w_i, w_e are the self-inhibition and self-excitation of the neurons whereas w_i^e and w_i^e represent the excitation to the inhibitory neuron and the inhibition to the excitatory neuron respectively. (B) A nullcline schematic illustrating the cause of the paradoxical effect in two dimensions where $\mathcal{N}_{e,i}$ represent the nullclines for the neurons. The first image shows the "non-paradoxical" case. Increasing the external input current to the x_i shifts its nullcline upward and increases its steady state activity. However, under the proper orientation of the nullclines, this upward shift $\mathcal{N}_i$ decreases the steady state activity of x_i.

$$\frac{dx_i}{dt} = -x_i + [w_e^i x_e + w_i x_i + \theta_i]_+$$

In particular, focus on the chamber $R_{[2]}$ with the corresponding ODE system $L_{[2]}$:

$$\frac{dx_e}{dt} = -x_e + w_e x_e + w_i^e x_i + \theta_1 = (w_e - 1)x_e + w_i^e x_i + \theta_e$$
$$\frac{dx_i}{dt} = -x_i + w_e^i x_e + w_i x_i + \theta_2 = w_e^i x_e + (w_i - 1)x_i + \theta_i$$

The fixed point of this system is:

$$x^* = \left(\frac{\theta_i w_i^e + \theta_e(1 - w_i)}{(1 - w_i)(1 - w_e) - w_e^i w_i^e}, \frac{\theta_e w_e^i + \theta_i(1 - w_e)}{(1 - w_i)(1 - w_e) - w_e^i w_i^e} \right)$$

Notice that if:

$$\frac{1 - w_e}{(1 - w_i)(1 - w_e) - w_e^i w_i^e} < 0$$

then x_i^* decreases as θ_i increases. That is to say that increasing the external drive on the inhibitory neuron will reduce its steady state value.

The fixed point of $L_{[2]}$ is only a fixed point of the the system as a whole if it lies in

$R_{[2]}$. It can be shown that $y_e(x^*), y_i(x^*) > 0$ precisely when $(1-w_i)(1-w_e) - w_e^i w_i^e > 0$.

It is clear then that if x^* is a fixed point of the TLN, then the paradoxical effect arises when $w_e > 1$. This realizes the paradoxical effect.

To understand what is happening consider how the nullclines change. For this two-dimensional system $L_{[2]}$ we have the nullclines:

$$\mathcal{N}_e : x_i = \frac{1-w_e}{w_i^e} x_e - \frac{\theta_e}{w_i^e}$$

$$\mathcal{N}_i : x_i = \frac{w_e^i}{1-w_i} x_e + \frac{\theta_i}{1-w_i}$$

The condition $w_e > 1$ change the sign of the slope for $\mathcal{N}_e$ and produces the paradoxical effect as depicted in Fig 2.2B. Now let us consider how the nature of TLNs lets us study the paradoxical effect in higher dimensions.

2.1.1 The Paradoxical Effect in Larger TLNs

We now consider TLNs with additional inhibitory neurons identical to the first (see Fig 2.3). As the inhibitory neurons are identical we can reuse the four synaptic weight parameters w_e, w_i^e, w_e^i, w_i and add the inhibition between inhibitory neurons w_i^i. As before, x_1 will be the firing rate of the excitatory neuron while $x_2, \ldots, x_n$ are the inhibitory neurons.

We consider the chamber with all the ReLU functions active, $R_{[n]}$, and study the dynamics of the linear system $L_{[n]}$. The nullclines are then as follows:

$$\mathcal{N}_1 : -x_1 + w_e x_1 + w_i^e \sum_{k=2}^{n} x_k + \theta_1 = 0$$

$$\mathcal{N}_j : -x_j + w_e^i x_1 + w_i x_j + w_i^i \sum_{k=2, k\neq j}^{n} x_k + \theta_j = 0 \text{ if } j > 1$$

These can be rewritten as:

$$\mathcal{N}_1 : x_1 = \frac{w_i^e}{1-w_e} \sum_{k=2}^{n} x_k + \frac{\theta_1}{1-w_e}$$

$$\mathcal{N}_j : x_1 = \frac{1-w_i}{w_e^i} x_j - \frac{w_i^i}{w_e^i} \sum_{k=2, k\neq j}^{n} x_k - \frac{\theta_j}{w_e^i} \text{ if } j > 1$$

The fixed point associated with $L_{[n]}$ lies at the intersection of the nullclines. The computations will get a bit involved, so we introduce a useful lemma.

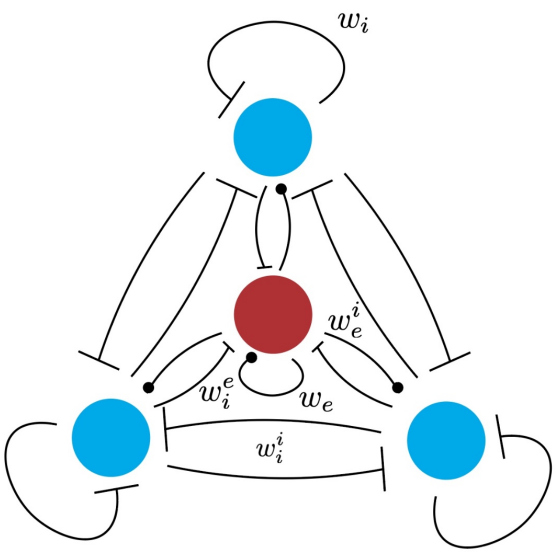


Figure 2.3. Higher-dimensional E-I networks with identical inhibitory neurons. An architecture of a four neuron E-I network in higher dimensions with 3 identical inhibitory neurons. The new parameter w_i^i corresponds to the inhibition between inhibitory neurons and is symmetric among them.

Lemma 1. *For:*

$$\mathcal{N}_{j_1} : x_1 = \frac{1-w_i}{w_e^i}x_{j_1} - \frac{w_i^i}{w_e^i}\sum_{k=2,k\neq j_1}^{n} x_k - \frac{\theta_{j_1}}{w_e^i}$$

$$\mathcal{N}_{j_2} : x_1 = \frac{1-w_i}{w_e^i}x_{j_2} - \frac{w_i^i}{w_e^i}\sum_{k=2,k\neq j_2}^{n} x_k - \frac{\theta_{j_2}}{w_e^i}$$

$$\textit{Then } x \in \mathcal{N}_{j_1} \cap \mathcal{N}_{j_2} \implies x_{j_1} = x_{j_2} + \frac{\theta_{j_1} - \theta_{j_2}}{1 - w_i + w_i^i}$$

Proof. If x lies on both nullclines, then the two nullcline equations can be set equal to one another.

$$\frac{1-w_i}{w_e^i}x_{j_1} - \frac{w_i^i}{w_e^i}\sum_{k=2,k\neq j_1}^{n} x_k - \frac{\theta_{j_1}}{w_e^i} = \frac{1-w_i}{w_e^i}x_{j_2} - \frac{w_i^i}{w_e^i}\sum_{k=2,k\neq j_2}^{n} x_k - \frac{\theta_{j_2}}{w_e^i}$$

After multiplying through by w_e^i and eliminating identical terms from both sides, this simplifies to:

$$(1-w_i)x_{j_1} - w_i^i x_{j_2} - \theta_{j_1} = (1-w_i)x_{j_2} - w_i^i x_{j_1} - \theta_{j_2}$$

The rearranging of which yields the desired:

$$E_{j_1 j_2} : x_{j_1} = x_{j_2} + \frac{\theta_{j_1} - \theta_{j_2}}{1 - w_i + w_i^i}$$

□

Lemma 1 is a useful technical tool which we can use to rewrite the entries of the fixed point in terms of one another. From here the fixed point of $L_{[n]}$ can be calculated.

Proposition 1. *The fixed point x^* associated with the system $L_{[n]}$ of a threshold linear network consisting of one excitatory neuron, x_1, and multiple identical inhibitory neurons is:*

$$x_1^* = \frac{w_i^e\left(\gamma(n-1)+\beta\right)}{1-w_e}\sum_{k=2}^{n}\theta_k + \frac{1 - w_i^e\alpha(n-1)}{1-w_e}\theta_1$$

$$x_j^* = \gamma\sum_{k=2}^{n}\theta_k + \beta\theta_j - \alpha\theta_1 \text{ if } (j>1)$$

where:

$$\alpha = \frac{w_e^i}{w_e^i w_i^e(n-1) + (1-w_e)(-1+w_i+w_i^i(n-2))}$$

$$\gamma = \frac{-w_e^i w_i^e - w_i^i(1-w_e)}{(1-w_i+w_i^i)(w_e^i w_i^e(n-1)+(1-w_e)(-1+w_i+w_i^i(n-2)))}$$

$$\beta = \frac{1}{1-w_i+w_i^i}$$

and the synaptic weights as defined in Fig 2.3.

Proof. For the fixed point x^*, we can apply Lemma 1, to rewrite each x_j^* such that $j > 1$ in the form:

$$x_j^* = x_n^* + \frac{\theta_j - \theta_n}{1 - w_i + w_i^i} \tag{2.1}$$

Then, since x^* lies on the nullcline $\mathcal{N}_1$:

$$x_1^* = \frac{w_i^e}{1-w_e}\sum_{k=2}^{n}\left(x_n^* + \frac{\theta_k - \theta_n}{1-w_i+w_i^i}\right) + \frac{\theta_1}{1-w_e} \tag{2.2}$$

Thus, all x_j^* such that $j \neq n$ can be rewritten in terms of x_n^*.
As x^* lies on $\mathcal{N}_n$:

$$x_1^* = \frac{1 - w_i + w_i^i}{w_e^i} x_n^* - \frac{w_i^i}{w_e^i} \sum_{k=2}^{n} x_k^* - \frac{\theta_n}{w_e^i} \tag{2.3}$$

Using (2.1) and (2.2), equation (2.3) can be rewritten purely in terms of x_n^*:

$$\frac{w_i^e}{1 - w_e} \sum_{k=2}^{n} \left(x_n^* + \frac{\theta_k - \theta_n}{1 - w_i + w_i^i} \right) + \frac{\theta_1}{1 - w_e}$$

$$= \frac{1 - w_i + w_i^i}{w_e^i} x_n^* - \frac{w_i^i}{w_e^i} \sum_{k=2}^{n} \left(x_n^* + \frac{\theta_k - \theta_n}{1 - w_i + w_i^i} \right) - \frac{\theta_n}{w_e^i}$$

Solving for x_n^* yields:

$$x_n^* = \gamma \sum_{k=2}^{n} \theta_k + \beta \theta_n - \alpha \theta_1$$

Then, for $j > 1$:

$$x_j^* = x_n^* + \frac{\theta_j - \theta_n}{1 - w_i + w_i^i} = \gamma \sum_{k=2}^{n} \theta_k + \beta \theta_j - \alpha \theta_1$$

Similarly, using (2), x_1^* is:

$$x_1^* = \frac{w_i^e \left(\gamma(n-1) + \beta \right)}{1 - w_e} \sum_{k=2}^{n} \theta_k + \frac{1 - w_i^e \alpha (n-1)}{1 - w_e} \theta_1$$

□

Theorem 1. *In a TLN of one excitatory neuron, x_1, and $n-1$ identical inhibitory neurons, $x_{j>1}$, let x^* be the fixed point of $L_{[n]}$. Then, for $j \in [2, n]$, increasing θ_j will decrease x_j^* when $\frac{\tilde{\gamma}}{\tilde{\beta}} < 0$ where:*

$\tilde{\gamma} = w_e^i w_i^e (n-2) + (1 - w_e)(-1 + w_i + w_i^i (n-3))$

$\tilde{\beta} = (1 - w_i + w_i^i)(w_e^i w_i^e (n-1) + (1 - w_e)(-1 + w_i + w_i^i (n-2)))$

and the synaptic weights as defined in Fig 2.3.

Proof. Recall that we seek the condition such that x_j^* decreases as θ_j increases for $j > 1$. Focusing on the role of θ_j in the expression of x_j^*, we have:

$$x_j^* = \gamma \sum_{k=2}^{n} \theta_k + \beta \theta_j - \alpha \theta_1 = (\gamma + \beta) \theta_j + \gamma \sum_{j \neq k=2}^{n} \theta_k - \alpha \theta_1$$

We would then expect that the paradoxical effect would arise when $\gamma + \beta < 0$.

Now notice that:

$$\gamma + \beta = \frac{-w_e^i w_i^e - w_i^i(1-w_e)}{(1-w_i+w_i^i)(w_e^i w_i^e(n-1)+(1-w_e)(-1+w_i+w_i^i(n-2)))} + \frac{1}{1-w_i+w_i^i}$$
$$= \frac{w_e^i w_i^e(n-2)+(1-w_e)(-1+w_i+w_i^i(n-3))}{(1-w_i+w_i^i)(w_e^i w_i^e(n-1)+(1-w_e)(-1+w_i+w_i^i(n-2)))} = \frac{\tilde{\gamma}}{\tilde{\beta}} < 0.$$

□

We can find a simpler corollary if we assume that inhibitory neurons inhibit themselves to the same extent as they inhibit other neurons, i.e. $w_i = w_i^i$. Then we have:

Corollary 1. *In a TLN of one excitatory neuron, x_1, and $n-1$ identical inhibitory neurons, $x_{j>1}$, let x^* be the fixed point of $L_{[n]}$. Then, for $j \in [2,n]$, increasing θ_j will decrease x_j^* when $\frac{\tilde{\gamma}}{\tilde{\beta}} < 0$ where:*

$\tilde{\gamma} = w_e^i w_i^e(n-2) + (1-w_e)(-1+w_i(n-2))$

$\tilde{\beta} = w_e^i w_i^e(n-1) + (1-w_e)(-1+w_i(n-1))$

and the synaptic weights as defined in Fig 2.3.

Notice that this analysis involved treating the nonlinear TLN as a linear system, which we could do precisely because of the piecewise linearity of the ReLU function. In a network which used a more complex firing rate function this analysis could have been far more challenging. The idea of using a chamber by chamber linear dynamical analysis is an approach we will come back to repeatedly and is at the heart of the rich body of theoretical results in TLN literature.

2.2 Binary Competition Model

A strategy that has been used in attractor network models of decision-making is to reduce a larger network of neurons down to a smaller network of neural populations [23]. Part of this involves treating populations of excitatory neurons as if they are effectively inhibiting one another, something we already see briefly with the discussion of CTLNs in the introduction.

Regions of the cerebral cortex associated with decision-making, such as the prefrontal cortex and the LIP, are thought to consist of modular networks of excitatory neurons immersed within a sea of inhibitory interneurons [24]. This means that we can think of a decision-making task as being handled by a modular decision-making circuit of excitatory neurons. We take as a modeling assumption that the excitatory neurons are effectively inhibiting each other through the interneurons, with excitatory connections

merely reducing the inhibition. What this would mean mathematically is that all the synaptic weights are negative. This yields the notion of *competitive TLNs*.

Definition 4 (Definition 5.1 in [25])**.** *We say that a TLN with weight matrix W and input current vector $\vec{\theta}$ is* ***competitive*** *if $W_{ij} \leq 0$ and $W_{ii} = 0$ $\forall i, j \in [n]$ with $\vec{\theta} \geq 0$. The TLN is further said to be* ***non-degenerate*** *if [25]:*

- $\theta_i > 0$ *for at least one* $i \in [n]$
- $\det(I - W|_\sigma) \neq 0$ *for every* $\sigma \subseteq [n]$.
- *For each* $\sigma \subseteq [n]$ *such that* $\forall i \in \sigma, \theta_i > 0$*, the corresponding Cramer's determinant is nonzero:* $\det((I - W|_\sigma)_i; \vec{\theta}|_\sigma)$

where the notation $(A_i; \vec{b})$ represents the matrix with the i-th column removed and replaced by $\vec{b}$. The first non-degeneracy condition ensures that the origin is not a fixed point. The second non-degeneracy condition makes sure that no chamber has a non-degenerate linear system. The third non-degeneracy condition ensures that the fixed points of linear systems in adjacent chambers do not both lie on the shared wall of the chambers [25].

We will be considering competitive, non-degenerate TLNs of the form:

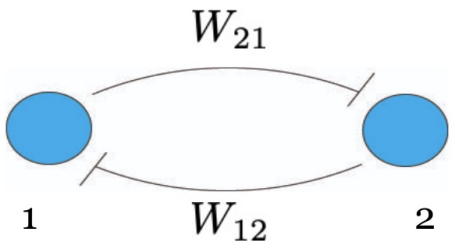


Figure 2.4. Binary Competition Model. Competitive model of two neural populations.

$$\frac{dx_1}{dt} = -x_1 + [W_{12}x_2 + \theta_1]_+$$
$$\frac{dx_2}{dt} = -x_2 + [W_{21}x_1 + \theta_2]_+$$

where $W_{12}, W_{21} < 0$ and $\theta_1, \theta_2 > 0$. The network architecture is depicted in Fig 2.4. The idea here is that each of the two neural populations corresponds to one choice in a binary choice decision-making task. We refer to this as the Binary Competition Model.

The qualitative nature of a bistable model is hardly original and there exist multistable models in mathematical biology with similar dynamics (e.g. the competitive exclusion

case of the Lotka-Volterra equations as described in section 6.4 of [26] and genetic toggle switch models such as that described in [27]), such models tend to have nonlinearities in their differential equations which make them more challenging to use for studying the basins of attraction [28]. Using the piecewise linearity of TLN equations, we will not only be able to calculate the seperatrix, but also integrate beneath it to determine the sizes of the basins of attraction relative to one another.

A challenge with basins of attraction is that they require an understanding of the global dynamics of a dynamical system and to understand how they evolve we need to be able to track those global dynamics through parameter space. Our approach will be to appeal to *combinatorial dynamics* where the strategy is to impose a combinatorial structure of the state space. We will partition it into regions with parameter dependent boundaries and then see how trajectories beginning in a region flow into others. We represent the regions as the vertices on a graph and draw directed edges between them if there exists a trajectory starting in one region that exits into another, creating a state transition graph. If a state transition graph is such that each vertex has only one outgoing edge, i.e. trajectories beginning in one region flow into only one one other, then we call such a state transition graph a **trajectory graph**.

The first partitions we make will naturally be using the ReLU hyperplane arrangement $\{H_i\}_{i=1}^2$ (Fig 2.5). We have:

$$H_1 : x_2 = -\frac{\theta_1}{W_{12}}$$

$$H_2 : x_1 = -\frac{\theta_2}{W_{21}}$$

Remark 3. *The intersection of the two lines H_1 and H_2 corresponds to the balanced state, x_{bs}, for the Binary Competition Model.*

Through standard ODE analysis we solve for the solutions of the linear systems in each of these chambers:

Lemma 2. *For the Binary Competition Model class of TLNs, the chamber by chamber linear ODE systems induced by the ReLU hyperplane partition and given in Figure 2.6,*

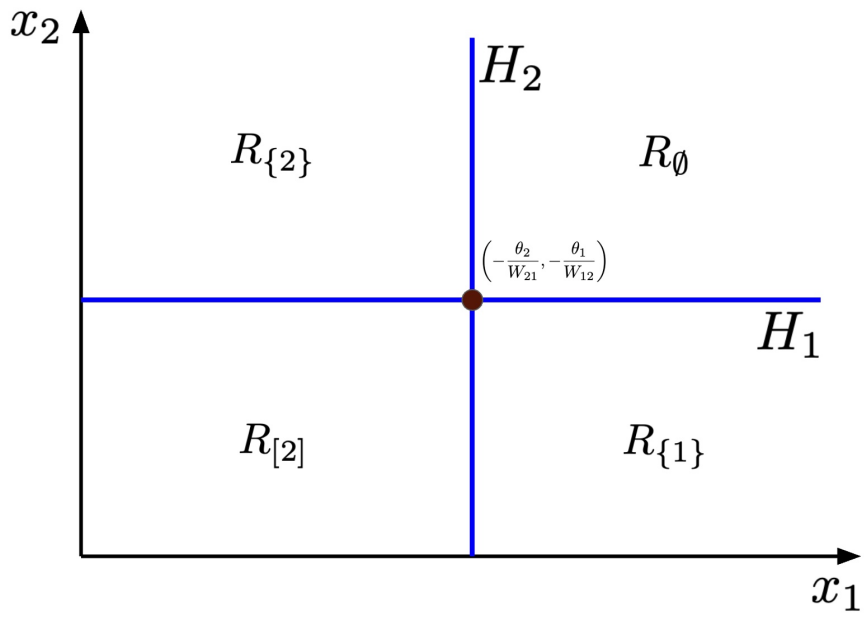


Figure 2.5. ReLU Partition. The hyperplane arrangement $\{H_i\}_{i=1}^2$ divides the state space into regions with linear dynamics.

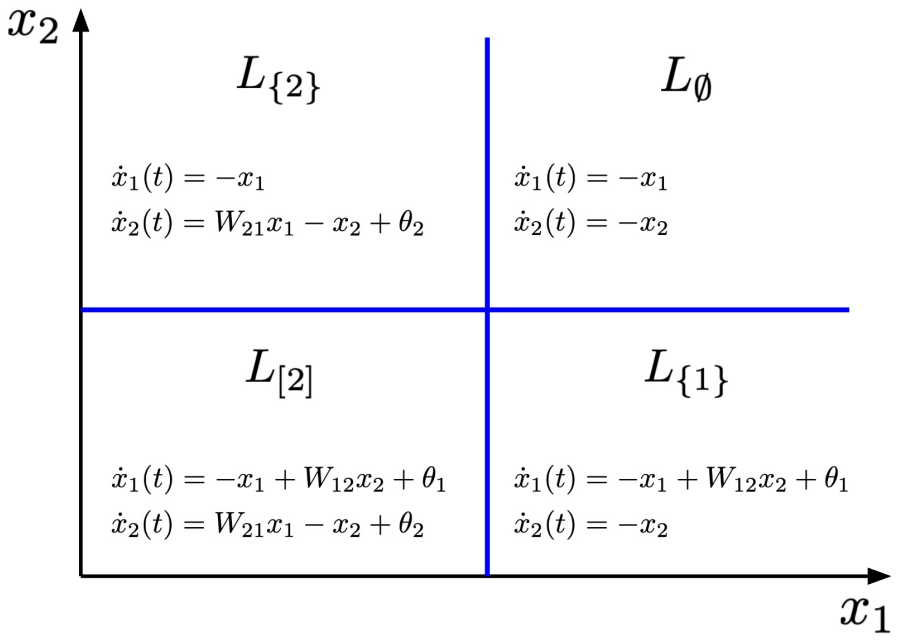


Figure 2.6. Linear ODE systems L_σ for Binary Competition Model.

have solutions:

$L_\emptyset$*:*

$x_1(t) = x_1^0 e^{-t}$

$x_2(t) = x_2^0 e^{-t}$

$L_{\{1\}}$**:**

$x_1(t) = x_2^0 W_{12} t e^{-t} + (x_1^0 - \theta_1)e^{-t} + \theta_1$

$x_2(t) = x_2^0 e^{-t}$

$L_{\{2\}}$**:**

$x_1(t) = x_1^0 e^{-t}$

$x_2(t) = x_1^0 W_{21} t e^{-t} + (x_2^0 - \theta_2)e^{-t} + \theta_2$

$L_{\{1,2\}}$**:**

$$x_1(t) = c_1\sqrt{-W_{12}}e^{(-1-\sqrt{W_{12}W_{21}})t} - c_2\sqrt{-W_{12}}e^{(-1+\sqrt{W_{12}W_{21}})t} + \frac{\theta_1 + W_{12}\theta_2}{1+|W|}$$

$$x_2(t) = c_1\sqrt{-W_{21}}e^{(-1-\sqrt{W_{12}W_{21}})t} + c_2\sqrt{-W_{21}}e^{(-1+\sqrt{W_{12}W_{21}})t} + \frac{W_{21}\theta_1 + \theta_2}{1+|W|}$$

$$c_1 = \frac{(1+|W|)(x_1^0 + x_2^0) - (\theta_1 + \theta_2 + W_{21}\theta_1 + W_{12}\theta_2)}{(1+|W|)(\sqrt{-W_{12}} + \sqrt{-W_{21}})}$$

$$c_2 = \frac{(1+|W|)(x_2^0 - x_1^0) - (\theta_2 - \theta_1 + W_{21}\theta_1 - W_{12}\theta_2)}{(1+|W|)(\sqrt{-W_{12}} + \sqrt{-W_{21}})}$$

where $|W| = -W_{12}W_{21}$ *is the determinant of* W*.*

We will also introduce a technical lemma which we will use across our calculations.

Lemma 3. *Let* $(a, b) \in \mathbb{R}^2_+$ *and define a vector field according to the linear system* $L_{\{1\}}$ *of the Binary Competition Model. Then, the implicitization of the trajectory passing through* (a, b) *is:*

$$x_1 = \left(\frac{a-\theta_1}{b}\right) x_2 - W_{12} x_2 \ln\left(\frac{x_2}{b}\right) + \theta_1.$$

Alternatively, if the vector field were defined according to $L_{\{2\}}$*, the implicitization of the alternative trajectory passing through* (a, b) *would be:*

$$x_2 = \left(\frac{b-\theta_2}{a}\right) x_1 - W_{21} x_1 \ln\left(\frac{x_1}{a}\right) + \theta_2.$$

Proof. The solutions for $L_{\{1\}}$ are of the form:

$x_1(t) = W_{12} x_2^0 t e^{-t} + (x_1^0 - \theta_1)e^{-t} + \theta_1$

$x_2(t) = x_2^0 e^{-t}$

Let $(a, b) \in \mathbb{R}^2_+$. For the trajectory of $L_{\{1\}}$ passing through (a, b), $\exists t^*$ such that:

$a = W_{12} x_2^0 t^* e^{-t^*} + (x_1^0 - \theta_1)e^{-t^*} + \theta_1$

$b = x_2^0 e^{-t^*}$.

The first expression can be manipulated into the following form after multiplying on both sides by e^{t^*}:

$$x_1^0 = (a - \theta_1)e^{t^*} - W_{12}x_2^0 t^* + \theta_1.$$

Manipulating the second expression, we have $e^{t^*} = \frac{x_2^0}{b}$ and $t^* = \ln\left(\frac{x_2^0}{b}\right)$. Substituting, we have:

$$x_1^0 = \left(\frac{a - \theta_1}{b}\right) x_2^0 - W_{12}x_2^0 \ln\left(\frac{x_2^0}{b}\right) + \theta_1.$$

This relation defines the set of initial conditions passing through (a, b) in positive or negative time. But this is exactly the set of points of the trajectory. Thus we have the implicitization of the trajectory as:

$$x_1 = \left(\frac{a - \theta_1}{b}\right) x_2 - W_{12}x_2 \ln\left(\frac{x_2}{b}\right) + \theta_1.$$

Performing this same analysis for $L_{\{2\}}$, we obtain the implicitization:

$$x_2 = \left(\frac{b - \theta_2}{a}\right) x_1 - W_{21}x_1 \ln\left(\frac{x_1}{a}\right) + \theta_2.$$

□

For convenience, we define functions for these expressions:

Definition 5.

$$f_1(a, b, x) := \left(\frac{a - \theta_1}{b}\right) x - W_{12}x \ln\left(\frac{x}{b}\right) + \theta_1.$$

$$f_2(a, b, x) := \left(\frac{b - \theta_2}{a}\right) x - W_{21}x \ln\left(\frac{x}{a}\right) + \theta_2.$$

Lemma 3 makes it easy to piece trajectories across chambers and its utility will quickly become evident.

2.2.1 The Bistable Symmetric Case

A critical aspect of our strategy for determining trajectory graphs will be nullcline analysis. We will get a feel for the nullclines in this system by looking at a perfectly symmetric TLN with $\theta_1 = \theta_2 = \theta$ and $W_{12} = W_{21} = -1 - \delta$ taking $\delta > 0$. Note that this corresponds to a CTLN derived from an independent set of two neurons. This is known

to be a bistable system with one attractor supported on each neuron and a saddle point supported on both of them. The connectivity matrix has the form:

$$W = \begin{bmatrix} 0 & -1-\delta \\ -1-\delta & 0 \end{bmatrix}.$$

The next step will be to introduce the nullclines into our partition where the nullclines are given by the equations:

$$\mathcal{N}_1 : x_1 = [W_{12}x_2 + \theta_1]_+$$

$$\mathcal{N}_2 : x_2 = [W_{21}x_1 + \theta_1]_+$$

Looking at our partition, we can start labeling our chambers using a canonical labelling scheme. For a partial partition using just chamber boundaries and nullclines associated with a network of n neurons, the region R will be labelled $S_1(R)S_2(R)\dots S_{2n}(R)$ derived from the string $H_1\mathcal{N}_1H_2\mathcal{N}_2\dots H_n\mathcal{N}_n$ where:

$$S_i(R) = \begin{cases} 0 & \text{if } R \text{ lies above the corresponding nullcline or chamber boundary} \\ 1 & \text{if } R \text{ lies below the corresponding nullcline or chamber boundary} \end{cases}$$

So, consider the case for a network of two neurons and in particular the region such that:

$$x_2 > \frac{-\theta_1}{W_{12}}, x_1 \leq \frac{-\theta_2}{W_{21}}$$

and

$$x_1 > [W_{12}x_2 + \theta_1]_+, x_2 \leq [W_{21}x_1 + \theta_2]_+$$

That is to say, the region lying above H_1 and $\mathcal{N}_1$ while lying below H_2 and $\mathcal{N}_2$.

The label for this region will then be:

$$\begin{array}{cccc} H_1 & \mathcal{N}_1 & H_2 & \mathcal{N}_2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 1 \end{array}$$

$$S_1S_2S_3S_4 = 0011$$

Our convention will also be to treat the label as a binary representation read left to right rather than the usual right to left.

So, for the region 0011, the corresponding vertex in the graph structure will be numbered:

$$0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 = 0 + 0 + 4 + 8 = 12$$

So, in the case of the bistable, symmetric TLN, we have the labelling in Figure 2.7

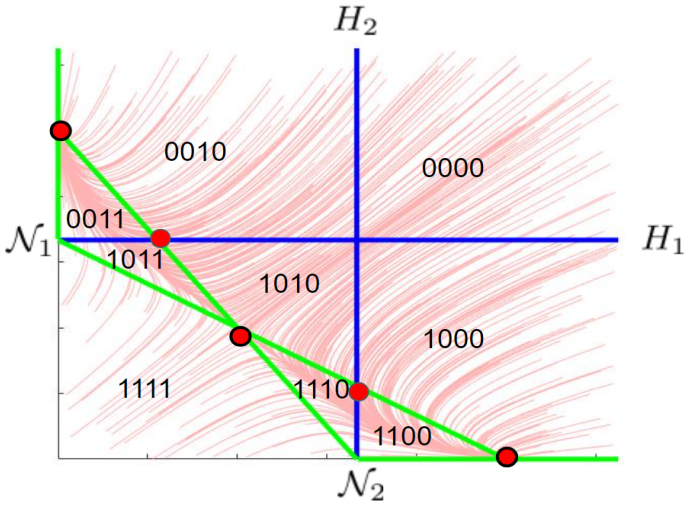


Figure 2.7. ReLU hyperplane and nullcline partition. Labelling of $H_i/\mathcal{N}_i$ partition for two neuron independent set CTLN.

From here we can build a state transition graph (Figure 2.8) using a nullcline analysis which tells us when x_1 and x_2 are increasing and decreasing respectively.

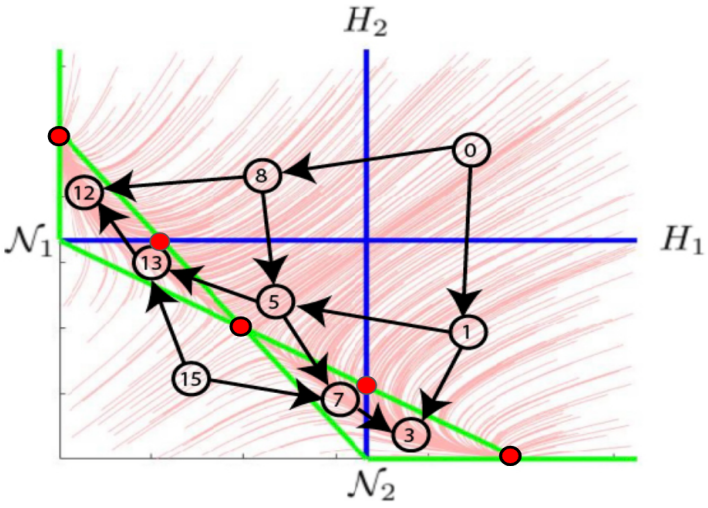


Figure 2.8. Canonical numbering of $H_i/\mathcal{N}_i$ partition. Canonical numbering scheme for partial trajectory graph of two neuron independent set CTLN

But notice that this state transition graph is not refined enough to give the basins of attraction for each attractor. We will have to break it down further. To obtain a fully refined trajectory graph, $R_{\{2\}}$ needs to be subdivided into the trajectories dropping into $R_{[2]}$ and those staying in the chamber with the situation similar for $R_{\{1\}}$. That division, which corresponds to a trajectory hitting the nullcline/chamber boundary intersection must then be traced through the other chambers as needed. Additionally, within $R_{[2]}$ and $R_{\emptyset}$ a diagonal separation is required.

Proposition 2. *The trajectory graph of the CTLN associated with a graph consisting of an independent set with two nodes takes the form of Figure 2.9 where:*

The Chamber Boundary equations are:

$$H_1 : x_2 = \frac{\theta}{1+\delta}$$

$$H_2 : x_1 = \frac{\theta}{1+\delta}$$

The nullclines are:

$$\mathcal{N}_1 : x_1 = [(-1-\delta)x_2 + \theta]_+$$

$$\mathcal{N}_2 : x_2 = [(-1-\delta)x_1 + \theta]_+$$

And the refining curves which cross through the nullcline/chamber boundary intersec-

tion are given by:

$$\mathcal{M}_1 : x_1 = \begin{cases} f_1\left(\frac{\theta}{1+\delta}, \frac{\theta\delta}{(1+\delta)^2}, x_2\right) & \text{if } \frac{\theta\delta}{(1+\delta)^2} \leq x_2 \leq \frac{\theta}{1+\delta} \\ \frac{(1+\delta)}{\theta} f_1\left(\frac{\theta}{1+\delta}, \frac{\theta\delta}{(1+\delta)^2}, \frac{\theta}{1+\delta}\right) x_2 & \text{if } \frac{\theta}{1+\delta} < x_2 \end{cases}$$

$$\mathcal{M}_2 : x_2 = \begin{cases} f_2\left(\frac{\theta\delta}{(1+\delta)^2}, \frac{\theta}{1+\delta}, x_1\right) & \text{if } \frac{\theta\delta}{(1+\delta)^2} \leq x_1 \leq \frac{\theta}{1+\delta} \\ \frac{(1+\delta)}{\theta} f_2\left(\frac{\theta\delta}{(1+\delta)^2}, \frac{\theta}{1+\delta}, \frac{\theta}{1+\delta}\right) x_1 & \text{if } \frac{\theta}{1+\delta} < x_1 \end{cases}$$

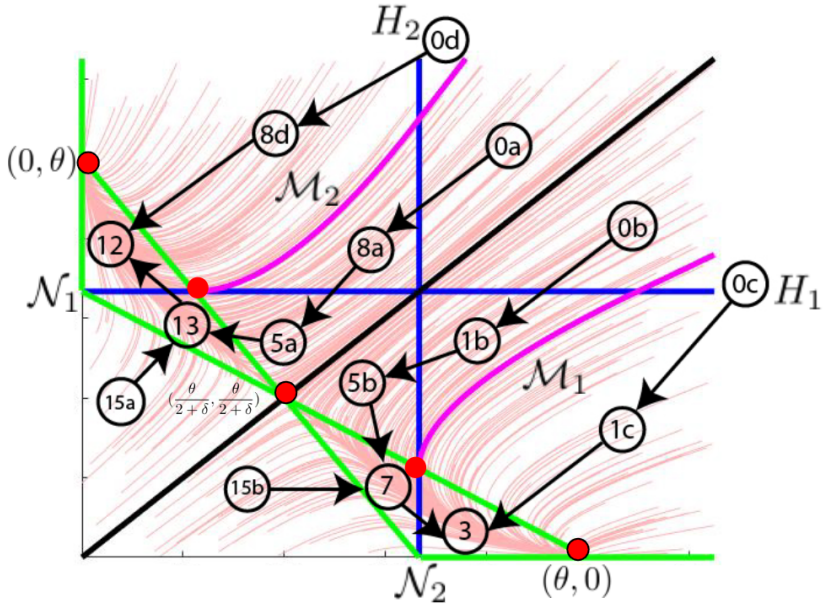


Figure 2.9. Trajectory graph for symmetric bistable TLN. The shown curves refine the state space into chambers which yield a trajectory graph. The light red curves in the background are simulated trajectories, confirming the accuracy of our trajectory graph.

