Research Notes on Self-Coupled Spiking Neural Networks

Chris Fritz

today

Contents

1 The self-coupled SNN Model **2**

2 Basic Model **3**
2.1 Derivation . . . 3
2.2 Simulation of Basic Equations . . . 11

3 Analysis: RMSE vs Spike Rate for Constant Driving Force **14**

4 Derivation: The Predictive Coding Framework and Gap-Junction Network **22**

5 Analysis: PCF and Gap-Junction Response to Constant Stimulus **28**
5.1 PCF Network Response to Constant Stimulus: . . . 28
5.2 Gap-Junction Network Response to Constant Stimulus: . . . 30
5.3 Per-spike RMSE of the PCF and Gap-Junction Networks for a Constant Stimulus . . . 33
5.4 Comparison of Self-Coupled, Gap-Junction, and PCF Networks for a Constant Stimulus . . . 35

6 Extensions of the Basic Model **39**
6.1 Dynamical Systems with Complex Eigenvalues . . . 39

1 The self-coupled SNN Model

Problem Statement

Given:

- A Linear Dynamical System $\frac{dx}{dt} = Ax(t) + Bc(t)$, $x \in \mathbf{R}^d$
- A Decoder Matrix $D \in \mathbf{R}^{d \times N}$ specifying the preferred directions of N neurons in d-dimensional space,

synthesize a spiking neural network that implements the linear dynamical system.

Features

1. ***Long-Term Network Accuracy*** The Deneve network assumes $\hat{x} = x$. We show this assumption produces estimation error between the network and its target system that increases with time. By avoiding this assumption, the self-coupled network remains accurate over time.

2. ***Tuning Curve Rotation*** To most efficiently use neurons, we use orthogonal coding directions via SVD. The dynamics matrix A is diagonalized by an orthonormal basis $\mathcal{U}$ in d-dimensional space, while the decoder matrix D is chosen such that $\mathcal{U}$ gives its left singular vectors. This choice of coding directions eliminates the need for coupling.

   At least two neurons per dimension (2d in total) are required since voltage thresholds are strictly positive. N-neuron ensembles can thus represent systems with $\frac{N}{2}$ dimensions or less.

3. ***Post-synaptic Spike Dropping*** At each synapse, neurotransmitter release due to an action potential is probabilistic. We incorporate probabilistic spike transmission by thinning at every synaptic connection. The pre-synaptic neuron's membrane potential is still deterministically reset by an action potential.

4. ***Dimensionless Time*** We describe both the network and target system in dimensionless time. Time is normalized by the synapses' time constant, τ_s. This dimensionless representation ensures consistent numerical simulation independent of simulation timestep. Furthermore, τ_s is implicitly specified as 1, reducing the model's parameters by one.

2 Basic Model

2.1 Derivation

1. Let τ_s be the synaptic time constant of each synapse in the network. Define dimensionless time as:

$$\xi \triangleq \frac{t}{\tau_s}.$$

We now assume our Linear Dynamical System is expressed in dimensionless time, i.e

$$\frac{dx}{d\xi} = Ax(\xi) + Bc(\xi). \tag{2.1}$$

To describe the neuron dynamics in dimensionless time, let $o(\xi) \in \mathbf{R}^N$ be the spike trains of N neurons composing the network with components

$$o_j(\xi) = \sum_{k=1}^{n_j \text{ spikes}} \delta(\xi - \xi_j^k),$$

where ξ_j^k is the time at which neuron j makes its k^{th} spike. Define the network's estimate of the state variable as

$$\hat{x}(\xi) \triangleq Dr(\xi), \tag{2.2}$$

where $D \in \mathbf{R}^{d\times N}$ and

$$\frac{dr}{d\xi} = -r + o(\xi). \tag{2.3}$$

When the probability of synaptic transmission is 1, component r_j is the total received post-synaptic current (PSC) from neuron j by the network estimator. Define the network error as

$$e(\xi) \triangleq x(\xi) - \hat{x}(\xi). \tag{2.4}$$

2. From equations (2.3) and (2.2), we have

$$D\dot{r} + Dr = Do$$

$$\implies \dot{\hat{x}} + \hat{x} = Do,$$

where the dot denotes derivative w.r.t dimensionless time ξ.

Subtract $\dot{\hat{x}}$ from $\dot{x}$ to get $\dot{e}$:

$$\begin{aligned}
\dot{e} &= \dot{x} - \dot{\hat{x}} \\
&= (Ax + Bc) - (Do - \hat{x}) \\
&= A\left(e + \hat{x}\right) + Bc - Do + \hat{x} \\
&= Ae + (A + I)\hat{x} + Bc - Do \\
&= Ae + (A + I)\left(Dr\right) + Bc - Do \\
\implies D^T\dot{e} &= D^T Ae + D^T(A + I)\left(Dr\right) + D^T Bc - D^T Do.
\end{aligned}$$

The quantity $D^T e$ defines the membrane voltage of the predictive coding framework (PCF), a precursor to this model:

$$v_{pcf} \triangleq D^T e.$$

Note that the definition implies $e = D^{T\dagger} v_{pcf}$. The voltage dynamics are thus

$$\dot{v}_{pcf} = D^T A D^{T\dagger} v_{pcf} + D^T (A + I)(Dr) + D^T Bc - D^T Do, \tag{2.5}$$

where $D^{T\dagger}$ is the left pseudo-inverse of $D^T \in \mathbf{R}^{N \times d}$. The PCF thus defines a mapping between two vector spaces: the d-dimensional state space of the target system, and the N-dimensional voltage space of the spiking neural network. This mapping is visualized in figure (1).

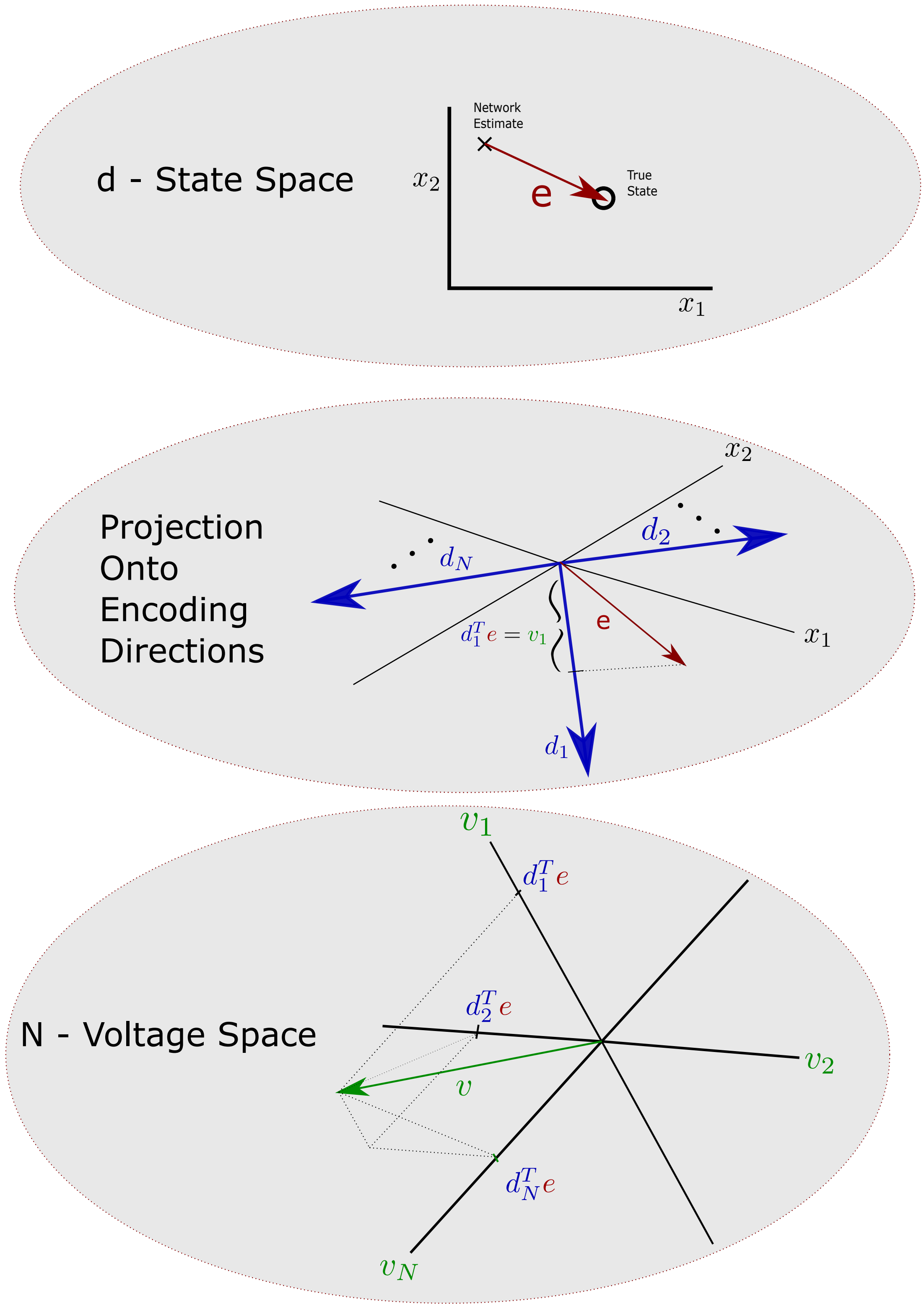

Figure 1: Mapping Between State and Voltage Spaces: ***Top:*** The estimation error e is computed by comparing the decoded network estimate to the true state of the target dynamical system. ***Middle:*** The e is projected onto the encoding directions of the neurons composing the network. The projection of error onto encoding direction j gives the membrane voltage of neuron j, $v_j = d_j^T e$. ***Bottom:*** The voltages form a N-dimensional vector contained in voltage space.

3. The self-coupled network is derived via a change of bases. Assuming both D and A are full rank, diagonalize each to a common left basis:

$$A = \mathcal{U}\Lambda\mathcal{U}^T = \sum_{j=1}^{d} \Lambda_j \mathcal{U}_j \mathcal{U}_j^T,$$

$$D = \mathcal{U}\,[S \quad 0]\,V^T = \sum_{j=1}^{d} S_j \mathcal{U}_j V_j^T,$$

$$D^T = V \begin{bmatrix} S \\ 0 \end{bmatrix} \mathcal{U}^T = \sum_{j=1}^{d} S_j V_j \mathcal{U}_j^T,$$

$$D^T D = V \begin{bmatrix} S \\ 0 \end{bmatrix} \begin{bmatrix} S & 0 \end{bmatrix} V^T = \sum_{j=1}^{d} S_j^2 V_j V_j^T,$$

with $\mathcal{U} \in \mathbf{R}^{d\times d}$ and $V \in \mathbf{R}^{N\times N}$, and $S \in \mathbf{R}^{d\times d}$.

In the original basis, the state is x. In the rotated basis we denote this quantity as y. It is the projection of x onto the d-dimensional $\mathcal{U}$ basis:

$$y \triangleq \mathcal{U}^T x \tag{2.6}$$

The rotated target dynamics are thus

$$\begin{aligned}
\dot{y} &= \mathcal{U}^T \dot{x} \\
&= \Lambda y(\xi) + \mathcal{U}^T B c(\xi) \\
&= \Lambda y(\xi) + \mathcal{U}^T B \mathcal{U}\mathcal{U}^T c(\xi) \\
&= \Lambda y(\xi) + \beta \tilde{c}(\xi)
\end{aligned} \tag{2.7}$$

where

$$\beta \triangleq \mathcal{U}^T B \mathcal{U},$$

and

$$\tilde{c} \triangleq \mathcal{U}^T c,$$

give the projections of B and c respectively. The network estimate in the rotated basis is

$$\hat{y} \triangleq \mathcal{U}^T \hat{x}.$$

From equation (2.2),

$$
\begin{aligned}
\hat{y} &= \mathcal{U}^T \hat{x} \\
&= \mathcal{U}^T D r \\
&= \begin{bmatrix} S & 0 \end{bmatrix} V^T r \\
&= \begin{bmatrix} S & 0 \end{bmatrix} \rho \\
\implies \dot{\hat{y}} &= \begin{bmatrix} S & 0 \end{bmatrix} V^T \dot{r} \\
&= \begin{bmatrix} S & 0 \end{bmatrix} \left(-V^T r + V^T o \right).
\end{aligned}
$$

Note that $V^T r$ and $V^T o$ are projections of the N-neuron network's post-synaptic current and spike train respectively onto the rotated basis, denoted by

$$\rho \triangleq V^T r, \tag{2.8}$$

$$\tilde{o} \triangleq V^T o. \tag{2.9}$$

The preceding equality also gives $\hat{y}$ in terms of ρ:

$$\hat{y} = \begin{bmatrix} S & 0 \end{bmatrix} \rho. \tag{2.10}$$

With these definitions, the last equality above also implies

$$\dot{\rho} = -\rho + \tilde{o}. \tag{2.11}$$

To finish describing the basic network quantities in terms of the rotated basis, let ϵ be the error in the rotated basis:

$$
\begin{aligned}
\epsilon &\triangleq y - \hat{y} \\
&= \mathcal{U}^T e.
\end{aligned} \tag{2.12}
$$

4. Repeat the derivation of equation (2.5) but with y, $\hat{y}$, and ϵ:

$$
\begin{aligned}
\dot{\epsilon} &= \dot{y} - \dot{\hat{y}} \\
&= \Lambda y + \beta c - \begin{bmatrix} S & 0 \end{bmatrix} (-\rho + \tilde{o}) \\
&= \Lambda \left(\epsilon + \begin{bmatrix} S & 0 \end{bmatrix} \rho \right) + \beta c - \begin{bmatrix} S & 0 \end{bmatrix} (-\rho + \tilde{o}) \\
&= \Lambda \epsilon + (\Lambda + I) \begin{bmatrix} S & 0 \end{bmatrix} \rho - \begin{bmatrix} S & 0 \end{bmatrix} \tilde{o} \\
\implies \begin{bmatrix} S \\ 0 \end{bmatrix} \dot{\epsilon} &= \begin{bmatrix} S \\ 0 \end{bmatrix} \Lambda \epsilon + \begin{bmatrix} S \\ 0 \end{bmatrix} (\Lambda + I) \begin{bmatrix} S \\ 0 \end{bmatrix} \rho - \begin{bmatrix} S \\ 0 \end{bmatrix} \begin{bmatrix} S & 0 \end{bmatrix} \tilde{o}.
\end{aligned}
$$

The last equality gives a system of N equations of which only d of are nontrivial. A comparison with equation (2.5) suggests the N-dimensional rotated membrane potential v is best defined as:

$$
v \triangleq \begin{bmatrix} S \\ 0 \end{bmatrix} \epsilon \in \mathbf{R}^N. \tag{2.13}
$$

This mapping is not invertible unless we only consider the first d components and neglect the remaining, trivial components. Abusing notation, we write

$$
\epsilon = S^{-1} v,
$$

giving an N vector whose first d elements are well defined, and the remaining components of v are assumed to be zero. Using a similar abuse for the ρ and $\tilde{o}$ terms, we arrive at the system of d equations describing the nontrivial network voltage dynamics:

$$
\begin{aligned}
\dot{v} &= S \Lambda S^{-1} v + S (\Lambda + I) S \rho - S^2 \tilde{o} \\
\implies \dot{v} &= \Lambda v + S (\Lambda + I) S \rho - S^2 \tilde{o}.
\end{aligned} \tag{2.14}
$$

We can also write all N dimensions explicitly to respect the dimensionality of v and ρ:

$$
\dot{v} = \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} v + \begin{bmatrix} S (\Lambda + I_d) S & 0 \\ 0 & 0 \end{bmatrix} \rho - \begin{bmatrix} S^2 & 0 \\ 0 & 0 \end{bmatrix} \tilde{o}.
$$

To summarize conceptually, there are 4 vector spaces in total: the error space which tracks the dynamical system and the network estimate, the voltage space which tracks the membrane potentials, and the transformed counterparts of each in the $\mathcal{U} - V$ bases. Figure (2) shows the relationships derived between these subspaces.

