Research Notes on Self-Coupled Spiking Neural Networks

Chris Fritz

July 2020

Contents

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 $D \in \mathbf{R}^{d\ x\ N}$ specifying The tuning curve 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 maintains numerical accuracy over time.
2. ***Tuning Curve Rotation*** To most efficiently use N neurons, we use orthogonal bases to choose tuning cures. The decoder and dynamics matrices D, A are rotated to a common orthonormal basis in d-dimensional space via singular value decomposition (SVD).

   The rotation eliminates off-diagonal elements of the network connectivity matrices. This decoupling prevents numerical instability when spikes simultaneously occur. Multiple neurons may now spike in the same simulation time step whereas Deneve networks forbid simultaneous spiking.

   Two neurons per dimension 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 stochastic thinning of the post-synaptic potential (PSP) at every synaptic connection. The pre-synaptic neuron's membrane potential is still reset by an action potential.
4. ***Dimensionless Time*** We describe both the network and target system in dimensionless time. Time is normalized by the neuron's synaptic time constant, τ_s. This dimensionless representation ensures consistent numerical simulation independent of simulation timestep. Furthermore, the PSP decay rate is implicitly specified as 1, reducing the required simulation parameters.

2 Derivation: Basic Model

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 train 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 $D \in \mathbf{R}^{d \text{ x } N}$ 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 $\dot{e}$:

$$\begin{aligned}
\dot{e} &= \dot{x} - \dot{\hat{x}} \\
&= (Ax + Bc) - (Do - \hat{x}) \\
&= A\left(e + \hat{x}\right) + Bc - Do + \hat{x} \\
&= Ae + (A + I)\hat{x} + Bc - Do \\
&= Ae + (A + I)\left(Dr\right) + Bc - Do \\
\implies A^{-1}\dot{e} &= e + (I + A^{-1})Dr + A^{-1}Bc - A^{-1}Do \\
\implies D^T A^{-1}\dot{e} &= D^T e + D^T(I + A^{-1})Dr + D^T A^{-1}Bc - D^T A^{-1}Do
\end{aligned} \tag{2.5}$$

where the third equality follows from equation (2.4) and the fifth from equation (2.2).

3. Assuming both D and A are full rank, diagonalize each to a common left basis:

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

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

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

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

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

To express equation (2.5) with the $\mathcal{U}$ and V bases, first note

$$\begin{aligned} D^T A^{-1} &= V \begin{bmatrix} S \\ 0 \end{bmatrix} \mathcal{U}^T \mathcal{U} \Lambda^{-1} \mathcal{U}^T \\ &= V \begin{bmatrix} S \\ 0 \end{bmatrix} \Lambda^{-1} \mathcal{U}^T \\ &= \sum_{j=1}^{d} \frac{S_j}{\Lambda_j} V_j \mathcal{U}_j^T, \end{aligned}$$

and

$$\begin{aligned} D^T A^{-1} D &= V \begin{bmatrix} S \\ 0 \end{bmatrix} \mathcal{U}^T \mathcal{U} \Lambda^{-1} \mathcal{U}^T \mathcal{U}\,[S \; 0]\,V^T \\ &= V \begin{bmatrix} S \\ 0 \end{bmatrix} \Lambda^{-1}\,[S \; 0]\,V^T \\ &= \sum_{j=1}^{d} \frac{S_j^2}{\Lambda_j} V_j V_j^T. \end{aligned}$$

Consequently,

$$\sum_{j=1}^{d} \frac{S_j}{\Lambda_j} V_j \mathcal{U}_j^T \dot{e} = \sum_{j=1}^{d} S_j V_j \mathcal{U}_j^T e + \sum_{j=1}^{d} S_j^2 (1 + \Lambda_j^{-1}) V_j V_j^T r + \sum_{j=1}^{d} \frac{S_j}{\Lambda_j} V_j \mathcal{U}_j^T Bc - \sum_{j=1}^{d} \frac{S_j^2}{\Lambda_j} V_j V_j^T o. \qquad (2.6)$$

Left-multiply both sides of the preceding equation by V_j^T to arrive at the system of equations

$$\frac{S_j}{\Lambda_j}\mathcal{U}_j^T\dot{e} = S_j\mathcal{U}_j^T e + S_j^2(1+\Lambda_j^{-1})V_j^T r + S_j\Lambda_j^{-1}\mathcal{U}_j^T Bc - S_j^2\Lambda_j^{-1}V_j^T o$$

$$\implies S_j\mathcal{U}_j^T\dot{e} = S_j\Lambda_j\mathcal{U}_j^T e + S_j^2(\Lambda_j+1)V_j^T r + S_j\mathcal{U}_j^T Bc - S_j^2 V_j^T o,$$

for $j = 1, \ldots, d$.