Proof. In the chamber $R_{\{2\}}$ the solution to the ODE system is:

$$x_1(t) = x_1^0 e^{-t}$$

$$x_2(t) = (-1-\delta)x_1^0 t e^{-t} + \theta + (x_2^0 - \theta)e^{-t}$$

Since x_1 is strictly decreasing, trajectories in this chamber cannot enter $R_\emptyset$.

To see which trajectories would enter chamber $R_{[2]}$, we take the point where the nullcline $\mathcal{N}_2$ intersects H_1 as this separates the H_1 wall of $R_{\{2\}}$ into where trajectories flow beneath it and above it. This point is $\left(\frac{\theta\delta}{(1+\delta)^2}, \frac{\theta}{1+\delta}\right)$ and by applying Lemma 3 we obtain $\mathcal{M}_2$.

This boundary, restricted to values of x_1^0 where t^* (as defined in the proof of Lemma 3) is positive, is the magenta boundary which separates **8a**. Thus, **8a** $\longrightarrow$ **5a**. Additionally, since **8d** is above $\mathcal{M}_2$, its trajectories remain in $R_{\{2\}}$, So trajectories beginning there will go to the fixed point $(0, \theta)$. To see that they go to the fixed point via **12** simply resolve $x_2(t^*) < \theta$ and notice that the resulting inequality is trivial. So, **8d** $\longrightarrow$ **12**.

Without loss of generality, we can extend these results to $R_{\{1\}}$ as well to conclude **1b** $\longrightarrow$ **5b** and **1c** $\longrightarrow$ **3**.

Now consider chamber $R_{[2]}$. The solution to the ODE system in this chamber is:

$$x_1(t) = -\frac{1}{2}(x_2^0 - x_1^0)e^{\delta t} + \frac{1}{2}\left(x_2^0 + x_1^0 - \frac{2\theta}{2+\delta}\right)e^{-(2+\delta)t} + \frac{\theta}{2+\delta}$$

$$x_2(t) = \frac{1}{2}(x_2^0 - x_1^0)e^{\delta t} + \frac{1}{2}\left(x_2^0 + x_1^0 - \frac{2\theta}{2+\delta}\right)e^{-(2+\delta)t} + \frac{\theta}{2+\delta}$$

Since $e^{\delta t}$ is strictly increasing to ∞, this indicates that the trajectories cannot remain in this chamber. Additionally $x_2(t) > x_1(t)$ precisely when $\frac{1}{2}(x_2^0 - x_1^0)e^{\delta t} > -\frac{1}{2}(x_2^0 - x_1^0)e^{\delta t}$. This holds true when $x_2^0 > x_1^0$. So if a trajectory begins above the diagonal it must remain above the diagonal, and if a trajectory begins below the diagonal it must remain below the diagonal. Thus, trajectories cannot cross the diagonal as well.

For **5a**, this means that trajectories cannot move into **5b** and since the region lies above both nullclines the trajectories must decrease in x_1 and x_2 into **13**. Thus **5a** $\longrightarrow$ **13**. By similar reasoning, **5b** $\longrightarrow$ **7**.

Since **15a** lies below both nullclines, it must increase into **13**. So, **15a** $\longrightarrow$ **13** and without loss of generality **15b** $\longrightarrow$ **7**.

Now, for **13**, to leave from above through the x_2-nullcline it would have to be through a point where $\frac{dx_1}{dt}$ is positive, which is impossible since **13** lies abov**12**e the x_1-nullcline. Similarly, to leave from below through the x_1-nullcline it must be through a point where $\frac{dx_2}{dt}$ is negative, which is again impossible as the **13** lies below the x_2-nullcline. Since trajectories cannot remain within the chamber, **13** $\longrightarrow$ **12** and again without loss of generality **7** $\longrightarrow$ **3**.

We finally turn our attention to $R_\emptyset$. The solution in this chamber is

$$x_1(t) = x_1^0 e^{-t}$$

$$x_2(t) = x_2^0 e^{-t}$$

Then, $x_2 = \dfrac{x_2^0}{x_1^0} x_1$ which is linear and in particular is above the diagonal if $x_2^0 > x_1^0$ and below if $x_2^0 < x_1^0$.

So, if $x_1^0 > x_2^0$, then the trajectory hits H_1 first and enters $R_{\{1\}}$. If the other way around, it enters $R_{\{2\}}$.

The only remaining question is which trajectories beginning in the upper diagonal enter **8a** and which enter **8d**.

Since $\mathcal{M}_2$ intersects H_2 at $\left(\dfrac{\theta}{1+\delta}, f_2\left(\dfrac{\theta\delta}{(1+\delta)^2}, \dfrac{\theta}{1+\delta}, \dfrac{\theta}{1+\delta}\right)\right)$ and in $R_\emptyset$ $x_2 = \dfrac{x_2^0}{x_1^0} x_1$, we can extend $\mathcal{M}_2$ into $R_\emptyset$ accordingly.

So, **0d** $\longrightarrow$ **8d** and **0a** $\longrightarrow$ **8a**. By similar arguments, we also conclude that **0c** $\longrightarrow$ **1c** and **0b** $\longrightarrow$ **1b**.

□

These trajectory graphs are a combinatorial representation of the dynamics and in particular the basins of attraction. Given an initial condition (x_1^0, x_2^0), by finding which region it falls within and tracing the path through the trajectory graph to the end region, we can determine to which point attractor it converges.

For the independent set, the basin for the upper-left fixed point is the region above the diagonal while the basin for the bottom-right fixed point is the region below the diagonal. For $x_1^0 = x_2^0$, you have convergence to the unstable fixed point on the diagonal. The diagonal is the stable manifold separating the basins of attraction for the attractors.

2.2.2 Trajectory Graphs of the Binary Competition Model

The Binary Competition Model is a patchwork of four linear systems: $L_\emptyset, L_1, L_2$, and L_{12}. The $R_\emptyset$ chamber will never have a fixed point as the fixed point of $L_\emptyset$ is $(0, 0)$. This means the TLNs in this class can have up to three fixed points. The $H_i/\mathcal{N}_i$ arrangements are closely related to the fixed points of the TLN.

Lemma 4. *For the Binary Competition Model, assume without loss of generality that $\theta_2 \geq \theta_1$. Then we have the bifurcation diagram on fixed point supports depicted in Figure 2.10.*

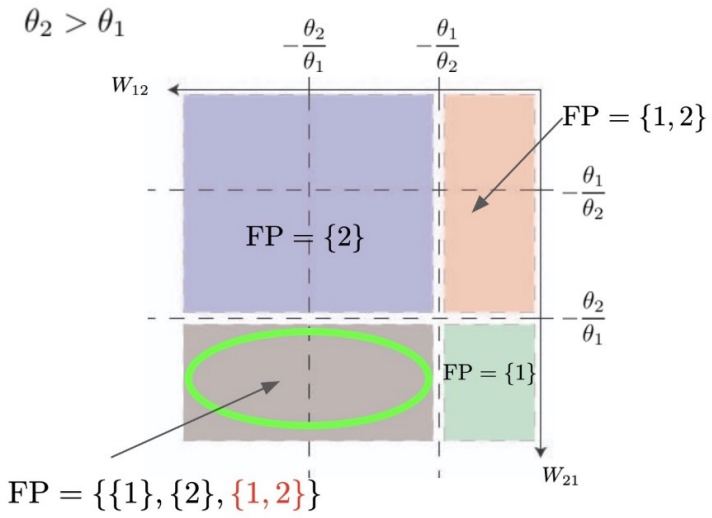


Figure 2.10. Bifurcations on fixed point supports of the Binary Competition Model.. This bifurcation diagram depicts the set of fixed point supports in the various regions of the parameter space.

Proof. From standard fixed point analysis we know that the fixed points of linear systems L_1, L_2, and $L_{1,2}$ are as follows:

$L_1 : (\theta_1, 0)$

$L_2 : (0, \theta_2)$

$L_{1,2} : \left(\dfrac{\theta_1 + W_{12}\theta_2}{1 - W_{12}W_{21}}, \dfrac{W_{21}\theta_1 + \theta_2}{1 - W_{12}W_{21}} \right)$

Each of these will be fixed points of the overall TLN if they lie in the chambers $R_{\{1\}}$, $R_{\{2\}}$, and $R_{[2]}$ respectively. These chambers are generated by the hyperplane arrangement:

$$H_1 : W_{12}x_2 + \theta_1 = 0 \implies x_2 = -\frac{\theta_1}{W_{12}}$$

$$H_2 : W_{21}x_1 + \theta_2 = 0 \implies x_1 = -\frac{\theta_2}{W_{21}}.$$

Now we consider each case separately:

Case 1: $R_{\{1\}}$

In this case, we would need the L_1 fixed point to lie inside H_1 and outside H_2. Being inside H_1 is trivially met as $0 < -\dfrac{\theta_1}{W_{12}}$. Alternatively, to lie outside H_2 requires $\theta_1 > -\dfrac{\theta_2}{W_{21}}$. Bearing in mind that W_{21} is negative, this rearranges to $W_{21} < -\dfrac{\theta_2}{\theta_1}$. So, we conclude that the $R_{\{1\}}$ fixed point exists when $W_{21} < -\dfrac{\theta_2}{\theta_1}$.

Case 2: $R_{\{2\}}$

Applying the arguments from Case 1 without loss of generality, we conclude that the $R_{\{2\}}$ fixed point exists when $W_{12} < -\frac{\theta_1}{\theta_2}$.

Case 3: $R_{[2]}$

For $R_{[2]}$ to have a fixed point, the $L_{1,2}$ fixed point must lie within both H_1 and H_2. So, we must have:

$$\frac{\theta_1 + W_{12}\theta_2}{1 - W_{12}W_{21}} < -\frac{\theta_2}{W_{21}} \text{ and } \frac{W_{21}\theta_1 + \theta_2}{1 - W_{12}W_{21}} < -\frac{\theta_1}{W_{12}}.$$

For the first condition, we can rearrange as follows:

$$\frac{\theta_1 + W_{12}\theta_2}{1 - W_{12}W_{21}} < -\frac{\theta_2}{W_{21}} \implies \frac{\theta_1 + W_{12}\theta_2}{1 - W_{12}W_{21}} + \frac{\theta_2}{W_{21}} < 0 \implies$$

$$\frac{W_{21}\theta_1 + W_{12}W_{21}\theta_2 + \theta_2 - W_{12}W_{21}\theta_2}{W_{21}(1 - W_{12}W_{21})} < 0 \implies \frac{W_{21}\theta_1 + \theta_2}{1 - W_{12}W_{21}} > 0.$$

Through similar manipulation the second condition becomes:

$$\frac{W_{21}\theta_1 + \theta_2}{1 - W_{12}W_{21}} < -\frac{\theta_1}{W_{12}} \implies \frac{\theta_1 + W_{12}\theta_2}{1 - W_{12}W_{21}} > 0.$$

Then, it is clear that we have two regions of the parameter space where this dual support fixed point occurs. If $W_{21} < \frac{1}{W_{12}}$ (i.e. $1 - W_{12}W_{21} > 0$), then the fixed point exists when $W_{21} > -\frac{\theta_2}{\theta_1}$ and $W_{12} > -\frac{\theta_1}{\theta_2}$. Alternatively, if $W_{21} > \frac{1}{W_{12}}$, the fixed point exists when $W_{21} < -\frac{\theta_2}{\theta_1}$ and $W_{12} < -\frac{\theta_1}{\theta_2}$.

Our last consideration is the stability of the $R_{[2]}$ fixed point. We find the eigenvalues of the $-I + W$ matrix.

$$\det\left(\begin{bmatrix} -1-\lambda & W_{12} \\ W_{21} & -1-\lambda \end{bmatrix}\right) = \lambda^2 + 2\lambda + 1 - W_{12}W_{21} = 0.$$

The characteristic polynomial has roots $\lambda_1 = -1 - \sqrt{W_{12}W_{21}}$ and $\lambda_2 = -1 + \sqrt{W_{12}W_{21}}$. The eigenvalue λ_1 is clearly negative, but the sign of λ_2 is parameter dependent:

$$\lambda_2 > 0 \iff \sqrt{W_{12}W_{21}} > 1.$$

Squaring both sides and rearranging, we have:

$$\sqrt{W_{12}W_{21}} > 1 \iff W_{21} > \frac{1}{W_{12}}.$$

Thus, we conclude that in the $W_{21} > -\frac{\theta_2}{\theta_1}$ and $W_{12} > -\frac{\theta_1}{\theta_2}$ regime, the $R_{[2]}$ fixed point is stable whereas in the $W_{21} < -\frac{\theta_2}{\theta_1}$ and $W_{12} < -\frac{\theta_1}{\theta_2}$ regime it is a saddle point. □

Corollary 2. *For the Binary Competition Model, assume without loss of generality that $\theta_2 \geq \theta_1$. Then we have the bifurcation diagram on $H_i/\mathcal{N}_i$ arrangements depicted in Figure 2.11.*

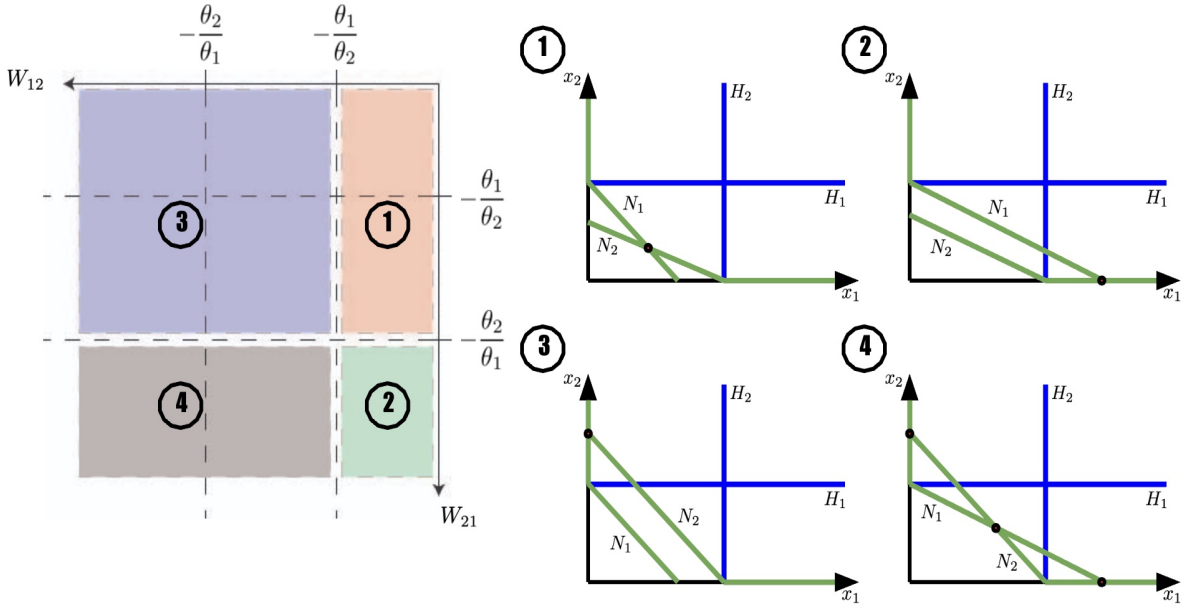


Figure 2.11. Bifurcation of $H_i/\mathcal{N}_i$ arrangement for Binary Competition Model Within marked regions of the parameter space, the $H_i/\mathcal{N}_i$ arrangement is as indicated. By convention, $\theta_2 \geq \theta_1$.

From this we can see that the key difference in trajectory graphs between the CTLN cases and the more general Binary Competition Model is the placement of the positively oriented stable eigenspace of the $L_{1,2}$ fixed point, where it exists. In the symmetric case it was the diagonal, but now it can be more freely moved. Our key consideration is whether it continues from $R_{[2]}$ into $R_{\{1\}}$ or $R_{\{2\}}$.

Lemma 5. *For the Binary Competition Model, with parameters in Zone 4 of Fig 2.11, if $W_{21} < \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$, then the stable manifold of the saddle point in $R_{[2]}$ intersects the H_1 wall of the chamber. If $W_{21} > \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$, then it intersects the H_2 wall of the chamber.*

For parameters in Zone 1 of Fig 2.11, if $W_{21} > \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$*, then the stable manifold of the saddle point in* $R_{[2]}$ *intersects the* H_1 *wall of the chamber. If* $W_{21} < \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$*, then it intersects the* H_2 *wall of the chamber.*

Proof. For the linear system $L_{[2]}$, the stable manifold corresponds to the eigenspace created by the eigenvector $\begin{bmatrix} \sqrt{-W_{12}} \\ \sqrt{-W_{21}} \end{bmatrix}$. Then, the stable manifold in $R_{[2]}$ is a line with slope $\sqrt{\frac{W_{21}}{W_{12}}}$ passing through the fixed point $x^* = \left(\frac{\theta_1 + W_{12}\theta_2}{1+|W|}, \frac{W_{21}\theta_1 + \theta_2}{1+|W|}\right)$.

Now consider the slope of the line passing through both x^* and the intersection of the ReLU lines (the corner of the $R_{[2]}$ chamber), the point $x_{bs} = \left(-\frac{\theta_2}{W_{21}}, -\frac{\theta_1}{W_{12}}\right)$. After some calculation we find that:

$$\frac{(x_{bs})_2 - x_2^*}{(x_{bs})_1 - x_1^*} = \frac{W_{21}(\theta_1 + W_{12}\theta_2)}{W_{12}(W_{21}\theta_1 + \theta_2)}.$$

Then, if the slope of the stable manifold is less than this slope, the stable manifold intersects the H_2 wall of $R_{[2]}$ and continues into $R_{\{1\}}$. That is to say, it does so if and only if:

$$\sqrt{\frac{W_{21}}{W_{12}}} < \frac{W_{21}(\theta_1 + W_{12}\theta_2)}{W_{12}(W_{21}\theta_1 + \theta_2)} \implies \sqrt{\frac{W_{12}}{W_{21}}} < \frac{\theta_1 + W_{12}\theta_2}{W_{21}\theta_1 + \theta_2}.$$

Alternatively, the stable manifold continues into $R_{\{2\}}$ if and only if:

$$\sqrt{\frac{W_{12}}{W_{21}}} > \frac{\theta_1 + W_{12}\theta_2}{W_{21}\theta_1 + \theta_2}.$$

Consider the first inequality. Since we are in Zone 4 of Fig 2.11, we have $W_{21}\theta_1 + \theta_2 < 0$. Thus,

$$\sqrt{\frac{W_{12}}{W_{21}}} < \frac{\theta_1 + W_{12}\theta_2}{W_{21}\theta_1 + \theta_2} \implies \sqrt{-W_{12}}(W_{21}\theta_1 + \theta_2) > \sqrt{-W_{21}}(\theta_1 + W_{12}\theta_2).$$

For the time being, we use the more compact notation $\alpha = \sqrt{-W_{12}}$ and $\beta = \sqrt{-W_{21}}$ and we note that $\alpha, \beta > 0$. We can then rewrite the inequality as:

$$\alpha(-\beta^2\theta_1 + \theta_2) > \beta(\theta_1 - \alpha^2\theta_2) \implies (\beta\theta_2)\alpha^2 + (\theta_2 - \beta^2\theta_1)\alpha - \beta\theta_1 > 0.$$

This is a quadratic inequality in α with a positive leading coefficient. So, the inequality

is satisfied for $\alpha < \alpha_1$ and $\alpha > \alpha_2$ where α_1, α_2 are the solutions of $(\beta\theta_2)\alpha^2 + (\theta_2 - \beta^2\theta_1)\alpha - \beta\theta_1 = 0$ such that $\alpha_1 \leq \alpha_2$.

Solving the quadratic, we obtain $\alpha_1 = -\beta^{-1}$ and $\alpha_2 = \dfrac{\theta_1}{\theta_2}\beta$. As $\alpha > 0$, the inequality $\alpha < -\beta^{-1}$ yields no solutions. That leaves $\alpha > \dfrac{\theta_1}{\theta_2}\beta$. This is equivalent to:

$$\sqrt{-W_{12}} > \frac{\theta_1}{\theta_2}\sqrt{-W_{21}}.$$

Squaring both sides and rearranging, we obtain:

$$W_{21} > \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}.$$

This is the condition for a TLN with parameters in Zone 4 of Fig 2.11 to have the stable manifold of the $R_{[2]}$ fixed point intersect the H_2 wall of the chamber and continue into $R_{\{1\}}$.

Similarly, taking the inequality in the other direction, we get the condition to intersect the H_1 wall.

Using the same process, we get the conditions for Zone 1 of Fig 2.11 as well. □

Based on this we can see that the bistable regime can yield two possible trajectory graphs.

Proposition 3. *For the Binary Competition Model, assume without loss of generality that $\theta_2 \geq \theta_1$ and that the conditions $W_{12} < -\dfrac{\theta_1}{\theta_2}$ and $W_{21} < -\dfrac{\theta_2}{\theta_1}$.*

If $W_{21} > \left(\dfrac{\theta_2}{\theta_1}\right)^2 W_{12}$, then the TLN has the trajectory graph in Figure 2.12A.

If $W_{21} < \left(\dfrac{\theta_2}{\theta_1}\right)^2 W_{12}$, then it has the trajectory graph in Figure 2.12B.

where, for $m = \sqrt{\dfrac{W_{12}}{W_{21}}}$, the refining curves are:

$\mathcal{M}_{(i)}$: Initial conditions for trajectories passing through $H_i/\mathcal{N}_i$ intersections in positive time (Magenta)

$$\mathcal{M}_1 : x_1 = \begin{cases} f_1\left(-\dfrac{\theta_2}{W_{21}}, \dfrac{W_{21}\theta_1 + \theta_2}{|W|}, x_2\right) & \text{if } \dfrac{W_{21}\theta_1 + \theta_2}{|W|} \leq x_2 \leq -\dfrac{\theta_1}{W_{12}} \\ -\dfrac{W_{12}}{\theta_1} f_1\left(-\dfrac{\theta_2}{W_{21}}, \dfrac{W_{21}\theta_1 + \theta_2}{|W|}, -\dfrac{\theta_1}{W_{12}}\right) x_2 & \text{if } -\dfrac{\theta_1}{W_{12}} < x_2 \end{cases}$$

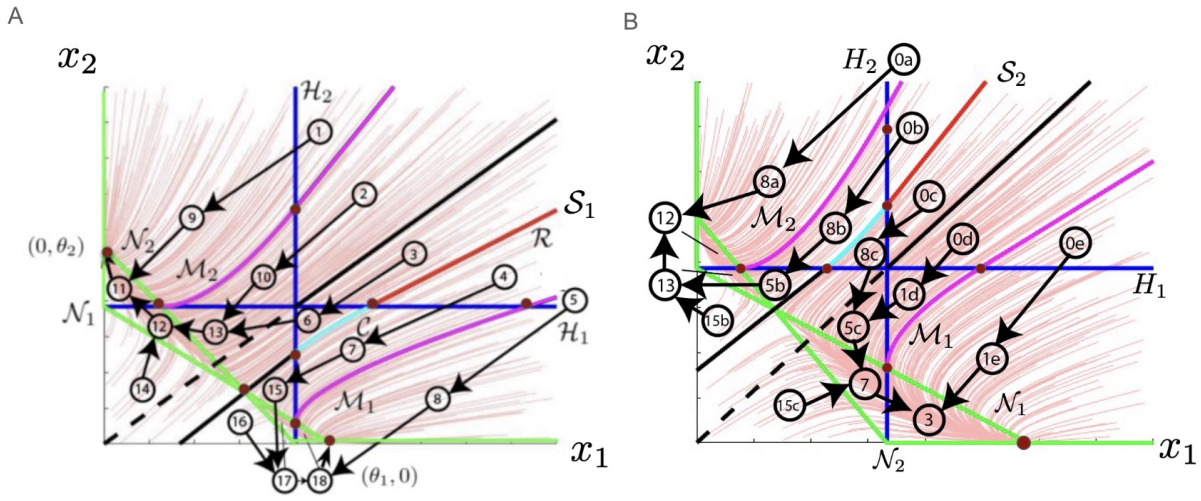


Figure 2.12. Trajectory graphs under bistable parameter regime. (A) Trajectory graph if stable manifold crosses $R_{\{1\}}$. (B) Trajectory graph if stable manifold crosses $R_{\{2\}}$.

$$\mathcal{M}_2 : x_2 = \begin{cases} f_2\left(\dfrac{\theta_1 + W_{12}\theta_2}{|W|}, -\dfrac{\theta_1}{W_{12}}, x_1\right) & \text{if } \dfrac{\theta_1 + W_{12}\theta_2}{|W|} \leq x_1 \leq -\dfrac{\theta_2}{W_{21}} \\ -\dfrac{W_{21}}{\theta_2} f_2\left(\dfrac{\theta_1 + W_{12}\theta_2}{|W|}, -\dfrac{\theta_1}{W_{12}}, -\dfrac{\theta_2}{W_{21}}\right) x_1 & \text{if } -\dfrac{\theta_2}{W_{21}} < x_1 \end{cases}$$

$\mathcal{S}_{(i)}$: Chamber by chamber breakdown of the positively oriented stable manifold $\mathcal{S}_1/\mathcal{S}_2$ for the $R_{[2]}$ fixed point (Black, Cyan, and Red)

$$\mathcal{S}_1 : x_1 = \begin{cases} mx_2 + \frac{(1-mW_{21})\theta_1 + (W_{12}-m)\theta_2}{1+|W|} & \text{if } 0 \leq x_2 \leq \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)} \\ f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)}, x_2\right) & \text{if } \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)} < x_2 \leq -\frac{\theta_1}{W_{12}} \\ -\frac{W_{12}}{\theta_1} f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)}, -\frac{\theta_1}{W_{12}}\right) x_2 & \text{if } -\frac{\theta_1}{W_{12}} < x_2 \end{cases}$$

$$\mathcal{S}_2 : x_2 = \begin{cases} m^{-1}x_1 + \frac{(1-m^{-1}W_{12})\theta_2 + (W_{21}-m^{-1})\theta_1}{1+|W|} & \text{if } 0 \leq x_1 \leq \frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)} \\ f_2\left(\frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)}, -\frac{\theta_1}{W_{12}}, x_1\right) & \text{if } \frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)} < x_1 \leq -\frac{\theta_2}{W_{21}} \\ -\frac{W_{21}}{\theta_2} f_2\left(\frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)}, -\frac{\theta_1}{W_{12}}, -\frac{\theta_2}{W_{21}}\right) x_1 & \text{if } -\frac{\theta_2}{W_{21}} < x_1 \end{cases}$$

Proof. These cases are very similar to that of Proposition 2 and most of the arguments

regarding nullclines carry over without loss of generality. We now find the $H_1/\mathcal{N}_2$ and $H_2/\mathcal{N}_1$ intersections to be $\left(\frac{\theta_1 + W_{12}\theta_2}{|W|}, -\frac{\theta_1}{W_{12}}\right)$ and $\left(-\frac{\theta_2}{W_{21}}, \frac{W_{21}\theta_1 + \theta_2}{|W|}\right)$ respectively. We then analogously find the curves $\mathcal{M}_{1,2}$ using Lemma 3.

The main difference from the previous case is the stable manifold. In the earlier case, the diagonal was the stable manifold, but by adjusting the parameters it is now free to move around. We begin with the $R_{[2]}$ chamber where we have the saddle point $x^* = \left(\frac{\theta_1 + W_{12}\theta_2}{1+|W|}, \frac{W_{21}\theta_1 + \theta_2}{1+|W|}\right)$. The eigenvector associated with the negative eigenvalue of W is $\begin{bmatrix} \sqrt{-W_{12}} \\ \sqrt{-W_{21}} \end{bmatrix}$. The stable manifold according to the $L_{[2]}$ system will then be of the form:

$$x_1 = mx_2 + b$$

where $m = \sqrt{\frac{W_{12}}{W_{21}}}$ and this line passes through x^*. So, $b = x_1^* - mx_2^*$. Then, we have:

$$x_1 = mx_2 + \frac{(1 - mW_{21})\theta_1 + (W_{12} - m)\theta_2}{1 + |W|}.$$

We can also rewrite this as:

$$x_2 = m^{-1}x_1 + \frac{(1 - m^{-1}W_{12})\theta_2 + (W_{21} - m^{-1})\theta_1}{1 + |W|}.$$

Applying Lemma 5, we determine whether to continue it into $R_{\{1\}}$ or $R_{\{2\}}$. Accordingly we apply Lemma 3 to further trace the stable manifold into the appropriate chamber.

Case 1: The stable manifold proceeds through $R_{\{1\}}$.

We write the stable manifold segment of $R_{[2]}$ as

$$x_1 = mx_2 + \frac{(1 - mW_{21})\theta_1 + (W_{12} - m)\theta_2}{1 + |W|}.$$

Then, its intersection with the H_2 $\left(\text{recall that this is } x_1 = -\frac{\theta_2}{W_{21}}\right)$ wall will be when:

$$-\frac{\theta_2}{W_{21}} = mx_2 + \frac{(1 - mW_{21})\theta_1 + (W_{12} - m)\theta_2}{1 + |W|}.$$

Resolving this, we have:

$$-\frac{\theta_2}{W_{21}} = mx_2 + \frac{(1 - mW_{21})\theta_1 + (W_{12} - m)\theta_2}{1 + |W|}.$$

Then,

$$mx_2 = \frac{(W_{21}m - 1)(W_{21}\theta_1 + \theta_2)}{W_{21}(1 + |W|)} \implies x_2 = \frac{(W_{21} - m^{-1})(W_{21}\theta_1 + \theta_2)}{W_{21}(1 + |W|)}.$$

Applying Lemma 3, we obtain the segment in $R_{\{1\}}$:

$$x_1 = f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21} - m^{-1})(W_{21}\theta_1 + \theta_2)}{W_{21}(1 + |W|)}, x_2\right)$$

It intersects the H_1 wall at the point:

$$\left(f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21} - m^{-1})(W_{21}\theta_1 + \theta_2)}{W_{21}(1 + |W|)}, -\frac{\theta_1}{W_{12}}\right), -\frac{\theta_1}{W_{12}}\right).$$

The linear continuation into $R_\emptyset$ is:

$$x_1 = -\frac{W_{12}}{\theta_1} f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21} - m^{-1})(W_{21}\theta_1 + \theta_2)}{W_{21}(1 + |W|)}, -\frac{\theta_1}{W_{12}}\right) x_2.$$

Thus, we have stable manifold $x_1 = \mathcal{S}_1(x_2)$ where:

$$\mathcal{S}_1(x_2) = \begin{cases} mx_2 + \frac{(1-mW_{21})\theta_1 + (W_{12}-m)\theta_2}{1+|W|} & \text{if } 0 \le x_2 \le \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)} \\ f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)}, x_2\right) & \text{if } \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)} < x_2 \le -\frac{\theta_1}{W_{12}} \\ -\frac{W_{12}}{\theta_1} f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21}-m^{-1})(W_{21}\theta_1+\theta_2)}{W_{21}(1+|W|)}, -\frac{\theta_1}{W_{12}}\right) x_2 & \text{if } -\frac{\theta_1}{W_{12}} < x_2 \end{cases}.$$

There exists a final technical point in confirming that the stable manifold intersects the x_1-axis wall of $R_{[2]}$ rather than the x_2-axis wall. This will hold true if, for the segment in $R_{[2]}$:

$$x_1 = mx_2 + \frac{(1 - mW_{21})\theta_1 + (W_{12} - m)\theta_2}{1 + |W|}.$$

the constant term is positive. That is:

$$\frac{(1 - mW_{21})\theta_1 + (W_{12} - m)\theta_2}{1 + |W|} > 0.$$

Now, from the proof of Lemma 4, we know that $-1 + \sqrt{W_{12}W_{21}} > 0$ in this parameter regime and so $1 + |W| = 1 - W_{12}W_{21} < 0$. So, we need to confirm that $(1 - mW_{21})\theta_1 + (W_{12} - m)\theta_2 < 0$ as well. Again using the compact notation $\alpha = \sqrt{-W_{12}}, \beta = \sqrt{-W_{21}}$, we can rewrite this as:

$$\theta_1 + \frac{\alpha\beta^2}{\beta}\theta_1 - \alpha^2\theta_2 - \frac{\alpha}{\beta}\theta_2 < 0 \implies (\beta\theta_2)\alpha^2 + (\theta_2 - \beta\theta_1)\alpha - \beta\theta_1 > 0.$$

But as we saw in the proof for Lemma 5, this condition translates to $W_{21} > \left(\frac{\theta_2}{\theta_2}\right)^2 W_{12}$ which is true by assumption.

Case 2: The stable manifold proceeds through $R_{\{2\}}$.

We write the stable manifold segment of $R_{[2]}$ as

$$x_2 = m^{-1}x_1 + \frac{(1 - m^{-1}W_{12})\theta_2 + (W_{21} - m^{-1})\theta_1}{1 + |W|}.$$

By symmetry, we obtain the stable manifold $x_2 = \mathcal{S}_2(x_1)$:

$$\mathcal{S}_2(x_1) = \begin{cases} m^{-1}x_1 + \frac{(1-m^{-1}W_{12})\theta_2 + (W_{21}-m^{-1})\theta_1}{1+|W|} & \text{if } 0 \le x_1 \le \frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)} \\ f_2\left(\frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)}, -\frac{\theta_1}{W_{12}}, x_1\right) & \text{if } \frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)} < x_1 \le -\frac{\theta_2}{W_{21}} \\ -\frac{W_{21}}{\theta_2} f_2\left(\frac{(W_{12}-m)(W_{12}\theta_2+\theta_1)}{W_{12}(1+|W|)}, -\frac{\theta_1}{W_{12}}, -\frac{\theta_2}{W_{21}}\right) x_1 & \text{if } -\frac{\theta_2}{W_{21}} < x_1 \end{cases}.$$

□

From here, it becomes easy to see that we have a bifurcation on the combinatorial dynamics of the Binary Competition Model.