5. The spike trains $\tilde{o}$ are chosen minimize the network estimation error

$$
\mathcal{L}(\xi) = ||x(\xi + d\xi) - \hat{x}(\xi + d\xi)||^2. \tag{2.15}
$$

The network greedily minimizes $\mathcal{L}(\xi)$ an instant $d\xi$ ahead in time. If no spike occurs at time ξ, then the objective is given above. If neuron j spikes, the estimate $\hat{x} \leftarrow \hat{x} + d_j$, where d_j is column j of D. The objective is now

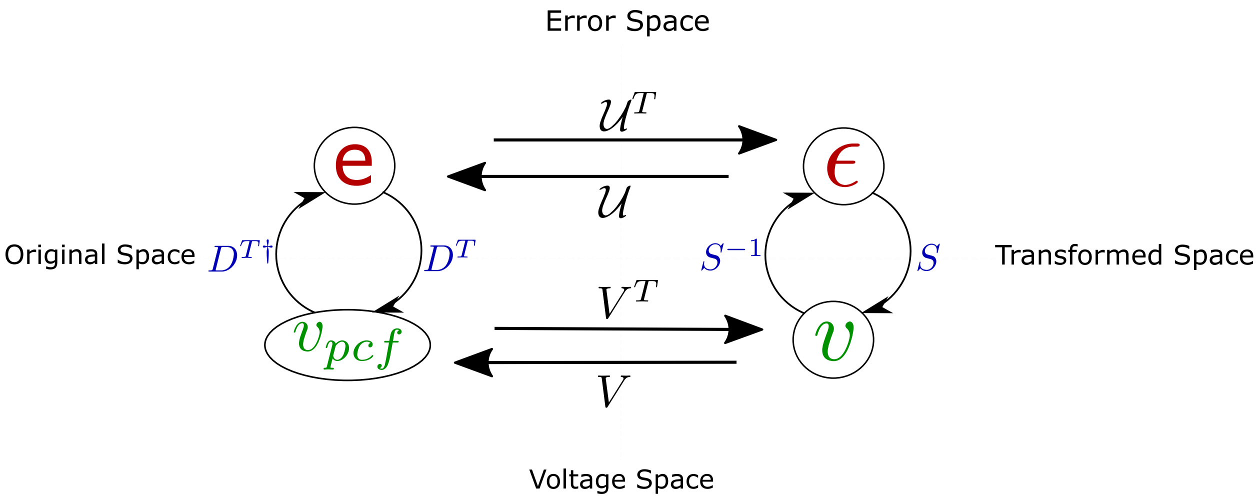

Figure 2: Depiction of the relationship between original and transformed spaces and their respective error and voltage spaces. An arrow represents left multiplication by the given matrix. The zeros in the full $N \times N$ matrices mapping between v and ϵ are omitted for clarity.

$$
\begin{aligned}
\mathcal{L}_{sp}(\xi) &= ||x - (\hat{x} + d_j)||^2 \\
&= x^T x - 2\, x^T \hat{x} - 2x^T d_j + \hat{x}^T \hat{x} + 2\hat{x}^T d_j + d_j^T d_j \\
&= x^T x - 2x^T \hat{x} + \hat{x}^T \hat{x} - 2d_j^T (x - \hat{x}) + d_j^T d_j \\
&= ||x - \hat{x}||^2 - 2d_j^T (x - \hat{x}) + d_j^T d_j \\
&= \mathcal{L}_{ns}(\xi) - 2d_j^T (x - \hat{x}) + d_j^T d_j,
\end{aligned}
$$

where $\mathcal{L}_{ns}(\xi)$ is the objective if no spike occurs. Spiking occurs when the objective decreases or

$$
\begin{gathered}
\mathcal{L}_{sp} < \mathcal{L}_{ns} \\
\implies -2d_j^T (x - \hat{x}) + d_j^T d_j < 0 \\
\implies d_j^T (x - \hat{x}) > \frac{||d_j||^2}{2}.
\end{gathered}
$$

Since $d_j^T (x - \hat{x}) = d_j^T e$ is already defined as membrane voltage, the right hand side gives neuron $j's$ spike threshold voltage $v_t h$,

$$
v_{th}^{pcf} = \frac{1}{2} \begin{bmatrix} d_1^T d_1 \\ \vdots \\ d_N^T d_N \end{bmatrix}.
$$

For the rotated network, note $\mathcal{U}^T$ is an orthonormal matrix by definition. Thus it is norm-preserving:

$$\begin{aligned}\mathcal{L}_{sp}(\xi) &= ||x - \hat{x}||^2 \\ &= ||\mathcal{U}^T(x - \hat{x})||^2 \\ &= ||y - \hat{y}||^2.\end{aligned}$$

If we define the rotated network objective as

$$\tilde{L}(\xi) \triangleq ||y(\xi + d\xi) - \hat{y}(\xi + d\xi)||^2,$$

it is equal to the original network objective when no spike occurs. However, a spike alters the readout by $\hat{y} \leftarrow \hat{y} + S_l$, where S_l is the l^{th} column of $\begin{bmatrix} S & 0 \end{bmatrix}$. With the same approach as above, the objective when neuron l spikes is

$$\tilde{L}_{sp} = \tilde{L}_{ns} + 2S_l^T \epsilon + S_l^T S_l$$

$$\implies v_l > \frac{||S_l||^2}{2}.$$

This leads to voltage thresholds

$$v_{th} = \frac{1}{2} \begin{bmatrix} S_1^T S_1 \\ \vdots \\ S_d^T S_d \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

6. Equations (2.14) and (2.11) describe how we implement a network with d neurons that produces an accurate estimate $\hat{x}$ of the given target system.

   When neuron l spikes, a vector S_l is added to the network estimate, $\hat{y}$. A spike has a strictly positive area so that the network is only able to modify its estimate by adding from a fixed set of vectors. This restricts the space representable by the network to strictly positive state-space, or only $\frac{1}{2^d}$ dimensions of the desired state-space. To remove this restriction, we add an additional d neurons whose preferred directions $-S_l$ are anti-parallel to neurons l for $l = 1, \ldots, d$. Thus the number of neurons required to represent a d-dimensional system is $2d$. We update U, S, Λ and v_{th} to reflect the additional neurons:

$$U \leftarrow [U \quad -U] \in \mathbf{R}^{d\times 2d},$$

$$S \leftarrow \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} \in \mathbf{R}^{2d\times 2d},$$

$$\Lambda \leftarrow \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda \end{bmatrix} \in \mathbf{R}^{2d\times 2d},$$

$$v_{th} \leftarrow \begin{bmatrix} v_{th} \\ v_{th} \end{bmatrix} \in \mathbf{R}^{2d},$$

and afterward recompute $\beta \in \mathbf{R}^{2d\times d}$.

2.2 Simulation of Basic Equations

Here we simulate the above equations (2.14) and (2.11) with the $N = 2d$ neurons. The parameters are

$$\begin{aligned} A &= -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathcal{U}\Lambda\mathcal{U}^T, \\ B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ c(\xi) &= \begin{bmatrix} cos(\frac{\pi}{4}\xi) \\ sin(\frac{\pi}{4}\xi) \end{bmatrix} \\ D &= \mathcal{U}\begin{bmatrix} S & 0 \end{bmatrix} V^T = \mathcal{U}\begin{bmatrix} .1\, I_d & 0 \end{bmatrix} I_N, \\ d\xi &= 10^{-6}, \\ N &= 4, \\ x(0) &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \end{aligned} \tag{2.16}$$

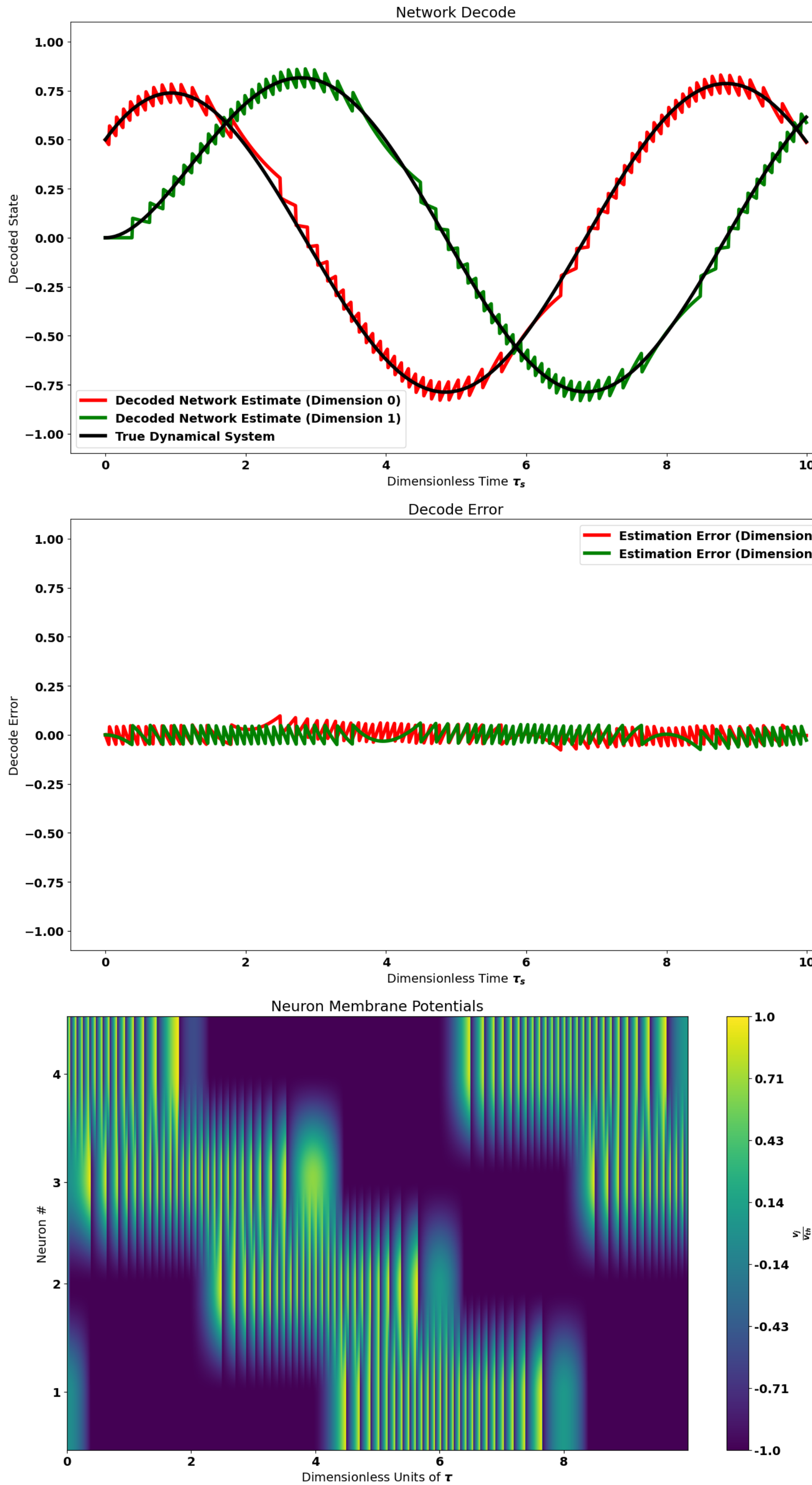
Network Decode
Decoded State
Decoded Network Estimate (Dimension 0)
Decoded Network Estimate (Dimension 1)
True Dynamical System
Dimensionless Time τ_s
Decode Error
Decode Error
Estimation Error (Dimension 0)
Estimation Error (Dimension 1)
Dimensionless Time τ_s
Neuron Membrane Potentials
Neuron #
Dimensionless Units of τ
$\frac{v_j}{v_{th}}$

Figure 3: Simulation of equations (2.14) and (2.11) with parameters listed in equation (2.16). ***Top:*** The decoded network estimate plotted alongside the target dynamical system. ***Middle:*** The estimation error along each state-space dimension. ***Bottom:*** The membrane potentials of the 4 neurons during the same time period.

For the numerical implementation, the matrix exponential was used to integrate the continuous terms over a simulation time step. Continuous terms include all equation terms excepting the delta functions ω handled separately. After integrating over a timestep, any neuron above threshold was manually reset (action of fast inhibition). If multiple neurons are above threshold, the system is integrated backwards in time until only one neuron is above threshold before spiking. The matrix exponential was computed using a Padé approximation via the Python package Scipy: *scipy.linalg.expm()*.