4. To simplify notation, we note that our preceding left-multiply has transformed the equations to the basis V^T. The transformed neuron's spike train, membrane voltage, PSC, and input matrix are respectively:

$$\begin{aligned}
\Omega_j &\triangleq S_j^2 V_j^T o \\
v_j &\triangleq S_j\mathcal{U}_j^T e, \\
\rho_j &\triangleq S_j^2 V_j^T r \\
\beta_j &\triangleq S_j\mathcal{U}_j^T B.
\end{aligned} \tag{2.7}$$

The system of equations simplifies to the membrane voltage dynamics

$$\dot{v}_j = \Lambda_j v_j + (\Lambda_j + 1)\rho_j + \beta_j c - \Omega_j,$$

or in matrix form,

$$\dot{v} = \Lambda v + (\Lambda + I)\rho + \beta c - \Omega. \tag{2.8}$$

Here, v is a d vector which describes the dynamics of the d-neurons needed to implement the dynamical system. The remaining $N - d$ neurons are unused and do not contribute to the network readout at present.
From equation (2.3) the PSC dynamics are

$$\dot{\rho} = -\rho + \Omega. \tag{2.9}$$

Similar to equation (2.8), ρ describes a d-vector.

5. The spike trains $\Omega(\xi)$ are chosen minimize the network estimation error

$$\mathcal{L}(\xi) = ||x(\xi + d\xi) - \hat{x}(\xi + d\xi)||^2. \tag{2.10}$$

The network greedily minimizes $\mathcal{L}$ an instant $d\xi$ ahead in time. Writing $\hat{x}$ in terms of Ω and ρ, equations (2.2) and (2.7) imply

$$
\begin{aligned}
\hat{x} &= Dr \\
&= \sum_{j=1}^{d} S_j \mathcal{U}_j V_j^T r \\
&= \sum_{j=1}^{d} (S_j^{-1} S_j^2) \mathcal{U}_j V_j^T r \\
&= \sum_{j=1}^{d} (S_j^{-1} \mathcal{U}_j)(S_j^2 V_j^T r) \\
&= \sum_{j=1}^{d} (S_j^{-1} \mathcal{U}_j) \rho_j \\
&= \mathcal{U} S^{-1} \rho \\
&= \Delta \rho,
\end{aligned}
$$

Where

$$\Delta \triangleq \mathcal{U} S^{-1}. \tag{2.11}$$

If neuron j does not spike, the objective is

$$\mathcal{L}_{ns} = ||x - \hat{x}||^2$$

If neuron j spikes at time ξ, then $\Omega \leftarrow \Omega + \hat{e}_j$. The estimate $\hat{x}$ is updated so that the objective is now

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

where Δ_j is the the j^{th} column of Δ. A spike occurs when it lowers the objective more than not spiking. Our spiking condition is therefore

$$\mathcal{L}_{sp} < \mathcal{L}_{ns}$$

$$\implies 2\Delta_j^T \left(\hat{x} - x \right) + \Delta_j^T \Delta_j < 0$$

$$\implies \Delta_j^T \left(x - \hat{x} \right) > \frac{\Delta_j^T \Delta_j}{2}$$

$$\implies \Delta_j^T e > \frac{\Delta_j^T \Delta_j}{2}.$$

Note $\Delta_j = \mathcal{U}_j S_j^{-1}$ so that

$$\Delta_j^T \Delta_j = S_j^{-1} \mathcal{U}_j^T \mathcal{U}_j S_j^{-1}$$

$$= S_j^{-2}$$

$$\implies S_j^{-1} \mathcal{U}_j^T e > \frac{S_j^{-2}}{2}$$
$$\implies S_j \mathcal{U}_j^T e > \frac{1}{2}$$
$$\implies v_j > \frac{1}{2}, \frac{1}{2} + v_{th}$$