Theorem 2. *For $\theta_2 \ge \theta_1$, the combinatorial structure of the trajectory graphs for a two neuron symmetric TLN with varying inputs bifurcates as depicted in Figure 2.13 with:*

$\mathcal{H}_i$: Rectifier Chamber Boundary for x_i (Blue)

$$\mathcal{H}_1 : x_2 = \frac{-\theta_1}{W_{12}}$$

$$\mathcal{H}_2 : x_1 = \frac{-\theta_2}{W_{21}}$$

$\mathcal{N}_i$: Nullclines for x_i (Green)

$$\mathcal{N}_1 : x_1 = [W_{12}x_2 + \theta_1]_+$$

$$\mathcal{N}_2 : x_2 = [W_{21}x_1 + \theta_2]_+$$

$\mathcal{M}_{(j)}$: Initial conditions for trajectories passing through $H_i/\mathcal{N}_j$ intersections in positive time (Magenta)

$\mathcal{S}_{(i)}$: *Chamber by chamber breakdown of the positively oriented stable manifold* $\mathcal{S}_1/\mathcal{S}_2$ *for the* $R_{[2]}$ *fixed point (Black, Cyan, and Red)*

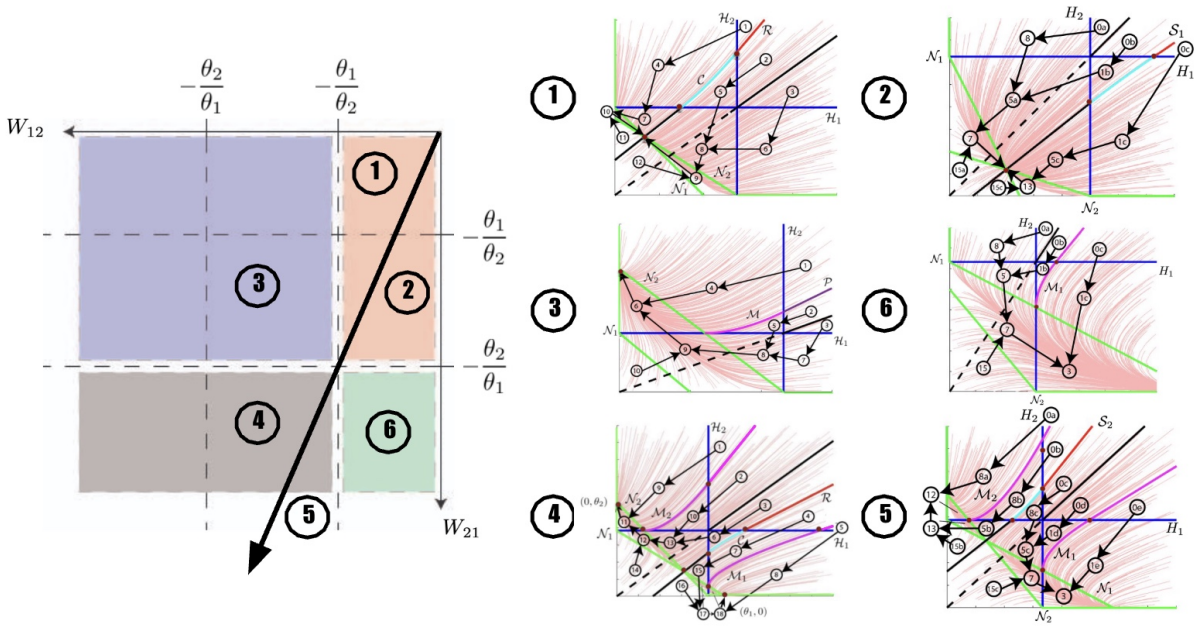


Figure 2.13. Trajectory graph bifurcation. Bifurcation diagram on trajectory graph structure of state space.

Proof. Drawing on Corollary 2, nullcline analysis yields most of the trajectory graph. Where there is an $H_i/\mathcal{N}_j$ intersection we draw in the curve $\mathcal{M}_j$. Additionally, we fill in the Perron-Frobenius stable manifold for Zones 1 and 4 of Figure 2.11 as prescribed by Lemma 5 and Proposition 3. The final piece is the black diagonal line restricted to $R_\emptyset$ which separates trajectories that flow into $R_{\{1\}}$ and $R_{\{2\}}$ respectively.

□

2.2.3 Basins of Attraction

We concern ourselves primarily with the bistable case as it is the only one with distinct basins of attraction.

In the case where $W_{21} > \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$, we have the separatrix:

$$x_1 = \mathcal{S}_1(x_2).$$

In the case of $W_{21} < \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$, we have the separatrix:

$$x_2 = \mathcal{S}_2(x_1).$$

Both are depicted in Figure 2.14.

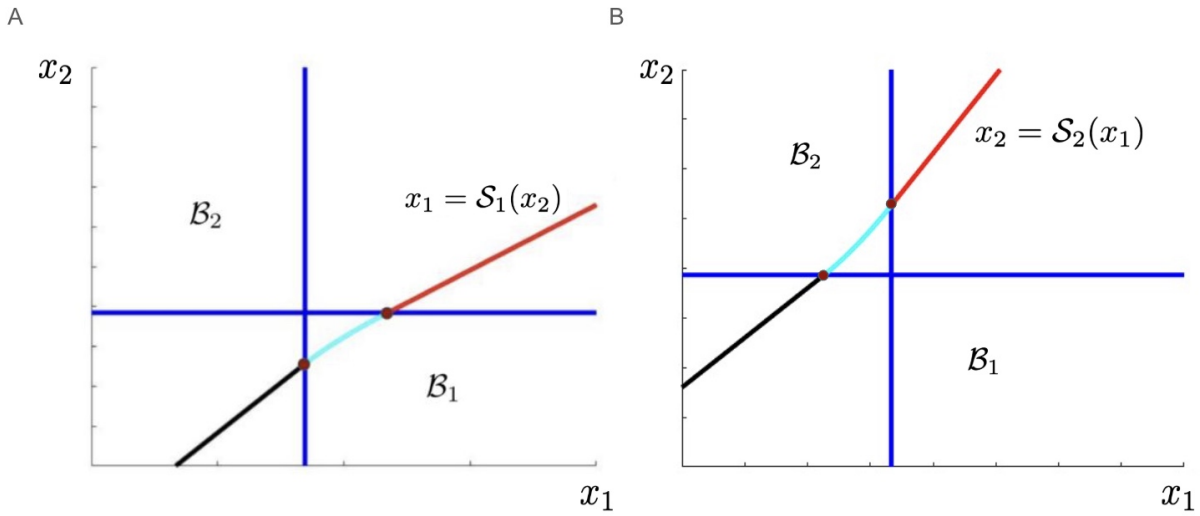


Figure 2.14. Basins of attraction under bistable parameter regime. (A) Separatrix and basins of attraction under Zone 4 parameter regime as indicated in Fig 2.13. (B) Separatrix and basins of attraction under Zone 5 parameter regime.

Corollary 3. *For the Binary Competition Model where without loss of generality* $\theta_2 \geq \theta_1 > 0$*, the basins of attraction are as follows:*

Zones 1 and 2, $W_{21} > -\frac{\theta_2}{\theta_1}$ *and* $W_{12} > -\frac{\theta_1}{\theta_2}$*:* $\mathrm{FP} = \{\{1,2\}\}$*, The basin of attraction for the single fixed point supported on* $\{1,2\}$ *is the entire phase space.*

Zone 3, $W_{21} > -\frac{\theta_2}{\theta_1}$ *and* $W_{12} < -\frac{\theta_1}{\theta_2}$*:* $\mathrm{FP} = \{1\}$*, The basin of attraction for the single fixed point supported on* $\{2\}$ *is the entire phase space.*

Zone 4, $W_{21} > \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$ *and* $W_{21} < -\frac{\theta_2}{\theta_1}$*:* $\mathrm{FP} = \{1,2,\{1,2\}\}$*, For the fixed point supported on* $\{1\}$*, the basin of attraction is:*

$$\mathcal{B}_1 = \{(x_1^0, x_2^0) \mid x_1^0 > \mathcal{S}_1(x_2^0)\}$$

and similarly the basin of attraction for the fixed point supported on $\{2\}$ *is given by:*

$$\mathcal{B}_2 = \{(x_1^0, x_2^0) \mid x_1^0 < \mathcal{S}_1(x_2^0)\}$$

Zone 5, $W_{21} < \left(\frac{\theta_2}{\theta_1}\right)^2 W_{12}$ *and* $W_{12} < -\frac{\theta_1}{\theta_2}$*:* FP $= \{1, 2, \{1, 2\}\}$*, For the fixed point supported on* $\{1\}$*, the basin of attraction is:*

$$\mathcal{B}_1 = \{(x_1^0, x_2^0) \mid x_2^0 < \mathcal{S}_2(x_1^0)\}$$

and similarly the basin of attraction for the fixed point supported on $\{2\}$ *is given by:*

$$\mathcal{B}_2 = \{(x_1^0, x_2^0) \mid x_2^0 > \mathcal{S}_2(x_1^0)\}$$

Zone 6, $W_{21} < -\frac{\theta_2}{\theta_1}$ *and* $W_{12} > -\frac{\theta_1}{\theta_2}$*:* FP $= \{2\}$*, The basin of attraction for the single fixed point supported on* $\{2\}$ *is the entire phase space.*

While these expressions are unwieldy, notice that it is not especially challenging to integrate beneath these seperatrices. The segments in $R_{[2]}$ and $R_\emptyset$ are linear and the segments in $R_{\{1\}}$ $R_{\{2\}}$, of the form $f_1(a, b, x)$ and $f_2(a, b, x)$ respectively, are both easily resolved using integration by parts on the non-linear term. To find the relative sizes of the basins, we restrict the window to $[0, B]$ x $[0, B]$, find the fractional area of the basin as a function of B, and then take the limit $B \to \infty$.

Definition 6. *Let* A *be a measurable set in* $\mathbb{R}_+^N$*. Call* A *an* $\mathcal{F}$***-set*** *if:*

$$\lim_{B\to\infty} \frac{\lambda(A \cap [0, B]^n)}{B^n}$$

exists (where λ *is the standard Lebesgue measure).*

If A *is an* $\mathcal{F}$***-set****, define as its* ***fractional area(volume)*** $\mathcal{F}(A)$*:*

$$\mathcal{F}(A) = \lim_{B\to\infty} \frac{\lambda(A \cap [0, B]^n)}{B^n}.$$

Theorem 3. *For the Binary Competition Model with parameters in Zone 4 of Fig 2.13:*

(a) The fractional area of the basin of attraction $\mathcal{B}_1$ *for the fixed point supported on* $\{1\}$ *is:*

$$\mathcal{F}(\mathcal{B}_1) = 1 + \frac{W_{12}}{2\theta_1} f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21} - m^{-1})(W_{21}\theta_1 + \theta_2)}{W_{21}(1 + |W|)}, -\frac{\theta_1}{W_{12}}\right)$$

(b) The fractional area of the basin of attraction $\mathcal{B}_2$ *for the fixed point supported on* $\{2\}$ *is:*

$$\mathcal{F}(\mathcal{B}_2) = -\frac{W_{12}}{2\theta_1} f_1\left(-\frac{\theta_2}{W_{21}}, \frac{(W_{21} - m^{-1})(W_{21}\theta_1 + \theta_2)}{W_{21}(1 + |W|)}, -\frac{\theta_1}{W_{12}}\right)$$

For parameters in Zone 5 of Fig 2.13:

(a) The fractional area of the basin of attraction $\mathcal{B}_1$ for the fixed point supported on $\{1\}$ *is:*

$$\mathcal{F}(\mathcal{B}_1) = -\frac{W_{21}}{2\theta_2} f_2\left(\frac{(W_{12} - m)(W_{12}\theta_2 + \theta_1)}{W_{12}(1 + |W|)}, -\frac{\theta_1}{W_{12}}, -\frac{\theta_2}{W_{21}}\right)$$

(b) The fractional area of the basin of attraction $\mathcal{B}_2$ for the fixed point supported on $\{2\}$ *is:*

$$\mathcal{F}(\mathcal{B}_2) = 1 + \frac{W_{21}}{2\theta_2} f_2\left(\frac{(W_{12} - m)(W_{12}\theta_2 + \theta_1)}{W_{12}(1 + |W|)}, -\frac{\theta_1}{W_{12}}, -\frac{\theta_2}{W_{21}}\right)$$

where $m = \sqrt{\dfrac{W_{12}}{W_{21}}}$.

Proof. Generally consider a function of the form:

$$\mathcal{S}(x) = \begin{cases} k_1 x & \text{if } 0 \le x \le p \\ f_1(a, b, x) & \text{if } p < x \le q \\ k_2 x & \text{if } q < x \end{cases}$$

Then,

$$\lim_{B\to\infty} \frac{\int_0^B \mathcal{S}(x)dx}{B^2} = \lim_{B\to\infty} \left(\frac{\int_0^p k_1 x dx}{B^2} + \frac{\int_p^q f_1(a, b, x)dx}{B^2} + \frac{\int_q^B k_2 x dx}{B^2} \right)$$

$$= 0 + 0 + \lim_{B\to\infty} \frac{k_2 B^2 - k_2 q^2}{2B^2} = \frac{k_2}{2}$$

.

This would also be true if we used $f_2(a, b, x)$ instead of $f_1(a, b, x)$.

Then, for Zone 4, we apply this for $\mathcal{S}_1(x)$ to find $\mathcal{F}(\mathcal{B}_2)$. To obtain $\mathcal{F}(\mathcal{B}_1)$, notice that:

$$\mathcal{F}(\mathcal{B}_1) = \lim_{B\to\infty} \frac{B^2 - \int_0^B \mathcal{S}_1(x)dx}{B^2} = 1 - \mathcal{F}(\mathcal{B}_2).$$

We apply this same approach to $\mathcal{S}_2(x)$ to obtain the result for Zone 5.

□

2.2.4 Decision-Making Bias in the Binary Competition Model

We return now to our three paradigms of how decision-making bias may be encoded in the basins of attraction. In Fig 2.15, we demonstrate each of these for this two-dimensional TLN model, assuming that we are in the bistable regime in parameter space.

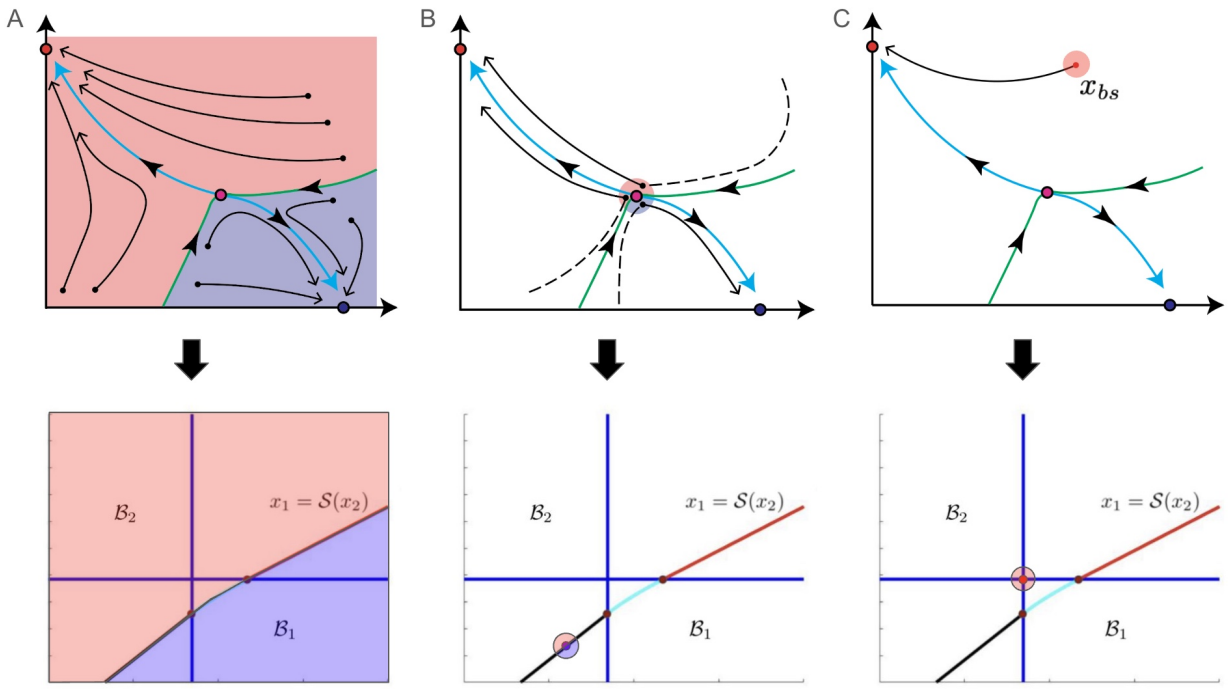


Figure 2.15. Encoding of decision-making bias in the Binary Competition Model. (A) Under the assumption that the full basins of attraction are of relevance we take the fractional area of the basins area. (B) Under the paradigm that the basins in the vicinity of the saddle point matter most, we restrict to the intersection between the basins and a small unit disk around the saddle point, which is problematic as the separatrix is locally linear near the saddle point. (C) Focusing on the balanced state trajectory, the basin the balanced state lies in is controlled strictly by whether the separatrix exits $R_{[2]}$ through H_1 or through H_2.

As depicted in Fig 2.15A, using Theorem 3 lets us understand bias under the hypothesis of high dimensional dynamics where we want to compare the relative sizes of the basins of attraction. While the expressions of the theorem are unwieldy they do precisely quantify this relationship. However, if we were merely concerned with the balanced state trajectory, we need only use Lemma 5 which indicates on which side of the seperatrix the balanced state lies, as depicted in Fig 2.15C.

Where problems majorly arise is in the case where we emphasize trajectories along

the decision-boundary, sampling initial conditions near the saddle point. As calculated, the seperatrix is locally linear near the saddle point, so will separate any small disc drawn around it in half as depicted in Fig 2.15B. This means that no matter how we modulate the parameters, the relative sizes of the localized basins would always be equivalent.

Another problem with this two-dimensional model is that it is highly simplified and obscures the larger connectivity structure of the network. While we could simulate a phenomenon like the Decoy Effect by setting $W_{12} = W_{21}$ and making say $\theta_2 > \theta_1$, the nuances of a more complex network structure may be destroyed by such a reduction. What this all makes clear is that this model cannot be the end of the story and that analyzing higher dimensional systems will often be necessary to properly understand bias in decision-making circuits.

Chapter 3
Challenges of Combinatorial Dynamics in Higher Dimensions

It goes without saying that an analytical approach to working out combinatorial dynamics across higher dimensional TLNs is not tractable. However, might there be a computer assisted approach that is viable? Even if we are not able to get a precise trajectory graph, we might still get a state transition graph with gives us rough lower bounds on the sizes of basins. One challenge of a computer assisted approach is that of attractors. While in the case of the Binary Competition Model all the attractors were fixed points, this is not true of TLNs in general and it is possible to have dynamic attractors such as limit cycles in higher dimensions [29]. This means a chamber by chamber linear system analysis is simply not enough, even with computer assistance, as dynamic attractors will span various chambers. How can we talk about basins of attraction if we do not understand what the attractors are?

One approach to this problem is in the application of Conley Index Theory. We briefly review the key ideas of applied Conley Index Theory with the exposition being drawn primarily from a review by Konstantin Mischaikow [30].

We will describe how it has been applied to state transition graphs to detect and classify attractors, but also show the challenges that this approach has when dealing with TLNs. The primary novelty of this chapter is the development of an algorithm which exploits the piecewise linear dynamics of TLNs to computationally determine a state transition graph without requiring brute force methods of simulation to determine edge directions.

3.1 Introduction to Conley Index Theory

Conley Index Theory can be thought of as a coarse graining of traditional dynamical systems theory. Instead of looking at invariant sets, such as attractors, directly, the objects of focus are instead the *isolating neighborhoods* of invariant sets.

Definition 7. *Let $\phi(t,x)$ be a flow in $\mathbb{R}^n$. A compact set $N \subset \mathbb{R}^n$ is an **isolating neighborhood** if:*

$$\mathrm{Inv}(N,\phi) := \{x \in N \mid \phi(\mathbb{R},x) \subset N\} \subset \mathrm{int}\, N.$$

Studying attractors directly is mathematically precise whereas studying isolating neighborhoods has more freedom and can be easier. To be particular, isolating neighborhoods allow us to study *isolated invariant sets* i.e. invariant sets S such that $S = \mathrm{Inv}\, N$ for some isolating neighborhood N.

The next construction to introduce is that of the *index pair* and the *exit set* L of an isolating neighborhood.

Definition 8. *Let S be an isolated invariant set. A pair of compact sets (N,L) with $L \subset N$ is an **index pair** for S if:*

1. $S = \mathrm{Inv}(\mathrm{cl}(N \backslash L))$ and $N \backslash L$ is a neighborhood of S.

2. Given $x \in L$ and $\phi([0,t],x) \subset N$, then $\phi([0,t],x) \subset L$.

3. L is an exit set for N i.e. given $x \in N$ and $t_0 > 0$ such that $\phi(t_0,x) \notin N$, then there exists $0 \leq t_1 < t_0$ such that $\phi(t_1,x) \in L$.

Essentially, these conditions mandate that L be an "outer" component of the isolating neighborhood such that it does not include any of the underlying invariant set, trajectories beginning in L do not enter $N \backslash L$, and that any trajectory leaving N must go through L (Fig 3.1A). The Conley index is then a topological invariant assigned to the pointed topological space obtained by collapsing L to a point. For example the *homotopy Conley index* of an invariant set S with index pair (N,L) is the homotopy type:

$$h(S) \sim (N/L, [L]).$$

We can use the Betti numbers, β_i, of the homology groups as a topological signature of the Conley index.

$$\beta_\bullet = \mathrm{rank}(H_\bullet(N/L,[L])).$$

The Conley Index has three properties that make it a useful tool for studying dynamical systems:

- If N and N' are isolating neighborhoods such that $\text{Inv}(N) = \text{Inv}(N')$, then Conley Index$(N)$ = Conley Index(N').
- If Conley Index(N) is not trivial, then $\text{Inv}(N) \neq \emptyset$.
- If there are a smooth family of flows $\phi^\lambda(t, x)$ with $\lambda \in [0, 1]$. such that N is an isolating neighborhood $\forall \phi^\lambda$. Then, the Conley Index for $S_\lambda = \text{Inv}(N, \phi^\lambda)$ is independent of λ.

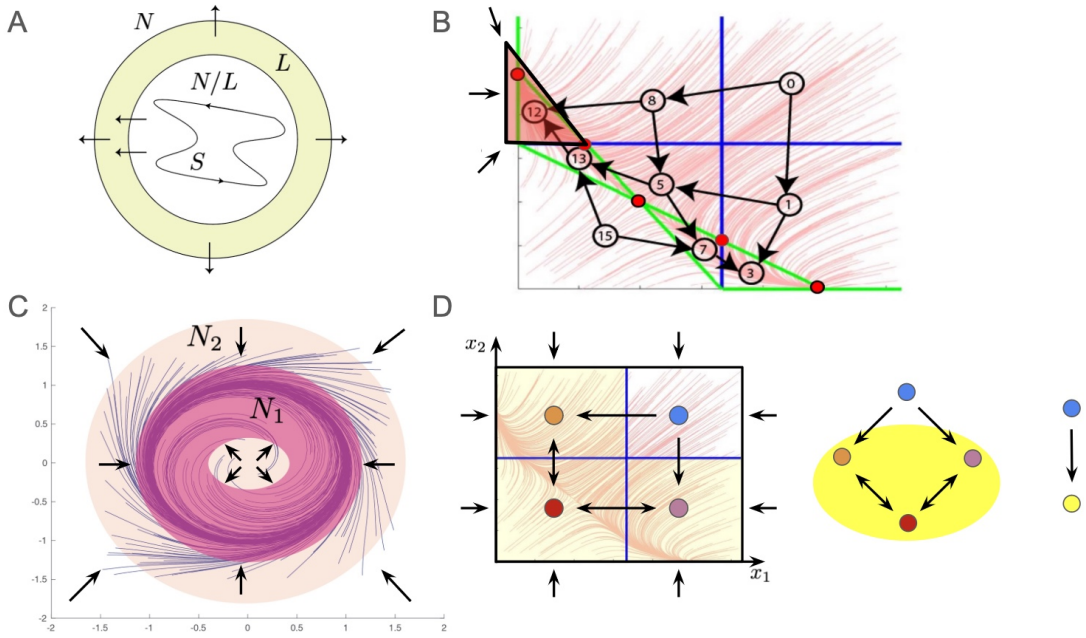


Figure 3.1. Conley Index Theory and TLNs. (A) Schematic showing an invariant set S, an isolating neighborhood N with a corresponding exit set L. (B) There exist arbitrarily tight compact neighborhoods of chamber 12 which are isolating neighborhoods for the point attractor inside. (C) A limit cycle with two choices of isolating neighborhood N_1 and N_2. Even though they are both isolating neighborhoods enclosing the limit cycle, they yield different Conley indices. (D) An example of the state transition graph analysis algorithm being applied to the ReLU hyperplane partition. Three chambers are collapsed into one strongly connected component which does not separate the two point attractors.

This allows us to study the attractors indirectly. From state transition graphs, we can identify isolating neighborhoods and compute their Conley indices to classify the

attractors. The state transition graph, even if not a full trajectory graph, would then hopefully give us some rough sense of the basins of attraction.

As an exercise for understanding, let us briefly consider the two-dimensional case. In Fig 3.1B we have highlighted that arbitrarily tight neighborhoods of chamber 12 are an isolating neighborhood for a point attractor in the independent set CTLN. Call it N. No trajectories exit this chamber so we have the exit set $L = \emptyset$. So, the pointed topological space $(N/L, [L])$ is simply N itself. As N is a contractible set, we have the Betti numbers $\beta_0 = 1, \beta_1 = 0$. Thus, the Conley index would be $(1, 0, 0)$. Alternatively, if we had an isolating neighborhood N_1 as given in Fig 3.1C, the Conley Index would be $(1, 1, 0)$ which characterizes the limit cycle inside. This is how the Conley index can be used to detect and categorize attractors.

Once we have a state transition graph, we can collapse strongly connected components to a single vertex, taking the union of the chambers, to allow for attractors that span multiple chambers. Then we find the sinks of this condensed graph which indicates an attracting set lying within. Finally, by calculating the Conley Index of that isolating neighborhood, we can try and determine the kind of attractor that lurks inside. This technique has been used successfully in studying dynamics in systems biology [31].

Note that in TLNs fixed points often lie on chamber boundaries in which case these chambers are not isolating neighborhoods in the strictest sense. However, as discussed, an arbitrarily tight neighborhood of the chamber would be an isolating neighborhood, and so, with this understanding, we informally treat these chambers for now as if they are as well. A technique of dealing with this issue is that the neighborhood around these fixed points can be "blown-up" into their own chambers [31]. Regardless, the first step is the development of a state transition graph.

The two dimensional case also shows us the limitations of Conley Index Theory. Consider the second isolating neighborhood N_2 in Fig 3.1C. The set N_2 is still an isolating neighborhood where the only attractor inside is the limit cycle, but will have Conley index $(1, 0, 0)$. How can this be as the Conley Index is meant to be invariant of the choice of isolating neighborhood? Notice the subtlety that the Conley index is defined in terms of invariant sets, not attractors. For N_2, the set $\text{Inv}(N_2)$ also includes everything inside of the limit cycle, hence the difference in Conley Index.

What this indicates is that the isolating neighborhood does need to be sufficiently tight to the attractor to properly categorize it. A unit ball isolating neighborhood could have a point attractor or even multiple chaotic attractors inside and would still yield the same Conley index as long as no trajectories exited the ball.

The key is then of course is to find a partition scheme that gives us a useable state transition graph. A problem that can arise is that too coarse a partition will result in too many bidirectional edges in the graph. Enough bidirectional edges will yield a strongly connected component which will then get collapsed together into a fairly loose isolating neighborhood for the attractors. Consider the state transition graph which would have arisen in the bistable case of the Binary Choice Model if we had not included the nullclines (Fig 3.1D) and see how such such a graph structure would tell us very little about the attractors or the basins of attraction. A key technical point is that our analysis will only be applied to competitive TLNs as the property $W \leq 0$ and $\vec{\theta} > 0$ confines activity in the positive orthant to a bubble within the state space, preventing solutions from blowing up to infinity. That lets us produce our state transition graph on a finite collection of compact sets (refer to Fig 3.1D).

3.2 Building a State Transition Graph

Before delving into partition schemes we will first discuss the construction of the state transition graph for a given partition.

A very naive way to approach this would be to sample initial conditions randomly from partition chambers and track their trajectories into other chambers, drawing edges accordingly. This is not only inefficient computationally, but it is also not particularly rigorous as we have to hope that our random sampling was large enough to properly approximate the dynamics within the chamber. Fortunately the linearity of the component dynamical systems of a TLN presents with an alternative approach as long as a partition separates the linear system regions R_σ and the chambers are convex polytopes generated by a hyperplane arrangement.

As a TLN is a continuous dynamical system, if on a chamber face there is a area where the vector field points inward and an area where it points outward, they are separated by a boundary where the vector field lies within the corresponding hyperplane of the face.

Proposition 4. *Let $\vec{b} \in \mathbb{R}^n$. Then the points such that the vector field given by TLN linear ODE system L_σ are orthogonal to $\vec{b}$ is $R_\sigma \cap B_\sigma$ where:*

$$B_\sigma := \sum_{k=1}^{n} \left(-b_k + \sum_{j \in \sigma} b_j W_{jk} \right) x_k + \sum_{j \in \sigma} b_j \theta_j = 0$$

Proof. We seek $\vec{x} \in R_\sigma$ such that $\vec{b} \cdot \frac{d\vec{x}}{dt}|_{R_\sigma} = 0$.

$$\vec{b} \cdot \frac{d\vec{x}}{dt} = -\sum_{j \notin \sigma} b_j x_j + \sum_{j \in \sigma} b_j \dot{x}_j \tag{3.1}$$

$$= -\sum_{j \notin \sigma} b_j x_j + \sum_{j \in \sigma} b_j (-x_j + \sum_{k=1}^{n} W_{jk} x_k + \theta_j) \tag{3.2}$$

$$= \sum_{k=1}^{n} \left(-b_k + \sum_{j \in \sigma} b_j W_{jk} \right) x_k + \sum_{j \in \sigma} b_j \theta_j \tag{3.3}$$

□

Let a chamber, call it K^μ, be governed by L_σ and bounded by the hyperplanes $\{K_i\}_{i=1}^m$ where $K_i : k_{i0} + \sum_{j=1}^n k_{ij} x_j = 0$. Then, the chamber is the intersection of half spaces associated with the hyperplanes and its closure can be expressed as the feasible region of the linear program:

$$\left[\begin{array}{ccc} k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{p1} & \cdots & k_{pn} \\ \hline -k_{p+1,1} & \cdots & -k_{p+1,n} \\ \vdots & \ddots & \vdots \\ -k_{m1} & \cdots & -k_{mn} \end{array}\right] \vec{x} \geq \left[\begin{array}{c} k_{10} \\ \vdots \\ k_{p0} \\ \hline -k_{p+1,0} \\ \vdots \\ -k_{m0} \end{array}\right]$$

where $\vec{x} \geq 0$.

To assign an edge through a chamber wall associated with K_i, where K^{μ_i} is the adjacent chamber, we simply turn the associated constraint into an equality to restrict to the face. We additionally apply Proposition 4 to the normal vector of K_i, to also add the constraint:

$$(K_i)_\sigma := \sum_{\ell=1}^{n} \left(-k_{i\ell} + \sum_{j \in \sigma} k_{ij} W_{j\ell} \right) x_\ell + \sum_{j \in \sigma} k_{ij} \theta_j > 0.$$

We see if this linear program has a solution and if so, we have an edge in one direction. Then we can see if there is a solution when the constraint is changed to:

$$\sum_{\ell=1}^{n} \left(-k_{i\ell} + \sum_{j \in \sigma} k_{ij} W_{j\ell} \right) x_\ell + \sum_{j \in \sigma} k_{ij} \theta_j < 0.$$

This would check if there is a directed edge in the other direction. If both linear

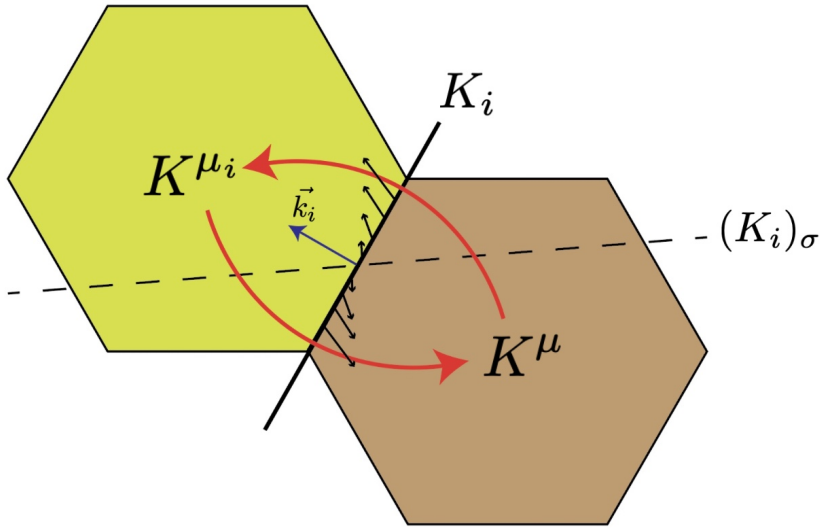


Figure 3.2. State transition graph construction. This diagram depicts the approach towards labeling directed edges of the state transition graph. The hyperplane $(K_i)_\sigma$ divides the hyperplane K_i into regions of inward and outward flow. In the diagram here outward flows exist on both sides of this chamber wall so edges are drawn in both directions.

programs have a solution, then there is a bidirectional edge over that wall of the chamber. The bidirectional case is depicted in Fig 3.2. Now, which edge is the one pointing out of the chamber and which is the one pointing in depends on the sign of the normal vector. Without loss of generality, call the linear program corresponding to the outward pointing edge $K^{\mu}_{i_+}$. Using this approach for each wall of each chamber, we can build a state transition graph for the hyperplane arrangement generated by $\{K_i\}_{i=1}^{m}$. We start by taking an undirected graph showing adjacency between chambers and then iterate through the adjacent pairs of chambers to determine the directed edges of the state transition graph.

Algorithm 1 State Transition Graph

$V(G_1) = \{\mu \mid K^\mu$ is a chamber of the partition$\}$
$V(G_2) = \{\mu \mid K^\mu$ is a chamber of the partition$\}$
$E(G_1) = \{(\mu, \mu') \mid K^\mu, K^{\mu'}$ share a face$\}$
for $\mu \in V(G_1)$ **do**
  for $\mu' \in V(G_1)$ **do**
    if $(\mu, \mu_i) \in E(G_1)$ **then**
      $K_i = \mu \cap \mu'$
      $\mu_i = \mu'$
      if $K^\mu_{i_+}$ has a solution **then**
        $(\mu, \mu') \in E(G_2)$
      end if
    end if
  end for
end for
return G_2

3.3 Devising a Partition

A natural first choice of partition would be our standard $H_i/\mathcal{N}_i$ partition. An implementation of Algorithm 1for this partition is used for the remainder of this chapter.. Let us begin with our two dimensional case, particularly the independent set CTLN. We first show an undirected graph of the adjacent chambers. For ease of comparison we have set aside our "canonical" labeling scheme and use the notation $+/-$ for each hyperplane in the arrangement. The $+$ symbol is used if the chamber is on the side of the hyperplane containing the origin and the $-$ symbol is used if the chamber is on the other side of the hyperplane. This label will be written in the form $\mathcal{N}_1 \dots \mathcal{N}_n, H_1 \dots H_n$. As an example the chamber lying inside the second and third nullcline and within all three of the ReLU hyperplanes will be labelled $-++, +++$. Using our algorithm, we compute the state transition graph and then collapse the simply connected components.

Beginning with the two-dimensional independent set, we recover our analytical results without much issue (Fig 3.3). However when we try to move to higher dimensional cases, we find things both intriguing and problematic (Fig 3.4 and Fig 3.5). The intriguing aspect is the sheer diversity of state transition graph structures that we find.