3 Analysis: RMSE vs Spike Rate for Constant Driving Force

We analyse the network described by equations (2.14) and (2.11) for the case of a constant (in time) driving force $c(\xi) = k$. First we derive explicit expressions for the network estimate, then we compute the resulting RMSE for various driving strengths k.

1. Let

$$A = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \Lambda,$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I,$$

$$D = \mathcal{U} \begin{bmatrix} S & 0 \end{bmatrix} V^T = \mathcal{U} \begin{bmatrix} I_d & 0 \end{bmatrix} I_N,$$

$$c(\xi) = k \in \mathbf{R}^d, \text{expressed in the } \mathcal{U} - V \text{ basis},$$

$$d\xi = 10^{-4},$$

$$N = 4,$$

$$x(0) = \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix}.$$

The system simplifies to one neuron by noting the following: The voltage of the $2d$ non trivial neurons is

$$v = S\epsilon.$$

Voltage j spikes i.f.f. $v_j = \frac{||S_j||^2}{2}$. Let 1 index the neuron whose encoding direction S_j is closest in angle to k, i.e

$$S_1^T e \geq S_j^T e \;\; \forall\, j$$

$$\implies v_1 \geq v_k \;\; \forall\, j.$$

It follows from equation (2.14) that v_1 will reach its threshold first. Neuron 1 spikes and its voltage is reset, and sequence repeats. Let us integrate until the first spike, avoiding the discontinuity from $\tilde{o}$. The nontrivial dynamics simplify to

$$\begin{aligned} \dot{v_1}(\xi) &= \Lambda_1 v_1(\xi) + (\Lambda_1 + 1)||S_1||^2 \rho_1(\xi) + S_1^T k, \\ \dot{\rho_1}(\xi) &= -\rho_1(\xi). \end{aligned} \tag{3.1}$$

With initial conditions $v_1(0) = v_1^0$ and $\rho_1(0) = \rho_1^0$, the system of equations has the general solution

$$\begin{aligned} \rho_1(\xi) &= \rho_1^0 e^{-\xi}, \\ v_1(\xi) &= e^{\Lambda_1 \xi} \left(\frac{S_1^T k}{\Lambda_1} + ||S_1||^2 \rho_1^0 + v_1^0 \right) - e^{-\xi} \, ||S_1||^2 \, \rho_1^0 - \frac{S_1^T k}{\Lambda_1}. \end{aligned} \tag{3.2}$$

Between spikes, neuron 1's voltage is independent from other neurons j. The voltage v_1 only depends on its own history and feedback from its own spike train, ρ_1. Replace index 1 we see this is true for all neurons so that all neurons are *self-coupled*, hence the name.

A spike occurs when $v(\xi_{spike}) = \frac{||S_1||^2}{2}$, or

$$\frac{||S_1||^2}{2} = e^{\Lambda_1 \xi_{spike}} \left(\frac{S_1^T k}{\Lambda_1} + ||S_1||^2 \rho_1^0 + v_1^0 \right) - e^{-\xi_{spike}} \, ||S_1||^2 \, \rho_1^0 - \frac{S_1^T k}{\Lambda_1}.$$

This equation is transcendental in that a closed form expression for ξ_{spike} does not exist. However, under certain initial conditions we can obtain a solution. Let $r_1^0 = v_1^0 = -v_{th} = \frac{||S||^2}{2}$. Moreover, for a sufficiently small angle between S_1 and k, $S_1^T k \simeq ||S_1|| \, ||k||$. Under these conditions,

$$\xi_{spike} = \frac{1}{\Lambda} \, ln \left(\frac{1 + \frac{\Lambda_1 ||S_1||}{2 \, ||k||}}{1 - \frac{\Lambda_1 ||S_1||}{2 \, ||k||}} \right)$$

The preceding expression is the amount of time required for neuron 1 to spike starting from its membrane reset potential. This expression is exact for the case of no slow-synaptic feedback $\rho_1^0 = 0$. The intrinsic firing rate of the neuron is the inverse:

$$\phi(s, k) \triangleq \Lambda_1 \, ln \left(\frac{1 + \frac{\Lambda_1 S_1}{2 \, k}}{1 - \frac{\Lambda_1 S_1}{2 \, k}} \right)^{-1}, \tag{3.3}$$

where the vertical bars for the norms are omitted for clarity.

The network will encode the constant driving force by spiking at a fixed rate determined by equation (3.3). Figure (4) shows a plot of equation (3.3) along with numerically computed spike rates for a simulated network driven with constant drive strength ratio $\frac{||k||}{||s||}$. Similar to membrane voltage, the resulting slow feedback and readout dynamics are reduced to one neuron periodically spiking:

$$\dot{\rho_1} = -\rho_1 + \tilde{o}_1$$

$$\implies \dot{\hat{x}} = -S_1 \rho_1 + S_1 \tilde{o}_1$$

$$= -\hat{x} + S_1 \tilde{o}_1,$$

where $S_1 \in \mathbf{R}^d$.

2. The spike train $\tilde{o}_1$ is a periodic sequence of impulses spaced in time by $\frac{1}{\phi}$. If the first spike occurs at ξ_1^1, then $\tilde{o}_1(\xi) = \sum_{l=0}^{\infty} \delta \left(\xi - \xi_1^1 - \frac{l}{\phi} \right)$. The network estimate therefore has dynamics

$$\dot{\hat{x}} = -\hat{x} + S_1 \sum_{l=0}^{\infty} \delta \left(\xi - \frac{l}{\phi} \right). \tag{3.4}$$

The target dynamical system is

$$\dot{x} = -x + k$$
$$x(0) = \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix},$$

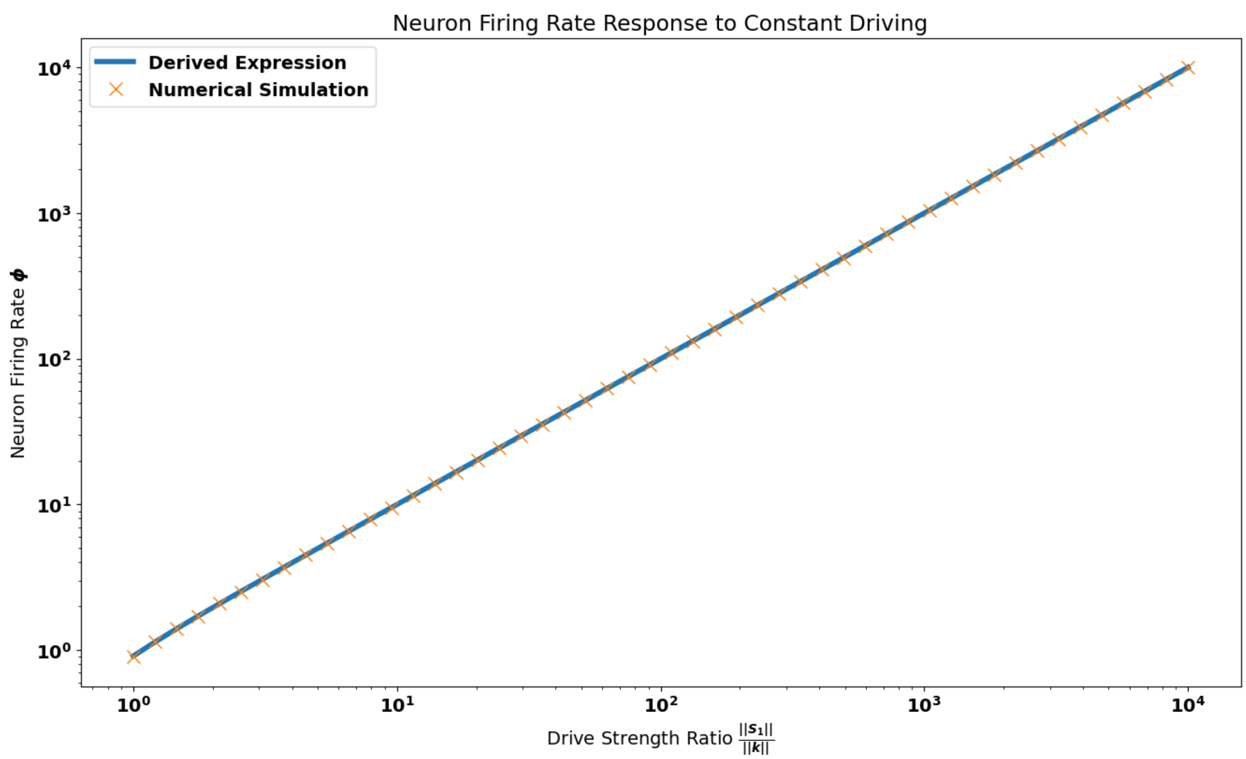

Figure 4: A log-log plot of equation (3.3) alongside the rates measured from numerical simulations. The simulation parameters are described at the beginning of this section. The rate was measured as the number of spike resets divided by the duration of the simulation.

which has a stable fixed point at

$$x = k. \tag{3.5}$$

Equation (3.4) implies that the network estimate $\hat{x}$ will decay until the first spike ξ_1^1 occurs:

$$\hat{x}(\xi) = x(0)e^{-\xi}, \quad 0 \le \xi < \frac{1}{\phi}.$$

At this instant, the vector S_1 is added to the network estimate.

$$\hat{x}(\frac{1}{\phi}) = x(0)e^{-\frac{1}{\phi}} + S_1.$$

Decay again occurs until the next spike

$$\hat{x}(\xi) = \hat{x}(\frac{1}{\phi})e^{-(\xi - \frac{1}{\phi})},$$

$$= \left(x(0)e^{-\frac{1}{\phi}} + S_1\right) e^{-(\xi-\frac{1}{\phi})}, \quad \frac{1}{\phi} \le \xi < \frac{2}{\phi}$$

$$\implies \hat{x}(\frac{2}{\phi}) = \left(x(0)e^{-\frac{1}{\phi}} + S_1\right) e^{-(\frac{1}{\phi})} + S_1$$

$$= x(0)e^{-\frac{2}{\phi}} + S_1 e^{-\frac{1}{\phi}} + S_1.$$

The third spike more clearly shows the recursive behavior

$$\hat{x}(\frac{3}{\phi}) = \left[x(0)e^{-\frac{2}{\phi}} + S_1 e^{-\frac{1}{\phi}} + S_1\right] e^{-\frac{1}{\phi}} + S_1$$

$$= x(0)e^{-\frac{3}{\phi}} + S_1 e^{-\frac{2}{\phi}} + S_1 e^{-\frac{1}{\phi}} + S_1$$

Let us consider the n^{th} spike sufficiently far from $\xi = 0$ such that the transient term $x(0)e^{-\frac{n}{\phi}}$ can be neglected. This leads to the expression

$$\hat{x}(\frac{n}{\phi}) = \sum_{l=0}^{n-1} S_1 e^{-\frac{l}{\phi}}$$

$$= S_1 \frac{1 - e^{-\frac{n}{\phi}}}{1 - e^{-\frac{1}{\phi}}}.$$

For sufficiently large n, this converges to

$$\hat{x}(\xi_1^n) = \frac{S_1}{1 - e^{-\frac{1}{\phi}}}. \tag{3.6}$$

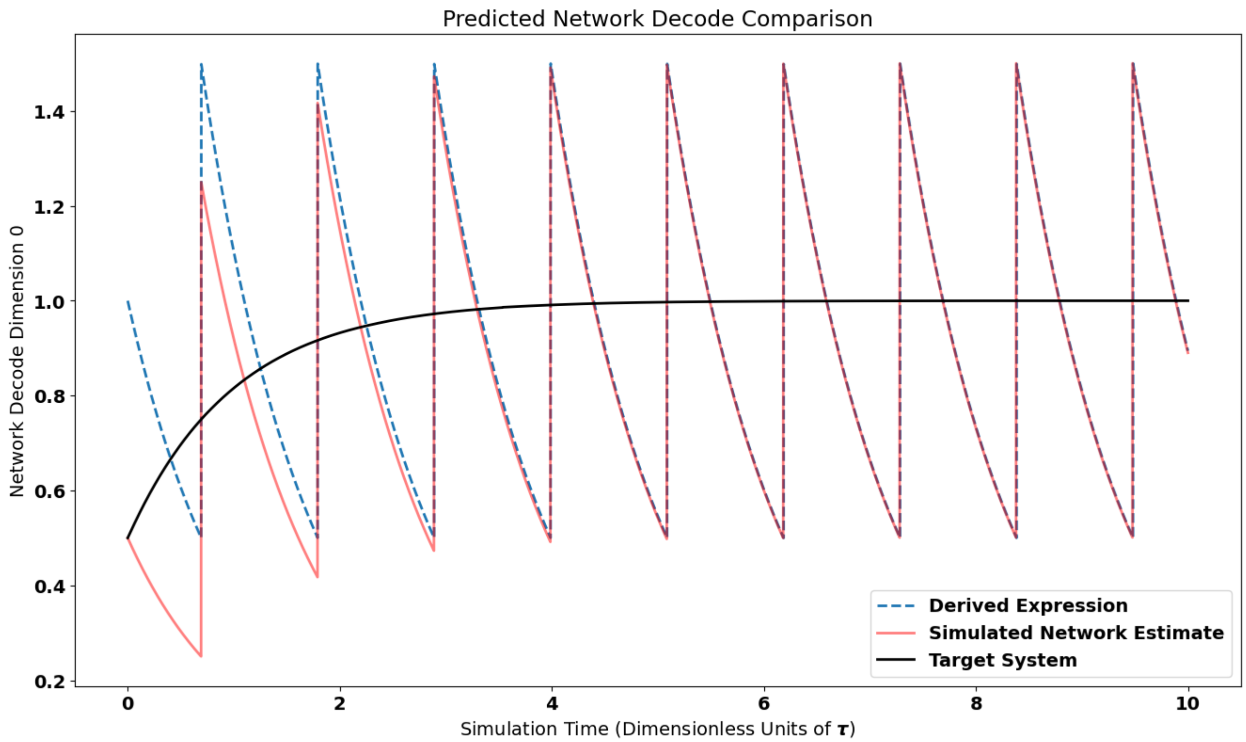

Figure 5: Comparison of the derived long-term network estimate equation (3.8) to numerical simulation. Parameters are the same as the previous figure, with $\frac{||S_1||}{||k||} = 1$.