where the last inequality follows from applying the voltage definition from equation (2.7). Thus neuron j spikes when its membrane voltage v_j exceeds the threshold of $\frac{1}{2}$. Consequently the spiking behavior of each neuron in the network is given by

$$\begin{aligned} & v^{th} = \frac{1}{2} 1_N, \\ & \text{if } v_j > v_j^{th}, \\ & \text{then } v_j^{'} = v_j - \int \delta(\tau) d\tau, \\ & \text{and } \rho_j^{'} = \rho_j + \int \delta(\tau) d\tau \end{aligned} \tag{2.12}$$

where

$$v_j^{'} = lim_{\xi \to \xi_{spike}^{+}} v_j(\xi),$$

$$\rho_j^{'} = lim_{\xi \to \xi_{spike}^{+}} \rho_j(\xi),$$

and 1_N is the N-vector with entries 1.

6. Equations (2.8), (2.9), and (2.12) describe how we implement a network with d neurons that produces an accurate estimate $\hat{x}$ of the given target system.

   When neuron j spikes, a vector $\Delta_j = S_j^{-1} \mathcal{U}_j$ is added to the network estimate, $\hat{x}$. 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 $\frac{1}{2^d}$ of the desired state-space. To remove this restriction, we add an additional d neurons whose tuning curves $\mathcal{U}_j$ are anti-parallel to neurons j for $j = 1, \ldots, d$. Such vectors are required in order to allow subtraction, defined as addition of the additive inverse. 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:

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

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

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

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

and afterward recompute $\beta \in \mathbf{R}^{2d \text{ x } d}$ and $\Delta \in \mathbf{R}^{d \text{ x } 2d}$.

Simulation of Basic Equations

Here we simulate the above equations (2.8), (2.9), and (2.12) with the $N = 2d$ neurons. The parameters are

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

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

$$c(\xi) = 10 \begin{bmatrix} cos(\frac{\pi}{4}\xi) \\ sin(\frac{\pi}{4}\xi) \end{bmatrix} \tag{2.13}$$

$$D_{\text{ij}} \sim \mathcal{N}(0,1) \text{ Columns Normalized to Unit Length}$$