We analytically know that the decoy effect DAG CTLN in Fig 3.4 should have multiple

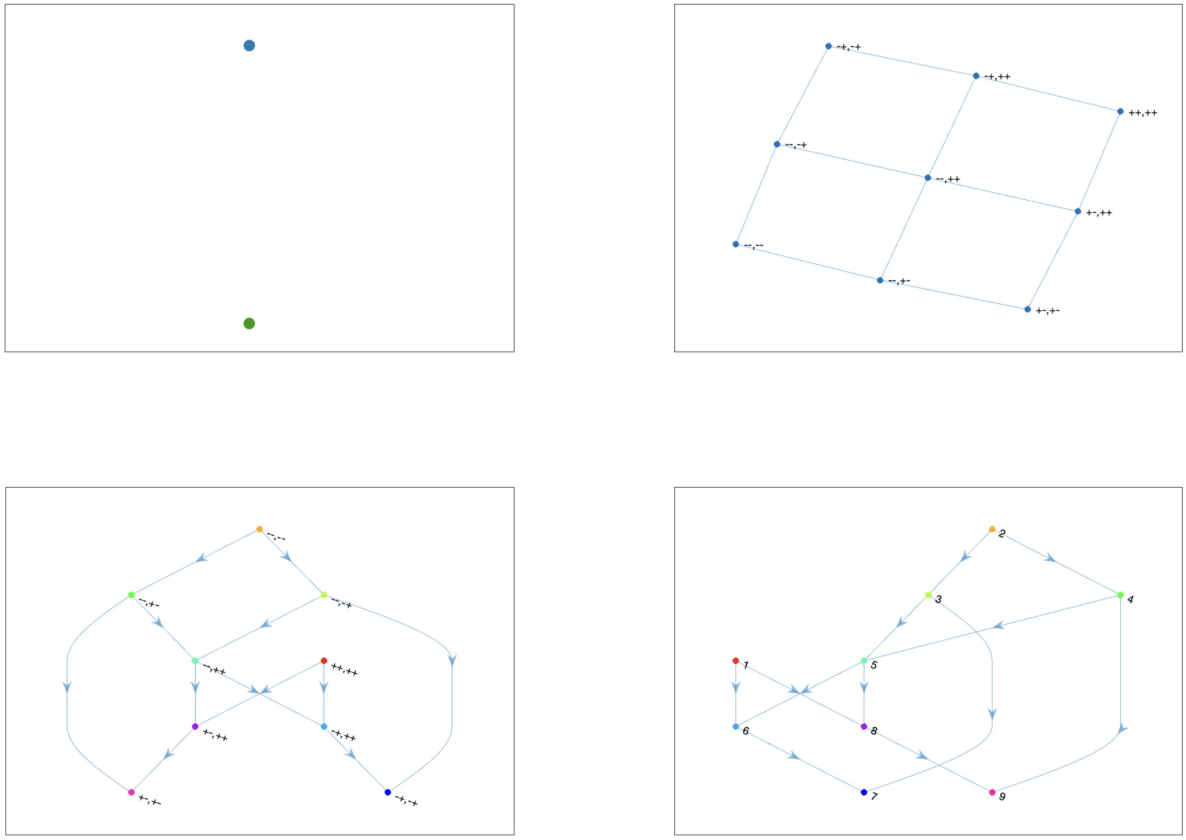


Figure 3.3. State transition graph of independent set CTLN. Upper left image depicts the graph from which the CTLN is derived with the neurons numbered clockwise starting from the top. The upper right shows the chambers of the $H_i/\mathcal{N}_i$ partition with the undirected edges representing a shared face of the polytope chambers. The lower left gives the state transition graph color coded to strongly connected components. The lower right is the condensed graph of strongly connected components.

attractors as it is a DAG with multiple sinks, but only a single sink is produced in the condensed state transition graph. This partition is simply inadequate to distinguish attractors let alone classify them.

Looking at more examples, we find that while the state transition graphs vary considerably and display a rich diversity of structure, they nonetheless only seem to produce one attracting strongly connected component. Looking at the chambers which compose that sink node in the condensed graph, they are those which are in between nullclines, i.e. which are on the $+$ side of some nullclines and the $-$ side of others. By and large it appears that these chambers have enough bidirectional edges between that they are collapsed into one strongly connected component which is the single attracting

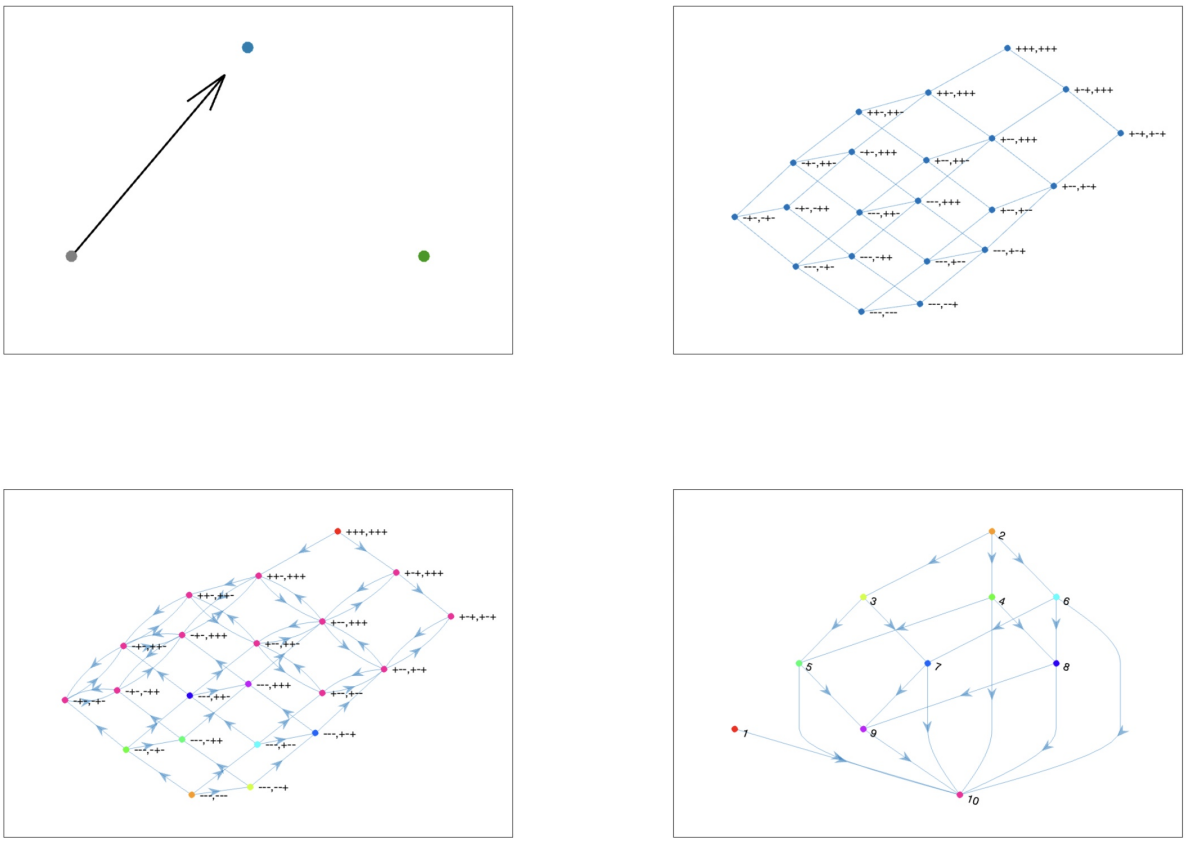


Figure 3.4. State transition graph of decoy effect CTLN.

isolating neighborhood in larger CTLNs. The various attractors of the CTLN seem to lurk within this union and collectively are part of the invariant set for which this union of chambers is an isolating neighborhood. This is aligned with prior theoretical results on competitive TLNs which indicate that the union of mixed sign nullcline chambers is attracting (Theorem 9.1 in [15]). Unfortunately, it seems that even this computational approach often doesn't tell us much more about the attractors of the CTLN than that. Nonetheless, the state transition graphs do capture a kind of combinatorial dynamics and help us to understand how the trajectories beginning outside this attracting set approach it.

While interesting, if our state transition graph is not able to separate out the attractors, it definitely does not give us information about the basins of attraction. If we want a combinatorial dynamics that guides us to better understanding the basins, we would need something different.

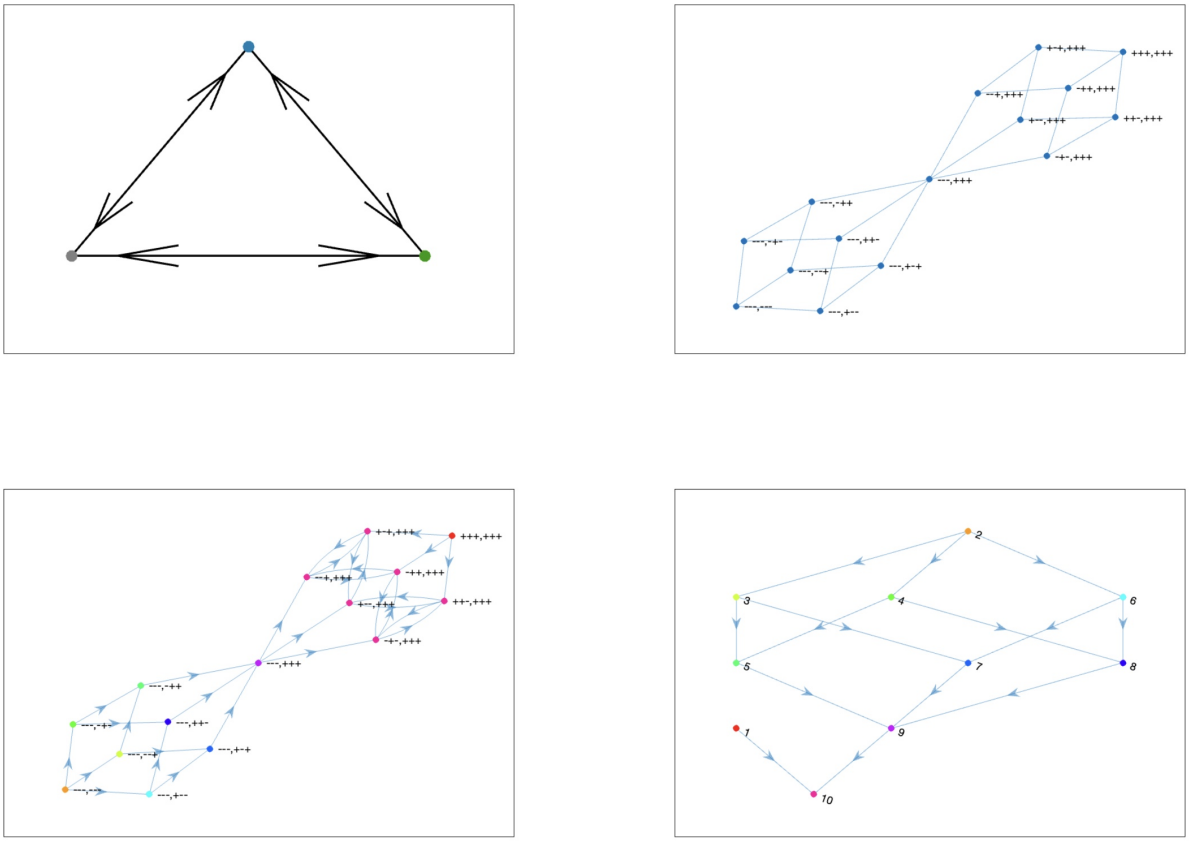


Figure 3.5. State transition graph of clique CTLN.

We necessarily need separation into the linear systems. This means we begin our partition with the hyperplanes $\{H_i\}_{i=1}^n$. However, as we saw, this is not refined enough even in the two neuron case. What we need to do is introduce a new set of hyperplanes that that reduce the bidirectional edges over the R_σ chamber boundaries. What we can do is add the hyperplanes which separate the inward and outward flows through the chamber walls H_i

Corollary 4. *The points in R_σ such that the vector field given by the linear ODE system L_σ is orthogonal to the normal vector of H_i is $R_\sigma \cap B_i^\sigma$ where:*

$$B_i^\sigma := \sum_{k=1}^{n} \left(-W_{ik} + \sum_{j \in \sigma} W_{ij} W_{jk} \right) x_k + \sum_{j \in \sigma} W_{ij} \theta_j = 0$$

So now, our hyperplane arrangement consists of $\{H_i\}_{i=1}^n$ and also for each $H_i \cap R_\sigma$ chamber wall with a bidirectional edge, we also include B_i^σ to separate the wall into

a half where the the vector field is "inward" facing and a half where the vector field is "outward" facing. While all the bidirectional edges with respect to the H_i hyperplanes have been resolved, each B_i^σ is a new chamber wall which can have a bidirectional edge it its own right. Of course we can inductively apply the same process to the new hyperplane, terminating when we no longer have bidirectional edges. But a problem arises. At a fixed point x_σ, the vector field is zero, and so is always orthogonal to any normal vector. Therefore, chambers with fixed points will continue to be endlessly partitioned and the process will never terminate.

There remains one hope. The hyperplane B_i^σ is unimportant in and of itself, but rather it is the intersection $B_i^\sigma \cap H_i$ and $\text{codim}(B_i^\sigma \cap H_i)$ is generally 2. This leaves one dimension of freedom which allows us to draw different hyperplanes which have the same intersection with H_i.

Proposition 5. *Let B be the hyperplane given by $b_0 + \sum_{k=1}^n b_k x_k = 0$. Then $B \cap B_\sigma = B \cap B_\sigma^*$ where:*

$$B_\sigma^* = \sum_{k=1}^{n} \left(\sum_{j \in \sigma} b_j W_{jk} \right) x_k + b_0 + \sum_{j \in \sigma} b_j \theta_j = 0$$

and

$$B_\sigma := \sum_{k=1}^{n} \left(-b_k + \sum_{j \in \sigma} b_j W_{jk} \right) x_k + \sum_{j \in \sigma} b_j \theta_j = 0$$

Proof. On B, we have the identity $b_0 + \sum_{k=1}^n b_k x_k = 0$. We use this to replace the expression $-\sum_{k=1}^n b_k x_k$ in B_σ with b_0. This yields the new B_σ^*. □

Corollary 5. *For a TLN chamber R_σ, let $H_i \cap R_\sigma$ be one of the chamber walls. Then, $(B_i^\sigma)^*$ is a hyperplane such that $H_i \cap (B_i^\sigma)^*$ separates the inward and outward flows of $H_i \cap R_\sigma$.*

$$(B_i^\sigma)^* := \sum_{k=1}^{n} \left(\sum_{j \in \sigma} W_{ij} W_{jk} \right) x_k + \theta_i + \sum_{j \in \sigma} W_{ij} \theta_j = 0$$

Proposition 5 gives a new set of hyperplanes which can be used instead of B_i^σ. This yields a different partition and potentially a different state transition graph. However, we find that chambers with saddle points still seem to require continual partitioning. Again, this is not terribly surprising upon some thought because as long as both the stable and unstable manifold of the saddle point are intersecting a hyperplane, then there would be a bidirectional edge through it.

While an unsatisfying conclusion, this is ultimately where this analysis stands right now. Hopefully there exists some way to exploit this additional dimension of freedom to obtain a hyperplane arrangement which aligns with either the stable or unstable manifold within a chamber and produces a viable state transition graph, but it remains out of reach at this time.

To summarize this chapter, we have explored computer assisted ways of extending our combinatorial dynamics into higher dimensions, but what we have primarily found are their limitations. Our takeaway is that approximating full basins of attraction in higher dimensional TLNs through combinatorial dynamics is a challenging task and now we consider other approaches.

Chapter 4
Localized Path Polynomials and the Properties of DAG CTLNs

While the attempts to extend a combinatorial dynamics of TLNs into higher dimensional proved unsuccessful, we did see in the introduction that the attractors of DAG CTLNs can be controlled by fixing the number of sinks. The following two theorems establish this concretely:

Theorem 4 (Theorem 10.2 in [15])**.** *A CTLN derived from a directed acyclic graph G will have no dynamic attractors.*

Theorem 5 (Rule 7 in [12])**.** *The set of fixed points of a CTLN derived from a directed acyclic graph G will be supported on sinks and the unions of sinks.*

$$\mathrm{FP}(G) = \{\bigcup s_i \mid s_i \text{ is a sink in } G\}$$

Moreover, each stable fixed point will be supported on exactly one of the sinks.

Once the set of sinks is fixed, so are the attractors, with one for each sink. The unstable saddle points supported on the unions of sinks are associated with the seperatrices which serve as the boundaries between the basins of attraction (Fig 4.1A-B). Once the sinks are fixed, altering the rest of the network allows us to shape the basins of attraction (Fig 4.1).

These basins can be highly complex and non-trivial as shown in the numerical simulation in Fig 4.2. in Fig 4.2A, it is clear that the choice of x_3^0 and x_4^0 are deeply connected to the likelihood of being in either basin. In addition, even though there is only a single path from the source to each of the sinks, the longer path seems to have a greater biasing effect. So it seems that the length of the paths from source to sink may

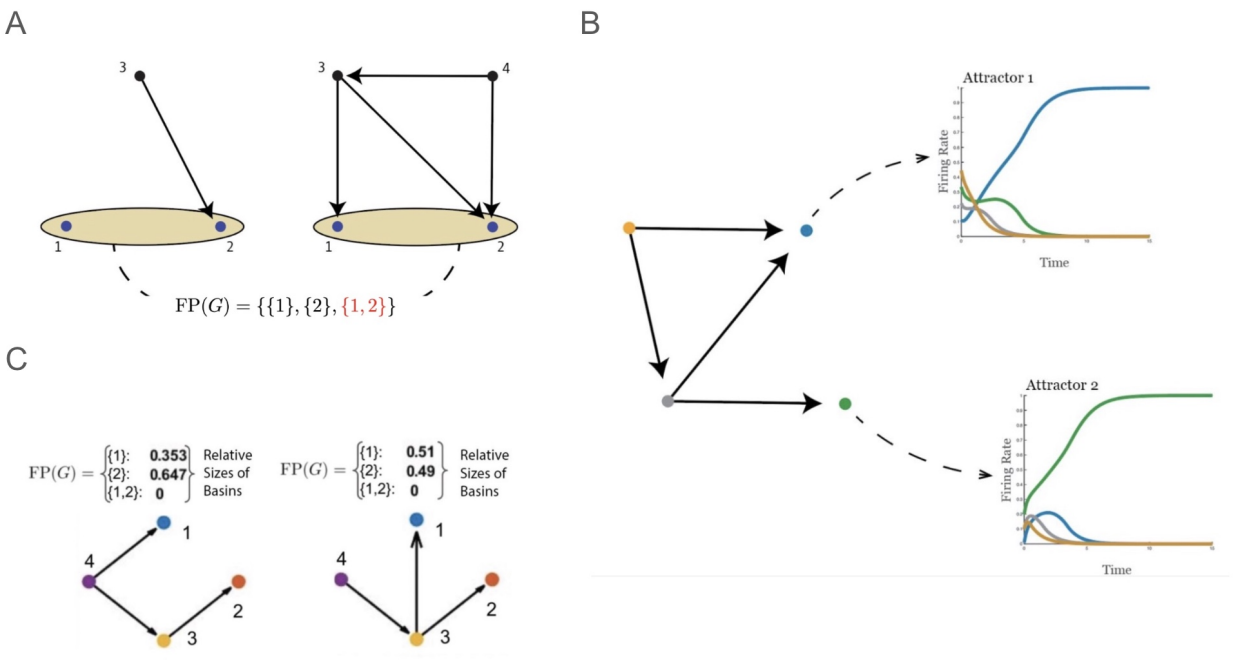


Figure 4.1. Attractors and basins of attraction in DAG CTLNs. (A) Directed acyclic graphs have fixed points supported only on sinks and the unions of sinks. Those which are the unions of sinks will be unstable (B) Only the fixed points supported on the sinks themselves yield attractors, one for each sink. (C) As the sinks control the attractors of DAG CTLNs, the only role of the rest of the graph is in shaping the basins of attraction. The relative sizes of the basins of attraction were found through a Monte Carlo approach by numerically simulating a random set of initial conditions and tracking their trajectories, seeing what fraction converge to each attractor.

have some role in the skewing of basins. The directed graph in Fig 4.2B modifies the previous DAG architecture by having node 4 connect directly to the sink 2 rather than indirectly via 3. A quick look at the relative areas shows that this seems to result in even greater shifting of the basins. It seems likely from this that the basins of attraction for DAG CTLNs factor in the full extent of the network and understanding them will entail unraveling its global dynamics.

Is there any way that we can take advantage of the structure of DAGs to reveal more about the dynamics of these systems? What we will show in this chapter is that we can use the combinatorial structure of the DAG to analytically find the solutions of the linear systems, L_σ, composing the CTLN. The key constructions linking them are *localized path polynomials*.

Definition 9. *For a DAG G of size n, let the vertices be numbered from 1 to n i.e.*

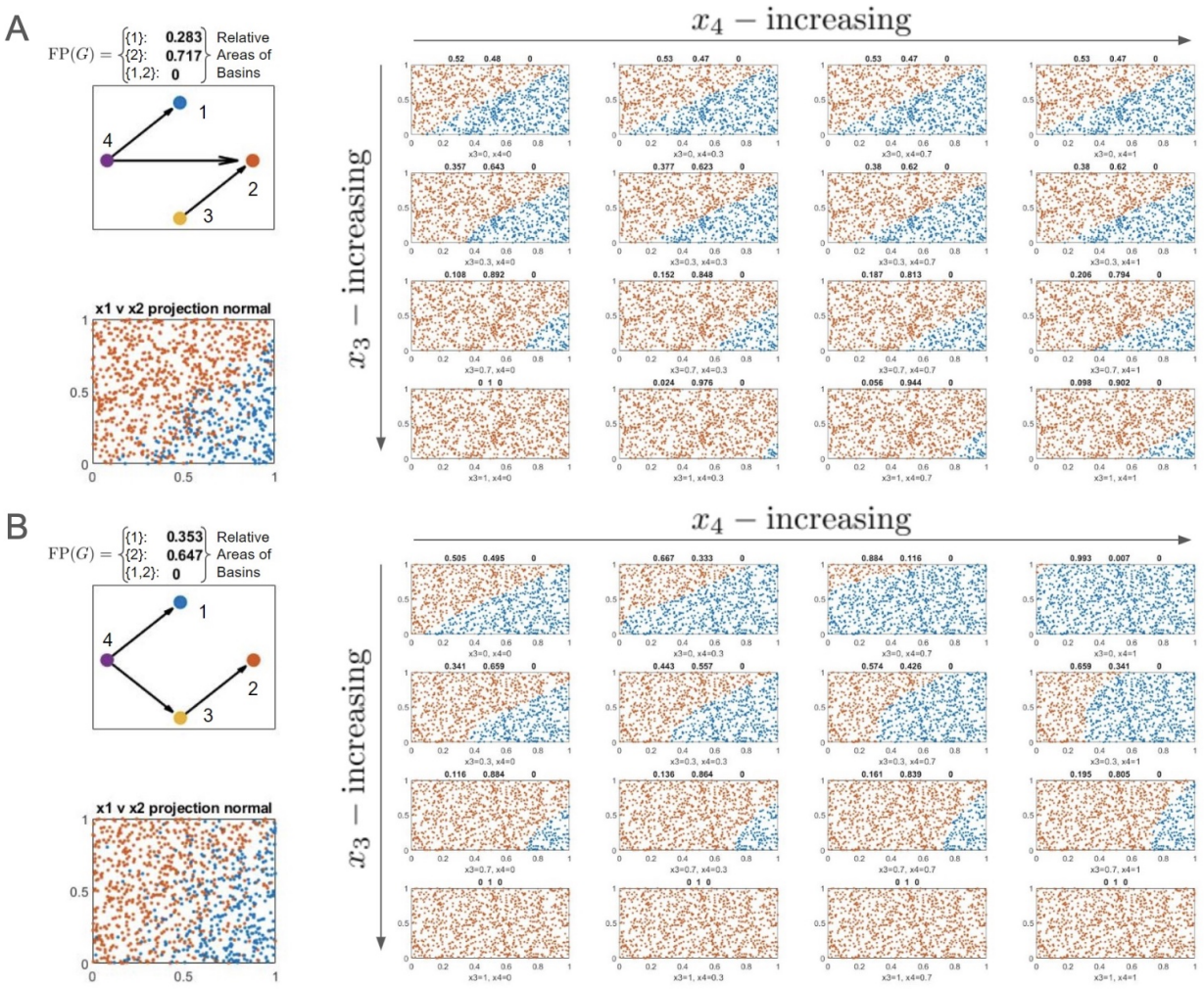
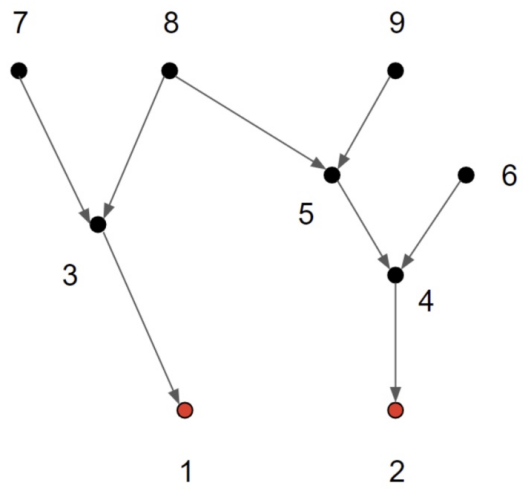


Figure 4.2. Simulation of basins of attraction in DAG CTLNs. A Monte Carlo simulation of the basins of attraction for two DAG CTLNs. As before, we randomly sample initial conditions for a CTLN and we color code that point in state space according to the sink attractor to which it converges. Each slice is a cross section of the state space, and so a cross section of the basins, which fix x_3^0 and x_4^0.

$V(G) = [n]$. *Then, the* ***i-th localized path polynomial****,* $p_i^G(z)$*, is defined to be:*

$$p_i^G(z) = 1 + \sum_{k=1}^{n} n_k^i z^k$$

where n_k^i *is the number of paths to* i *of length* k *(finite because* G *is acyclic). Since* G *is acyclic,* $deg(p_i^G(z))$ *is finite.*



$$p_1^G(z) = 1 + z + 2z^2$$
$$p_2^G(z) = 1 + z + 2z^2 + 2z^3$$
$$p_3^G(z) = 1 + 2z$$
$$p_4^G(z) = 1 + 2z + 2z^2$$
$$p_5^G(z) = 1 + 2z$$
$$p_6^G(z) = 1$$
$$p_7^G(z) = 1$$
$$p_8^G(z) = 1$$
$$p_9^G(z) = 1$$

Figure 4.3. Localized path polynomials. An example of a DAG CTLN with its associated set of localized path polynomials.

Remark 4. *The localized path polynomial construction is only of finite degree for DAGs as the presence of a cycle would give the vertices paths of arbitrarily large length.*

Consider Fig 4.3 where we have depicted a DAG and its associated set of localized path polynomials, one for each vertex.

Definition 10. *Let G be a DAG such that $|G| = n$ and let $\sigma \subseteq [n]$. Define the* ***σ-path function*** *$\vec{p_\sigma} : \mathbb{R} \to \mathbb{R}^n$ to be:*

$$(\vec{p_\sigma}(z))_i = p_i^{G|_\sigma}(z) \ \forall i \in \sigma \text{ and } (\vec{p_\sigma}(z))_i = 0 \ \forall i \notin \sigma.$$

Many of the key properties of the linear systems L_σ comprising a DAG CTLN can be expressed in terms of localized path polynomials. To show this, we first transform the matrices $(-I+W)|_\sigma$ into a more workable form. Taking the function $g(x) = -\frac{1}{1+\delta}(x-\delta)$, we have that $B_\sigma = g((-I+W)|_\sigma) = \mathbb{1}\mathbb{1}^T - \frac{\varepsilon+\delta}{1+\delta}A|_\sigma$ where $A|_\sigma$ is the adjacency matrix of the subgraph $G|_\sigma$. By the Spectral Mapping Theorem, the spectrum of $(-I+W)|_\sigma$ and B_σ, denoted $\rho((-I+W)|_\sigma)$ and $\rho(B_\sigma)$ respectively, are related in the same way:

$$\rho(B_\sigma) = g(\rho((-I+W)|_\sigma)).$$

Additionally, it is not difficult to see that they will share the same eigenvectors. The insight that undergirds many of the results in this chapter is that through this simple

transformation, the CTLN matrices $(-I+W)|_\sigma$ can be studied as rank one updates to scaled adjacency matrices. Recall that the general solution of a non-degenerate, diagonalizable linear system of equations is written in terms of its eigenvectors, eigenvalues, and its fixed point in the following way:

$$\vec{x}(t) = \sum_{i=1}^{n} c_i \vec{v}_i e^{\lambda_i t} + x^*$$

where $\vec{v_i}$ and λ_i are the eigenvectors and eigenvalues of the associated matrix with x^* being the fixed point of the linear system.

The theory developed in this chapter will allow us to find the general solutions of the linear systems L_σ when the subgraph $G|_\sigma$ is an *analytic DAG*.

Definition 11.

$$\mathrm{SE}(G) := \{(i,j) \in V(G) \times V(G) \mid i \neq j \text{ and, } \forall k \in V(G),\ i \to k \iff j \to k\}$$

Definition 12. *A DAG G is said to be* ***analytic*** *if $\exists \widetilde{\mathrm{SE}}(G) \subseteq \mathrm{SE}(G)$, a subset such that $|\widetilde{SE}(G)| = n - (m+1)$ and $\{e_i - e_j \mid (i,j) \in \widetilde{\mathrm{SE}}(G)\}$ is a linearly independent set where m is the maximum path length in G.*

The results of this chapter are summarized within the following theorem:

Theorem 6. *Let G be a DAG and let W be the weight matrix for an associated CTLN with parameters $\varepsilon, \delta, \theta$. Let $\sigma \subseteq [n]$ be such that $G|_\sigma$ is analytic, $(-I+W)|_\sigma$ is diagonalizable, and the polynomial:*

$$f(\lambda) = (-\lambda+\delta)^{m+1} - (1+\delta)(|\sigma|(-\lambda+\delta)^m + n_1^\sigma c(-\lambda+\delta)^{m-1} + ... + n_m^\sigma c^m)$$

has distinct roots $\{\lambda_k\}_{k=1}^{m+1}$ where $n_{j>0}^\sigma$ is the number of paths of length j in $G|_\sigma$ and m is the maximum path length in $G|_\sigma$.

Then, the general solution of L_σ is of the following form:

$$\vec{x}(t) = \sum_{k=1}^{m+1} c_k \vec{p_\sigma}(\alpha_k) e^{\lambda_k t} + \sum_{(i,j) \in \widetilde{\mathrm{SE}}(G|_\sigma)} c_{(i,j)}(e_i - e_j)e^{\delta t} + \sum_{k=1}^{n-|\sigma|} c_k \vec{v}_k e^{-t} + \vec{p_\sigma}(\beta)\Gamma(\sigma)$$

where $n = |G|$, $\alpha_k = \dfrac{\varepsilon+\delta}{\lambda_k - \delta}$, $\beta = \dfrac{-\varepsilon-\delta}{\delta}$, and $\Gamma(\sigma) = \dfrac{\theta}{-\delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}$.

4.1 Eigenvalues and Eigenvectors of L_σ

We begin by studying the relationship between localized path polynomials of the graph $G|_\sigma$ and the eigenvalues/eigenvectors of $(-I+W)|_\sigma$. First, we determine the characteristic polynomial for the matrix. To do this we will make use of the Matrix Determinant Lemma, which we restate here.

Lemma 6 (Matrix Determinant Lemma: Lemma 1.1 in [32])**.** *Suppose that A is an invertible square matrix and u and v are column vectors. Then, the determinant of $uv^T + M$ is given by:*

$$\det(uv^T + M) = (1 + v^T M^{-1} u) \det(M).$$

We use this lemma to find the characteristic polynomial of matrices of the form $\mathbb{1}\mathbb{1}^T + cA$ where A is the adjacency matrix of a DAG. An important property that we will need to make use of is the nilpotency of DAG adjacency matrices.

Proposition 6. *Let B be a matrix derived from directed acyclic graph G with maximum path length m and adjacency matrix A such that:*

$$B = \mathbb{1}\mathbb{1}^T + cA.$$

Then, the characteristic polynomial of B is:

$$f(\lambda) = \lambda^{n-(m+1)}(\lambda^{m+1} - n_0\lambda^m - n_1 c\lambda^{m-1} - ... - n_{m-1}c^{m-1}\lambda - n_m c^m)$$

where $n_0 = |G|$ and $n_{j>0}$ is the number of paths of length j in G.

Also, if $c < 0$, then the real roots of $\lambda^{m+1} - n_0\lambda^m - n_1 c\lambda^{m-1} - ... - n_{m-1}c^{m-1}\lambda - n_m c^m$ are positive.

Proof. The characteristic polynomial is $f(\lambda) = \det(\mathbb{1}\mathbb{1}^T + cA - \lambda I)$. We apply the matrix determinant lemma taking $u, v = \mathbb{1}$ and $M = cA - \lambda I$.

So, we have $f(\lambda) = (1 + \mathbb{1}^T(cA - \lambda I)^{-1}\mathbb{1}) \det(cA - \lambda I)$.

Note that a DAG can be indexed from sink to source according to its topological ordering i.e. such that if $i \geq j$ then $j \not\to i$. In this indexing, the DAG's adjacency matrix is strictly upper triangular. Then, this means that there exists a matrix P such that PAP^{-1} is strictly upper triangular. Since $P(\lambda I)P^{-1} = \lambda I$ for any invertible P:

$$\det(cA - \lambda I) = \det(P) \det(cA - \lambda I) \det(P^{-1})$$

$$= \det(cPAP^{-1} - P(\lambda I)P^{-1}) = \det(cPAP^{-1} - \lambda I).$$

Then, $cPAP^{-1} - \lambda I$ is upper triangular with $-\lambda$ on the diagonals, so $\det(cA - \lambda I) = -\det(\lambda I - cA) = -\lambda^n$. We conclude from this that:

$$f(\lambda) = -\lambda^n(1 + \mathbb{1}^T(cA - \lambda I)^{-1}\mathbb{1}).$$

We will now further analyze $((cA - \lambda I)^{-1}) = -\frac{1}{\lambda}(I - \frac{c}{\lambda}A)^{-1}$. As the adjacency matrix of a DAG is nilpotent with index $m+1$, $(\frac{c}{\lambda}A)^{m+1} = 0$ so $I - (\frac{c}{\lambda}A)^{m+1} = I$.

Then, since $1 - x^{m+1} = (1-x)\sum_{i=0}^{m} x^i$, we have:

$$I = I - \left(\frac{c}{\lambda}A\right)^{m+1} = \left(I - \left(\frac{c}{\lambda}A\right)\right)\sum_{i=0}^{m}\left(\frac{c}{\lambda}A\right)^i.$$

Multiplying by $(I - \frac{c}{\lambda}A)^{-1}$ on both sides, we obtain:

$$\left(I - \frac{c}{\lambda}A\right)^{-1} = \sum_{i=0}^{m}\left(\frac{c}{\lambda}A\right)^i.$$

So, we can rewrite the expression $1 + \mathbb{1}^T(cA - \lambda I)^{-1}\mathbb{1}$ as $1 - \sum_{i=0}^{m}\frac{c^i}{\lambda^{i+1}}\mathbb{1}(A^i)\mathbb{1}^T$.

Finally, recognizing that $\mathbb{1}(A^i)\mathbb{1}^T = n_i$, we conclude:

$$f(\lambda) = -\lambda^n\left(1 - \sum_{i=0}^{m}\frac{c^i}{\lambda^{i+1}}n_i\right) = -\lambda^{n-(m+1)}(\lambda^{m+1} - n_0\lambda^m - n_1c\lambda^{m-1} - ... - n_{m-1}c^{m-1}\lambda - n_mc^m).$$

Without loss of generality, we change the sign and have the proposed characteristic polynomial.

To find that the real roots of $q(\lambda) = \lambda^{m+1} - n_0\lambda^m - n_1c\lambda^{m-1} - ... - n_{m-1}c^{m-1}\lambda - n_mc^m$ are positive for $c < 0$, we apply Descartes' Rule of Signs. Notice that for $q(-x)$ there are no variations in this sign (there are two cases here, m even or m odd, but in both cases we end up with no variations in sign). As there are no negative real roots of $q(\lambda)$, if λ is real, $\lambda > 0$.

□

Corollary 6. *Let G be a Directed Acyclic Graph and let W be the derived CTLN weight matrix. Then, let $\sigma \subseteq [n]$ and m the maximum path length of $G|_\sigma$. Then, for $c = -\varepsilon - \delta$, the characteristic polynomial of $(-I + W)|_\sigma$ is:*

$$f(\lambda) = (-\lambda+\delta)^{|\sigma|-(m+1)}((-\lambda+\delta)^{m+1} - (1+\delta)(|\sigma|(-\lambda+\delta)^m + n_1^\sigma c(-\lambda+\delta)^{m-1} + ... + n_m^\sigma c^m)$$

where $n^{\sigma}_{j>0}$ is the number of paths of length j in $G|_{\sigma}$.

Also, the real roots of $(-\lambda+\delta)^{m+1} - |\sigma|(1+\delta)(-\lambda+\delta)^{m} - n^{\sigma}_{1}c(1+\delta)^{2}(-\lambda+\delta)^{m-1} - ... - n^{\sigma}_{m}c^{m}(1+\delta)^{m+1}$ satisfy $\lambda < \delta$.