3. The preceding argument states that after a transient interval, the network estimate at any spike time ξ_1^n is given by equation (3.6). As shown in figure (5), this convergence occurs after roughly 5 spikes under the given parameters.

   We know from equation (3.4) that the estimate will decay exponentially from this value over an interval $\frac{1}{\phi}$ until a spike returns it returns to the initial value. Thus the network estimate between two consecutive spikes is given by

$$\hat{x}(\xi) = \frac{S_1}{1 - e^{-\frac{1}{\phi}}} e^{-(\xi - \xi_1^n)}, \quad 0 \le \xi - \xi_1^n < \frac{1}{\phi}.$$

   Combine this expression with equation (3.6), we have an explicit expression for the long-term behavior of the network estimate given by

$$\hat{x}(\xi) = \frac{S_1}{1 - e^{-\frac{1}{\phi}}} e^{-(\xi - \xi_1^1) \mod \frac{1}{\phi}}, \tag{3.7}$$

   where $x \mod y$ denotes the fractional remainder of x after division by y.

   Writing this equation in terms of provided network parameters, we use (3.3), to first obtain:

$$e^{-\frac{1}{\phi}} = \left(\frac{2k - \Lambda_1 S_1}{2k + \Lambda_1 S_1} \right)^{1/\Lambda_1},$$

which then gives

$$\hat{x}(\xi) = \frac{S_1}{\left(\frac{2k-\Lambda_1 S_1}{2k+\Lambda_1 S_1}\right)^{1/\Lambda_1}} e^{-(\xi-\xi_1^1) \mod \left(\frac{2k+\Lambda_1 S_1}{2k-\Lambda_1 S_1}\right)^{1/\Lambda_1}}. \tag{3.8}$$

4. Assume the true system dynamics have settled to their fixed point $x = k$. From equation (3.8) the network estimate $\hat{x}$ and therefore error $e = x - \hat{x}$ is a periodic function of ξ with period $\frac{1}{\phi}$. The RMSE over any integer number of spike periods is easily calculated from the RMSE over a single spike period.

   We compute the per-spike RMSE of the error signal e by

$$RMSE_{spike} \triangleq \sqrt{\phi \int_0^{\frac{1}{\phi}} ||e(\tau)||^2 \, \mathrm{d}\tau}. \tag{3.9}$$

   The integrand $||e(\tau)||^2$ simplifies to

$$\begin{aligned}
e^T e &= (x - \hat{x})^T (x - \hat{x}) \\
&= x^T x - 2x^T \hat{x} + \hat{x}^T \hat{x} \\
&= ||k||^2 \; - 2S_1^T k \frac{e^{-\tau}}{1 - e^{-\frac{1}{\phi}}} + ||S_1||^2 \left(\frac{e^{-\tau}}{1 - e^{-\frac{1}{\phi}}}\right)^2 \\
&= ||k||^2 \; - 2||S_1|| \, ||k|| \frac{e^{-\tau}}{1 - e^{-\frac{1}{\phi}}} + ||S_1||^2 \left(\frac{e^{-\tau}}{1 - e^{-\frac{1}{\phi}}}\right)^2.
\end{aligned}$$

   Note that

$$\int_0^{\frac{1}{\phi}} e^{-\tau} \, \mathrm{d}\tau = 1 - e^{-\frac{1}{\phi}},$$

   while

$$\begin{aligned}
\int_0^{\frac{1}{\phi}} \left(e^{-\tau}\right)^2 &= \frac{1 - e^{-\frac{2}{\phi}}}{2} \\
&= \frac{1}{2}\left(1 - e^{-\frac{1}{\phi}}\right)\left(1 + e^{-\frac{1}{\phi}}\right).
\end{aligned}$$

   Therefore the integral is

$$\phi \int_0^{\frac{1}{\phi}} ||e(\tau)||^2 \, \mathrm{d}\tau = ||k||^2 - 2\,\phi\,||S_1||\,||k|| + \phi \,\frac{||S_1||^2}{2} \frac{1 + e^{-\frac{1}{\phi}}}{1 - e^{-\frac{1}{\phi}}}$$

   The per-spike RMSE of the network estimate is thus

$$RMSE_{spike}(k, S_1, \phi) = \sqrt{||k||^2 - 2\,\phi\,||S_1||\,||k|| + \phi \,\frac{||S_1||^2}{2} \frac{1 + e^{-\frac{1}{\phi}}}{1 - e^{-\frac{1}{\phi}}}}. \tag{3.10}$$

To write the RMSE explicitly as a function of given parameters S_1, k, Λ_1, we substitute our earlier expression for $e^{-\frac{1}{\phi}}$ and use equation (3.3)to obtain

$$RMSE_{spike}(k, S_1, \Lambda_1) = \sqrt{||k||^2 - 2\frac{\Lambda_1||S_1||\,||k||}{ln\left(\frac{2\,||k||+\Lambda_1||S_1||}{2\,||k||-\Lambda_1||S_1||}\right)} + \frac{\Lambda_1}{ln\left(\frac{2\,||k||+\Lambda_1||S_1||}{2\,||k||-\Lambda_1||S_1||}\right)}\frac{||S_1||^2}{2}\left(\frac{1+\left(\frac{2||k||-\Lambda_1||S_1||}{2||k||+\Lambda_1||S_1||}\right)^{1/\Lambda_1}}{1-\left(\frac{2||k||-\Lambda_1||S_1||}{2||k||+\Lambda_1||S_1||}\right)^{1/\Lambda_1}}\right)} \tag{3.11}$$

5. For the case where $\Lambda_1 = -1$ and $S_1 = 1$, we can solve for the per-spike RMSE as a simple function of ϕ. First note that from equation (3.3),

$$\frac{||k||}{||S_1||}(\phi) = \frac{1}{2}\frac{1+e^{-\frac{1}{\phi}}}{1-e^{-\frac{1}{\phi}}}.$$

From the preceding expression equation (3.11) simplifies to

$$RMSE_{spike}\,(k, S_1, \phi) = \sqrt{||k||^2 - \phi||S_1||\,||k||}.$$

If we normalize by the driving force $||k||$, we can isolate the change in network accuracy due to the intrinsic neuron firing rate. Divide the preceding expression by $||k||$ to get

$$NRMSE_{spike}\,(\phi) = \sqrt{1 - 2\phi\frac{1-e^{-\frac{1}{\phi}}}{1+e^{-\frac{1}{\phi}}}}. \tag{3.12}$$

Equations (3.11) and (3.12) are plotted in figure (6). Note that the drive strength varies the amplitude of the target system's steady state. Thus we have derived the the network performance over its dynamic range of representable state space.

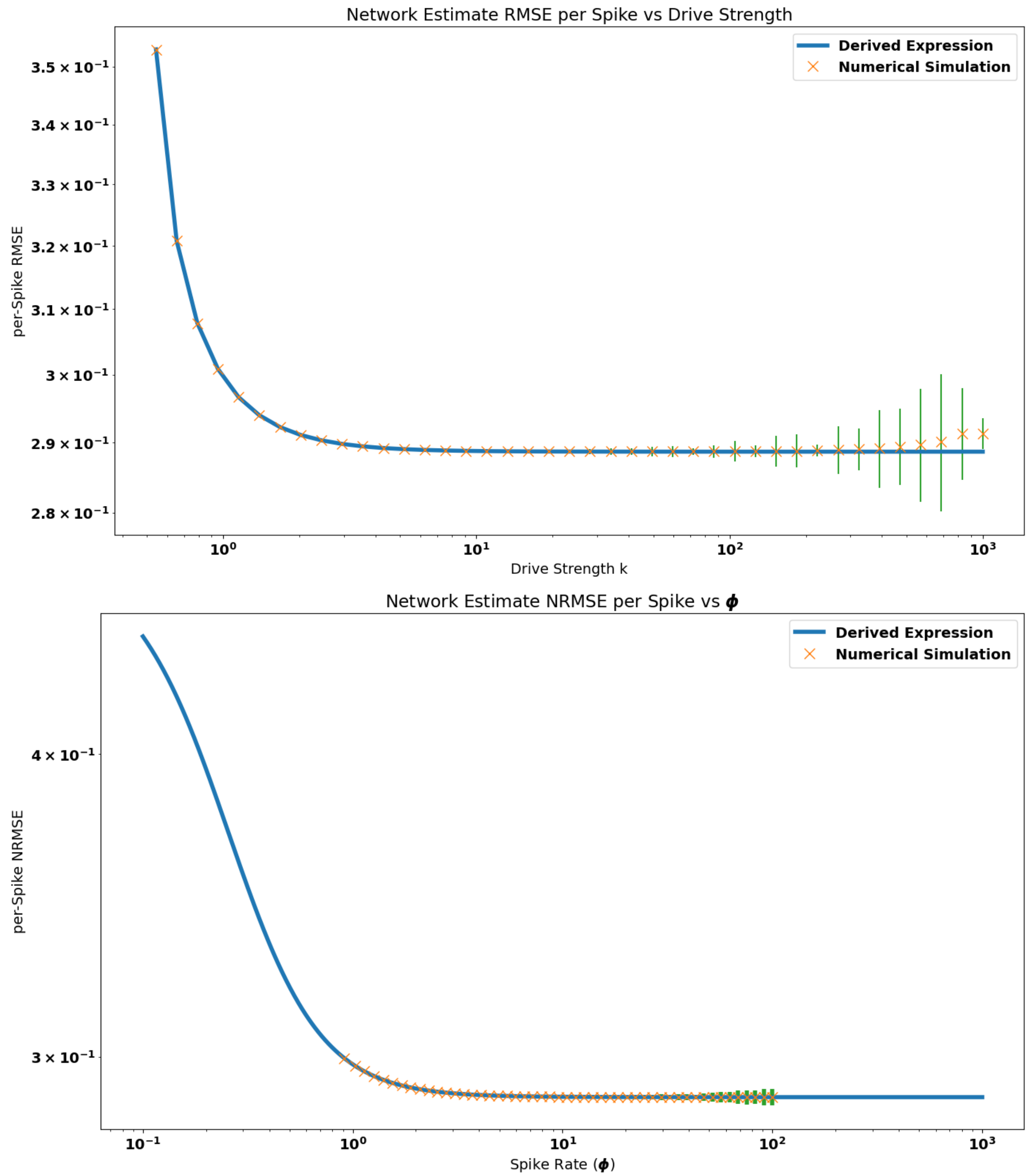

Figure 6: ***Top:*** A log-log plot of equation (3.11). ***Bottom:*** A log-log plot of equation (3.12). ***Both:*** Each simulated data point is the RMSE averaged over all inter-spike intervals in a simulation of length $T = 80\tau_s$ at a constant (in time) drive strength. Between simulations, the spike rates were varied by sweeping drive strength. Green vertical lines towards the larger values are +/- 1 standard deviation. The spike rates $\hat{\phi}$ were computed numerically via dividing the number of spikes in a simulation by the simulation duration. The RMSE between two adjacent spikes was computed by numerical integration as a discrete sum: $R\hat{M}SE = \sqrt{\hat{\phi}\sum_{\tau \text{ between spikes}} e(\xi)^T e(\xi)\ d\xi}$.

4 Derivation: The Predictive Coding Framework and Gap-Junction Network

Here we derive the a form of the predictive coding framework (PCF) as defined in Boerlin & Deneve, 2013. We note an assumption in this model that we later show leads to errant behavior in the network estimate. The correction of this assumption produces an intermittent mode featuring direct membrane voltage coupling. We loosely term this a gap-junction network. We compare the network estimate of all three models (PCF, gap-junction, and self-coupled) for the case of a constant driving stimulus.

1. ***The Predictive Coding Framework (PCF):*** The PCF synthesizes a spiking neural network that implements a given dynamical system. It is briefly derived as follows:

   Assume the following are given:

   - A Linear Dynamical System $\dot{x}(\xi) = Ax(\xi) + Bc(\xi)$, $x \in \mathbf{R}^d$
   - A Decoder Matrix $D \in \mathbf{R}^{d\ x\ N}$ specifying The tuning curve of N neurons in d-dimensional space.

   Let $o(t) \in \mathbf{R}^N$ describe the spike trains whose j^{th} component is given by

   $$o_j(t) \triangleq \sum_{k=0}^{\infty} \delta(t - t_j^k),$$

   where t_j^k is the time of the k^{th} spike of neuron j. Define the time-varying firing rate of the neurons by

   $$\frac{dr}{dt}(t) \triangleq -\tau_s^{-1} r(t) + \tau_s^{-1} o(t),$$

   where $\tau_s{-}1$ is the decay rate of $r(t)$ given by the inverse synaptic time constant τ_s. For consistency across models, we transform the preceding two equations to dimensionless time via $\xi = \frac{t}{\tau_s} \implies \tau_s\, d\xi = dt$. This gives

   $$o_j(\xi) \triangleq \sum_{k=0}^{\infty} \delta(\xi - \xi_j^k), \tag{4.1}$$

   where ξ_j^k is the k^{th} spike of neuron j in dimensionless time, and

   $$\frac{dr}{dt}(t) = -\tau_s^{-1} r(t) + \tau_s^{-1} o(t),$$

   $$\implies \frac{dr}{\tau_s\, d\xi}(\xi) = -\tau_s^{-1} r(\xi) + \tau_s^{-1} o(\xi),$$

   $$\implies \frac{dr}{d\xi}(\xi) = -r(\xi) + o(\xi).$$

   Letting $[\ \dot{}\ ]$ denote differentiation w.r.t. dimensionless time ξ, we arrive at

   $$\dot{r}(\xi) \triangleq -r(\xi) + o(\xi). \tag{4.2}$$

   The network estimate is defined as

   $$\hat{x}(\xi) \triangleq Dr(\xi), \tag{4.3}$$

which gives rise to the network estimation error

$$e(\xi) \triangleq x(\xi) - \hat{x}(\xi). \tag{4.4}$$

The network chooses spike times ξ_j^k to greedily optimize the objective function

$$\mathcal{L}(\xi) = ||x(\xi + d\xi) - \hat{x}(\xi + d\xi)||^2.$$

The PCF features regularized rate terms $r(\xi)$ for the sake of biological plausibility. At present we ignore these terms. They only increase the network estimation error e by sacrificing accuracy to minimize $r(\xi)$. Using an identical approach to the derivation of the self-coupled network in section (2), we arrive at

$$d_j^T (x - \hat{x}) = \frac{d_j^T d_j}{2}$$

where d_j is the j^{th} column of D. We define membrane voltage to get the spiking condition:

$$\begin{aligned} v_j &\triangleq d_j^T (x - \hat{x}) \\ \implies d_j^T e &= v_{th}, \end{aligned} \tag{4.5}$$

where $v^{th} = \frac{d_j^T d_j}{2}$.