Consider what we have demonstrated. We have made it so that we can read off a characteristic polynomial for the DAG CTLN matrices $(-I+W)|_{\sigma}$ from the combinatorial structure of the subgraph $G|_{\sigma}$. Moreover, we have also demonstrated that the real eigenvalues are less than or equal to δ. We will now take this result, and use it to express eigenvectors in terms of the localized path polynomials of $G|_{\sigma}$.

Proposition 7. *Let B be a matrix derived from directed acyclic graph G with upper triangular adjacency matrix A such that:*

$$B = \mathbb{1}\mathbb{1}^{T} + cA.$$

Let λ be an eigenvalue of B such that $\lambda \neq 0$. Then, the following is an associated eigenvector $\vec{v}$ of λ

$$\vec{v}_{j} = p^{G}_{j}\left(\frac{c}{\lambda}\right).$$

Proof. If $\lambda \neq 0$, it satisfies:

$$\lambda^{m+1} - n_{0}\lambda^{m} - n_{1}c\lambda^{m-1} - ... - n_{m-1}c^{m-1}\lambda - n_{m}c^{m} = 0$$

.

Now, rearranging this, we have:

$$\lambda^{m+1} = n_{0}\lambda^{m} + n_{1}c\lambda^{m-1} + ... + n_{m-1}c^{m-1}\lambda + n_{m}c^{m}$$

.

Then, we divide on both sides by λ^{m} and obtain:

$$\lambda = n_{0} + n_{1}\left(\frac{c}{\lambda}\right) + ... + n_{m-1}\left(\frac{c}{\lambda}\right)^{m-1} + n_{m}\left(\frac{c}{\lambda}\right)^{m} = \sum_{i=1}^{n} p^{G}_{i}\left(\frac{c}{\lambda}\right)$$

.

We directly insert and verify that, $\forall i$, $\sum_{j=1}^{n} B_{ij}\vec{v}_{j} = \lambda\vec{v}_{i}$.

$$\sum_{j=1}^{n} B_{ij}\vec{v}_{j} = \sum_{j=1}^{n}\vec{v}_{j} + c\sum_{k\to i}\vec{v}_{k} = \sum_{j=1}^{n} p^{G}_{j}\left(\frac{c}{\lambda}\right) + c\sum_{k\to i} p^{G}_{k}\left(\frac{c}{\lambda}\right) = \lambda + c\sum_{k\to i} p^{G}_{k}\left(\frac{c}{\lambda}\right).$$

If $k \to i$, a path to k of length m corresponds to a path to i of length $m+1$. With this we recognize that $c\sum_{k\to i} p_k^G\left(\frac{c}{\lambda}\right) = \lambda\left(-1 + p_i^G\left(\frac{c}{\lambda}\right)\right)$.

Then:

$$\sum_{j=1}^{n} B_{ij}\vec{v}_j = \lambda + \lambda\left(-1 + p_i^G\left(\frac{c}{\lambda}\right)\right) = \lambda p_i^G\left(\frac{c}{\lambda}\right) = \lambda\vec{v}_i.$$

□

Corollary 7. *Let G be a DAG and W be the weight matrix of an associated CTLN.*

Let λ be an eigenvalue of $(-I+W)|_\sigma$ such that $\lambda \neq \delta$. Then, the following is an associated eigenvector of the matrix for λ

$$\vec{v}_j = p_j^G(\alpha).$$

where $\alpha = \dfrac{\varepsilon+\delta}{\lambda-\delta}$.

Now, the localized path polynomials are only able to capture the eigenvectors when $\lambda \neq \delta$. How much of a problem does this present? Since we know the characteristic polynomial of the matrices $(-I+W)|_\sigma$, we also know that the algebraic multiplicity of $\lambda = \delta$ is $n-(m+1)$ where m is the maximum path length in the DAG subgraph $G|_\sigma$ (Fig 4.4). While a way to find all of the associated eigenvectors and generalized eigenvectors is lacking, there is an obvious way to identify some of them.

Proposition 8. *Let $G|_\sigma$ be a subgraph of a DAG with multiple sinks. Then, δ is an eigenvalue of $(-I+W)|_\sigma$. Moreover, corresponding eigenvectors of the form $\vec{v}_j = e_i - e_j$ exist if and only if i, j are independent vertices such that if $k \neq i, j$, then $i \to k \iff j \to k$.*

Proof. The graph theoretic condition translates to having identical columns i, j in the matrix $-I+W-\delta I$, which means that $e_i - e_j$ is in $\ker(-I+W-\delta I)$. As the pair of sinks satisfies the graph theoretic condition, then $e_1 - e_2$ is an eigenvector corresponding which means that δ is an eigenvalue. □

While not a solution in all cases, there are many DAGs where this approach helps us to obtain a full set of linearly independent eigenvectors (Fig 4.4B).

Definition 13. *We call a CTLN derived from DAG G* ***totally analytic*** *if, $\forall\sigma \subseteq [n]$ such that $|\sigma| > 2$, $(-I+W)|_\sigma$ is diagonalizable, $G|_\sigma$ is analytic, and the polynomial:*

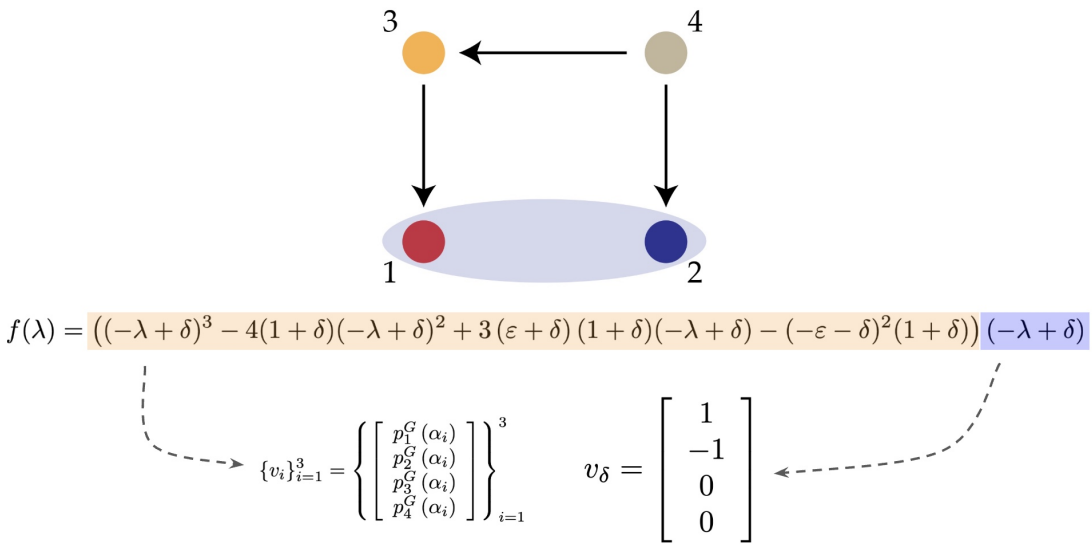


Figure 4.4. Eigenvectors of DAG CTLNs. An example of a DAG CTLN where four eigenvectors can be found for the $L_{[4]}$ system in the $R_{[4]}$ chamber. Note that $\alpha_i = \dfrac{\varepsilon+\delta}{\lambda_i - \delta}$.

$$f(\lambda) = (-\lambda+\delta)^{m+1} - (1+\delta)(|\sigma|(-\lambda+\delta)^m + n_1^\sigma c(-\lambda+\delta)^{m-1} + \ldots + n_m^\sigma c^m)$$

has distinct roots $\{\lambda_k\}_{k=1}^{m+1}$ *where* $n_{j>0}^\sigma$ *is the number of paths of length* j *in* $G|_\sigma$ *and* m *is the maximum path length in* $G|_\sigma$.

The observant reader will at this point notice an oversight. For $\sigma \neq [n]$, the system L_σ is generally of the form:

$$\begin{bmatrix} \dot{x}_\sigma \\ \dot{x}_{[n]\setminus\sigma} \end{bmatrix} = \left[\begin{array}{c|c} (-1+W)|_\sigma & C \\ \hline 0 & -I \end{array}\right] \begin{bmatrix} x_\sigma \\ x_{[n]\setminus\sigma} \end{bmatrix} + \begin{bmatrix} \theta \\ 0 \end{bmatrix}$$

What we notice here is that there is the additional eigenvalue of $\lambda = -1$ with algebraic multiplicity $n - |\sigma|$. We have not introduced a way of determining their eigenvectors. While for $|\sigma| = 1$ the solutions are easy to find, this is certainly not true for the other chambers. There is a way to find these eigenvectors, but to do so will require some more sophisticated machinery. We will return to this in the final chapter and for now will concern ourselves primarily for the time being with $L_{[n]}$. In Fig 4.5, we present an example of a non-analytic DAG.

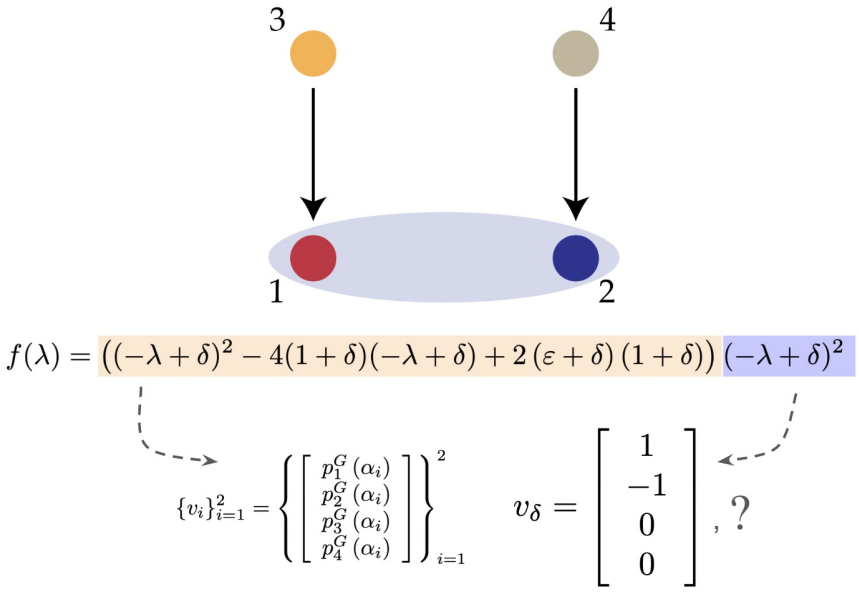


Figure 4.5. Non-analytic DAGs. An example of a DAG CTLN where only three eigenvectors can be found for the $L_{[4]}$ system in the $R_{[4]}$ chamber.

4.2 Fixed Points of L_σ: The Chamber Mapping Function

While eigenvalues and eigenvectors are sufficient to find the homogenous solution for a linear system of ODEs, we will also need a particular solution, i.e. the fixed point. We will now show the relationship between the localized path polynomial construction and the fixed points of the linear systems L_σ. The first step in this is introducing a critical lemma

Lemma 7 (DAG Lemma)**.** *Let B be a matrix derived from directed acyclic graph G with adjacency matrix A such that:*

$$B = \mathbb{1}\mathbb{1}^T + cA$$

Then the solution to the following linear system:

$$B\vec{x} = \gamma\mathbb{1} + a\vec{x}$$

is:

$$x_j = p_j^G\left(\frac{c}{a}\right)\Gamma$$

.

where $\Gamma = \dfrac{\gamma}{-a + \sum_{i=1}^{n} p_i^G(\frac{c}{a})}$

Proof. For now, assume Γ is defined. We will show $\forall i \in [n]$, the specified $\vec{x}$ satisfies the equation $\sum_{j=1}^{n} B_{ij}x_j = \gamma + ax_i$.

First, we expand out the entries of B:

$$\sum_{j=1}^{n} B_{ij}x_j = \sum_{j=1}^{n} x_j + c\sum_{k\to i} x_k.$$

Next, we will insert our proposed solution $\vec{x}$:

$$\sum_{j=1}^{n} x_j + c\sum_{k\to i} x_k = \Gamma\sum_{j=1}^{n} p_j^G\left(\frac{c}{a}\right) + c\Gamma\sum_{k\to i} p_k^G\left(\frac{c}{a}\right).$$

We now process the two sums separately:

$$\Gamma\sum_{j=1}^{n} p_j^G\left(\frac{c}{a}\right) = \frac{\gamma\sum_{j=1}^{n} p_j^G(\frac{c}{a})}{-a + \sum_{l=1}^{n} p_l^G(\frac{c}{a})} = \frac{\gamma(-a + \sum_{j=1}^{n} p_j^G(\frac{c}{a})) + a\gamma}{-a + \sum_{l=1}^{n} p_l^G(\frac{c}{a})} = \gamma + \Gamma a$$

$$c\Gamma\sum_{k\to i} p_k^G\left(\frac{c}{a}\right) = c\Gamma\sum_{k\to i}\left(1 + \sum_{m=1}^{n} n_m^k\left(\frac{c}{a}\right)^m\right) = \Gamma\sum_{k\to i}\left(c + \sum_{m=1}^{n} n_m^k\frac{c^{m+1}}{a^m}\right).$$

Now for the key idea: if $k \to i$, then a path to k of length m corresponds to a path to i of length $m+1$. So, then $\sum_{k\to i}(c + \sum_{m=1}^{n} n_m^k \frac{c^{m+1}}{a^m}) = \sum_{m=0}^{n} n_{m+1}^i \frac{c^{m+1}}{a^m}$. The second sum can then be manipulated further:

$$\Gamma\sum_{k\to i}\left(c + \sum_{m=1}^{n} n_m^k\frac{c^{m+1}}{a^m}\right) = \Gamma\sum_{m=0}^{n} n_{m+1}^i\frac{c^{m+1}}{a^m} = \Gamma a\sum_{m=0}^{n} n_{m+1}^i\frac{c^{m+1}}{a^{m+1}} = \Gamma a\left(-1 + p_i^G\left(\frac{c}{a}\right)\right).$$

At last, we combine the two sums to obtain our desired result.

$$\sum_{j=1}^{n} B_{ij}x_j = \gamma + \Gamma a - \Gamma a + ap_i^G\left(\frac{c}{a}\right)\Gamma = \gamma + ax_i.$$

We conclude by establishing that Γ is defined. The key issue is whether the denominator is non-zero. The equation $B\vec{x} = \gamma\mathbb{1} + a\vec{x}$ can be rearranged into $(B - aI)\vec{x} = \gamma\mathbb{1}$. So, we need only confirm that $\det(B - aI) \neq 0 \implies -a + \sum_{i=1}^{n} p_i^G\left(\frac{c}{a}\right) \neq 0$. By Lemma 6, we see that:

$$\det(B - aI) = a^{n-(m+1)}(a^{m+1} - n_0 a^m - n_1 c a^{m-1} - \ldots - n_{m-1} c^{m-1} a - n_m c^m).$$

Then, this means $a^{m+1} - n_0 a^m - n_1 c a^{m-1} - \ldots - n_{m-1} c^{m-1} a - n_m c^m \neq 0$. Dividing by a^m on both sides, this yields:

$$a - n_0 - n_1 \left(\frac{c}{a}\right) - \ldots - n_{m-1} \left(\frac{c}{a}\right)^{m-1} - n_m \left(\frac{c}{a}\right)^m = a - \sum_{i=1}^{n} p_i^G \left(\frac{c}{a}\right) \neq 0$$

Multiplying by -1 on both sides gives the desired $-a + \sum_{i=1}^{n} p_i^G \left(\frac{c}{a}\right) \neq 0$. □

We can now use this lemma to solve for the fixed points of the systems L_σ.

Proposition 9. *Let G be a DAG of size n, $\sigma \subseteq [n]$, and $\beta = \dfrac{-\varepsilon - \delta}{\delta}$. Then, for a CTLN associated with G, the fixed point of L_σ is:*

$$x_\sigma^* = \vec{p_\sigma}(\beta)\Gamma(\sigma)$$

where $\Gamma(\sigma) = \dfrac{\theta}{-\delta + (1+\delta)\sum_{i \in \sigma} p_i^{G|_\sigma}(\beta)}$.

Proof. The matrix corresponding to the system in this chamber is of the form $-I + W$. Restricting to the set of neurons with active rectifiers, we have $(-I + W)|_\sigma$. Call this matrix Z. Then, find the fixed point we need to solve $Z\vec{x} = -\theta\mathbb{1}$. This is equivalent to $(Z - \delta I)\vec{x} = -\theta\mathbb{1} - \delta\vec{x}$.

Notice that $\frac{-1}{1+\delta}(Z - \delta I) = \mathbb{1}\mathbb{1}^T + cA|_\sigma = \frac{\theta}{1+\delta}\mathbb{1} + \frac{\delta}{1+\delta}\vec{x}$ where $c = \frac{-\varepsilon-\delta}{1+\delta}$ and A is the adjacency matrix of G. The result follows by applying Lemma 7. □

Let us emphasize what was accomplished. For any chamber R_σ of the DAG CTLN, we are able to write the fixed point of L_σ in terms of the combinatorial structure of the DAG subgraph and see precisely how it depends on its paths. Even though these are often not fixed points of the CTLN as a whole, they nonetheless shape the dynamics of the CTLN within R_σ. Taking a DAG, we could break it down into its subgraphs $G|_\sigma$ and quickly determine the fixed points associated with each L_σ as shown in Figure 4.6.

This result begs a natural question. We know the fixed points of DAG CTLNs are supported on the sinks and the unions of sinks, so most of these fixed points are not located in their own chambers. Can we find a function that tells us what chamber they do lie in? A Chamber Mapping Function?

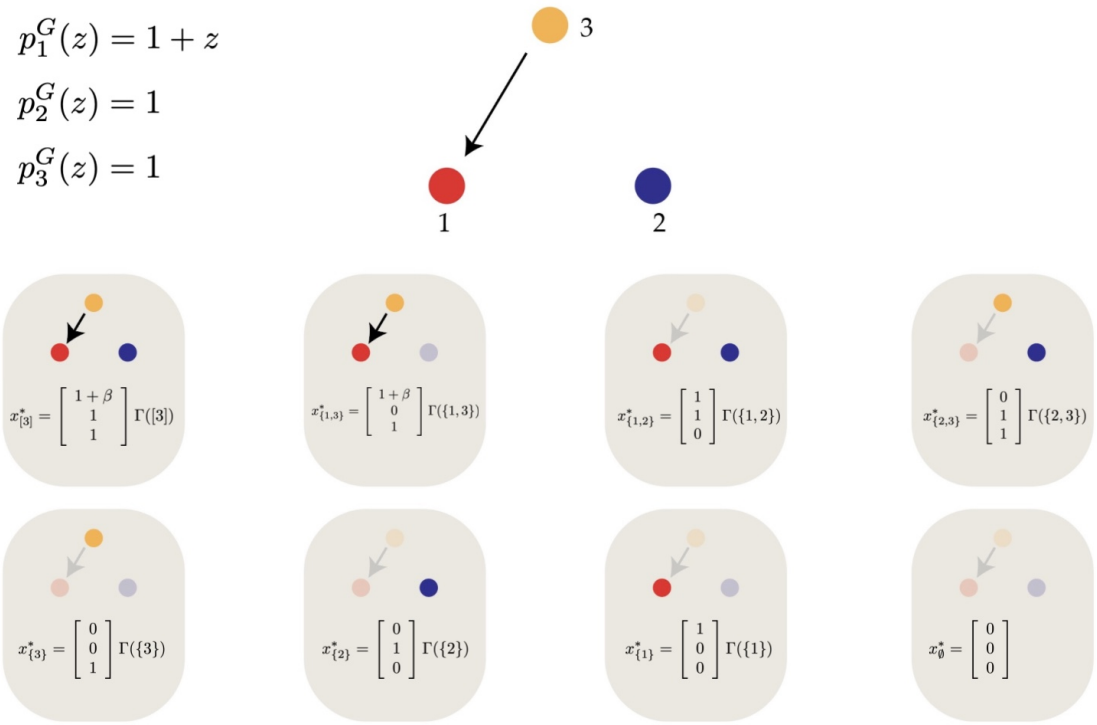


Figure 4.6. Virtual fixed points of L_σ. Fixed points of the linear systems L_σ. Most of these are not fixed points of the CTLN as a whole, but nonetheless play a role in shaping the dynamics within the corresponding R_σ.

Definition 14. *For a CTLN, define the Chamber Mapping Function to be:*

$$\mathcal{G} : 2^{[n]} \to 2^{[n]} \text{ such that } \mathcal{G}(\sigma) = \rho \iff x^*_\sigma \in R_\rho$$

Lemma 8. *For a CTLN, define x^*_σ to be the fixed point associated with the ODE system L_σ.*

Then, $i \in \mathcal{G}(\sigma) \iff y_i(x^*_\sigma) > 0$

Proposition 10. *For a CTLN derived from a Directed Acyclic Graph G, let x^*_σ be the fixed point of L_σ.*

Define:

$$g_i^\sigma = \frac{p_i^{G|_{\sigma\cup\{i\}}}(\beta)}{-\frac{\delta}{1+\delta} + \sum_{k\in\sigma} p_k^{G|_\sigma}(\beta)}$$

where $\beta = \frac{-\varepsilon-\delta}{\delta}$.

Then, construct a codeword ρ such that:

$\rho = \{i \in \sigma | g_i^\sigma > 0\} \bigcup \{k \notin \sigma | -g_k^\sigma \geq 0\}$
Then, $x_\sigma^* \in R_\rho$ *and hence* $\mathcal{G}(\sigma) = \rho$

Proof. For $i \in \sigma$, $y_i(x_\sigma^*) = (x_\sigma^*)_i$ and the result is trivial.

For $i \notin \sigma$:

$y_i(x_\sigma^*) = (-1-\delta)\sum_{j\in\sigma} x_j^\sigma + (\varepsilon+\delta)\sum_{j\in\sigma\to i} x_j^\sigma + \theta$

Applying Proposition 9, this can be rewritten as:

$y_i(x_\sigma^*) = (-1-\delta)\Gamma\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta) + (\varepsilon+\delta)\Gamma\sum_{j\in\sigma\to i} p_j^{G|_\sigma}(\beta) + \theta$

Expanding and combining, we have:

$$y_i(x_\sigma^*) = \left(\frac{(-1-\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta) + (\varepsilon+\delta)\sum_{j\in\sigma\to i} p_j^{G|_\sigma}(\beta) - \delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}{-\delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}\right)\theta.$$

This simplifies to:

$$y_i(x_\sigma^*) = \left(\frac{(\varepsilon+\delta)\sum_{j\in\sigma\to i} p_j^{G|_\sigma}(\beta) - \delta}{-\delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}\right)\theta = \left(\frac{-\delta\beta\sum_{j\in\sigma\to i} p_j^{G|_\sigma}(\beta) - \delta}{-\delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}\right)\theta.$$

We use once again that if $j \to i$, then a path to j of length k corresponds to a path to i of length $k+1$ to replace $\beta\sum_{j\in\sigma\to i} p_j^{G|_\sigma}(\beta) = p_i^{G|_{\sigma\cup\{i\}}}(\beta) - 1$.

$$y_i(x_\sigma^*) = \left(\frac{-\delta p_i^{G|_{\sigma\cup\{i\}}}(\beta) + \delta - \delta}{-\delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}\right)\theta = \frac{-\delta\theta p_i^{G|_{\sigma\cup\{i\}}}(\beta)}{-\delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}.$$

To conclude, we have:

$$y_i(x_\sigma^*) = -g_i^\sigma\left(\frac{\delta\theta}{1+\delta}\right) \geq 0 \iff -g_i^\sigma \geq 0.$$

□

Figure 4.7 shows an example of this mapping process. An additional consequence of this is that we derive an alternative proof for Theorem 5, which we restate here:

Theorem 5. (Rule 7 in [12]) *The set of fixed points of a CTLN derived from a DAG G will be supported on sinks and the unions of sinks.*

$$\mathrm{FP}(G) = \{\bigcup s_i \mid s_i \text{ is a sink in } G\}$$

Moreover, each stable fixed point will be supported on exactly one of the sinks.

Proof. First, we show that sinks and unions of sinks are fixed point supports. Let $\sigma \subseteq [n]$ be a set of sinks. Then, for $i \in \sigma$, we have:

$$g_i^\sigma = \frac{1}{-\frac{\delta}{1+\delta} + |\sigma|} > 0$$

.

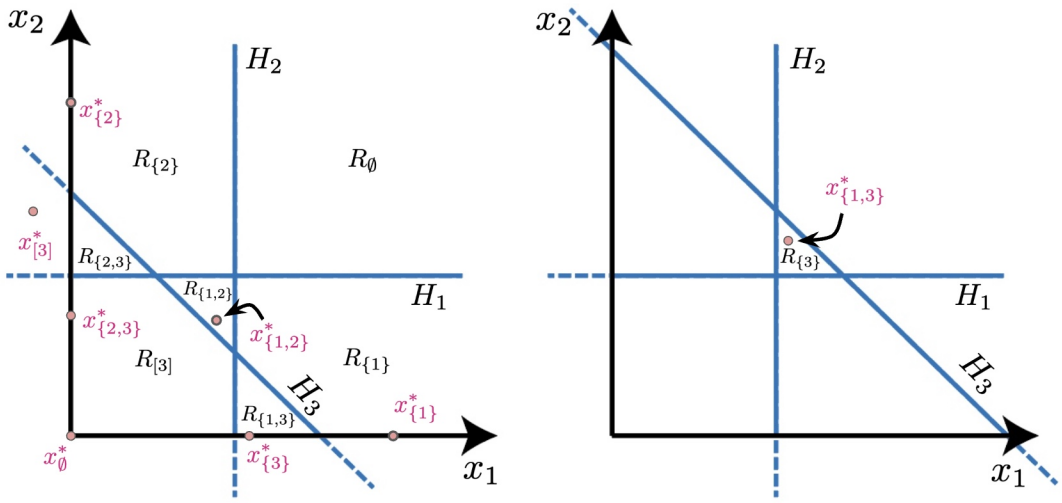


Figure 4.7. Chamber mapping of virtual fixed points. Schematic describing the chambers in which the fixed point solutions of the linear systems L_σ lie for the Decoy Effect CTLN. Unless they lie within their own chambers R_σ, these are not fixed points of the CTLN, but their locations still shape dynamics within R_σ. Each image is a different x_1, x_2-cross section of the three dimensional state space and both are needed to account for all chambers of the hyperplane partition (the first lacks R_3 while the second lacks $R_{1,2}$). While some effort has been made to give a rough sense of the state space location of the fixed points, this is not entirely accurate. The accuracy of the schematic is in which chamber the virtual fixed points would lie in.

From this we conclude that $\forall i \in \sigma$, we have $i \in \mathcal{G}(\sigma)$.

Since $\forall i \in \sigma$ are sinks of G, for any $k \notin \sigma$ we have:

$$g_k^\sigma = \frac{1}{-\frac{\delta}{1+\delta} + |\sigma|} > 0$$

.

Then $\forall k \notin \sigma$, $k \notin \mathcal{G}(\sigma)$. So, $\mathcal{G}(\sigma) = \sigma$ and we conclude that all sinks and the unions of sinks are fixed point supports of a DAG CTLN.

Now we show that no other $\sigma \subseteq [n]$ can support a fixed point. There are two cases.

Case 1: $G|_\sigma$ is not an independent set

In the case, there exist a source vertex of $G|_\sigma$, $i \in \sigma$, and some $j \in \sigma$ such that there exist paths in $G|_\sigma$ to j of length 1 but not length 2.

$$g_i^\sigma = \frac{1}{-\frac{\delta}{1+\delta} + \sum_{l\in\sigma} p_l^{G|_\sigma}(\beta)} \ , \ g_k^\sigma = \frac{1 + n_1^k \beta}{-\frac{\delta}{1+\delta} + \sum_{l\in\sigma} p_l^{G|_\sigma}(\beta)}$$

Since $\beta < -1$, necessarily $1 + n_1^k \beta < 0$. Then, if $-\frac{\delta}{1+\delta} + \sum_{l \in \sigma} p_l^{G|_\sigma}(\beta) > 0$, we would have $g_k^\sigma < 0$ and so $k \notin \mathcal{G}(\sigma)$. Otherwise, we would have $g_i^\sigma < 0$ and $i \notin \mathcal{G}(\sigma)$. Thus, $\mathcal{G}(\sigma) \neq \sigma$.

Case 2: $G|_\sigma$ is an independent set

Since by assumption σ is not a union of sinks, $\exists k \notin \sigma$ and $i \in \sigma$ such that $i \to k$. Then we have:

$$g_k^\sigma = \frac{1 + n_1^k \beta}{-\frac{\delta}{1+\delta} + |\sigma|} < 0.$$

Thus, $k \in \mathcal{G}(\sigma)$ so $\mathcal{G}(\sigma) \neq \sigma$.

We conclude from the above that $\mathcal{G}(\sigma) = \sigma \iff \sigma$ is a sink or the union of sinks.

The stability and instability of these fixed points follow from the fact that, using Corollary 6 the characteristic polynomials of the submatrices $(-I + W)|_\sigma$ in these cases are:

$$f(\lambda) = (-\lambda + \delta)^{|\sigma|-1}(-\lambda + \delta - (1+\delta)|\sigma|).$$

These have positive roots if and only if $|\sigma| > 1$. □

4.3 Combinatorial Solutions for DAG CTLNs: the Initial Value Problem

Generally speaking, diagonalizable linear dynamical systems are characterized by their fixed points and the eigenvectors/eigenvalues of their associated matrices. What we have done in this chapter is show how each of these components can be enumerated from the combinatorial structure of the DAG using the localized path polynomials. The final piece of the puzzle needed to combinatorially derive the solutions of the systems L_σ is a way of solving for the coefficients in the initial value problem.

In fact, the path polynomials do offer a way to do this as well in the case of analytic DAG CTLNs. However, the results are highly unpleasant most of the time and this show the limitations of this analytical approach. To illustrate, we will solve the system $L_{[4]}$ in the $R_{[4]}$ chamber for the DAG CTLN depicted in Fig 4.4. By convention we number the vertices from sink to source, respecting the topological ordering of the DAG such that the adjacency matrix is strictly upper triangular. Additionally, in the case we have multiple eigenvectors for $\lambda = \delta$ which share an entry (e.g. $v_1 = e_{i_1} - e_{j_1}$ and $v_2 = e_{i_2} - e_{j_2}$ such

that $i_1 = j_2$), we write them so that the j's are distinct.

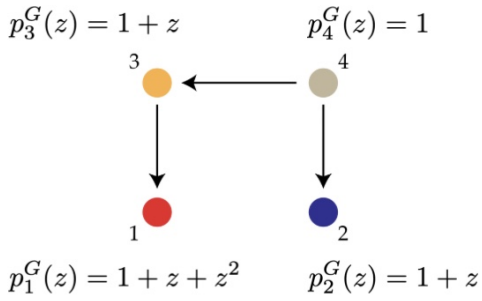


Figure 4.8. Localized path polynomials for G. Each of the four vertices has a localized path polynomial listed here.

4.3.1 Fixed Point

The fixed point is the first component in finding the solution of L_σ, serving as the particular solution. We simply apply Proposition 9 and obtain the fixed point:

$$\vec{x}^*(t) = \begin{bmatrix} p_1^G(\beta) \\ p_2^G(\beta) \\ p_3^G(\beta) \\ p_4^G(\beta) \end{bmatrix} \Gamma$$

where, as before, $\beta = \frac{-\varepsilon-\delta}{\delta}$ and $\Gamma = \dfrac{\theta}{-\delta + (1+\delta)\sum_{j=1}^4 p_j(\beta)}$.

4.3.2 Eigenvectors

From Corollary 6, we know the characteristic polynomial for the matrix $-I + W$ is:

$$f(\lambda) = (-\lambda+\delta)((-\lambda+\delta)^3 - 4(1+\delta)(-\lambda+\delta)^2 - 3c(1+\delta)^2(-\lambda+\delta) - c^2(1+\delta)^3).$$

For $\lambda = \delta$, as the algebraic multiplicity is 1 we need only one eigenvector. Applying Lemma 8 we have $v_\delta = e_1 - e_2$.

The remaining three eigenvalues are the roots of $(-\lambda+\delta)^3 - 4(1+\delta)(-\lambda+\delta)^2 - 3c(1+\delta)^2(-\lambda+\delta) - c^2(1+\delta)^3$. While in this particular case the roots could be obtained through the cubic formula, in general the problem of finding the roots of a sufficiently

high degree polynomial is not algebraically tractable. We will simply refer to the roots as λ_1, λ_2, and λ_3. We accordingly use the notation $\alpha_i = \frac{\varepsilon+\delta}{\lambda_i-\delta}$.

Then, using Corollary 7, the full set of eigenvectors and eigenvalues are:

$$\lambda = \delta : \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \lambda = \{\lambda_i\}_{i=1}^{3} : \left\{ \begin{bmatrix} p_1^G(\alpha_i) \\ p_2^G(\alpha_i) \\ p_3^G(\alpha_i) \\ p_4^G(\alpha_i) \end{bmatrix} \right\}_{i=1}^{3}.$$

This gives us the general solution:

$$\vec{x}(t) = \vec{x}_H(t) + \vec{x}_p(t)$$

$$= c_1 \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} e^{\delta t} + c_2 \begin{bmatrix} p_1^G(\alpha_1) \\ p_2^G(\alpha_1) \\ p_3^G(\alpha_1) \\ p_4^G(\alpha_1) \end{bmatrix} e^{\lambda_1 t} + c_3 \begin{bmatrix} p_1^G(\alpha_2) \\ p_2^G(\alpha_2) \\ p_3^G(\alpha_2) \\ p_4^G(\alpha_2) \end{bmatrix} e^{\lambda_2 t} + c_4 \begin{bmatrix} p_1^G(\alpha_3) \\ p_2^G(\alpha_3) \\ p_3^G(\alpha_3) \\ p_4^G(\alpha_3) \end{bmatrix} e^{\lambda_3 t} + \begin{bmatrix} p_1^G(\beta) \\ p_2^G(\beta) \\ p_3^G(\beta) \\ p_4^G(\beta) \end{bmatrix} \Gamma.$$

4.3.3 Solving the Initial Value Problem

We have up until this point the general solution:

$$x_1(t) = c_1 e^{\delta t} + c_2 p_1^G(\alpha_1) e^{\lambda_1 t} + c_3 p_1^G(\alpha_2) e^{\lambda_2 t} + c_4 p_1^G(\alpha_3) e^{\lambda_3 t} + p_1^G(\beta)\Gamma$$

$$x_2(t) = -c_1 e^{\delta t} + c_2 p_2^G(\alpha_1) e^{\lambda_1 t} + c_3 p_2^G(\alpha_2) e^{\lambda_2 t} + c_4 p_2^G(\alpha_3) e^{\lambda_3 t} + p_2^G(\beta)\Gamma$$

$$x_3(t) = c_2 p_3^G(\alpha_1) e^{\lambda_1 t} + c_3 p_3^G(\alpha_2) e^{\lambda_2 t} + c_4 p_3^G(\alpha_3) e^{\lambda_3 t} + p_3^G(\beta)\Gamma$$

$$x_4(t) = c_2 e^{\lambda_1 t} + c_3 e^{\lambda_2 t} + c_4 e^{\lambda_3 t} + \Gamma.$$

The last step then is to work out the initial value problem for this system. Taking the initial condition $\vec{x}_0$, we set up the system:

$$\begin{bmatrix} 1 & 1+\alpha_1+\alpha_1^2 & 1+\alpha_2+\alpha_2^2 & 1+\alpha_3+\alpha_3^2 \\ -1 & 1+\alpha_1 & 1+\alpha_2 & 1+\alpha_3 \\ 0 & 1+\alpha_1 & 1+\alpha_2 & 1+\alpha_3 \\ 0 & 1 & 1 & 1 \end{bmatrix} \vec{c} = \begin{bmatrix} x_1^0 - p_1^G(\beta)\Gamma \\ x_2^0 - p_2^G(\beta)\Gamma \\ x_3^0 - p_3^G(\beta)\Gamma \\ x_4^0 - \Gamma \end{bmatrix}$$

We will now employ a three step process to find $\vec{c}$.

Step 1: Turn columns from eigenvectors of $\lambda = \delta$ to distinct basis vectors

By construction , all of our eigenvectors for $\lambda = \delta$ are of the form $e_i - e_j$ such that each j is distinct. For each of these eigenvectors, we add the j-th row of the system to the i-th row, turning the corresponding column into $-e_j$. In our current system, we would then have:

$$\begin{bmatrix} 0 & 2+2\alpha_1+\alpha_1^2 & 2+2\alpha_2+\alpha_2^2 & 2+2\alpha_3+\alpha_3^2 \\ -1 & 1+\alpha_1 & 1+\alpha_2 & 1+\alpha_3 \\ 0 & 1+\alpha_1 & 1+\alpha_2 & 1+\alpha_3 \\ 0 & 1 & 1 & 1 \end{bmatrix} \vec{c} = \begin{bmatrix} x_1^0 + x_2^0 - (p_1^G(\beta) + p_2^G(\beta))\Gamma \\ x_2^0 - p_2^G(\beta)\Gamma \\ x_3^0 - p_3^G(\beta)\Gamma \\ x_4^0 - \Gamma \end{bmatrix}$$

Step 2: Restrict to the last $m+1$ entries of $\vec{c}$ and solve the Vandermonde Matrix

We will set aside rows that have non-zero entries in the first $n-m+1$ columns for the time being. So, we will have a $(m+1) \times (m+1)$ subsystem. In our example, we have the following subsystem:

$$\begin{bmatrix} 2+2\alpha_1+\alpha_1^2 & 2+2\alpha_2+\alpha_2^2 & 2+2\alpha_3+\alpha_3^2 \\ 1+\alpha_1 & 1+\alpha_2 & 1+\alpha_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} x_1^0 + x_2^0 - (p_1^G(\beta) + p_2^G(\beta))\Gamma \\ x_3^0 - p_3^G(\beta)\Gamma \\ x_4^0 - \Gamma \end{bmatrix}$$

Now for the key trick. We will transform the matrix on the left into the well known Vandermonde Matrix. By subtracting the bottom row we can eliminate the constant terms.

$$\begin{bmatrix} 2\alpha_1+\alpha_1^2 & 2\alpha_2+\alpha_2^2 & 2\alpha_3+\alpha_3^2 \\ \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} x_1^0 + x_2^0 - 2x_4^0 - (p_1^G(\beta) + p_2^G(\beta) - 2p_4^G(\beta))\Gamma \\ x_3^0 - x_4^0 - (p_3^G(\beta) - p_4^G(\beta))\Gamma \\ x_4^0 - \Gamma \end{bmatrix}$$

We can then repeat this process, with some potential scaling, moving up the rows with $\alpha_i, \alpha_i^2, ..., \alpha_i^{m-1}$ respectively until we are left with the Vandermonde Matrix:

$$\begin{bmatrix} \alpha_1^2 & \alpha_2^2 & \alpha_3^2 \\ \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} x_1^0 + x_2^0 - 2x_4^0 - 2x_3^0 - (p_1^G(\beta) + p_2^G(\beta) - 2p_3^G(\beta) - 2p_4^G(\beta))\Gamma \\ x_3^0 - x_4^0 - (p_3^G(\beta) - p_4^G(\beta))\Gamma \\ x_4^0 - \Gamma \end{bmatrix}$$

The benefit of this is that the inverse of the Vandermonde matrix of size $(m+1)\times(m+1)$ is known to have the entries:

$$(V^{-1})_{ij} = \frac{(-1)^{j-1} E_{j-1}(\{\alpha_1, \ldots, \alpha_{m+1}\} \backslash \{\alpha_i\})}{\Pi_{k=1, k\neq i}^{m+1}(\alpha_i - \alpha_k)}$$

where $E_m(\{y_1, ..., y_k\}) = \sum_{1 \leq j_1 < \cdots < j_m \leq k} y_{j_1} \ldots y_{j_m}$ are the elementary symmetric functions. [33]

So, we can then solve for the coefficients:

$$c_2 = \sum_{j=1}^{3} (V^{-1})_{1j} \phi_j$$

$$c_3 = \sum_{j=1}^{3} (V^{-1})_{2j} \phi_j$$

$$c_4 = \sum_{j=1}^{3} (V^{-1})_{3j} \phi_j$$

where:

$$\vec{\phi} = \begin{bmatrix} x_1^0 + x_2^0 - 2x_4^0 - 2x_3^0 - (p_1^G(\beta) + p_2^G(\beta) - 2p_3^G(\beta) - 2p_4^G(\beta))\Gamma \\ x_3^0 - x_4^0 - (p_3^G(\beta) - p_4^G(\beta))\Gamma \\ x_4^0 - \Gamma \end{bmatrix}.$$

Step 3: Solve for the remaining coefficients

We can now also solve for the remaining $n - (m + 1)$ coefficients. As a byproduct of **Step 1**, each of these coefficients corresponds to exactly one of the rows omitted in **Step 2**. In our case, c_1 corresponds to row 2.

$$-c_1 + c_2(1 + \alpha_1) + c_3(1 + \alpha_2) + c_4(1 + \alpha_4) = x_2^0 - p_2^G(\beta)\Gamma.$$

Finally we obtain our last coefficient:

$$c_1 = c_2(1+\alpha_1) + c_3(1+\alpha_2) + c_4(1+\alpha_4) - x_2^0 + p_2^G(\beta)\Gamma.$$

4.4 Takeaways

The results from this chapter demonstrate how finding analytical solutions for DAG CTLNs is surprisingly tractable on a theoretical level. The results in our final example were parameter independent up to some conditions and didn't require particularly difficult calculation to achieve. However, the obvious problem is that the expressions are simply too unwieldy. Trying to piece together analytical trajectories across chambers with these coefficients is not practical, especially since finding the points of intersection with the chamber walls will entail the solution of transcendental equations.

It would be natural to wonder whether an approximate solution, perhaps obtained through the method of matrix exponentials, might suffice. The challenge with this approach would be that such approximations are mainly reliable at small time scales and so would be less useful in tracking the long term dynamics of trajectories towards an attractor. Ultimately, our takeaway from this chapter is that, despite the considerable theory that can be developed about DAG CTLNs derived from the graph combinatorics, the limitations in higher dimensions are tough to overcome if the goal is to calculate the full basins of attraction.

Chapter 5
Decision-Making Bias Near Decision Boundaries

Our work so far has simultaneously demonstrated both the tractability of DAG CTLNs along with the challenges of even roughly determining their full basins of attraction. That said, we also had paradigms of decision-making that did not require analyzing the full basins of attraction. The focus of this chapter will be the paradigm of studying initial conditions near decision boundaries. This approach is aligned with an ontology of decision-making circuits that sees them as operating along branching manifolds [34]. In the case of a binary choice task, this consists of trajectories beginning along a manifold representing the decision boundary before being carried away along another manifold toward one of the attractors encoding a decision.

With respect to CTLNs of DAGs with two sinks, our branching manifolds are the codimension 1 stable manifold and the dimension 1 unstable manifold associated with the saddle point supported on the union of sinks. The question of course is how to study trajectories near decision boundaries without actually knowing the decision boundary itself. Consider that we are interested then in trajectories which hew close to these manifolds, traveling near the saddle point. As TLNs are autonomous systems, and we are ultimately only concerned with categorizing trajectories based on which attractors they converge to, we can begin tracking them when they are near the saddle point (Fig 5.1). Through this thinking, we can move from thinking about initial conditions near the decision boundary to initial conditions near the saddle point.

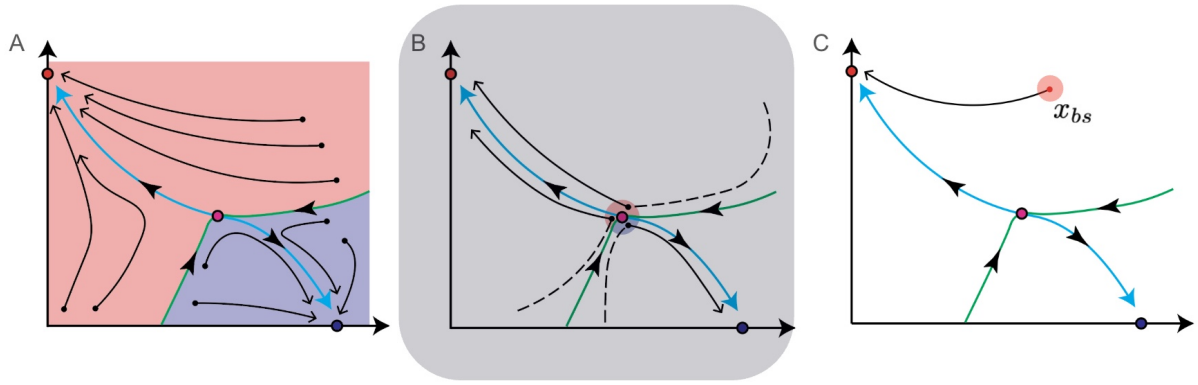


Figure 5.1. Decision-making dynamics near low dimensional submanifolds. Having been unsuccessful at analytically determining the basins of attraction for DAG CTLNs, in this chapter we explore the basins of attraction with the neighborhood of an unstable fixed point as a proxy for studying trajectories near the branching stable and unstable manifolds of the network.

5.1 DAGs with Two Sinks

Proposition 11. *Let G be a DAG with two sinks. For a CTLN derived from G such that the sinks correspond to x_1 and x_2, the stable manifold in the $L_{\{1,2\}}$ system for its saddle point is:*

$$-x_1 + x_2 + a_3x_3 + ... + a_nx_n = 0$$

where:

$$a_j = 0 \text{ if } j \not\to 1 \text{ and } j \not\to 2 \text{ or } j \to 1 \text{ and } j \to 2$$
$$a_j = \tfrac{-\varepsilon-\delta}{1+\delta} \text{ if } j \to 1 \text{ and } j \not\to 2$$
$$a_j = \tfrac{\varepsilon+\delta}{1+\delta} \text{ if } j \not\to 1 \text{ and } j \to 2$$

Proof. The matrix corresponding to the linear ODE system $L_{\{1,2\}}$ is:

$$A = \left[\begin{array}{cc|ccc} -1 & -1-\delta & w_{13} & \dots & w_{1n} \\ -1-\delta & -1 & w_{23} & \dots & w_{2n} \\ \hline 0 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -1 \end{array}\right]$$

Which has eigenvalues δ and $-2-\delta$ from the upper left block and repeated eigenvalue -1 from the lower right block. We claim the eigenvectors are as follows:

$$\lambda_1 = \delta : v_1 = \begin{bmatrix} -1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\lambda_2 = -2 - \delta : v_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\lambda_{3,\dots,n} = -1 : v_{3,\dots,n} = \begin{bmatrix} \frac{-w_{23}}{1+\delta} \\ \frac{-w_{13}}{1+\delta} \\ -1 \\ \vdots \\ 0 \end{bmatrix} \dots \begin{bmatrix} \frac{-w_{2n}}{1+\delta} \\ \frac{-w_{1n}}{1+\delta} \\ 0 \\ \vdots \\ -1 \end{bmatrix}$$

For λ_1 and λ_2 this is clear from inspection. For $\lambda_{k\geq 3}$, see that:

$$Av_k = \begin{bmatrix} \frac{w_{2k}}{1+\delta} + w_{1k} - w_{1k} \\ w_{2k} + \frac{w_{1k}}{1+\delta} - w_{2k} \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{w_{2k}}{1+\delta} \\ \frac{w_{1k}}{1+\delta} \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = -v_k.$$

Notice that there are $n-1$ eigenvectors corresponding to negative eigenvalues and so the stable manifold of the system $L_{\{1,2\}}$ will be the hyperplane spanned by those eigenvectors. We now claim that the normal vector to this stable manifold of $L_{\{1,2\}}$ is:

$$\vec{n} = \begin{bmatrix} -1 \\ 1 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \quad \text{where} \quad \begin{array}{c} a_j = 0 \text{ if } j \not\to 1 \text{ and } j \not\to 2 \text{ or } j \to 1 \text{ and } j \to 2 \\ a_j = \frac{-\varepsilon-\delta}{1+\delta} \text{ if } j \to 1 \text{ and } j \not\to 2 \\ a_j = \frac{\varepsilon+\delta}{1+\delta} \text{ if } j \not\to 1 \text{ and } j \to 2 \end{array} .$$

Again, for λ_2 this is clear from inspection. We again show via direct computation

that this is true for $\lambda_{k\geq 3}$.

$$\vec{n} \cdot v_k = \frac{w_{2k}}{1+\delta} - \frac{w_{1k}}{1+\delta} - a_k = \frac{w_{2k} - w_{1k}}{1+\delta} - a_k.$$

If $k \not\to 1$ and $k \not\to 2$ or $k \to 1$ and $k \to 2$:

$$w_{2k} - w_{1k} = 0.$$

If $k \to 1$ and $k \not\to 2$:

$$w_{2k} - w_{1k} = -\varepsilon - \delta.$$

If $k \not\to 1$ and $k \to 2$:

$$w_{2k} - w_{1k} = \varepsilon + \delta.$$

In each of these cases, $\frac{w_{2k} - w_{1k}}{1+\delta} = a_k$ as needed to have $\vec{n} \cdot v_k = 0$. So, we know that the stable manifold of $L_{\{1,2\}}$ is of the form $\vec{n} \cdot \vec{x} + c = 0$.

As the saddle point $x^* = \left(\frac{\theta}{2+\delta}, \frac{\theta}{2+\delta}, 0, \ldots, 0\right)$ must lie on this manifold, we use it to find c.

$$\vec{n} \cdot x^* + c = 0.$$

Since $\vec{n} \cdot x^* = -\frac{\theta}{2+\delta} + \frac{\theta}{2+\delta} = 0$, we conclude $c = 0$ and we are done.

□

What Proposition 11 indicates is that if we were to ignore the dynamics of other chambers and simply consider the local dynamics in the $L_{\{1,2\}}$ chamber for the saddle point $x^*_{\{1,2\}}$ supported on the union of the two sinks, then the sink with the greater in-degree should have the larger basin of attraction. Of course the actual stable manifold of the CTLN will be more complex and highly non-linear, even within the $R_{\{1,2\}}$ chamber. We conjecture that if we restrict our analysis to a small region around the fixed point, $S := B_\eta(x^*_{\{1,2\}}) \cap R^+_{\{1,2\}}$ where $\eta << 1$, then the linear stable manifold of the $L_{\{1,2\}}$ system should approximate this small piece of the actual stable manifold of the CTLN. For a basin of attraction of a sink, call it $\mathcal{B}_i$, we expect the fractional volume in this region:

$$\mathcal{F}_i := \frac{\lambda(\mathcal{B}_i \cap S)}{\lambda(S)}$$

to be influenced strongly by the sink's indegree.

To be mathematically clear, what we would expect is a strong correlation between $\mathcal{F}_i$ and the fractional indegree of the sink over that of all the sinks:

$$\text{indeg}_\text{f}(i) := \frac{\text{indeg}(i)}{\sum_{j \in \text{sinks}(G)} \text{indeg}(j)}.$$

Numerically simulating for 1000 random DAGs of 6 neurons with two sinks for $\eta = 0.01$, we obtain the results in Fig 5.2A. Notice the correlation is very strong which provides clear numerical evidence for our conjecture. Compare this with a weaker relationship if we were to sample initial conditions more broadly from the state space as in Fig 5.2B. Beyond the correlation coefficients, notice the sheer spread when considering the full basins of attraction

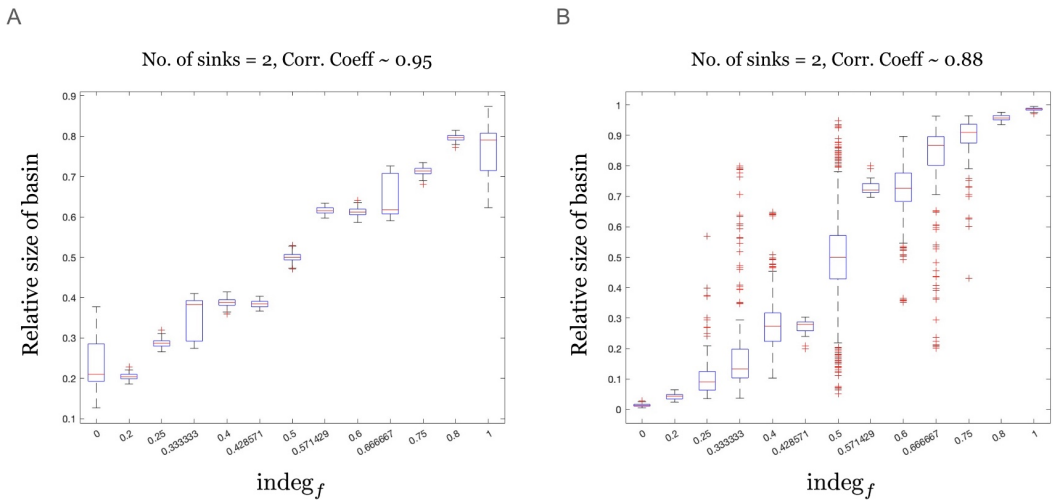


Figure 5.2. Correlation between basin size and sink fractional indegree. (A) Boxplots depicting the fraction of trajectories converging to a sink vs. the fractional indegree of the sink for initial conditions sampled near the saddle point. (B) Boxplots illustrating the same, but with initial conditions sampled from state space more broadly, the box $[0, 1.5]^6$. Both of these simulations use the same 1000 DAGs of size 6 with parameters $\delta = 0.5$, $\varepsilon = 0.25$, and $\theta = 1$. The number of initial conditions sampled per DAG was 2500.

What can we say about DAGs more broadly?

5.2 DAGs with Several Sinks

We can find a result similar to Proposition 11 for DAGs with several sinks by making use of the Sherman-Morrison Formula.

Lemma 9 (Sherman-Morrison Formula [35]). *Suppose M is an $n \times n$ matrix with u, v being $n \times 1$ column vectors. Then, $uv^T + M$ is invertible if and only if $1 + v^T M^{-1} u \neq 0$. In this case,*

$$(uv^T + M)^{-1} = M^{-1} - \frac{M^{-1} u v^T M^{-1}}{1 + v^T M^{-1} u}.$$

Proposition 12. *Let G be a DAG of size n and $\sigma \subseteq [n]$ such that $G|_\sigma$ is an independent set. Then, the linear system L_σ for an associated CTLN has fixed point $x^*_\sigma = \dfrac{\theta}{|\sigma| + (|\sigma| - 1)\delta} \mathbb{1}_\sigma$ with an unstable manifold of dimension $|\sigma| - 1$ which is :*

$$x^*_\sigma + \text{span}\left(\{e_{\sigma_1} - e_{\sigma_j} \mid \sigma_1, \sigma_j \in \sigma \text{ and } \sigma_j \neq \sigma_1\}\right)$$

and a stable manifold of codimension $|\sigma| - 1$ which is:

$$x^*_\sigma + \text{span}\left(\mathbb{1}_\sigma \cup \left\{\left(\frac{1}{|\sigma| - 1} - \gamma \frac{id_j(\sigma)}{|\sigma| - 1}\right)\mathbb{1}_\sigma + \gamma \mathbb{1}_{\sigma \leftarrow j} - e_j \mid j \notin \sigma\right\}\right)$$

where $\mathbb{1}_s = \sum_{i \in s} e_i$, $\gamma = \dfrac{\varepsilon + \delta}{1 + \delta}$, and $id_j(\sigma) = |\{k \in \sigma \mid j \to k\}|$.

Proof. Without loss of generality number the vertices in σ to be $1, \ldots, k$ where $k = |\sigma|$. Then the matrix for L_σ is of the form:

$$A = \left[\begin{array}{ccc|ccc} -1 & \ldots & -1-\delta & w_{1,k+1} & \ldots & w_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ldots & \vdots \\ -1-\delta & \ldots & -1 & w_{k,k+1} & \ldots & w_{kn} \\ \hline 0 & \ldots & 0 & -1 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & -1 \end{array}\right]$$

where, because $G|_\sigma$ is an independent set, the upper left block is of the form

$$(-I + W)|_\sigma = (-1 - \delta)\mathbb{1}\mathbb{1}^T + \delta I.$$

Recalling Lemma 6, we know that the characteristic polynomial of A is:

$$p(\lambda) = (\lambda + 1)^{n-k}(-\lambda + \delta)^{k-1}(-\lambda + \delta - (1 + \delta)|\sigma|)$$

and so its eigenvalues are $\lambda_1 = \delta - (1 + \delta)|\sigma|$, $\lambda_2 = \delta$, and $\lambda_3 = -1$ with algebraic multiplicities 1, $k - 1$, and $n - k$ respectively.

The set:

$$\{e_1 - e_j \mid 1 < j \leq k\}$$

is linearly independent and of size $k-1$ with each element being an eigenvector for $\lambda = \delta$ (Proposition 8). Therefore it is a basis for the unstable manifold.

Now we turn our attention to the stable manifold. For $\lambda = \delta - (1+\delta)|\sigma|$, applying Corollary 7, we know that $\vec{p_\sigma}(\alpha)$ is the associated eigenvector. However, since $G|_\sigma$ is an independent set, $p_i^{G|_\sigma}(z) = 1$, $\forall i \in \sigma$. We will thus refer to this eigenvector as $\mathbb{1}_\sigma$.

Finally, we find eigenvectors for $\lambda = -1$. We show that there are $n-k$ linearly independent eigenvectors by construction. We then take as an ansatz vectors of the form:

$$\vec{v} = \left[\begin{array}{c} c_1 \\ \vdots \\ c_k \\ \hline 0 \\ \vdots \\ -1 \\ \vdots \\ 0 \end{array}\right] = \left[\begin{array}{c} \vec{c} \\ \hline 0 \end{array}\right] - e_j$$

Then, we have:

$$A\vec{v} = \left[\begin{array}{c} ((-I+W)|_\sigma)\vec{c} - \vec{w}_{*j} \\ \hline 0 \end{array}\right] + e_j$$

where

$$\vec{w}_{*j} = \begin{bmatrix} w_{1j} \\ \vdots \\ w_{kj} \end{bmatrix}.$$

So, if $\vec{c}$ satisfies $((-I+W)|_\sigma)\vec{c} - \vec{w}_{*j} = -\vec{c}$ then $\vec{v}$ is an eigenvector. Rearranging this system can be rewritten as:

$$((-I+W)|_\sigma)\vec{c} + I\vec{c} = \vec{w}_{*j} \implies ((-1-\delta)\mathbb{1}\mathbb{1}^T + (1+\delta)I)\vec{c} = \vec{w}_{*j} \implies (\mathbb{1}\mathbb{1}^T - I)\vec{c} = -\frac{\vec{w}_{*j}}{1+\delta}.$$

Finally, we apply the Sherman-Morrison Formula with $M = -I$ and $u, v = \mathbb{1}$. Then

$M^{-1} = -I$ and so:

$$(\mathbb{1}\mathbb{1}^T - I)^{-1} = -I - \frac{1}{1-|\sigma|}\mathbb{1}\mathbb{1}^T = \left(\frac{1}{|\sigma|-1}\right)\mathbb{1}\mathbb{1}^T - I.$$

Finally, we have that:

$$\vec{c} = -\frac{1}{1+\delta}(\mathbb{1}\mathbb{1}^T - I)^{-1}\vec{w}_{*j} = -\frac{1}{1+\delta}\left(\frac{1}{|\sigma|-1}\sum_{k\in\sigma} w_{kj}\mathbb{1} - \vec{w}_{*j}\right) =$$

Substituting the matrix entries, we have:

$$\vec{c} = \left(\frac{|\sigma|}{|\sigma|-1} - \gamma\frac{id_j(\sigma)}{|\sigma|-1} - 1\right)\mathbb{1} + \gamma\mathbb{1}_{\sigma\leftarrow j} = \left(\frac{1}{|\sigma|-1} - \gamma\frac{id_j(\sigma)}{|\sigma|-1}\right)\mathbb{1} + \gamma\mathbb{1}_{\sigma\leftarrow j}.$$

The value of the fixed point follows from Proposition 9. □

What we see from Proposition 12 is that in larger DAGs with multiple saddle points, stable manifolds of the linear systems L_σ are of codimension $|\sigma|-1$ whereas the unstable manifold is of dimension $|\sigma|-1$. Moreover, the expression of each depends only on neurons which depends only on the activity of nonsink neurons which have contribute edges toward the sinks. While of course the actual stable and unstable manifolds will be more complex and connected, we might still expect that sink indegree continues to be a strong factor in shaping dynamics along the decision boundaries.

It is worth noting that up until this point we have somewhat been referring to the stable manifolds of saddle points and the decision boundaries interchangeably because in the case of two sink DAGs they are equivalent. It is clear that this relationship does not hold quite so directly when there are $k > 2$ sinks, i.e. k basins of attraction, and $2^k - k - 1$ saddle points. Instead we should expect that the trajectories composing the decision boundaries will pass near the unique saddle point supported on the union of the sinks, $x^*_{\text{sinks}(G)}$.

What we conjecture now is that, the fractional volume of a basin of attraction for a sink i:

$$\mathcal{F}_i := \frac{\lambda(\mathcal{B}_i \cap S)}{\lambda(S)}$$

where $S := B_\eta\left(x^*_{\text{sinks}(G)}\right) \cap R^+_{\text{sinks}(G)}$, correlates with $\text{indeg}_{\text{f}}(i)$.

Numerically simulating this for 1000 random DAGs of six neurons for $\eta = 0.01$ split

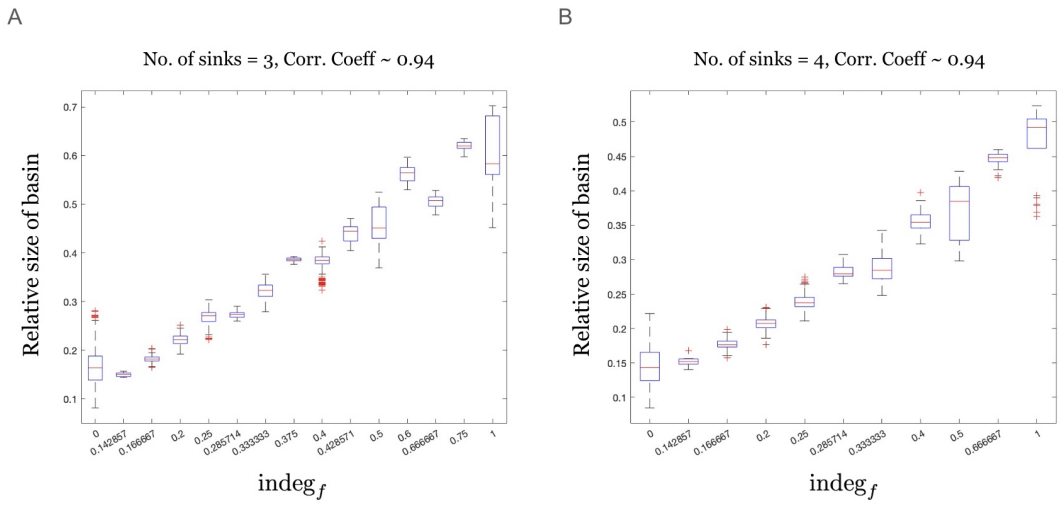


Figure 5.3. Correlation between basin size and sink fractional indegree. Simulation similar to Fig 5.1A sampling in the neighborhood of the fixed points supported on all sinks. 1000 DAGs were simulated of size $n = 6$ where 500 had 3 sinks and 500 had 4 sinks with parameters $\delta = 0.5$, $\varepsilon = 0.25$, and $\theta = 1$. The number of initial conditions sampled per DAG was 2500. (A) Restricted to the 500 DAGs with 3 sinks. (B) Restricted to the 500 DAGs with 4 sinks.

between 3 and 4 sinks, we again find a reasonably strong correlation as depicted in Fig 5.3A and Fig 5.3B respectively. While the spread is noticeably less tame than in the two dimensional case, and the monotonicity does not quite seem to hold, it is still a far superior than the relationship in Fig 5.2B. A point to note is that, in all the cases we have looked at, the accuracy drops considerably if we consider sinks where $\text{indeg}_\text{f} = 0$. This is likely because of a limitation in the indeg_f construction where in this case it makes no distinction between a neuron having indegree 0 whether $\sum_{j \in \text{sinks}(G)} \text{indeg}(j)$ is large or small. This could create considerable variance in this case.

In the above analysis we found strong computational results, justified by some theoretical results, that, in a DAG CTLN, if we restrict our basins of attraction to a neighborhood of the saddle point $x^*_{\text{sinks}(G)}$, the volumes of the basins relative to one another are strongly related to the fractional indegree, indeg_f, received by the sinks relative to one another. The significance of this is that, assuming a circuit architecture comparable to that of a DAG CTLN, under the hypothesis that the dynamics of a decision-making circuit operate along their decision-boundaries before arriving at a decision, we expect bias to be strongly affected by the direct excitation received by the neural population encoding a choice relative to that received by populations encoding other choices.

Chapter 6
Balanced States and their Decision-Making Dynamics in DAG CTLNs

We turn our attention now back to balanced states. As a reminder, the balanced state of a TLN is when the internal inhibition of the circuit is cancelled out by the excitation of external input current i.e. when $\sum_{i=1}^{n} W_{ij}x_j + \theta_j = 0$ for all $i \in [n]$. Combining each of these conditions, the balanced state is the state x_{bs} such that $Wx_{bs} + \vec{\theta} = 0$.

Remark 5. *Notice that the balanced state $\vec{x}_{bs}$ corresponds to the intersection of the hyperplanes $\{H_i\}_{i=1}^{n}$. It is the unique point adjacent to all chambers of the TLN.*

Balanced states are of great interest in the theory of attractor networks, but are often computed without considering further dynamics and are at times thought to be related to the stable states of the network. [19]. This is often not true in many attractor network models. Certainly this is not generally the case in TLNs. What we will consider here is the possibility that the balanced state represents the appropriate starting point of the network as it begins computation. Since the balanced state does not typically lie on a separatrix, it will generally converge to one of the attractors. In this paradigm, we will consider the bias of the network to be toward that attractor. What we seek to determine is which basin of attraction the balanced state lies in. To be more precise, as the balanced state is a computation on the external input received by a network, the vector $\vec{\theta}$, we might consider that the appropriate way of measuring the bias of the decision-making circuit is to evaluate to which attractor the balanced state converges in the event of uniform external input i.e. $\vec{\theta} = \theta\mathbb{1}$, which is the case in CTLNs. As shown in Fig 6.1, privileging this particular trajectory this aligns with the paradigm of path-following dynamics.

As seen in our discussion of the Binary Competition Model, two-dimensional compet-

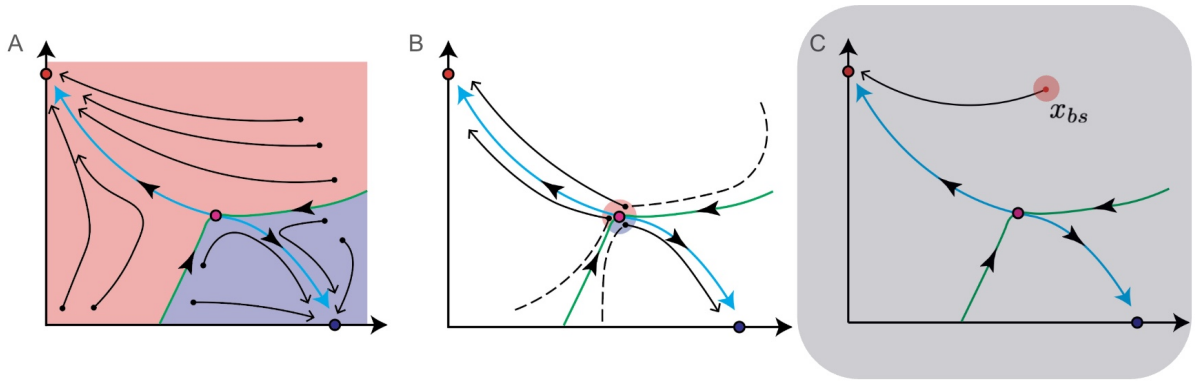


Figure 6.1. Decision-making dynamics along the balanced state trajectory. Within this paradigm, the bias of the decision-making circuit is aligned with the attractor to which the balanced state trajectory converges.

itive TLNs always have a balanced state in the positive quadrant, but this is certainly not the case in general. In higher dimensional TLNs, $\vec{x}_{bs}$ frequently lies outside of the positive orthant (Fig 6.2). This presents a challenge as it is nonsensical to have negative firing rates.

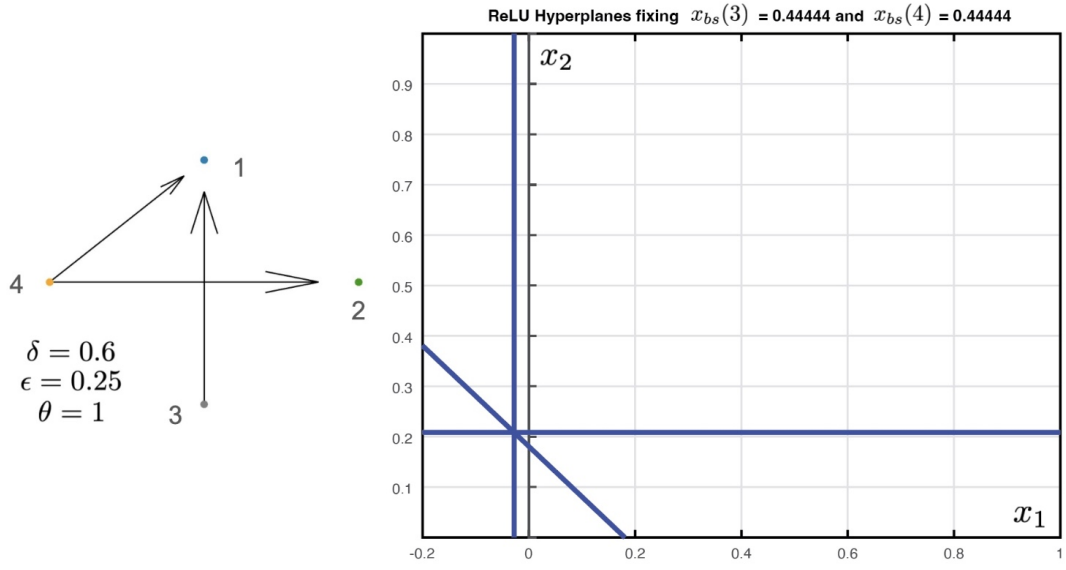


Figure 6.2. Unbalanced CTLN. For the depicted graph and parameter values, the associated CTLN is unbalanced with $x_{bs}(1)$ having a negative value. This is biologically nonsensical as the state variables are meant to represent firing rates of neurons.

Definition 15. *A TLN is said to be* ***balanced*** *if the point* $x_{bs} = -W^{-1}\vec{\theta}$ *lies in the positive orthant.*

Our goal in this chapter is twofold. First, we want to determine when we can use the balanced state as a reasonable initial condition. In other words, we want to determine when a CTLN is balanced. Our second goal will be to better understand to what attractor a trajectory beginning at the balanced state converges. We prove that there exists a sufficient condition for a CTLN to be balanced derived from the maximum in-degree of a graph. Employing again localized path polynomials, we find a sharper result for CTLNs derived from DAGs and use it to show that there exist graphs such that their CTLNs are always balanced, regardless of the choice of parameters. We then return to the question of basins of attraction, presenting an algorithm which aims to predict within which basin the balanced state trajectory lies.

6.1 Balanced CTLNs

We begin with a very general theorem to determine that a CTLN is balanced that works for any CTLN with no assumptions on graph structure. The proof for this result makes use of Farkas' Lemma which we restate here in a slightly adapted form.