Deriving the dynamics, the preceding equation defines voltage, which in matrix form is given by

$$\begin{aligned} V &= D^T (x - \hat{x}) \\ \implies \dot{V} &= D^T \dot{x} - D^T \dot{\hat{x}} \\ &= D^T (Ax + Bc) - D^T (D\dot{r}) \\ &= D^T Ax + D^T Bc - D^T D (-r + o) . \end{aligned}$$

The PCF makes the assumption that when the network performs correctly, $x = \hat{x}$. We later quantify the estimation error introduced by this assumption and correct it to form the gap-junction model. For now make the assumed substitution $x = \hat{x} = Dr$.

$$\begin{aligned} \dot{V} &= D^T A (Dr) + D^T Bc + D^T Dr - D^T Do \\ &= D^T (A + I) Dr + D^T Bc - D^T Do. \end{aligned}$$

The model is finalized by the addition of a voltage leakage term to ensure stability, giving the final dynamics equation

$$\dot{V} = -v + D^T (A + I) Dr + D^T Bc - D^T Do. \tag{4.6}$$

Equation (4.6) scales the spike train o_j by $d_j^T d_j$. Thus the spiking behavior is described by

$$\begin{aligned}
&v_{th} = \frac{d_j^T d_j}{2} \\
&\text{if } v_j > v_j^{th}, \\
&\text{then } v_j^{'} = v_j - d_j^T d_j \int \delta(\tau)\, d\tau, \\
&\text{and } r_j^{'} = r_j + \int \delta(\tau)\, d\tau.
\end{aligned} \tag{4.7}$$

Equations (4.6) and (4.7) specify the PCF model we compare against. Figure (7) shows simulations of the PCF model with the following parameters:

$$\begin{aligned}
A &= -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
c(\xi) &= 10 \begin{bmatrix} cos(\frac{\pi}{2}\xi) \\ sin(\frac{\pi}{2}\xi) \end{bmatrix} + 8 \\
D_{\text{ij}} &\sim \mathcal{N}(0, 1) \text{ Columns Normalized to Unit Length} \\
d\xi &= 10^{-5}, \\
N &= 32, \\
x(0) &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}.
\end{aligned} \tag{4.8}$$

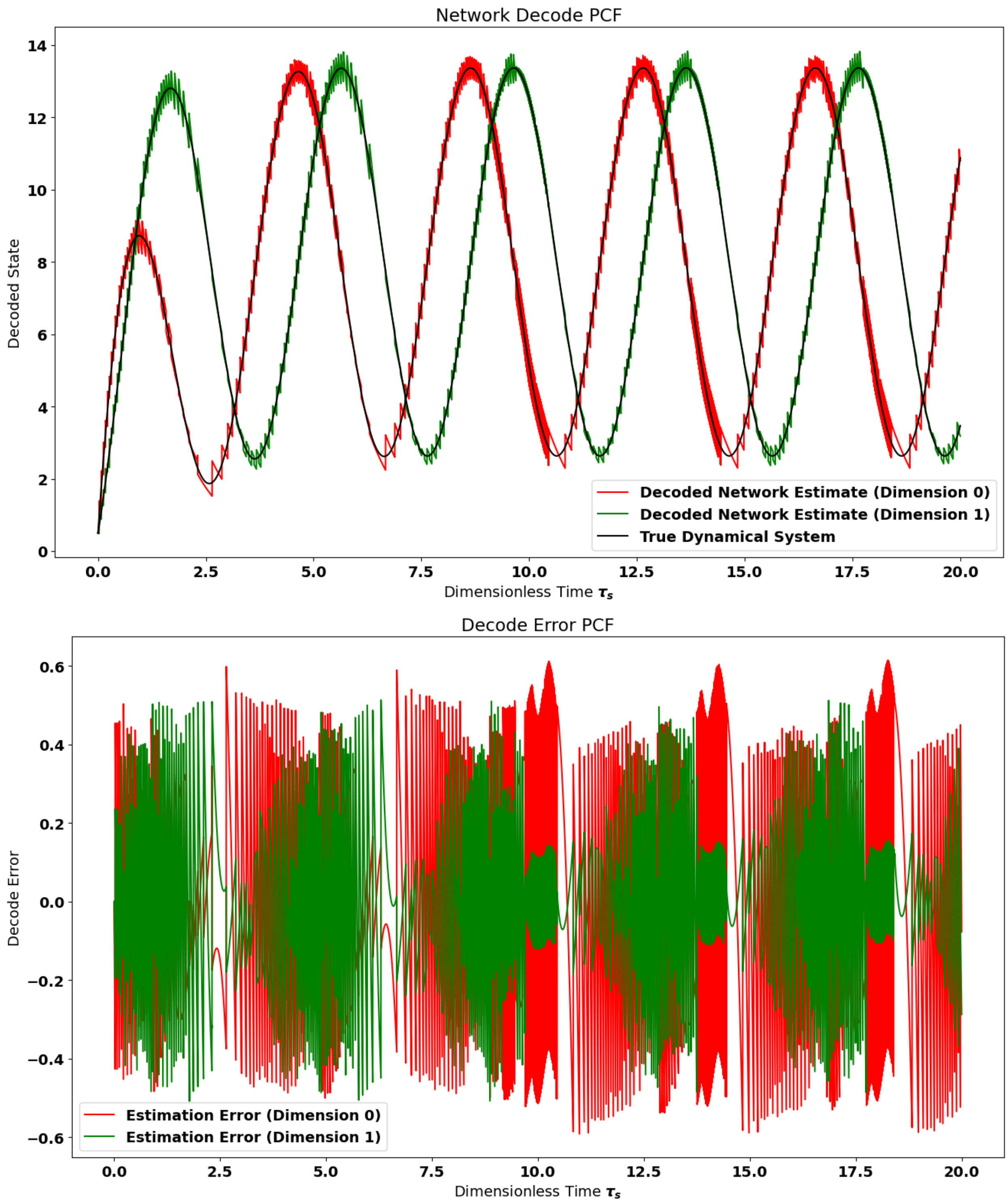

Figure 7: Simulation of PCF model given by equations (4.7) and (4.6). ***Top:*** Network estimate given by equation (4.3). ***Bottom:*** Estimation Error for PCF network from equation (4.4). The simulation parameters are given in equation (4.8). The numerical implementation is identical to that in section (2). A Padé approximation is used to compute a matrix exponential, then used to integrate the continuous terms of the differential equations. The spikes are handled separately at each time step by manually changing the values of neurons above threshold. For reasons of numerical stability, only one spike per time-step is allowed in the PCF model.

2. ***The Gap-Junction Correction:*** Here we correct the assumption that $\hat{x} = x$ made in the PCF model. We restart the previous derivation from this point and derive more a accurate form of equation (4.6) termed the gap-junction model.
   The derivation is identical as the PCF until we derive the voltage dynamics.

$$\dot{V} = D^T A x + D^T B c + D^T D r - D^T D o.$$

Instead of assuming $x = \hat{x}$, we apply the definition of voltage, equation (4.5) in matrix form.

$$\begin{aligned} v_j &= d_j^T e \\ \implies V &= D^T e \\ &= D^T (x - \hat{x}) \\ \implies x &= D^{T\dagger} V + \hat{x} \\ &= D^{T\dagger} V + Dr, \end{aligned}$$

where $D^{T\dagger}$ is the left Moore-Penrose pseudo-inverse of D^T. Substitute this for x in $\dot{V}$ above to get

$$\dot{V} = D^T A \left(D^{T\dagger} V + Dr \right) + D^T D r + D^T B c - D^T D o$$

$$\implies \dot{V} = D^T A D^{T\dagger} V + D^T (A + I) D r + D^T B c - D^T D o. \tag{4.9}$$

Equation (4.9) in conjunction with an identical spiking rule from PCF, equation (4.7) specifies the gap-junction model. It is simulated in figure (8). While the two simulations are similar, there are noticeable differences in their behavior e.g. $\tau_s \simeq 10$, 13.

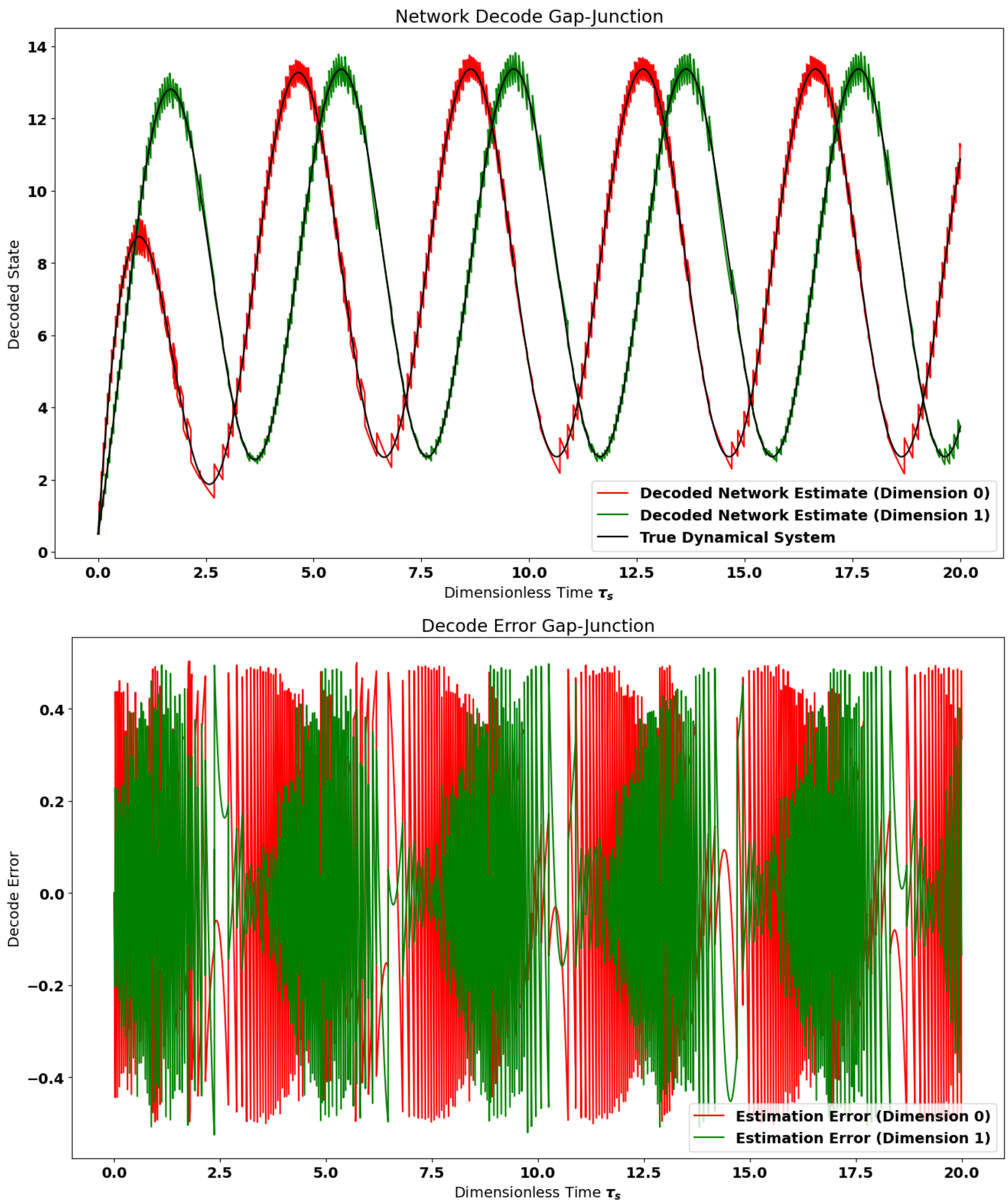

Figure 8: Simulation of the Gap-Junction model given by equations (4.7) and (4.9). ***Top:*** Network estimate given by equation (4.3). ***Bottom:*** Estimation Error for the Gap-Junction network from equation (4.4). The simulation parameters are the same as the previous figure. As with the PCF model, the network is only numerically stable if spikes are restricted to one per time step.

5 Analysis: PCF and Gap-Junction Response to Constant Stimulus

We compare the network estimate of all three models (PCF, gap-junction, and self-coupled) for the case of a constant driving stimulus.
Let all 3 models have the same parameters as given by equation (4.8) with the exception that

$$c(\xi) = c = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and $x(0) = \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix}$.

5.1 PCF Network Response to Constant Stimulus:

From equation (4.6), the PCF dynamics become

$$\dot{V}_{pcf} = -V_{pcf} + D^T \left(-I + I\right) D^T r + D^T \begin{bmatrix} 1 \\ 0 \end{bmatrix} - D^T D o$$

$$= -V_{pcf} + D^T \begin{bmatrix} 1 \\ 0 \end{bmatrix} - D^T D o.$$

All voltages are initially 0. From equation (4.7) the thresholds are identically $\frac{1}{2}$. Until the first spike, neuron j's voltage integrates the quantity $d_j^T \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Denote the neuron j whose tuning curve d_j is closest in angle to c by

$$j_{max} \triangleq \underset{i \in [1,\ldots,N]}{argmax} \quad d_j^T c.$$

Neuron j_{max} will receive the highest driving force and will therefore reach its threshold before any other neuron. It will then be reset by 1 to $-\frac{1}{2}$. Each other neuron k will also be reset (decremented) by $d_k^T d_{j_{max}}$, proportional to their angle relative to both neuron j_{max} and the driving strength c. This sequence will repeat periodically so that only neuron j_{max} fires at a constant rate.
We write the PCF network as the one-dimensional equation

$$v_{pcf} = -v_{pcf} + d_{j_{max}}^T c - o_{j_{max}}.$$

This is a form of the leaky integrate-and-fire (LIF) model, with drive term $d_j^T c(\xi)$. The neuron is driven by inner product $d_{j_{max}}^T c$. Note from equation (4.7) that the threshold voltage varies with $||d_{j_{max}}||^2$. For clarity, we drop the subscripts j, j_{max} in the following equations. It is understood that we are referring to the solely spiking neuron j_{max}. With initial condition $v_{pcf}(0) = -\frac{||d||^2}{2}$, the neuron's trajectory is integrated as

$$v_{pcf}(\xi) = d^T c - e^{-\xi} \left(d^T c + \frac{||d||^2}{2} \right). \tag{5.1}$$

The neuron spikes when it reaches the threshold $v_{pcf} = ||d||^2$. To compare with the self-coupled network, we note that the singular value associated with neuron j of the decoder matrix $S = ||d||^2$.