Lemma 10 (Farkas' Lemma: Theorem 4.6 in [36])**.** *Let* $W \in \mathbb{R}^{m\times n}$ *and* $\vec{b} \in \mathbb{R}^m$*. Then exactly one of the following two assertions is true:*

1. *There exists an* $\vec{x} \in \mathbb{R}^n$ *such that* $W\vec{x} = \vec{b}$ *and* $\vec{x} \geq 0$*.*
2. *There exists a* $\vec{y} \in \mathbb{R}^m$ *such that* $W^T\vec{y} \geq 0$ *and* $\vec{b}\cdot\vec{y} < 0$*.*

Using Farkas' Lemma with $\vec{b} = -\theta\mathbb{1}$, we obtain the following result:

Theorem 7. *(Balanced State Theorem) Let* G *be a directed graph with maximum in-degree* $d_{\max}$*. Any CTLN satisfying:*

$$\frac{\varepsilon+\delta}{1+\delta} \leq \frac{1}{d_{\max}}$$

associated with G *is balanced.*

Proof. The CTLN has a balanced state if alternative (1) of Lemma 10 holds true for $\vec{b} = -\theta\mathbb{1}$. We will assume alternative (2) of Lemma 10 and aim for a contradiction. We will show that if $d_{\max} = 1$ or if $d_{\max} > 1$ and $\delta \leq -1 + \sqrt{1 + \dfrac{1}{d_{\max}-1}}$, there does not exist $\vec{y} \in \mathbb{R}^n$ satisfying:

a. $-\theta \mathbb{1} \cdot \vec{y} < 0$

b. $(W^T \vec{y})_i \geq 0$, $\forall i \in [n]$

Dividing through by $-\theta$, (a) can be rewritten as:

a. $\sum_{i=1}^{n} y_i > 0$

We will show that if (a) is true, (b) is contradicted.

Since (b) requires the inequality to hold $\forall i \in [n]$, it would follow that $\forall S \subseteq [n]$:

$$\sum_{i \in S} (W^T \vec{y})_i \geq 0$$

Recall that $W = (-1-\delta)\mathbb{1}\mathbb{1}^T + (1+\delta)I + (\varepsilon+\delta)A$ where A is the adjacency matrix of G. It follows that $W^T = (-1-\delta)\mathbb{1}\mathbb{1}^T + (1+\delta)I + (\varepsilon+\delta)A^T$.

This allows for the above inequality to be rewritten as:

$$\sum_{i \in S} (W^T \vec{y})_i = (-1-\delta)|S| \sum_{i=1}^{n} y_i + (1+\delta) \sum_{i \in S} y_i + (\varepsilon+\delta) \sum_{i=1}^{n} d_i^S y_i \geq 0$$

where $d_i^S = |\{j \in S | j \to i\}|$

Dividing by $(-1-\delta)$ on both sides yields the equivalent:

$$|S| \sum_{i=1}^{n} y_i - \sum_{i \in S} y_i - \alpha \sum_{i=1}^{n} d_i^S y_i \leq 0$$

where $\alpha = \dfrac{\varepsilon+\delta}{1+\delta}$.

Fix $\vec{y}$ and construct the sets $P = \{i | y_i \geq 0\}$ and $N = \{j | y_j < 0\}$. Observe that $P \sqcup N = [n]$ and that since $\sum_{i=1}^{n} y_i > 0$, $P \neq \emptyset$.

Then we assume by way of contradiction that:

$$|N| \sum_{i=1}^{n} y_i - \sum_{i \in N} y_i - \alpha \sum_{i=1}^{n} d_i^N y_i \leq 0$$

By decomposing $[n]$ into $P \sqcup N$, this is equivalent to:

$$|N| \sum_{i \in P} y_i - \alpha \sum_{i \in P} d_i^N y_i + (|N|-1) \sum_{j \in N} y_j - \alpha \sum_{j \in N} d_j^N y_j \leq 0$$

Since $0 \leq d_i^N \leq \min(d_{\max}, |N|) \leq |N|$, it follows that:

$$(|N| - \alpha \min(d_{\max}, |N|)) \sum_{i \in P} y_i + (|N|-1) \sum_{j \in N} y_j \leq 0$$

Now we show the contradiction. Taking (a) to be true, $\sum_{i=1}^{n} y_i = \sum_{i \in P} y_i + \sum_{i \in N} y_j > 0$.

Then it follows that for $A > 0$ and $B \geq 0$ s.t. $A \geq B$, it must also be true that $A\sum_{i\in P} y_i + B\sum_{i\in N} y_j > 0$.

Clearly $|N| - 1 \geq 0$ to avoid a trivial contradiction, and, since $\alpha < 1$, it follows that $|N| - \alpha\min(d_{\max}, |N|) \geq |N|(1-\alpha) > 0$. So, if $|N| - \alpha\min(d_{\max}, |N|) \geq |N| - 1$, there is a contradiction.

If $d_{\max} = 1$, this is always true as $|N| - \alpha\min(1, |N|) \geq |N| - \alpha > |N| - 1$.

If $d_{\max} > 1$, then the contradiction could potentially be avoided if $|N| - \alpha\min(d_{\max}, |N|) < |N| - 1$ i.e. if $\dfrac{1}{\min(d_{\max}, |N|)} < \alpha$. However, $\dfrac{1}{d_{\max}} \leq \dfrac{1}{\min(d_{\max}, |N|)}$ and, by assumption, $\alpha \leq \dfrac{1}{d_{\max}}$. □

Corollary 8. *Let G be a directed graph with maximum in-degree $d_{\max}$. If $d_{\max} = 1$, then any CTLN associated with G has a balanced state. If $d_{\max} > 1$, then any CTLN satisfying:*

$$\delta \leq -1 + \sqrt{1 + \frac{1}{d_{\max} - 1}}$$

associated with G has a balanced state.

Proof. Again, If $d_{\max} > 1$, then the contradiction is avoided if $|N| - \alpha\min(d_{\max}, |N|) < |N| - 1$ i.e. if $\dfrac{1}{\min(d_{\max}, |N|)} < \alpha$. But, $\alpha = \dfrac{\varepsilon + \delta}{1 + \delta} < 1 - \dfrac{1}{(1+\delta)^2}$ and $\dfrac{1}{\min(d_{\max}, |N|)} < 1 - \dfrac{1}{(1+\delta)^2}$ only if:

$$\delta > -1 + \sqrt{1 + \frac{1}{\min(d_{\max}, |N|) - 1}} > -1 + \sqrt{1 + \frac{1}{d_{\max} - 1}}$$

But this contradicts the assumption that $\delta \leq -1 + \sqrt{1 + \dfrac{1}{d_{\max} - 1}}$ in the case of $d_{\max} > 1$. □

Corollary 9. *If G is a directed graph with number of sinks $s > 0$, then the CTLN associated with G is balanced for $\delta \leq -1 + \sqrt{1 + \dfrac{1}{n - s - 1}}$.*

Proof. If a graph G of size n has s sinks, then, since the sinks have out-degree 0, $d_{\max} \leq n - s$. □

There are some interesting takeaways from these results. The most important of these is that there is no graph such that associated CTLNs are not balanced for sufficiently weak inhibition. We can easily look at a graph and devise a choice of ε and δ such that

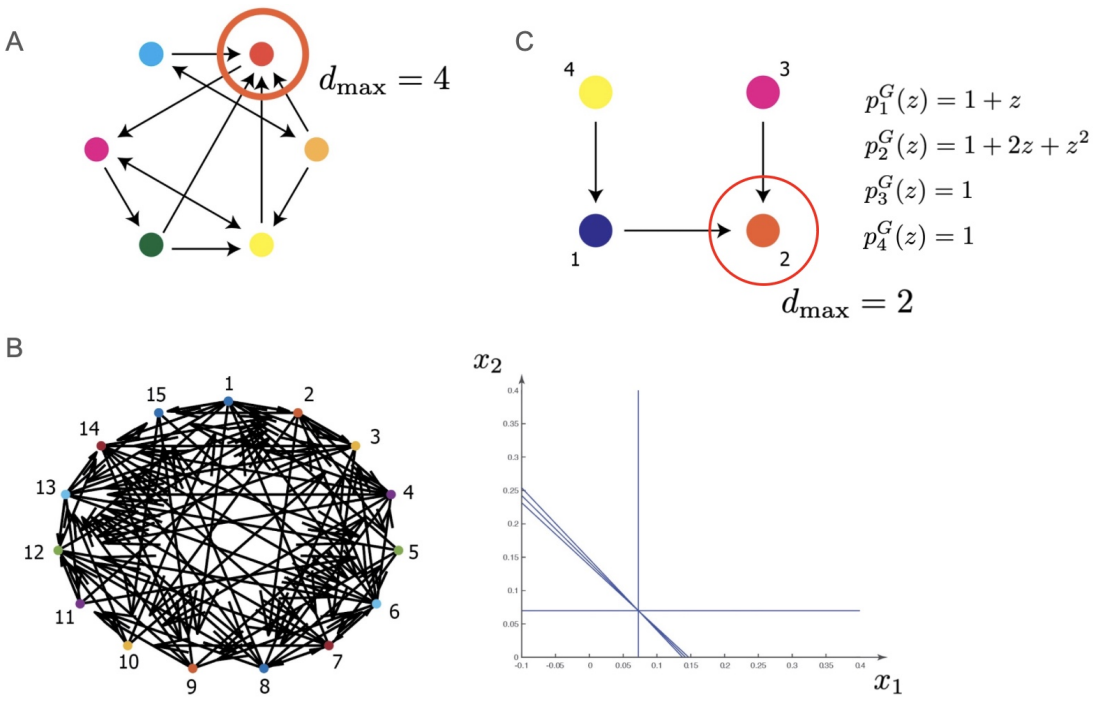


Figure 6.3. Balanced states of CTLNs. (A) As $d_{\max} = 4$ in this graph, as associated CTLN is balanced for $\frac{\varepsilon+\delta}{1+\delta} \leq \frac{1}{4}$. (B) Any directed graph of 15 neurons will have an associated balanced CTLN for $\frac{\varepsilon+\delta}{1+\delta} \leq \frac{1}{14}$. (C) Theorem 7 is a sufficient but not a necessary condition. Using Theorem 8 we know that any CTLN associated with this DAG is balanced.

the CTLN is balanced. In the case of the example depicted in Fig 6.3A, any choice of parameters such that $\frac{\varepsilon+\delta}{1+\delta} \leq 0.25$ will suffice. Even if we did not know a graph's structure, we could use the fact that $d_{\max} \leq n-1$ and quite generally choose ε, δ such that $\frac{\varepsilon+\delta}{1+\delta} \leq \frac{1}{n-1}$. In Fig 6.3B we have a random graph of 15 neurons, fairly dense with edges, and have chosen $\delta = 0.035$ and $\varepsilon = 0.03$ satisfying $\frac{\varepsilon+\delta}{1+\delta} \leq \frac{1}{14}$. The CTLN is balanced as can be seen in the balanced state cross-section of the state space.

While these results are useful and general, a natural concern presents itself. As the network gets larger and the probability of excitatory connections remains constant, the expected in-degree scales linearly and $d_{\max}$ should rise. We would need extremely weak inhibition to ensure balance in large networks.

The conditions derived from the above results are sufficient but not always necessary. Since our interest is mainly in CTLNs derived from DAGs, can we do any better in that case? In fact we can!

Lemma 11. *Let G be a DAG. Letting $\beta = \dfrac{-\varepsilon - \delta}{1+\delta}$, the point $x_{bs} = -W^{-1}(\theta \mathbb{1})$ of an associated CTLN is:*

$$(x_{bs})_j = \frac{p_j^G(\beta)}{-1 + \sum_{i=1}^n p_i^G(\beta)} \left(\frac{\theta}{1+\delta} \right) \text{ for } j \in [n]$$

Proof. We need to solve $W\vec{x} = -\theta\mathbb{1}$. This is equivalent to $(W-(1+\delta)I)\vec{x} = -\theta\mathbb{1} - (1+\delta)\vec{x}$.

Notice that $\dfrac{-1}{1+\delta}(W - (1+\delta)I) = \mathbb{1}\mathbb{1}^T + cA = \dfrac{\theta}{1+\delta}\mathbb{1} + \vec{x}$ where $c = \dfrac{-\varepsilon-\delta}{1+\delta}$ and A is the adjacency matrix of G. The result follows by applying Lemma 7. □

The power of using the Lemma 7 here is that the question of whether or not a CTLN is balanced can be transformed from a general problem of matrix inversion and can instead be approached via tracking the sign changes of the path polynomials i.e. a problem of root detection.

Definition 16. *Define $r_G = \max\{z \in \mathbb{R} \mid \exists i \in V(G) \text{ s.t. } p_i^G(z) = 0\}$ to be the greatest real root for the path polynomials of G. By convention, if $\{p_i^G(z)\}_{i \in V(G)}$ has no real roots we say $r_G = -\infty$.*

Remark 6. *Since $\{p_i^G(z)\}_{i \in V(G)}$ are polynomials with positive coefficients, $r_G < 0$.*

The relationship between the existence of the balanced state and the roots of the path polynomials is captured in the following result.

Theorem 8. *(Balanced State Theorem for DAGs) Let G be a DAG of size $n \geq 2$. A CTLN associated with G has a balanced state if*

$$\frac{\varepsilon + \delta}{1+\delta} \leq -r_G$$

.

Proof. From Lemma 11, It suffices to show that:

$$\frac{p_j^G(\beta)}{-1 + \sum_{k=1}^n p_k^G(\beta)} > 0$$

for each $j \in [n]$.

Define $p_0^G(z) = -1 + \sum_{k=1}^n p_k^G(z)$.

For $n \geq 2$, these polynomials have positive coefficients and so their greatest real root, if any real roots exist, are negative. Call them $r_0, r_1, ..., r_n$ respectively with $r_i = -\infty$ if no real roots exist for $p_i^G(z)$. Let $K = \sup_{i \in \{0\} \cup [n]} r_i$. Since $p_0^G(z) = -p_s^G(z) + \sum_{k=1}^n p_k^G(z) =$

$\sum_{i\neq s}^{n} p_i^G(z)$, where x_s is any one of the source neurons, $r_0 < r_G$ as one of the path polynomials must change sign before $p_0^G(z)$ has a root. So, $K = \sup_{i\in[n]} r_i = r_G$. Then, for $\alpha > r_G$, we have $p_0^G(\alpha), p_1^G(\alpha), ..., p_n^G(\alpha) > 0$. So, if $\beta = \dfrac{-\varepsilon-\delta}{1+\delta} \geq r_G$, the result holds. □

Corollary 10. *Let G be a DAG of size $n \geq 2$. If $r_G \leq -1$, then any CTLN associated with G has a balanced state. If $r_G > -1$, then a CTLN associated with G has a balanced state if*

$$\delta < -1 + \frac{1}{\sqrt{r_G + 1}}$$

.

Proof. Note that:

$$\frac{-\varepsilon-\delta}{1+\delta} > \frac{-\dfrac{\delta}{1+\delta} - \delta}{1+\delta} = \frac{-\delta^2-2\delta-1}{\delta^2+2\delta+1} + \frac{1}{(1+\delta)^2} = -1 + \frac{1}{(1+\delta)^2}$$

so it suffices to require that:

$$-1 + \frac{1}{(1+\delta)^2} > r_G$$

Some rearranging yields:

$$1 > (r_G + 1)(1+\delta)^2$$

This yields two cases:

Case 1: $r_G \leq -1$

In this case, $r_G + 1 \leq 0$ and the inequality is always true.

Case 2: $r_G > -1$

In this case, the inequality $1 > (r_G + 1)(1+\delta)^2$ produces the condition:

$$\delta < -1 + \frac{1}{\sqrt{r_G + 1}}$$

□

Consider the example depicted in Fig 6.4. The use of localized path polynomials allowed us to, without much difficulty, find conditions on CTLN balance, but also note that it was not any better than that obtained by Theorem 7.

That said, this result is indeed stronger and so let us pause briefly to look at an example illustrating the added strength of the above theorem. We will first evaluate the

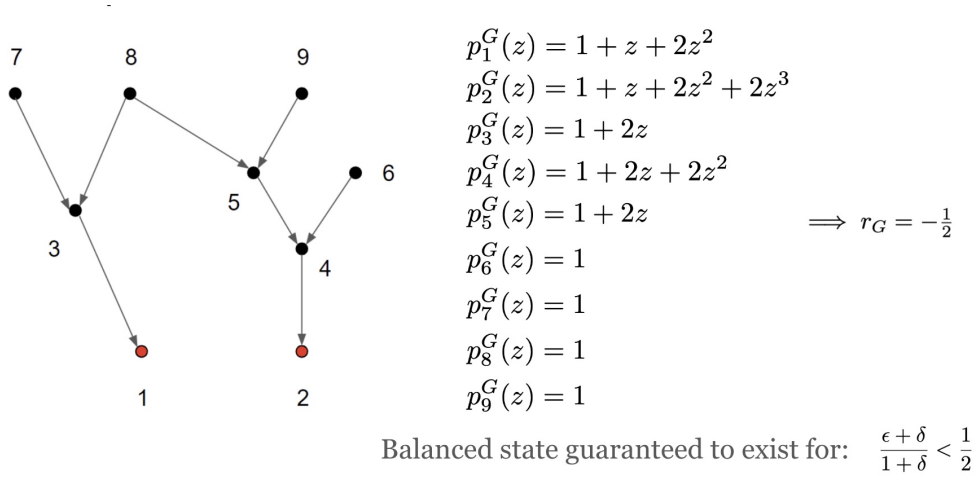


Figure 6.4. Localized path polynomials and CTLN balance. Each vertex of the DAG contributes a localized path polynomial and taking the greatest root r_G among them gives us a condition on CTLN balance.

following example using the original, general Balanced State Theorem and then with the new Theorem 8.

In the DAG depicted in Fig 6.3C, $d_{\max} = 2$, so Theorem 7 guarantees that an associated CTLN will be balanced for $\dfrac{\varepsilon+\delta}{1+\delta} \leq \dfrac{1}{2}$.

Now, applying the new theorem, we find the following localized path polynomials

$$p_1^G(z) = z + 1$$

$$p_2^G(z) = z^2 + 2z + 1 = (z+1)^2$$

$$p_3^G(z) = 1$$

$$p_4^G(z) = 1$$

It is clear to see for these polynomials that we have $r_G = 1$. Thus, we find that a CTLN derived from this DAG is in fact always balanced!

This begs another question: for what graphs are all associated CTLNs balanced?

Definition 17. *A directed graph G is said to be* ***balanced*** *if all CTLNs derived from G are balanced.*

One class of graphs which are balanced is those which are uniform in-degree i.e. all vertices have the same in-degree (Fig 6.5A).

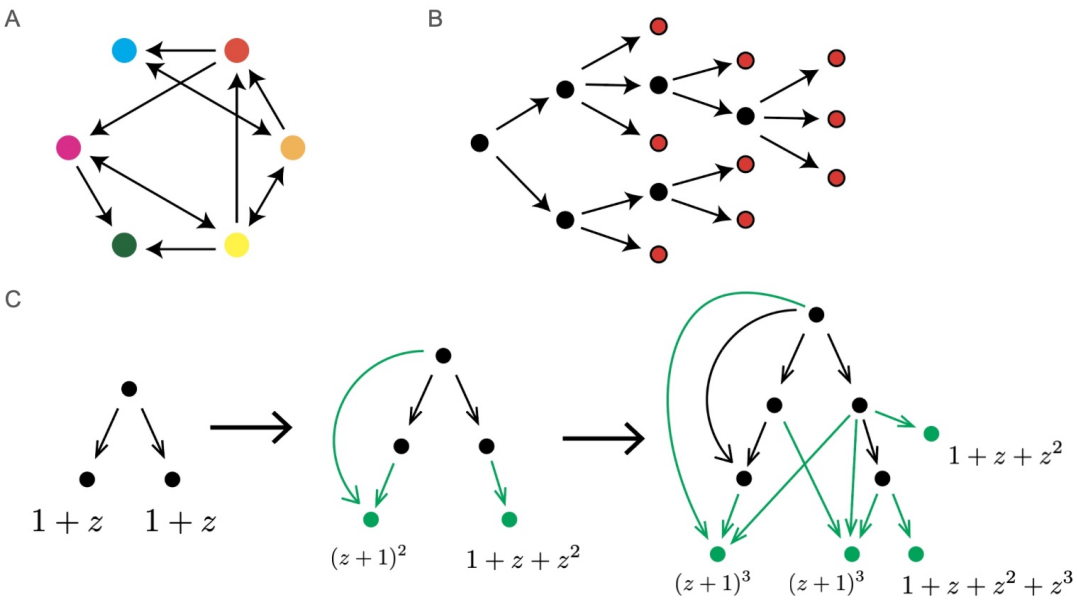


Figure 6.5. Balanced graphs. (A) Uniform in-degree graphs balanced. This graph has uniform in-degree 2. (B) Out-trees are balanced graphs. (C) Large balanced graphs can be constructed by adding vertices with localized path polynomials known to have no roots in $(-1, 0)$. As no edges are being drawn back to the extant vertices, their localized path polynomials remain unchanged.

Theorem 9. *(UFD Theorem for Balanced States) Let G be a uniform in-degree directed graph. Then any associated CTLN is balanced.*

Proof. Let W be the weight matrix for a CTLN associated with G.

Let d be the uniform in-degree of the vertices. Observe that the row sum for each row of W will be uniform and equal to $d(-1+\varepsilon)+(n-1-d)(-1-\delta)$.

Then, construct:

$$x_{bs} = \frac{-\theta \mathbb{1}}{d(-1+\varepsilon)+(n-1-d)(-1-\delta)}.$$

Since $d \leq n-1$, both terms of the bottom sum are negative. As θ is positive $x_{bs} > 0$. Also observe that since the denominator is the uniform rom sum, $Wx_{bs} = -\vec{\theta}$. Thus, x_{bs} lies in the positive orthant. □

To be uniform in-degree is a very strong condition on a graph, but fortunately our results suggest other classes of balanced graphs. Theorem 7 suggests that any graph such that $d_{\max} \leq 1$ is balanced. While this also seems quite narrow, it includes an interesting

class of graphs. Out-trees (Fig 6.5B), directed trees with in-degree at most 1, satisfy this condition and so are balanced.

Corollary 11. *Let G be a directed graph which is an out-tree. Then, any CTLN associated with G is balanced.*

There have been studies which propose decision-making as potentially a sequence of binary choices [37] which suggests a DAG network architecture. All out-trees are DAGs and would fit well with this paradigm.

Concentrating further on DAGs, an additional consequence of the path polynomial formulation of balanced states is that it gives us the tools for generating balanced DAGs.

Corollary 12. *Let G be a DAG such that the maximum path length in G is 2. Then, the CTLN associated with G has a parameter independent balanced state if and only if each vertex with maximum path length 1 has in-degree 1 and each vertex i with max path length 2 has incoming paths obeying* $n_1^i < 2\sqrt{n_2^i}$ *or, if* $n_2^i = 1$, $n_1^i \leq 2$.

Proof. The path polynomials of G are of the form $p_i^G(z) = 1$, $p_i^G(z) = n_1^i z + 1$, and $p_i^G(z) = n_2^i z^2 + n_1^i + 1$. Note that $p_i^G(z) = 1$ trivially has no roots in $(-1, 0)$ and $p_i^G(z) = n_1^i z + 1$ has no roots in the interval if and only if $n_1^i = 1$.

Since the constant term of $p_i^G(z) = n_2^i z^2 + n_1^i + 1$ is 1, the roots of the polynomial, r_1 and r_2, are such that $r_1 r_2 = \dfrac{1}{n_2^i} \leq 1$. This leads to two cases.

Case 1: $n_2^i = 1$

Then, if $n_2^i = 1$ the polynomial has a root in $(-1, 0)$ if and only if both roots are real and distinct. Then, there are no roots in the interval if and only if there is a repeated root or if there are imaginary roots. What is required then is that the discriminant be non-positive i.e. $(n_1^i)^2 - 4 \leq 0$, which is rearranged to $n_1^i \leq 2$.

Case 2: $n_2^i > 1$

If $n_2^i > 1$, then the polynomial has a root in $(-1, 0)$ if it has a real root. This is avoided if the roots are imaginary i.e. if $(n_1^i)^2 - 4n_2^i < 0$, which is rearranged to $n_1^i < 2\sqrt{n_2^i}$. □

If one wished to find similar conditions for parameter independent balance for more complex networks, there exists a vast literature around root detection of polynomials in real intervals, such as Sturm's Theorem (refer to section 8.4 of [38] for a discussion of Sturm's Theorem), which could potentially be exploited to these ends.

Additionally, by taking localized path polynomials which are known to not have roots in the interval $(-1, 0)$ we can quickly generate balanced DAGs. Two classes of polynomials which are useful in this enterprise are finite geometric sums of the form

$p(z) = \sum_{i=0}^{m} z^k$ and the binomial expansions $p(z) = (z+1)^m$. Since localized path polynomials only take into account incoming paths, larger balanced DAGs can be built up from smaller ones. In Fig 6.5C, we start by taking a balanced graph of maximum path length 1 and progressively adding neurons in lower and lower topological layers, building larger and larger graphs, all of which are balanced.

6.2 Balanced States as Initial Conditions

Now that we have a detailed understanding of balanced states in CTLNs, let us discuss their relevance to the problem at hand. The particular trajectory is challenging to follow analytically. It is not hard to see that for any TLN, the first chamber a trajectory beginning at $\vec{x}_{bs}$ enters is the full support chamber $R_{[n]}$. This means that even the beginning dynamics are governed by the most complex chamber, making analytical results difficult to obtain, while still being theoretically possible using Theorem 6 and piecing the trajectory across the chambers R_σ. What we will now discuss is an imperfect computational algorithm with which we have had considerable success at predicting the attractor to which a trajectory beginning at the balanced state converges.

Definition 18. $G^i = G|_\sigma$ *where*

$$\sigma = \{j \in [n] \mid \textit{there does not exist a directed path of length } i \textit{ starting from } j\}$$

Notice that $G^i \subseteq G^{i+1}$, G^1 is the set of sinks, and $G^{m+1} = G$, where m is the maximum path length in G, and so this layering of the DAG is a filtration.

Definition 19. *For a vertex i of a DAG G, let d_k^i be the in-degree of vertex i in the subgraph G^k. We call this the* ***k-th filtered in-degree of i****.*

The algorithm begins by assembling a list of the sinks of the DAG as the possible attractor candidates. We then consider G^2 and eliminate any sink from the list if its second filtered in-degree is smaller than that of any of the other sinks. We then repeat the process, iterating through G^k and comparing values of d_k^i until we are left with one sink, which will be our prediction. If multiple sinks survive all the way through, the algorithm deems the case inconclusive with the correct fixed point lying among the remaining candidates.

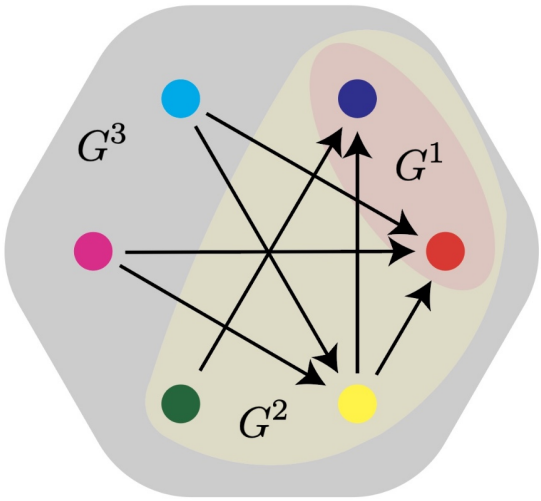


Figure 6.6. G^i filtration of a DAG. An example of a DAG showing the construction of the subgraph filtration $\{G^i\}_{i=1}^3$.

Algorithm 2 Balanced State Attractor Prediction

$C = \{i \in G |\ i \text{ is a sink of } G\}$
for $k \leftarrow 2$ to n **do**
    $D = \{d_k^i | i \in C\}$
    $C \leftarrow \{i \in C | d_k^i = \max(D)\}$
end for
if length$(C) = 1$ **then**
    return C
end if

Fig 6.7A-C illustrate this process for three different DAGs, progressing through the filtration until reaching a prediction. Comparing each of these with the actual result, we see accurate prediction in each case. However, there are cases where this algorithm can make an incorrect prediction. This begs the question: how often does this happen and to what extent does it depend on the parameters ε and δ? If we test a set of DAGs at various grid points in parameter space we find the results in Figure 6.8.

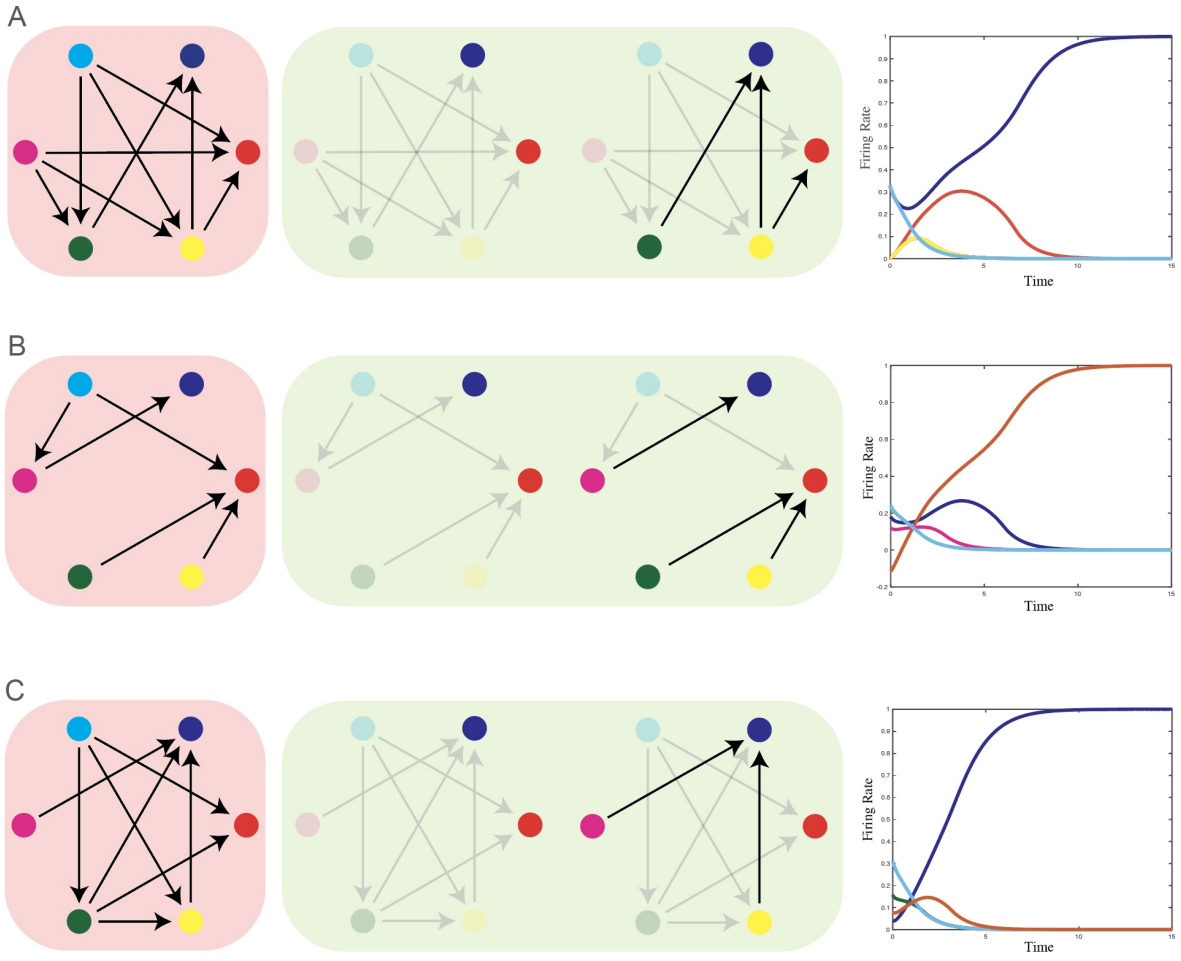


Figure 6.7. Balanced state attractor prediction. An application of the prediction algorithm to three cases with parameters $\delta = 0.5$, $\varepsilon = 0.24$, and $\theta = 1$. On the left are the graphs, in the center is the filtration of the algorithm, stopping when one of the sink filtered in-degrees is dominant, and on the right is the numerical simulation of the firing rates, colored according to the neuron, confirming the results. Notably, (B) is even an unbalanced CTLN and yet we have success with the prediction.

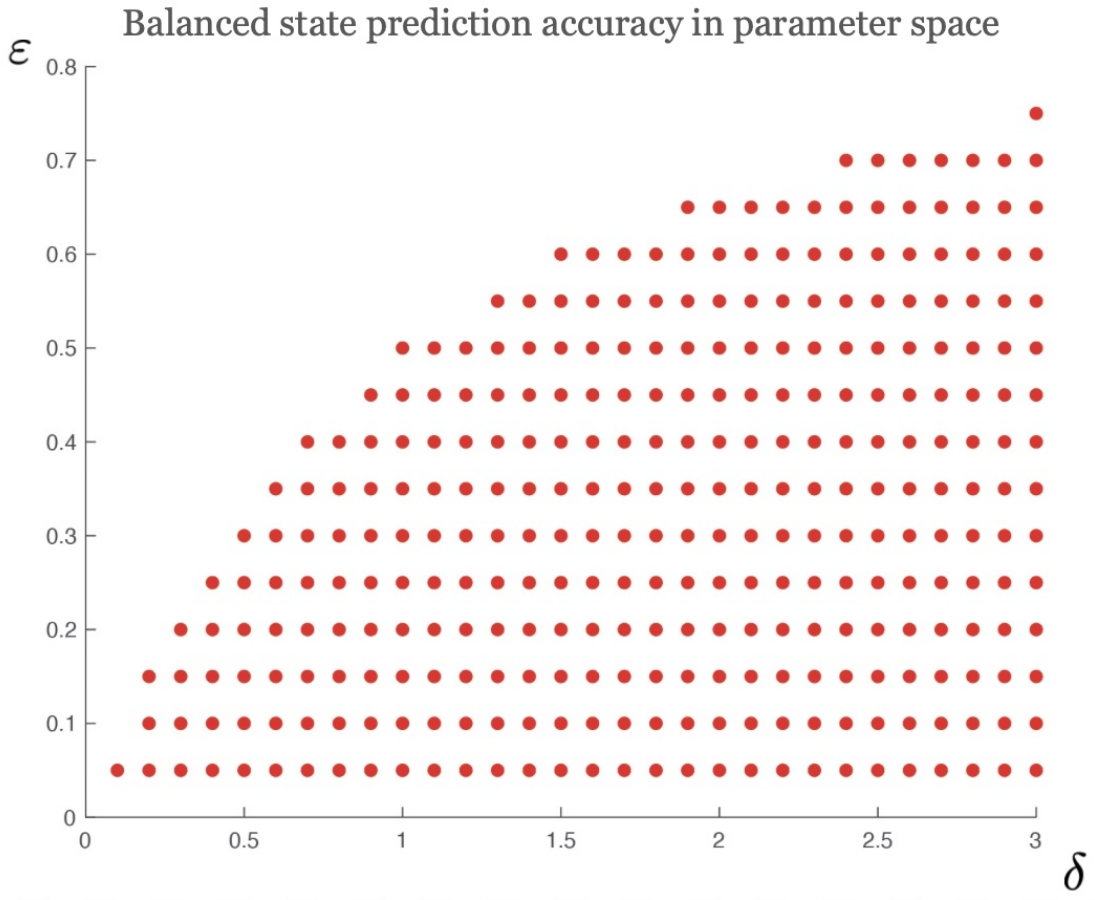


Figure 6.8. Balanced state attractor prediction accuracy. A list of 1000 DAG (n=8) CTLN balanced state trajectories were numerically tested for different parameter values and compared with the prediction from the balanced state attractor prediction algorithm. The color intensity of the red point in parameter space marks the percentage correct out of the predicted set of the 1000 CTLNs. We see minimal variation in color because accuracy was similar for various parameter values at approx. 92%.

Chapter 7
Heterogeneous DAG CTLNs

We conclude this thesis with a primarily theoretical chapter that will give us tools to generalize some of our earlier results beyond strict CTLNs and also to resolve a shortcoming of our analytical solutions in DAG CTLNs. We begin by noting that the CTLN conditions are quite strict and it is their strictness that makes the class so tractable. The easiest condition to weaken is that of external drive symmetry, i.e. $\vec{\theta} = \theta\mathbb{1}$. Instead, we will allow $\vec{\theta} > 0$ to be an arbitrary positive vector. We will refer to this as a *heterogeneous CTLN* (hCTLN).

The reason this is the easiest condition to weaken is that doing so does not affect the eigenvalues or eigenvectors of the $(-I + W)|_\sigma$ matrices. This means that the general homogeneous solution within the chamber is intact. What is changed however is the hyperplane arrangement and the particular solutions of L_σ i.e. the fixed points. A consequence of this is that Theorem 1 no longer applies and we are not even sure what the fixed points of the system are anymore. What theory can we develop about hCTLNs?

Fortunately for hCTLNs derived from DAGs, many of the results we had for DAG CTLNs in Chapter 7 can be generalized. To develop them, we need to first introduce the *pinned path polynomials*.

Definition 20. *For a DAG G of size n, let the vertices be numbered from 1 to n i.e. $V(G) = [n]$. Then, the **i,j pinned path polynomial**, $p^G_{i,j}(z)$, is defined to be:*

$$p^G_{i,j}(z) = \delta_{ij} + \sum_{k=1}^{n} n^{i,j}_k z^k$$

where $n^{i,j}_k$ is the number of paths from j to i of length k (finite because G is acyclic) and δ_{ij} is the Kronecker delta. Since G is acyclic, $deg(p^G_{i,j}(z))$ is finite.

Definition 21. *Let G be a labeled directed graph with adjacency matrix A. Then, G^T is*

the labeled directed graph induced by the adjacency matrix A^T.

Remark 7. *Note that $\sum_{j=1}^{n} p_{i,j}^{G}(z) = p_i^G(z)$ and $\sum_{i=1}^{n} p_{i,j}^{G}(z) = p_j^{G^T}(z)$. Also, $\sum_{i=1}^{n} p_i^G(z) = \sum_{j=1}^{n} p_j^{G^T}(z)$.*

Using this new construction we can both strengthen Theorem 6 and generalize it to the case of hCTLNs. We will also develop an appropriate generalization for Lemma 11.

7.1 Virtual Fixed Points in hCTLNs

The component linear systems L_σ will have the same homogenous solution as $(-I + W)|_\sigma$ remains unchanged, but the particular solution is now different as we have the heterogenous external input vector $\vec{\theta}|_\sigma$.

Using pinned path polynomials we have a generalization of Lemma 7.

Lemma 12. *Let B be a matrix derived from a DAG G and adjacency matrix A such that:*

$$B = \mathbb{1}\mathbb{1}^T + cA$$

Then the solution to the following linear system:

$$B\vec{x} = \vec{\phi} + a\vec{x}$$

can be written as:

$$x_j = -\sum_{i=1}^{n} \Phi_i p_{j,i}^{G}\left(\frac{c}{a}\right) + p_j^G\left(\frac{c}{a}\right)\Gamma$$

where $\Gamma = \left(\dfrac{\sum_{i=1}^{n} \Phi_i p_i^{G^T}\left(\frac{c}{a}\right)}{-a + \sum_{i=1}^{n} p_i^G\left(\frac{c}{a}\right)}\right)$ *and* $\Phi_i = \dfrac{\phi_i}{a}$.

Proof. The proof will be similar to that of Lemma 7. We will again proceed by simply showing that, $\forall i \in [n]$, the specified $\vec{x}$ satisfies $\sum_{j=1}^{n} B_{ij}x_j = \phi_i + ax_i$. For compactness of notation, we will define $\kappa = \dfrac{c}{a}$.

Recall that:

$$\sum_{j=1}^{n} B_{ij}x_j = \sum_{j=1}^{n} x_j + c\sum_{j\to i} x_j.$$

Inserting our proposed solution:

$$\sum_{j=1}^{n} x_j = \sum_{j=1}^{n} \left(-\sum_{\ell=1}^{n} \Phi_\ell p_{j,\ell}^G(\kappa) + p_j^G(\kappa)\,\Gamma \right) = -\sum_{\ell=1}^{n} \Phi_\ell \sum_{j=1}^{n} p_{j,\ell}^G(\kappa) + \Gamma \sum_{j=1}^{n} p_j^G(\kappa)$$

$$= -\sum_{\ell=1}^{n} \Phi_\ell p_\ell^{G^T}(\kappa) + \Gamma \sum_{j=1}^{n} p_j^G(\kappa)$$

$$= \frac{a\sum_{\ell=1}^{n} \Phi_\ell p_\ell^{G^T}(\kappa) - \left(\sum_{j=1}^{n} p_j^G(\kappa)\right)\left(\sum_{\ell=1}^{n} \Phi_\ell p_\ell^{G^T}(\kappa)\right) + \left(\sum_{j=1}^{n} p_j^G(\kappa)\right)\left(\sum_{\ell=1}^{n} \Phi_\ell p_i^{G^T}(\kappa)\right)}{-a + \sum_{\ell=1}^{n} p_\ell^G(\kappa)}.$$

But after simplifying we notice that this is equal to $a\Gamma$. From this we see that:

$$\sum_{j=1}^{n} B_{ij}x_j = a\Gamma + c\sum_{j\to i}\left(-\sum_{\ell=1}^{n}\Phi_\ell p_{j,\ell}^G(\kappa) + p_j^G(\kappa)\Gamma\right) = a\Gamma - c\sum_{\ell=1}^{n}\Phi_\ell \sum_{j\to i} p_{j,\ell}^G(\kappa) + c\Gamma\sum_{j\to i} p_j^G(\kappa).$$

Recall from previous proofs that $c\sum_{j\to i} p_j^G(\kappa) = a(p_i^G(\kappa) - 1)$. So, we can conclude:

$$\sum_{j=1}^{n} B_{ij}x_j = a\Gamma - c\sum_{\ell=1}^{n}\Phi_\ell \sum_{j\to i} p_{j,\ell}^G(\kappa) - a\Gamma + ap_i^G(\kappa)\Gamma = -c\sum_{\ell=1}^{n}\Phi_\ell \sum_{j\to i} p_{j,\ell}^G(\kappa) + ap_i^G(\kappa)\Gamma.$$

Now notice similarly that $c\sum_{j\to i} p_{j,\ell}^G(\kappa) = ap_{i,\ell}^G(\kappa)$. Then:

$$-c\sum_{\ell=1}^{n}\Phi_\ell \sum_{j\to i} p_{j,\ell}^G(\kappa) = a\Phi_i p_{i,i}^G(\kappa) - a\sum_{\ell=1}^{n}\Phi_\ell p_{i,\ell}^G(\kappa) = \phi_i - a\sum_{\ell=1}^{n}\Phi_\ell p_{i,\ell}^G(\kappa).$$

Thus,

$$\sum_{j=1}^{n} B_{ij}x_j = \phi_i - a\sum_{\ell=1}^{n}\Phi_\ell p_{i,\ell}^G(\kappa) + ap_i^G(\kappa)\Gamma = \phi_i + ax_i.$$

□

This allows us to generalize the results on the fixed points of L_σ analogously as we did with Proposition 9.

Proposition 13. *Let G be a DAG of size n, $\sigma \subset [n]$, and $\beta = \frac{-\epsilon-\delta}{\delta}$. Then, for an hCTLN associated with G, the fixed point of L_σ is:*

$$(x^*_\sigma)_j = -\sum_{i\in\sigma} \Theta_i p^{G|_\sigma}_{j,i}(\beta) + p^{G|_\sigma}_j(\beta)\,\Gamma(\sigma),\ \forall j \in \sigma \text{ and } (x^*_\sigma)_j = 0,\ \forall j \notin \sigma.$$

$$\text{where } \Gamma(\sigma) = \left(\frac{\sum_{i\in\sigma}\Theta_i p_i^{(G|_\sigma)^T}(\beta)}{-\frac{\delta}{1+\delta} + \sum_{i\in\sigma} p_i^{G|_\sigma}(\beta)}\right) \text{ and } \Theta_i = \frac{\theta_i}{\delta}.$$

Proof. The proof is identical to that of Proposition 9 with the only change that we use Lemma 12 rather than Lemma 7. □

We have nearly reconstructed a version of Theorem 6 from Chapter 4 in the setting of hCTLNs. Still, there is one more piece of the puzzle and resolving it will fill in an oversight of our analysis of CTLNs as well.

7.2 Revisiting CTLNs

Recall a gap that was left in the statement of Theorem 6. For the linear systems L_σ, we did not provide the eigenvectors for the eigenvalue $\lambda = -1$ of multiplicity $n - |\sigma|$. At the time we said that this is something we would revisit. These eigenvectors can be obtained using Lemma 12.

Proposition 14. *Let G be a directed graph such that $|G| = n$ and let $\sigma \subset [n]$. Then, the matrix for the linear system L_σ has eigenvalue $\lambda = -1$ with algebraic multiplicity $n - |\sigma|$. Moreover there are $n - |\sigma|$ distinct eigenvectors of the form:*

$$\vec{v_j} = -e_j + \vec{g_j},\ j \notin \sigma$$

such that, if $k \in \sigma$:

$$(\vec{g_j})_k = -\sum_{j\not\to\ell\in\sigma} p^{G|_\sigma}_{k,\ell}(\alpha) + \frac{1-\varepsilon}{1+\delta}\sum_{j\to\ell\in\sigma} p^{G|_\sigma}_{k,\ell}(\alpha) + p^{G|_\sigma}_k(\alpha)\Gamma_j(\sigma)$$

$$\text{where } \Gamma_j(\sigma) = \frac{\sum_{j\not\to\ell\in\sigma}(1+\delta)p_\ell^{(G|_\sigma)^T}(\alpha) + \sum_{j\to\ell\in\sigma}(1-\varepsilon)p_\ell^{(G|_\sigma)^T}(\alpha)}{-(1+\delta) + (1+\delta)\sum_{\ell=1}^n p_\ell^{G|_\sigma}(\alpha)} \text{ and } \alpha = \frac{-\varepsilon-\delta}{1+\delta}$$

and, if $k \notin \sigma$, $(\vec{g_j})_k = 0$.

Proof. This proof will take a similar approach to that of Proposition 12, but with the notable modification of using Lemma 12 rather than the Sherman-Morrison Formula.

Without loss of generality number the vertices in σ to be $1, \ldots, k$ where $k = |\sigma|$. Then the matrix for L_σ is of the form:

$$B = \left[\begin{array}{ccc|ccc} -1 & \ldots & -1-\delta & w_{1,k+1} & \ldots & w_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ldots & \vdots \\ -1-\delta & \ldots & -1 & w_{k,k+1} & \ldots & w_{kn} \\ \hline 0 & \ldots & 0 & -1 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & -1 \end{array}\right]$$

where, taking $A|_\sigma$ as the adjacency matrix of $G|_\sigma$, the upper left block is of the form

$$(-I + W)|_\sigma = (-1-\delta)\mathbb{1}\mathbb{1}^T + \delta I + (\varepsilon + \delta)A|_\sigma.$$

Recalling Lemma 6, we know that $\lambda = -1$ has algebraic multiplicities $n - k$.

We find eigenvectors for $\lambda = -1$. We show that there are $n - k$ linearly independent eigenvectors by construction. We then take as an ansatz vectors of the form:

$$\vec{v_j} = \left[\begin{array}{c} g_1 \\ \vdots \\ g_k \\ \hline 0 \\ \vdots \\ -1 \\ \vdots \\ 0 \end{array}\right] = \left[\begin{array}{c} \vec{g} \\ \hline 0 \end{array}\right] - e_j$$

Then, we have:

$$B\vec{v_j} = \left[\begin{array}{c} ((-I + W)|_\sigma)\vec{g} - \vec{w}_{*j} \\ \hline 0 \end{array}\right] + e_j$$

where

$$\vec{w}_{*j} = \left[\begin{array}{c} w_{1j} \\ \vdots \\ w_{kj} \end{array}\right].$$

So, if $\vec{g}$ satisfies $((-I + W)|_\sigma)\vec{g} - \vec{w}_{*j} = -\vec{g}$ then $\vec{v_j}$ is an eigenvector. Rearranging, this system can be rewritten as:

$$((-I+W)|_\sigma)\vec{g}+I\vec{g}=\vec{w}_{*j} \implies ((-1-\delta)\mathbb{1}\mathbb{1}^T+(\varepsilon+\delta)A|_\sigma)\vec{g}=\vec{w}_{*j}-(1+\delta)\vec{g}$$

$$\implies (\mathbb{1}\mathbb{1}^T+\alpha A|_\sigma)\vec{g}=-\frac{\vec{w}_{*j}}{1+\delta}+\vec{g}.$$

where $\alpha=\dfrac{-\varepsilon-\delta}{1+\delta}$.
Applying Lemma 12 we obtain:

$$g_k=\sum_{\ell\in\sigma}\frac{w_{\ell j}}{1+\delta}p_{k,\ell}^{G|_\sigma}(\alpha)+p_k^{G|_\sigma}(\alpha)\Gamma_j(\sigma).$$

where $\Gamma_j(\sigma)=\dfrac{-\sum_{\ell\in\sigma}w_{\ell j}p_\ell^{(G|_\sigma)^T}(\alpha)}{-(1+\delta)+(1+\delta)\sum_{\ell=1}^n p_\ell^{G|_\sigma}(\alpha)}$
Then, using that $w_{\ell j}=-1-\delta$ if $j\not\to\ell$ and $w_{\ell j}=-1+\varepsilon$ if $j\to\ell$, we finally obtain:

$$g_k=-\sum_{j\not\to\ell\in\sigma}p_{k,\ell}^{G|_\sigma}(\alpha)+\frac{1-\varepsilon}{1+\delta}\sum_{j\to\ell\in\sigma}p_{k,\ell}^{G|_\sigma}(\alpha)+p_k^{G|_\sigma}(\alpha)\Gamma_j(\sigma)$$

where $\Gamma_j(\sigma)=\dfrac{\sum_{j\not\to\ell\in\sigma}(1+\delta)p_\ell^{(G|_\sigma)^T}(\alpha)+\sum_{j\to\ell\in\sigma}(1-\varepsilon)p_\ell^{(G|_\sigma)^T}(\alpha)}{-(1+\delta)+(1+\delta)\sum_{\ell=1}^n p_\ell^{G|_\sigma}(\alpha)}$.

□

Now we can restate Theorem 6 in a more complete and comprehensive form.

Theorem 6. *Let G be a DAG and let W be the weight matrix for an associated CTLN with parameters $\varepsilon,\delta,\theta$. Let $\sigma\subseteq[n]$ be such that $G|_\sigma$ is analytic, $(-I+W)|_\sigma$ is diagonalizable, and the polynomial:*

$$f(\lambda)=(-\lambda+\delta)^{m+1}-(1+\delta)(|\sigma|(-\lambda+\delta)^m+n_1^\sigma c(-\lambda+\delta)^{m-1}+...+n_m^\sigma c^m)$$

has distinct roots $\{\lambda_k\}_{k=1}^{m+1}$ where $n_{j>0}^\sigma$ is the number of paths of length j in $G|_\sigma$ and m is the maximum path length in $G|_\sigma$.

Then, the general solution of L_σ is of the following form:

$$\vec{x}(t)=\sum_{k=1}^{m+1}c_k\vec{p_\sigma}(\alpha_k)e^{\lambda_k t}+\sum_{(i,j)\in\widetilde{\mathrm{SE}}(G|_\sigma)}c_{(i,j)}(e_i-e_j)e^{\delta t}+\sum_{k\notin\sigma}c_k(-e_k+\vec{g_k})e^{-t}+\vec{p_\sigma}(\beta)\Gamma(\sigma)$$

where $n = |G|$, $\alpha_k = \dfrac{\varepsilon + \delta}{\lambda_k - \delta}$, $\beta = \dfrac{-\varepsilon - \delta}{\delta}$, *and* $\Gamma(\sigma) = \dfrac{\theta}{-\delta + (1+\delta)\sum_{j\in\sigma} p_j^{G|_\sigma}(\beta)}$.
Additionally, $\vec{g_j}$ *is defined to be:*

$$(\vec{g_j})_k = -\sum_{j\not\to\ell\in\sigma} p_{k,\ell}^{G|_\sigma}(\gamma) + \frac{1-\varepsilon}{1+\delta}\sum_{j\to\ell\in\sigma} p_{k,\ell}^{G|_\sigma}(\gamma) + p_k^{G|_\sigma}(\gamma)\Gamma_j(\sigma),\ \forall k \in \sigma$$

where $\Gamma_j(\sigma) = \dfrac{\sum_{j\not\to\ell\in\sigma}(1+\delta)p_\ell^{(G|_\sigma)^T}(\gamma) + \sum_{j\to\ell\in\sigma}(1-\varepsilon)p_\ell^{(G|_\sigma)^T}(\gamma)}{-(1+\delta) + (1+\delta)\sum_{\ell=1}^{n} p_\ell^{G|_\sigma}(\gamma)}$ *and* $\gamma = \dfrac{-\varepsilon-\delta}{1+\delta}$
and, if $k \notin \sigma$, $(\vec{g_j})_k = 0$.

7.2.1 Revised Initial Value Problem

This new set of eigenvectors does change the initial value problems for the systems L_σ where $\sigma \subset [n]$, but not considerably. The same approach can still be employed. We can use the same CTLN as we did in Chapter 4, but instead of solving the initial value problem for the $L_{[4]}$ system, we will solve it for the $L_{[3]}$ system instead. In this system we have the localized path polynomials depicted in Figure 7.1, but we will also have an eigenvector associated with the eigenvalue $\lambda = -1$.

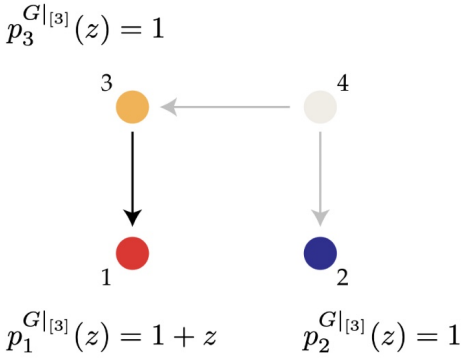


Figure 7.1. Localized path polynomials for $G|_{[3]}$. The localized path polynomials in the subgraph $G|_{[3]}$.

Applying Proposition 14, this eigenvector is:

$$\vec{v_4} = \begin{bmatrix} -1+\gamma\left(\dfrac{1-\varepsilon}{1+\delta}\right)+(1+\gamma)\Gamma_4([3]) \\ \left(\dfrac{1-\varepsilon}{1+\delta}\right)+\Gamma_4([3]) \\ \left(\dfrac{1-\varepsilon}{1+\delta}\right)+\Gamma_4([3]) \\ -1 \end{bmatrix}.$$

Taking $\kappa = \dfrac{1-\varepsilon}{1+\delta}$ and referring to $\Gamma_4([3]) = \Gamma_4$, the general solution for this system is of the form:

$$x(t) = c_1\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}e^{\delta t} + c_2\begin{bmatrix} 1+\alpha_1 \\ 1 \\ 1 \\ 0 \end{bmatrix}e^{\lambda_1 t} + c_3\begin{bmatrix} 1+\alpha_2 \\ 1 \\ 1 \\ 0 \end{bmatrix}e^{\lambda_2 t} + c_4\begin{bmatrix} -1+\gamma\kappa+(1+\gamma)\Gamma_4 \\ \kappa+\Gamma_4 \\ \kappa+\Gamma_4 \\ -1 \end{bmatrix}e^{-t} + x^*_{[3]}.$$

Then, the initial value problem can then be set up as:

$$\begin{bmatrix} x_1^0-(x^*_{[3]})_1 \\ x_2^0-(x^*_{[3]})_2 \\ x_3^0-(x^*_{[3]})_3 \\ x_4^0 \end{bmatrix} = \begin{bmatrix} 1 & 1+\alpha_1 & 1+\alpha_2 & -1+\gamma\kappa+(1+\gamma)\Gamma_4 \\ -1 & 1 & 1 & \kappa+\Gamma_4 \\ 0 & 1 & 1 & \kappa+\Gamma_4 \\ 0 & 0 & 0 & -1 \end{bmatrix}\vec{c}$$

Notice that the row corresponding to x_4 is empty except for the fourth column. By construction of the eigenvectors for $\lambda = -1$, this will be true for any of the rows corresponding to $j \notin \sigma$. We can then subtract that row to clear out that column.

$$\begin{bmatrix} x_1^0-(x^*_{[3]})_1+(-1+\gamma\kappa+(1+\gamma)\Gamma_4)x_4^0 \\ x_2^0-(x^*_{[3]})_2+(\kappa+\Gamma_4)x_4^0 \\ x_3^0-(x^*_{[3]})_3+(\kappa+\Gamma_4)x_4^0 \\ x_4^0 \end{bmatrix} = \begin{bmatrix} 1 & 1+\alpha_1 & 1+\alpha_2 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\vec{c}$$

Then we can conclude that $c_4 = -x_4^0$. Then c_1, c_2, c_3 satisfy:

$$\begin{bmatrix} x_1^0 - (x_{[3]}^*)_1 + (-1 + \gamma\kappa + (1+\gamma)\Gamma_4)x_4^0 \\ x_2^0 - (x_{[3]}^*)_2 + (\kappa + \Gamma_4)x_4^0 \\ x_3^0 - (x_{[3]}^*)_3 + (\kappa + \Gamma_4)x_4^0 \end{bmatrix} = \begin{bmatrix} 1 & 1+\alpha_1 & 1+\alpha_2 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

which can be solved using the approach described in Chapter 4.

7.3 Balanced hCTLNs

As Lemma 12 is an hCTLN analogue for Lemma 7, we should be able to say something about balanced states in DAG hCTLNs as well. Using the same process, we obtain a closed form expression of the balanced state.

Proposition 15. *Let G be a DAG. Letting $\beta = \frac{-\varepsilon-\delta}{1+\delta}$, the point $x_{bs} = -W^{-1}\vec{\theta}$ of an associated hCTLN can be written as:*

$$(x_{bs})_j = -\sum_{i=1}^{n} \left(\frac{\theta_i}{1+\delta} \right) p_{j,i}^G(\beta) + p_j^G(\beta)\Gamma$$

where $\Gamma = \left(\frac{-\sum_{i=1}^n \theta_i p_i^{G^T}(\beta)}{(1+\delta)(-1+\sum_{i=1}^n p_i^G(\beta))} \right)$.

Proof. Identical to that of Lemma 11 using Lemma 12 in the place of Lemma 7. □

Unfortunately the expressions for the entries of x_{bs} are now far more complicated, but we can still say something about whether an hCTLN is balanced.

Proposition 16. *Let G be a DAG. Let $\beta = \frac{-\varepsilon-\delta}{1+\delta}$ where ε and δ are taken from an associated hCTLN with external input vector $\vec{\theta}$. Then, that hCTLN is balanced if and only if:*

$$p_j^G(\beta)\Gamma' \geq \sum_{i=1}^{n} \theta_i p_{j,i}^G(\beta),\ \forall j \in [n]$$

where $\Gamma' = \left(\frac{-\sum_{i=1}^n \theta_i p_i^{G^T}(\beta)}{-1+\sum_{i=1}^n p_i^G(\beta)} \right)$.

While this result is not in and of itself terribly interesting, notice that, if $\beta > r_G$, the LHS of the inequality is guaranteed to be positive except for the term $-\sum_{i=1}^{n} \theta_i p_i^{G^T}(\beta)$. Both this and the RHS is $\sum_{i=1}^{n} \theta_i p_{j,i}^{G}(\beta)$. Both of these can be represented as inner products between the external input current vector $\vec{\theta}$ and a vector which is independent of $\vec{\theta}$. Perhaps, then using tools such as the Cauchy-Schwarz inequality, a sufficient condition can be obtained separating out the role of the connectivity parameters and $\vec{\theta}$. This is a problem for further study.

Chapter 8
Conclusions and Open Questions

We began this dissertation by asking how parameters of TLN attractor networks shape their basins of attraction while fixing the set of attractors. Our goal was to understand how decision-making bias was encoded in these models and we considered three ways of relating decision-making bias to basins of attraction.

1. The relative sizes of the basins of attraction restricted to the positive orthant.
2. The relative sizes of the basins within a neighborhood of the saddle point(s).
3. The basin of attraction within which the balanced state trajectory falls.

Recall that each of these corresponds to a paradigm of how neural dynamics behave. The first assumes that neural circuits display a high dimensional dynamics with diverse trajectories. The second assumes that the dynamics are confined to a lower dimensional submanifold on which the saddle point(s) lies. The third assumes a that the circuit operates along a particular trajectory.

In the context of a two neuron competitive TLN model, we worked the problem out completely, rigorously proving the basins of attraction and analytically calculating their sizes relative to one another. What we found however is that this model makes too many simplifications to be meaningful in encoding decision-making bias under a low dimensional paradigm.

We then considered the dynamics of CTLNs, particularly those derived from DAGs. While we were unable to find ways of determining the relative sizes of the basins of attraction, we did demonstrate how the dynamics could be rigorously worked out in great detail. Additionally, we were able to numerically demonstrate the relationship between the fractional indegree of sinks and the relative sizes of their basins in the vicinity of the saddle point(s), also providing a partially rigorous justification for the relationship. We finally explored the existence of balanced state trajectories rigorously in great detail, and offered numerical evidence for how the filtration of a DAG influences the attractor

within which the balanced state trajectory lies.

We will conclude by considering future lines of research which build on the results of this dissertation.

Open Questions

The following are a series of open questions which invite further inquiry and offer new avenues of study.

Question 1: What is a viable hyperplane arrangement for building a state transition graph approximating the basins of attraction in a competitive TLN?

This was the problem we left open at the end of Chapter 3. We showed how there exists a degree of freedom in finding a hyperplane which divides another into regions of inward and outward flow, but we were unable to give a way of exploiting that freedom so that the new hyperplane would not require additional separation. One approach to this might take inspiration from the two-dimensional trajectory graphs, where the stable manifold was an important separating line. While the stable manifold is curved, perhaps hyperplanes drawn to be roughly aligned could be used. In the case of DAG CTLNs, where the eigenvectors of the component systems L_σ have been worked out in great detail, those corresponding to the stable manifold could be incorporated using the degree of freedom that exists in the partition.

Question 2: Fixing a DAG and a corresponding CTLN, how does the network structure shape the change in the basins of attraction under a perturbation of parameters?

Among the key challenges with employing Theorem 6 to determine basins of attraction is tracking the trajectories across chambers and determining what chambers the seperatrix goes through. However, if a CTLN is fixed with a particular ε and δ, the chambers can be determined computationally and it may be easier to write out an analytical expression for the separatrix. Perturbing the parameters slightly will likely not change which chambers the separatrix passes through, so it may be possible to precisely understand how the basins are changing.

Question 3: How can the Balanced State Attractor Prediction Algorithm be improved?

We saw that there were cases where the attractor within which the balanced state trajectory lies was incorrectly predicted by Algorithm 2. This indicates that the filtration of the DAG is not the full story. Is there an algorithm with better accuracy?

Question 4: How can actual data be incorporated into these frameworks to make testable predictions about decision-making?

The goal of this dissertation was to understand how neural circuits encode decision-making biases. We found ways in which to theoretically model decision-making biases, but a natural next question is to consider whether the theoretical models can be fit to actual data and tested against experimental outcomes. The hope of course would be that these models can be used to infer new aspects of decision-making dynamics and inspire experimentation in their own right.

8.1 Numerical Methods Repository

A package of code for the recreation of figures and implementation of algorithms is available at: `https://github.com/sadiq-safaan/Dissertation-Companion`.

Bibliography

[1] Mas-Colell, A., M. Whinston, and J. Green (1995) *Microeconomic Theory*, 1 ed., Oxford University Press.

[2] Huber, J., J. Payne, and C. Puto (1982) "Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis," *Journal of Consumer Research*, **9**(1), pp. 90–98.

[3] Dorris, M. and P. Glimcher (2004) "Activity in Posterior Parietal Cortex Is Correlated with the Relative Subjective Desirability of Action," *Neuron*, **44**, pp. 365–378.

[4] Hopfield, J. (1982) "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. U.S.A.*, **79**(8), pp. 2554–2558.

[5] Kim, S., H. Rouault, S. Druckmann, and V. Jayaraman (2017) "Ring attractor dynamics in the Drosophila central brain," *Science*, **356**, pp. 849–853.

[6] Seung, H. (1996) "How the brain keeps eyes still," *PNAS*, **93**, p. 13339–13344.

[7] Wang, X. (2012) "Neural dynamics and circuit mechanisms of decision-making," *Current Opinion in Neurobiology*, **22**(6).

[8] Khona, M. and I. R. Fiete (2022) "Attractor and integrator networks in the brain," *Nature Reviews Neuroscience*, **23**(12), pp. 744–766.

[9] Gerstner, W., A. Kreiter, H. Markram, and A. Herz (1997) "Neural codes: firing rates and beyond," *Proc. Natl. Acad. Sci. U.S.A.*, **94**, p. 12740–12741.

[10] Abbott, L., S. Fusi, and K. Miller (2012) *Principles of Neural Science*, 5 ed., McGraw-Hill Education/Medical.

[11] Hahnloser, R., S. Seung, and J. Slotine (2003) "Permitted and forbidden sets in symmetric threshold-linear networks," *Neural Comput.*, **15**(3), pp. 621–638.

[12] Curto, C. and K. Morrison (2023) "Graph Rules for Recurrent Neural Network Dynamics," *Notices of the American Mathematical Society*, **70**(4), pp. 536–551.

[13] Churchland, A., R. Kiani, and M. Shadlen (2008) "Decision-making with multiple alternatives," *Nature Neuroscience*, **11**(6), pp. 693–702.

[14] Laboratory, T. I. B. (2021) "Standardized and reproducible measurement of decision-making in mice," *eLife*, **10:e63711**.

[15] Lienkaemper, C. (2022), "Combinatorial geometry of neural codes, neural data analysis, and neural networks," 2209.07583.
URL https://arxiv.org/abs/2209.07583

[16] Carnevale, F., V. de Lafuente, R. Romo, O. Barak, and N. Parga (2015) "Dynamic Control of Response Criterion in Premotor Cortex during Perceptual Detection under Temporal Uncertainty," *Neuron*, **86**(4), pp. 1067–1077.
URL https://www.sciencedirect.com/science/article/pii/S0896627315003645

[17] van Vreeswijk, C. and H. Sompolinsky (1996) "Chaos in neuronal networks with balanced excitatory and inhibitory activity," *Science*, **274**(5293), pp. 1724–1726.

[18] Deneve, S. and C. Machens (2016) "Efficient codes and balanced networks," *Nature Neuroscience*, **19**, pp. 375–382.

[19] Baker, C., V. Zhu, and R. Rosenbaum (2020) "Nonlinear stimulus representations in neural circuits with approximate excitatory-inhibitory balance," *PLOS Computational Biology*, **16**(9).

[20] O'Shea, D. J., L. Duncker, W. Goo, X. Sun, S. Vyas, E. M. Trautmann, I. Diester, C. Ramakrishnan, K. Deisseroth, M. Sahani, and K. V. Shenoy (2022) "Direct neural perturbations reveal a dynamical mechanism for robust computation," *bioRxiv*, https://www.biorxiv.org/content/early/2022/12/16/2022.12.16.520768.full.pdf.
URL https://www.biorxiv.org/content/early/2022/12/16/2022.12.16.520768

[21] Biák, M., T. Hanus, and D. Janovská (2013) "Some applications of Filippov's dynamical systems," *Journal of Computational and Applied Mathematics*, **254**, pp. 132–143, nonlinear Elliptic Differential Equations, Bifurcation, Local Dynamics of Parabolic Systems and Numerical Methods.
URL https://www.sciencedirect.com/science/article/pii/S0377042713001428

[22] Tsodyks, M. V., W. E. Skaggs, T. J. Sejnowski, and B. L. McNaughton (1997) "Paradoxical Effects of External Modulation of Inhibitory Interneurons," *Journal of Neuroscience*, **17**(11), pp. 4382–4388, https://www.jneurosci.org/content/17/11/4382.full.pdf.
URL https://www.jneurosci.org/content/17/11/4382

[23] Albantakis, L. and G. Deco (2011) "Changes of Mind in an Attractor Network of Decision-Making," *PLOS Computational Biology*, **7**(6), pp. 1–13.
URL https://doi.org/10.1371/journal.pcbi.1002086

[24] et al., R. Y. (2005) "The cortex as a central pattern generator," *Nature Reviews Neuroscience*, **6**, pp. 477–483.

[25] Morrison, K., A. Degeratu, V. Itskov, and C. Curto (2024) "Diversity of Emergent Dynamics in Competitive Threshold-Linear Networks," *SIAM Journal on Applied Dynamical Systems*, **23**(1), pp. 855–884, https://doi.org/10.1137/22M1541666.
URL https://doi.org/10.1137/22M1541666

[26] Strogatz, S. (2000) *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*, Studies in nonlinearity, Westview.
URL https://books.google.com/books?id=NZZDnQEACAAJ

[27] Lugagne, J.-B., S. Sosa Carrillo, M. Kirch, A. Köhler, G. Batt, and P. Hersen (2017) "Balancing a genetic toggle switch by real-time feedback control and periodic forcing," *Nature Communications*, **8**(1), p. 1671.

[28] Abdelkader, M. A. (1974) "Exact solutions of Lotka-Volterra equations," *Mathematical Biosciences*, **20**(3), pp. 293–297.
URL https://www.sciencedirect.com/science/article/pii/0025556474900054

[29] Bel, A., R. Cobiaga, W. Reartes, and H. G. Rotstein (2021) "Periodic Solutions in Threshold-Linear Networks and Their Entrainment," *SIAM Journal on Applied Dynamical Systems*, **20**(3), pp. 1177–1208, https://doi.org/10.1137/20M1337831.
URL https://doi.org/10.1137/20M1337831

[30] Mischaikow, K. (1999) "The Conley index theory: A brief introduction," *Banach Center Publications*, **47**(1), pp. 9 – 19.

[31] Gedeon, T., B. Cummins, S. Harker, and K. Mischaikow (2018) "Identifying robust hysteresis in networks," *PLOS Computational Biology*, **14**(4), pp. 1–23.
URL https://doi.org/10.1371/journal.pcbi.1006121

[32] Ding, J. and A. Zhou (2007) "Eigenvalues of rank-one updated matrices with some applications," *Applied Mathematics Letters*, **20**(12), pp. 1223–1226.
URL https://www.sciencedirect.com/science/article/pii/S0893965907000614

[33] Moya-Cessa, H. and F. Soto-Eguibar (2012), "Inverse of the Vandermonde and Vandermonde confluent matrices," 1211.1566.
URL https://arxiv.org/abs/1211.1566

[34] Langdon, C., M. Genkin, and T. Engel (2023) "A unifying perspective on neural manifolds and circuits for cognition," *Nature Reviews Neuroscience*, **24**(6), pp. 363 – 377.

[35] Bartlett, M. S. (1951) "An Inverse Matrix Adjustment Arising in Discriminant Analysis," *The Annals of Mathematical Statistics*, **22**(1), pp. 107 – 111.
URL https://doi.org/10.1214/aoms/1177729698

[36] Bertsimas, D. and J. Tsitsiklis (1997) *Introduction to Linear Optimization*, 1 ed., Athena Scientific.

[37] Sridhar, V., L. Li, D. Gorbonos, M. Nagy, B. Schell, T. Sorochkin, N. Gov, and I. Couzin (2021) "The geometry of decision-making in individuals and collectives," *Proc. Natl. Acad. Sci. U.S.A.*, **118**(50).

[38] Mishra, B. (1993) *Real Algebra*, Springer New York, New York, NY, pp. 297–383.
URL https://doi.org/10.1007/978-1-4612-4344-1_8

Vita

Syed Safaan Sadiq

Safaan Sadiq graduated from American University in 2017 with a BS in Mathematics and a BA in International Studies. He then graduated from Northwestern University in 2018 with an MS in Engineering Sciences and Applied Mathematics. In 2019, he entered the Ph.D. program at Penn State University where he has worked with Carina Curto.

Teaching

Instructor - MATH21 - College Algebra I	FA2020, SP2021
Instructor - MATH22 - College Algebra II	FA2021-SP2023 and SP2024-FA2024
Grader - MATH310 - Combinatorics	SP2020
Grader - MATH311 - Discrete Mathematics	SP2024