From the preceding equation with voltage at threshold $\frac{||d||^2}{2}$,

$$\frac{||d||^2}{2} = d^T c - e^{-\xi_{spike}} \left(d^T c + \frac{||d||^2}{2} \right)$$

$$\implies e^{-\xi_{spike}} = \frac{d^T c - \frac{||d||^2}{2}}{d^T c + \frac{||d||^2}{2}}$$

$$\implies \xi_{spike} = ln \left(d^T c + \frac{||d||^2}{2} \right) - ln \left(d^T c - \frac{||d||^2}{2} \right)$$

This leads to a firing rate

$$\phi_{pcf}(d) = \frac{1}{ln \left(d^T c + \frac{||d||^2}{2} \right) - ln \left(d^T c - \frac{||d||^2}{2} \right)} \tag{5.2}$$

Deriving the network estimate, suppose it begins at $x(0)$. The trajectory until the first spike at time ξ_1 is

$$\hat{x}(\xi) = x(0)e^{-\xi}, \quad 0 \leq \xi < \xi_1.$$

The spike adds d to the readout followed by exponential decay:

$$\hat{x}(\xi) = \left(x(0)e^{-\xi_1} + d \right) e^{-(\xi - \xi_1)}, \quad 0 \leq \xi - \xi_1 < \frac{1}{\phi}.$$

Until the third spike the readout is

$$\hat{x}(\xi) = \left(x(0)e^{-\xi_1} e^{-\frac{1}{\phi}} + de^{-\frac{1}{\phi}} + d \right) e^{-(\xi - \frac{1}{\phi} - \xi_1)}, \quad \frac{1}{\phi} \leq \xi - \xi_1 < \frac{2}{\phi}.$$

The recursive pattern is visible after the third spike

$$\hat{x}(\xi) = \left(x(0)e^{-\xi_1} e^{-\frac{2}{\phi}} + de^{-\frac{2}{\phi}} + de^{-\frac{1}{\phi}} + d \right) e^{-(\xi - \frac{2}{\phi} - \xi_1)}, \quad \frac{2}{\phi} \leq \xi - \xi_1 < \frac{3}{\phi}.$$

Consider the n^{th} term for n big enough so that the $x(0)$ term is approximately 0. The readout at spike time ξ_n is given by the sum

$$\hat{x}(\xi_n) = d \sum_{l=0}^{n-1} e^{-\frac{l}{\phi}}.$$

The series converges to

$$\hat{x}(\xi_n) = \frac{d}{1 - e^{-\frac{1}{\phi}}}.$$

Between the spikes the readout exponentially decays so that the network estimate is given by

$$\hat{x}_{pcf}(\xi) = \frac{d}{1 - e^{-\frac{1}{\phi}}} e^{-\ \left(\xi - \xi_1^1\right) \mod \frac{1}{\phi}}. \tag{5.3}$$

5.2 Gap-Junction Network Response to Constant Stimulus:

Here we derive the decoded estimate of a gap-junction network driven by a constant stimulus, $c(\xi) = \begin{bmatrix} 1 & 0 \end{bmatrix}$. All other parameters are identical to those in equation (4.8).

From the dynamics equation (4.9) gap-junction voltages are continuously coupled to one another via $D^T A D^{T\dagger}$. We thus need to solve the entire system between spikes rather than reducing it to a single dimension. Let $\tilde{[]}$ denote the Laplace transform of a variable. Assume neuron j has just spiked so that

$$V(0) = -\frac{1}{2} \begin{bmatrix} d_1^T d_j \\ \vdots \\ d_j^T d_j \\ \vdots \\ d_N^T d_j \end{bmatrix}.$$

Since $c(\xi) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $B = I$, and $o(\xi) = 0$ between spikes, we have

$$\dot{V} = D^T A D^{T\dagger} V + D^T (A + I) Dr + d_1,$$

where d_1 is the first column of D. Apply the one-sided Laplace Transform to both sides and use the Laplace derivative property:

$$s\tilde{V} - V(0) = D^T A D^{T\dagger} \tilde{V} + D^T (A + I) oD\tilde{r} + \mathcal{L}[d_1]$$

$$\implies \left(s\,I - D^T A D^{T\dagger}\right) \tilde{V} = V(0) + D^T (A + I) D\tilde{r} + \tilde{d_1}$$

$$\implies \tilde{V} = \left(s\,I - D^T A D^{T\dagger}\right)^{-1} \left[V(0) + D^T (A + I) D\tilde{r} + \tilde{d_1}\right]$$

$$= \left(sI - D^T A D^{T\dagger}\right)^{-1} V(0) + \left(sI - D^T A D^{T\dagger}\right)^{-1} D^T (A + I) D\tilde{r} + \tilde{(}d_1).$$

Now apply the inverse Laplace transform. Note that by definition of matrix exponential,

$$\mathcal{L}^{-1} \left(sI - D^T A D^{T\dagger}\right)^{-1} = e^{\xi D^T A D^{T\dagger}}.$$

Therefore,

$$V(\xi) = e^{\xi D^T A D^{T\dagger}} V(0) + \mathcal{L}^{-1} \left[\left(sI - D^T A D^{T\dagger}\right)^{-1} D^T B\tilde{c}\right] + \mathcal{L}^{-1} \left[\left(sI - D^T A D^{T\dagger}\right)^{-1} D^T (A + I) D\tilde{r}\right]. \quad (5.4)$$

To simplify the second term of equation (5.4), use the convolution-product property of the Laplace transform to get

$$\mathcal{L}^{-1} \left[\left(sI - D^T A D^{T\dagger}\right)^{-1} D^T B\tilde{c}\right] = \mathcal{L}^{-1} \left[\left(sI - D^T A D^{T\dagger}\right)^{-1}\right] * \mathcal{L}^{-1} \left[D^T B\tilde{c}\right].$$

Note $c(\xi) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and $B = I$. Therefore,

$$\mathcal{L}^{-1} \left[D^T B\tilde{c}\right] = D^T Bc = d_1,$$

where d_1 is the first column of D. The entire second term in $V(\xi)$ above becomes

$$\mathcal{L}^{-1}\left[\left(sI - D^T A D^{T\dagger}\right)^{-1} D^T B\tilde{c}\right] = e^{\xi D^T A D^{T\dagger}} * d_1.$$

Evaluating the convolution, bring d_1 outside the integral, a linear operator:

$$e^{\xi D^T A D^{T\dagger}} * d_1(\xi) = \int_{\tau=-\infty}^{\infty} e^{(\xi-\tau) D^T A D^{T\dagger}}\, d\tau\, d_1.$$

The state $V(\xi)$ depends only on the past up to $V(0)$ so that $0 < \xi - \tau \leq \xi$:

$$e^{\xi D^T A D^{T\dagger}} * d_1(\xi) = \int_{\tau=0}^{\xi} e^{(\xi-\tau) D^T A D^{T\dagger}}\, d\tau\, d_1.$$

The integral of the matrix exponential $\int_{t=0}^{T} e^{tX}\, dt = X^{-1}\left(e^{Tx} - I\right)$. Thus,

$$\mathcal{L}^{-1}\left[\left(sI - D^T A D^{T\dagger}\right)^{-1} D^T B\tilde{c}\right] = \left(D^T A D^{T\dagger}\right)^{-1}\left(e^{\xi D^T A D^{T\dagger}} - I\right) d. \tag{5.5}$$

Note the notation $d = D^T Bc$. Looking at the final term of equation (5.4), assume the network estimate is periodic with period $\frac{1}{\phi}$, where ϕ is the unknown spike rate. Between spikes, the dynamics of $r(\xi)$ are known from equation (4.2) solved as

$$r(\xi) = e^{-\xi I} r(0), \quad 0 < \xi \leq \frac{1}{\phi}.$$

Hence,

$$\begin{aligned}
\mathcal{L}^{-1}\left[\left(sI - D^T A D^{T\dagger}\right)^{-1} D^T (A+I) D\tilde{r}\right] &= \mathcal{L}^{-1}\left[\left(sI - D^T A D^{T\dagger}\right)^{-1}\right] * \mathcal{L}^{-1}\left[D^T (A+I) D\tilde{r}\right] \\
&= e^{\xi D^T A D^{T\dagger}} * D^T (A+I) D\, e^{-\xi I}\, r(0) \\
&= e^{\xi D^T A D^{T\dagger}} * \left(D^T A D e^{-\xi I}\, r(0) + D^T D e^{-\xi I}\, r(0)\right) \\
&= e^{\xi D^T A D^{T\dagger}} * D^T A D e^{-\xi I}\, r(0) + e^{\xi D^T A D^{T\dagger}} * D^T D e^{-\xi I}\, r(0).
\end{aligned}$$

The two convolutions are nearly identical so we solve the simpler of the two:

$$e^{\xi D^T A D^{T\dagger}} * D^T D e^{-\xi I}\, r(0) = \int_{\tau=0}^{\xi} e^{(\xi-\tau) D^T A D^{T\dagger}} D^T D e^{-\tau I}\, r(0)\, d\tau.$$

Note that $e^{-\tau I}$ simplifies as

$$\begin{aligned}
e^{-\tau I} &= \sum_{k=0}^{\infty} \frac{(-\tau I)^k}{k!} \\
&= \left(\sum_{k=0}^{\infty} \frac{(-\tau^k)}{k!}\right) I \\
&= e^{-\tau} I.
\end{aligned}$$

The scalar and identity matrix can both move to the beginning of the integral and reformed into a matrix:

$$\begin{aligned}\int_{\tau=0}^{\xi} e^{(\xi-\tau)D^T A D^{T\dagger}} D^T D e^{-\tau\, I}\, r(0)\, d\tau &= \int_{\tau=0}^{\xi} e^{-\tau\, I} e^{(\xi-\tau)D^T A D^{T\dagger}} D^T D\, r(0)\, d\tau \\ &= e^{\xi D^T A D^{T\dagger}} \int_{\tau=0}^{\xi} e^{-\tau\left(I + D^T A D^{T\dagger}\right)}\, d\tau\ D^T D\, r(0) \\ &= e^{\xi D^T A D^{T\dagger}} \left(I + D^T A D^{T\dagger}\right)^{-1} \left(e^{\xi\left(I + D^T A D^{T\dagger}\right)} - I\right) D^T D\, r(0).\end{aligned}$$

From this expression it follows that

$$e^{\xi D^T A D^{T\dagger}} * D^T (A+I)\, D\, e^{-\xi\, I}\, r(0) = e^{\xi D^T A D^{T\dagger}} \left(I + D^T A D^{T\dagger}\right)^{-1} \left(e^{\xi\left(I + D^T A D^{T\dagger}\right)} - I\right) D^T (A+I) D\, r(0).$$

Hence,

$$\mathcal{L}^{-1}\left[\left(sI - D^T A D^{T\dagger}\right)^{-1} D^T (A+I)\, D\tilde{r}\right] = e^{\xi D^T A D^{T\dagger}} \left(I + D^T A D^{T\dagger}\right)^{-1} \left(e^{\xi\left(I + D^T A D^{T\dagger}\right)} - I\right) D^T (A+I) D\, r(0). \tag{5.6}$$

Using equations (5.5) and (5.6), the voltage trajectory equation (5.4) becomes

$$\begin{aligned}V(\xi) = &\\ &e^{\xi D^T A D^{T\dagger}} V(0) \\ &+ \left(D^T A D^{T\dagger}\right)^{-1} \left(e^{\xi D^T A D^{T\dagger}} - I\right) d \\ &+ e^{\xi D^T A D^{T\dagger}} \left(I + D^T A D^{T\dagger}\right)^{-1} \left(e^{\xi\left(I + D^T A D^{T\dagger}\right)} - I\right) D^T (A+I) D\, r(0).\end{aligned} \tag{5.7}$$

In the case $A = -I$, equation (5.7) simplifies considerably:

$$V(\xi) = e^{-\xi D^T D^{T\dagger}} V(0) + \left(D^T D^{T\dagger}\right)^{-1} \left(I - e^{-\xi D^T D^{T\dagger}}\right) d. \tag{5.8}$$

Simplify the matrix $D^T D^{T\dagger}$ via its SVD:

$$D^T = V \begin{bmatrix} S \\ 0 \end{bmatrix} \mathcal{U}^T,$$

$$\implies D^{T\dagger} = \mathcal{U} \begin{bmatrix} S & 0 \end{bmatrix} V^T$$

$$\implies D^T D^{T\dagger} = V \begin{bmatrix} S \\ 0 \end{bmatrix} \mathcal{U}^T \mathcal{U} \begin{bmatrix} S & 0 \end{bmatrix} V^T$$

$$= V \begin{bmatrix} S \\ 0 \end{bmatrix} \begin{bmatrix} S & 0 \end{bmatrix} V^T$$

$$= V \begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix} V^T$$

$$= \begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix} \in \mathbf{R}^{N \text{ x } N},$$

where I_d denotes the d-dimensional identity matrix. Equation (5.8) becomes

$$V(\xi) = e^{-\xi I_d} V(0) + \begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix}^{-1} \left(I - e^{-\xi I_d}\right) d.$$

The matrix $\begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix}$ is not invertible. Consider instead only the first d equations of the preceding system.

$$V_j(\xi) = e^{-\xi} V_j(0) + \left(1 - e^{-\xi}\right) d_j, \quad j = 1, \ldots, d.$$

Note that $d_j = d^T c$, and consider neuron $j_{max} = j$ the first to reach the spike threshold. Recall its initial condition $v(0) = -\frac{||d||^2}{2}$ to arrive at

$$V(\xi) = -e^{-\xi} \frac{||d||^2}{2} + \left(1 - e^{-\xi}\right) d^T c$$

Compare with the corresponding PCF trajectory, equation (5.1). The preceding equation rearranges to

$$V(\xi) = d^T c - e^{-\xi} \left(d^T c + \frac{||d||^2}{2}\right),$$

which is identical to equation (5.1). Since only one neuron spikes, $r(\xi)$ and thus $\hat{x}(\xi)$ are identical for both PCF and gap-junction networks. The preceding, somewhat painful analysis shows that if the PCF and gap-junction models begin on their steady-state trajectories with the same initial conditions, their network estimates are identical in time. The statement is limited to the case of a constant driving stimulus. It does not, for example, show which network reaches the steady state trajectory first.

5.3 Per-spike RMSE of the PCF and Gap-Junction Networks for a Constant Stimulus

Equation (5.3) gives both gap-junction and PCF trajectories. We compute the per-spike RMSE by the integral

$$RMSE_{spike} \triangleq \sqrt{\phi \int_0^{\frac{1}{\phi}} e^T e(\tau) \, \mathrm{d}\tau}.$$

The target dynamical system over this interval is $x(\xi) = \mathcal{U}_1$. Assuming the first spike is at $\xi_1 = 0$, we have

$$\begin{aligned} e(\xi) &= x(\xi) - \hat{x}(\xi) \\ &= \mathcal{U}_1 - \frac{d}{1 - e^{-\frac{1}{\phi}}} e^{-\xi} \\ \implies e^T e &= \mathcal{U}_1^T \mathcal{U}_1 - 2\mathcal{U}_1^T \frac{d}{1 - e^{-\frac{1}{\phi}}} e^{-\xi} + \frac{d^T d}{\left(1 - e^{-\frac{1}{\phi}}\right)^2} e^{-2\xi} \\ &= 1 - 2\frac{c^T d}{1 - e^{-\frac{1}{\phi}}} e^{-\xi} + \frac{d^T d}{\left(1 - e^{-\frac{1}{\phi}}\right)^2} e^{-2\xi}. \end{aligned}$$

Integrate over a spike interval to arrive at

$$\begin{aligned} \int_0^{\frac{1}{\phi}} e^T e(\tau)\, \mathrm{d}\tau &= \frac{1}{\phi} - 2\frac{c^T d}{1 - e^{-\frac{1}{\phi}}} \left(1 - e^{-\frac{1}{\phi}}\right) + \frac{d^T d}{\left(1 - e^{-\frac{1}{\phi}}\right)^2} \frac{1}{2} \left(1 - e^{-\frac{2}{\phi}}\right) \\ &= \frac{1}{\phi} - 2\, c^T d + \frac{||d||}{2} \frac{1 - e^{-\frac{2}{\phi}}}{\left(1 - e^{-\frac{1}{\phi}}\right)^2}. \end{aligned}$$

The per-spike RMSE of both PCF and Gap-Junction Networks is therefore

$$RMSE_{spike} = \sqrt{1 - 2\,\phi c^T d + \phi \frac{||d||^2}{2} \frac{1 - e^{-\frac{2}{\phi}}}{\left(1 - e^{-\frac{1}{\phi}}\right)^2}}.$$

To write the RMSE as a function of only firing rate ϕ, we invert equation (5.2) to obtain $d(\phi)$:

$$\frac{1}{\phi} = ln\left(d^T c + \frac{||d||^2}{2}\right) - ln\left(d^T c - \frac{||d||^2}{2}\right).$$

Note that $d^T c = d^T \begin{bmatrix} 1 \\ 0 \end{bmatrix} = d_0$. Because d is the most parallel vector to c, for large enough networks with uniformly distributed directions d, we have that $d = d_0 c$. This implies that $||d||^2 = d_0^2$. The preceding equation becomes

$$\frac{1}{\phi} = ln\left(d_0 + \frac{d_0^2}{2}\right) - ln\left(d_0 - \frac{d_0^2}{2}\right)$$

$$\implies e^{-\frac{1}{\phi}} = \frac{d_0 - \frac{d_0^2}{2}}{d_0 + \frac{d_0^2}{2}}$$

$$= \frac{1 - \frac{d_0}{2}}{1 + \frac{d_0}{2}}$$

$$\implies e^{-\frac{1}{\phi}} + e^{-\frac{1}{\phi}}\frac{d_0}{2} = 1 - \frac{d_0}{2}$$

$$\implies d_0\frac{1 + e^{-\frac{1}{\phi}}}{2} = 1 - e^{-\frac{1}{\phi}}$$

$$\implies d_0(\phi) = 2\,\frac{1 - e^{-\frac{1}{\phi}}}{1 + e^{-\frac{1}{\phi}}}$$

$$= 2\,\tanh\frac{1}{2\phi}$$

Thus the per-spike RMSE simplifies to

$$RMSE_{spike} = \sqrt{1 - 4\,\phi\tanh\frac{1}{2\phi} + 2\,\phi\tanh^2\frac{1}{2\phi}\frac{1 - e^{-\frac{2}{\phi}}}{\left(1 - e^{-\frac{1}{\phi}}\right)^2}}. \tag{5.9}$$

5.4 Comparison of Self-Coupled, Gap-Junction, and PCF Networks for a Constant Stimulus

We now compare all three models as they respond to a constant driving stimulus, while varying their firing rate.

Let the parameters be

$$A = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$c(\xi) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$D = \mathcal{U} \begin{bmatrix} S & 0 \end{bmatrix} V^T = \mathcal{U} \begin{bmatrix} I_d & 0 \end{bmatrix} I_N,$$

$$d\xi = 10^{-4},$$

$$N = 8,$$

$$x(0) = \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix}.$$

We simulate the self-coupled, PCF, and gap-junction networks and compare their derived estimates given by equations (**??**) and (5.3) respectively. Figure (9) shows the network estimates of each model for the above parameters. We see that all trajectories are identical as predicted by our derived estimates, shown by the dotted line.

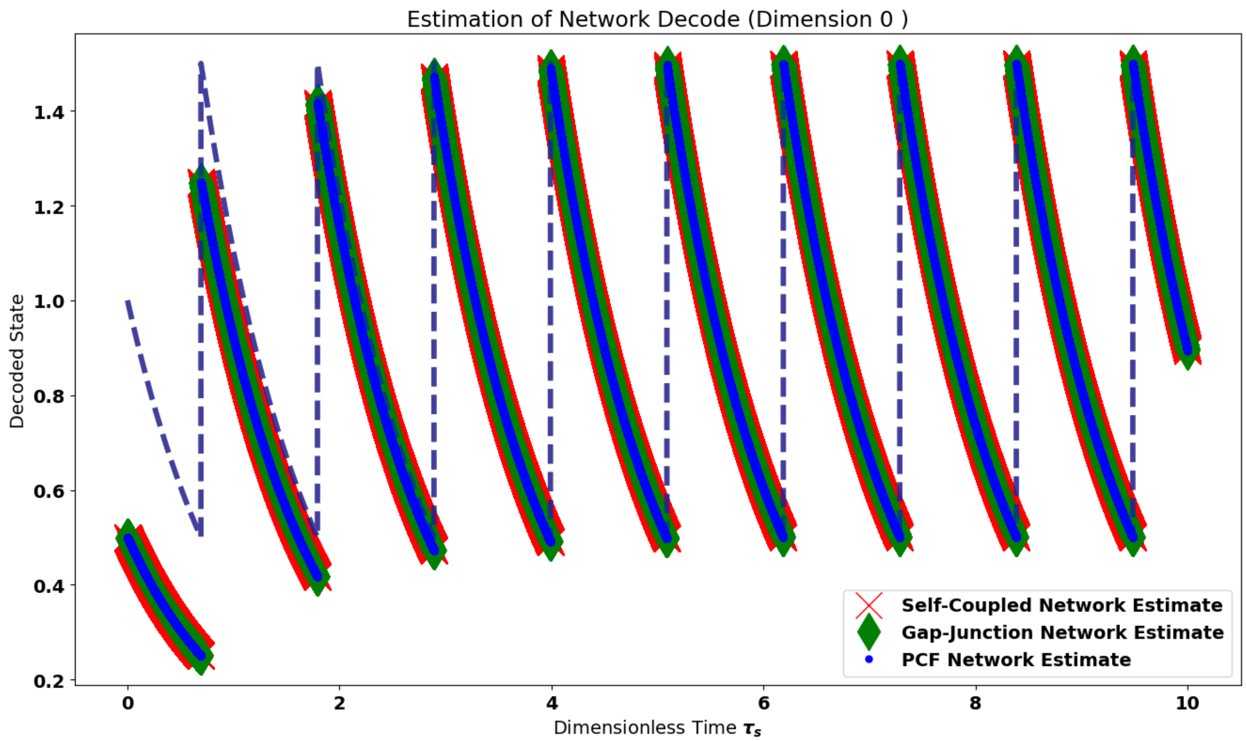

Figure 9: Simulation of self-coupled, gap-junction, and PCF networks. Network readouts for each are plotted. The dotted line is the derived expression(s) given by equations (**??**) and (5.3). Note that all three trajectories are identical.

Next we plot the per-spike RMSE of each model for the same parameters while varying the spike rate. Figure (10) shows the numerically measured per-spike RMSE for each model and their derived expressions, equations (**??**) and (5.9). As the firing rate approaches the simulation timestep, 10^4, the curves deviate from the derived expression.

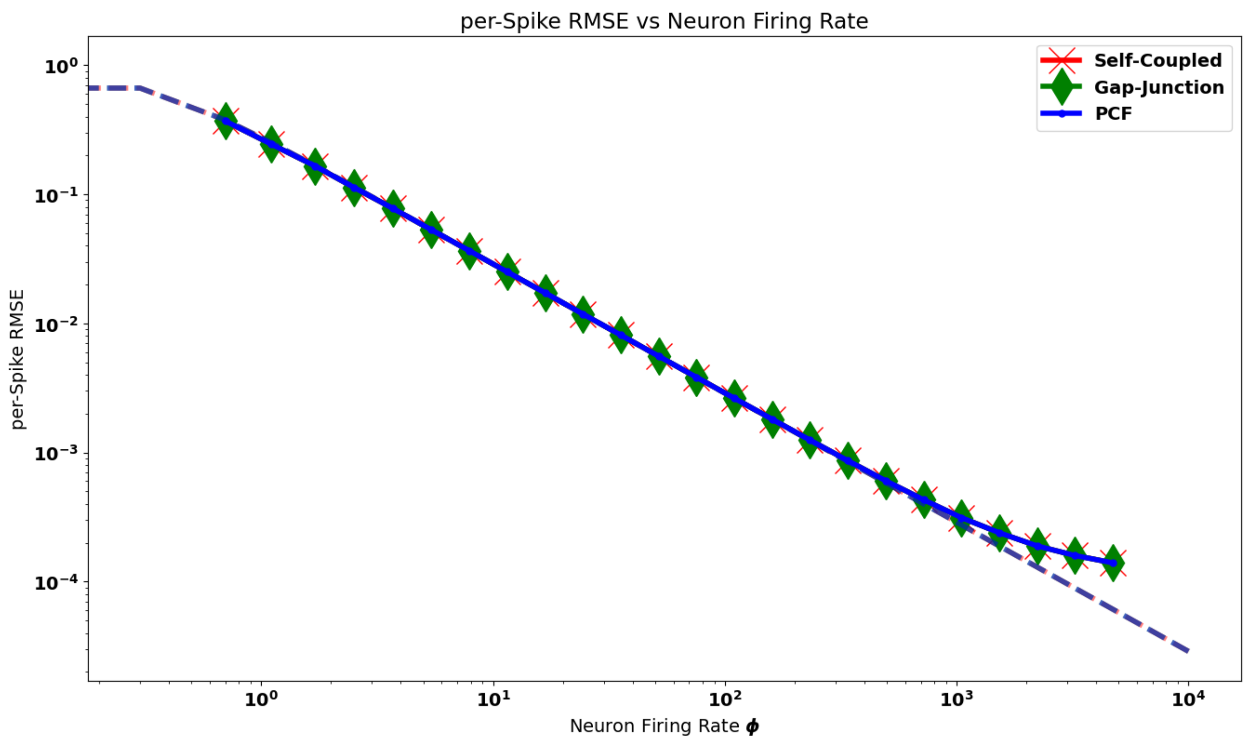

Figure 10: Simulated per-spike RMSE for self-coupled, gap-junction, and PCF networks. The dotted lines are the derived expression for each model given by given by equations (**??**) for the self-coupled and (5.9) for the gap-junction and PCF models respectively. Spike rates were estimated numerically by dividing the number of spikes by the simulation length. The RMSE was computed numerically by the discrete integral $R\hat{M}SE = \sqrt{\hat{\phi} \sum_{\tau \text{ between spikes}} e(\xi)^T e(\xi) \; d\xi}$. All computations used the numerically estimated spike rate.

6 Extensions of the Basic Model

The basic model presented in section (2) currently has the following limitations

1. Dynamical systems with complex eigenvalues are ill-defined
2. Only $2d$ of N neurons have nontrivial dynamics.
3. Network dynamics are inherently periodic and deterministic, not asynchronous.

We extend the basic model to address these limitations.

6.1 Dynamical Systems with Complex Eigenvalues

1. Recall the basic self-coupled network equations:

$$\dot{v} = \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} v + \begin{bmatrix} S\left(\Lambda + I_d\right) S & 0 \\ 0 & 0 \end{bmatrix} \rho + \beta \tilde{c} - \begin{bmatrix} S^2 & 0 \\ 0 & 0 \end{bmatrix} \tilde{o},$$

$$\dot{\rho} = -\rho + \tilde{o},$$

$$\hat{y} = \begin{bmatrix} S & 0 \end{bmatrix} \rho.$$

When the dynamical system $\dot{x} = Ax + Bc$ is oscillatory, the eigenvalues Λ of $A = \mathcal{U}\,\Lambda\,\mathcal{U}^T$ are complex, implying $\dot{v}$ is a system of complex equations. Currently, all network quantities are only defined for real-values. Here we generalize the self-coupled network to complex vector space so that it is well defined when A has complex eigenvalues.

The existence of complex eigenvalues implies that x is an element of a complex vector space $\mathbf{C}^d$. The spectral theorem assumes as much when proving the existence of an eigendecomposition for $A = \mathcal{U}\Lambda\mathcal{U}^T$. If we otherwise restrict x to $\mathbf{R}^d$, the eigendecomposition A would exist i.f.f. A was symmetric, i.e. $A = A^T$.

Complex quantities are often simpler to manipulate in polar coordinates than Cartesian, so we use them here. For any complex scalar $\alpha \in \mathbf{C}$, the relation between polar and Cartesian coordinates is

$$\alpha = a + ib = \mu e^{i\theta},$$

where

$$\mu = \sqrt{a^2 + b^2},$$

$$\theta = \tan^{-1} \frac{b}{a},$$

$$a + ib = \mu\, cos\, \theta + i\, \mu\, sin\, \theta = e^{i\theta}.$$

2. Let $\bar{\Lambda}$ denote the polar representation of Λ, the eigenvalues of A.

$$\bar{\Lambda}_j \triangleq \mu_j\, e^{i\omega_j},$$

where

$$\omega_j = tan^{-1}\frac{\Re\Lambda_j}{\Im\Lambda_j},$$

and

$$\mu_j = \sqrt{\Re\Lambda_j^2 + \Im\Lambda_j^2}.$$

A's eigenvalues are

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 & \dots & 0 & 0 \\ 0 & \Lambda_2 & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & \Lambda_d \end{bmatrix} = \begin{bmatrix} \mu_1 \, e^{i\omega_1} & 0 & \dots & 0 & 0 \\ 0 & \mu_2 \, e^{i\omega_2} & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & \mu_d \, e^{i\omega_d} \end{bmatrix} = \bar{\Lambda}.$$

Transforming the eigenvector, we denote the k^{th} complex component of $\mathcal{U}_j$ by $u^{\Re}_{kj} + i\, u^{\Im}_{kj}$. Let W_j denote the polar representation of $\mathcal{U}_j$.

$$\mathcal{U}_j = \begin{bmatrix} u^{\Re}_{1j} + iu^{\Im}_{1j} \\ \vdots \\ u^{\Re}_{dj} + iu^{\Im}_{dj} \end{bmatrix} = \begin{bmatrix} \alpha_{1j} e^{i\theta_{1j}} \\ \vdots \\ \alpha_{dj} e^{i\theta_{dj}} \end{bmatrix} = W_j,$$

where

$$\alpha_{ij} = \sqrt{(u^{\Re}_{ij})^2 + (u^{\Im}_{ij})^2},$$

and

$$\theta_{ij} = tan^{-1}\frac{u^{\Im}_{ij}}{u^{\Re}_{ij}}.$$

We now write A as

$$A = W\bar{\Lambda}W^*.$$

3. The given decoder matrix D maps integrated spikes from $\mathbf{R}^N$ to the network estimate $\hat{x} \in \mathbf{R}^d$. View $D = \mathcal{U}\begin{bmatrix} S & 0 \end{bmatrix} V^T$ as a sequence of linear maps between vector spaces as in figure (11). For complex $\hat{x}$, we require that D map from $\mathbf{C}^N$ to $\mathbf{C}^d$. By assumption, D and A share a common left basis, which is now $W \in \mathbf{C}^{d\times d}$. However the remaining real-valued matrices S, V discard any complex components, limiting the span of D to the complex basis W scaled only by real coefficients, as in figure (12). To ensure that D spans $\mathbf{C}^d$ and not a real-coefficient subspace, we extend S and V to the complex domain. Denote these complexified matrices by $\bar{S}$ and $\bar{V}$ respectively.

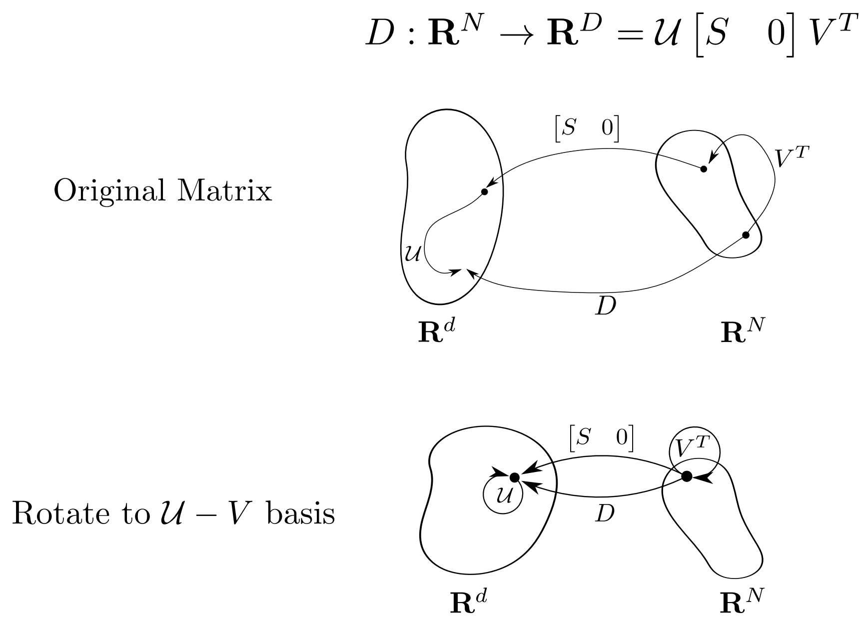

Figure 11: Visualizing D as a sequence of linear maps between subspaces. ***Top:*** The matrix $D \in \mathbf{R}^{d \times N}$ is decomposed via SVD into a sequence of 3 linear maps (matrices). The rightmost matrix $V^T \in \mathbf{R}^{N \times N}$ projects a vector x to give coefficients for the expansion in the basis V. The center matrix $\begin{bmatrix} S & 0 \end{bmatrix} \in \mathbf{R}^{d \times N}$ maps vectors from the V basis to a vector in $\mathbf{R}^d$ by scaling and truncation. The leftmost matrix $\mathcal{U} \in \mathbf{R}^{d \times d}$ gives the resultant vector $Dx \in \mathbf{R}^d$ by using the scaled vector $\begin{bmatrix} S & 0 \end{bmatrix} V^T$ as coefficients for a basis expansion in $\mathcal{U}$. ***Bottom:*** We rotate the basis for vectors in $\mathbf{R}^N$ and $\mathbf{R}^d$ to the $\mathcal{U}$ and V bases respectively. This negates the need of D to preemptively project and afterward rotate a vector, leaving only scaling by a diagonal matrix. The mapping D performs on a vector y simplifies to multiplication by a diagonal matrix S of y's first d components.

$$D : \mathbf{C}^N \rightarrow \mathbf{C}^d = W \begin{bmatrix} S & 0 \end{bmatrix} V^T$$

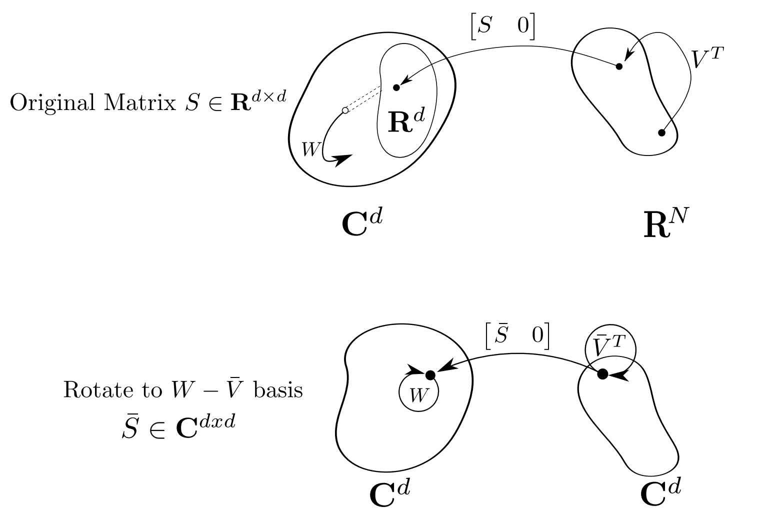

Figure 12: D projects vectors to $\mathbf{R}^d$ before expansion in the basis W. ***Top:*** The matrix $D = W \begin{bmatrix} S & 0 \end{bmatrix} V^T$ shares a complex left-basis W with A. However the remaining matrices S and V are real-valued. This limits the range of D to real linear combinations of the basis W. ***Bottom:*** We complexify $S \rightarrow \bar{S}$ and $V \rightarrow \bar{V}$ and rotate network quantities to the $W - \bar{V}$ basis. This ensures that the $span(D) = \mathbf{C}^d$.

A principled method of extending real functions in $\mathbf{R}^d$ to the complex plane $\mathbf{C}^d$ is to use the Hilbert Transform. Suppose our real function is $f(x)$ and we wish to find $g(x)$ such that $h(x) = f(x) + ig(x)$. The Hilbert transform of f gives a g such that h has at least two useful properties. 1) $h(x)$ is complex-differentiable in x; 2) The Fourier transform of h has no negative frequency components, which respects the conjugate-symmetric spectrum of the real-valued f.

Let $H_x(f)$ denote the Hilbert Transform of the function f over the domain x, i.e.

$$H_x(f) \triangleq \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{f(y)}{x-y} dy = f * \frac{1}{\pi x}(x).$$

We complexify the matrix D by applying the Hilbert transform to the columns of S and of V over the domain $k \in [1, \dots, d]$; i.e

$$\bar{S} = \begin{bmatrix} H_k(S_1) & \dots & H_k(S_d) \end{bmatrix} = H_k(S),$$

$$\bar{V} = \begin{bmatrix} H_k(V_1) & \dots & H_k(V_N) \end{bmatrix} = H_k(V).$$

Here $k \in \mathbf{Z}$ is a discrete domain of indices, so we use the discrete version of the Hilbert transform.

4. The neurons which implement the network have spike trains $\tilde{o}(\xi) \in \mathbf{C}^N$:

$$\tilde{o}_j(\xi) = \sum_{k=1}^{n_j \text{ spikes}} \delta(\xi - \xi_j^k) + i0,$$

which have no imaginary component.

The network's estimate is

$$\hat{y}(\xi) = \begin{bmatrix} \bar{S} & 0 \end{bmatrix} \rho(\xi),$$

for $\rho = \bar{V}^T r \in \mathbf{C}^N$ where

$$\frac{d\rho}{d\xi} = -\rho + o(\xi).$$

The network error $\epsilon \in \mathbf{C}^d$ is

$$\epsilon = y - \hat{y} = W^* e.$$

5. Before deriving the network voltage dynamics, we redefine voltage as complex using network optimization. Consider the previous optimization from which voltage was defined

$$\mathcal{L}(\xi) = ||y(\xi + d\xi) - \hat{y}(\xi + d\xi)||^2 = \epsilon^T \epsilon \in \mathbf{R}.$$

The network optimized the Euclidean norm of two real vectors given by the inner product of the error with itself. We generlize to complex vectors by using the complex inner product. I.e,

$$\mathcal{L}(\xi) = \epsilon^* \epsilon,$$

where $*$ denotes the Hermitian transpose.

When neuron j spikes, the vector $\bar{S}_j$ is added to the network estimate so that the objective is

$$\begin{aligned}
\mathcal{L}_{sp} &= (y - \hat{y} - \bar{S}_j)^*(y - \hat{y} - \bar{S}_j) \\
&= y^* y - 2y^*\hat{y} + \hat{y}^*\hat{y} - 2\bar{S}_j^*(y - \hat{y}) + \bar{S}_j{}^* \bar{S}_j \\
&= \mathcal{L}_{ns} - 2\bar{S}_j^*(y - \hat{y}) + \bar{S}_j{}^* \bar{S}_j.
\end{aligned}$$

We would like to say the spiking condition $\mathcal{L}_{sp} < \mathcal{L}_{ns}$ is then

$$\bar{S}_j{}^{*}\epsilon > \frac{\bar{S}_j^{*}\bar{S}_j}{2},$$

however the above equation is between two complex scalars, which have no well-defined ordering. Instead, we take the modulus of either side, which returns a real scalar. This gives the spike condition

$$|\bar{S}_j^{*}\epsilon| > \frac{|\bar{S}_j|^2}{2}.$$

This suggests that complex voltage is suitably defined by $\bar{S}_j^{*}\epsilon$ so that neuron j spikes when

$$|v_j| > \frac{\bar{S}_j}{2}.$$

6. We now derive the voltage dynamics as before. The rotated target dynamical system is

$$\dot{y} = \bar{\Lambda} y + \beta \tilde{c},$$

where

$$\beta = W^* B W,$$

$$\tilde{c} = W^* c.$$

The error has dynamics

$$\begin{aligned}\dot{\epsilon} &= \dot{y} - \dot{\hat{y}} \\ &= \bar{\Lambda} y + \beta \tilde{c} - \begin{bmatrix} S & 0 \end{bmatrix} \dot{\rho} \\ &= \bar{\Lambda} y + \beta \tilde{c} + \begin{bmatrix} S & 0 \end{bmatrix} \rho - \begin{bmatrix} S & 0 \end{bmatrix} \tilde{o}.\end{aligned}$$

Apply the matrix $\begin{bmatrix} \bar{S} \\ 0 \end{bmatrix}^*$ to both sides to get the voltage dynamics. We write the full set of N equations as

$$\dot{v} = \begin{bmatrix} \bar{\Lambda} & 0 \\ 0 & 0 \end{bmatrix} v + \begin{bmatrix} \bar{S}^* \left(I + \bar{\Lambda} \right) \bar{S} & 0 \\ 0 & 0 \end{bmatrix} \rho + \beta \tilde{c} - \begin{bmatrix} \bar{S}^* \bar{S} & 0 \\ 0 & 0 \end{bmatrix} \tilde{o}.$$

To summarize, the self-coupled network model is extended to complex-valued dynamical systems by the following:

1. Factorize $A = \mathcal{U} \Lambda \mathcal{U}^T$, by assumption Λ contains complex entries. Rewrite this matrix as $A = W^T \bar{\Lambda} W$ so that $\bar{\Lambda}$ contains only real entries.

2. The voltage contains the sum of real and imaginary components of the error projected onto the rotated (now complex) basis W.

3.