$$d\xi = 10^{-6},$$

$$N = 4,$$

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

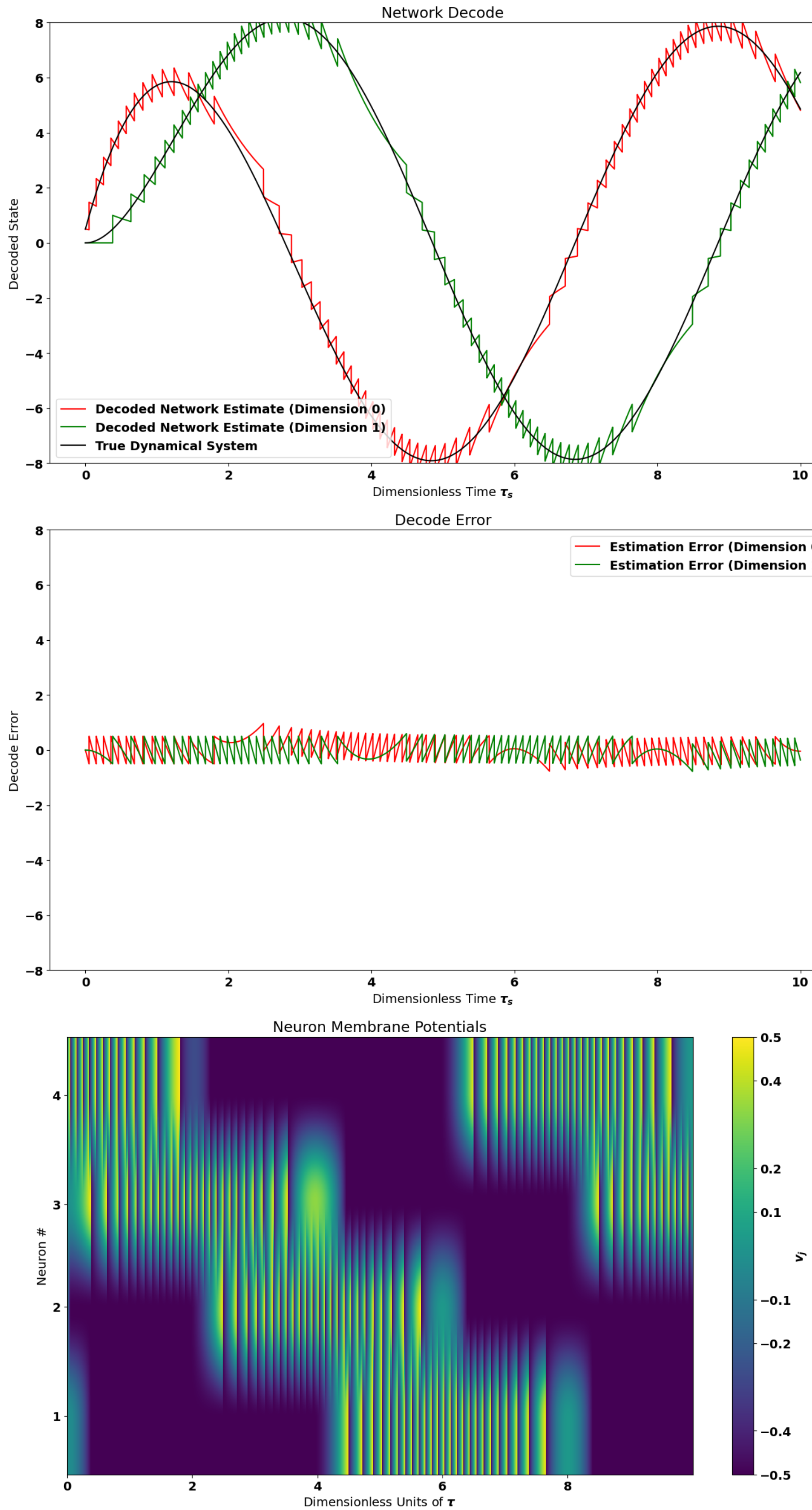

Network Decode
Decoded State
Dimensionless Time τ_s
Decoded Network Estimate (Dimension 0)
Decoded Network Estimate (Dimension 1)
True Dynamical System
Decode Error
Decode Error
Dimensionless Time τ_s
Estimation Error (Dimension 0)
Estimation Error (Dimension 1)
Neuron Membrane Potentials
Neuron #
Dimensionless Units of τ
v_j

Figure 1: Simulation of equations (2.8), (2.9), and (2.12), with parameters listed in equation (2.13). ***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 spike trains Ω handled separately. After integrating over a timestep, all neurons above threshold were manually reset according to the spiking rule (2.12). 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 (2.8), (2.9), and (2.12) for the case of a constant (in time) driving force $c(\xi) = k\mathcal{U}_j$. First we derive explicit expressions for the network estimate, then we compute the resulting RMSE for various driving strengths k.

1. Let

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

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

$$c(\xi) = k\mathcal{U}_1$$

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

$$N = 4,$$

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

   With the given initial conditions, $v_j = 0$ for $j \neq 1$ for all ξ. The dynamics simplify to

$$\dot{v_1} = \Lambda_1 v_1 + (\Lambda_1 + 1)\rho_1 + k - \Omega_1.$$

2. We assume that the decoding matrix D is chosen such that $S_1 = 1$. Because A is the negative identity matrix, it is also clear that $\Lambda_1 = -1$. The preceding equation simplifies to

$$\dot{v_1} = -v_1 + k - \Omega_1, \tag{3.1}$$

   which is a form of the well-known Leaky Integrate-and-Fire (LIF) model. Assuming a spike has occurred at $v_{th} = \frac{1}{2}$, the voltage has just been reset so that $v_1(0) = -\frac{1}{2}$. Until the next spike, the neuron's trajectory is integrated as

$$v(\xi) = k - e^{-\xi}(k + \frac{1}{2}).$$

   Neglecting any spike reset, the voltage will asymptotically approach $v_1 = k$. Thus for any spiking to occur, we must have $k \geq v_{th}$. In this case, the time required to reach a spike threshold v_{th} is

$$v_{th} = k - e^{-\xi_{spike}}(k + \frac{1}{2})$$

$$\implies e^{-\xi_{spike}} = \frac{k - v_{th}}{k + \frac{1}{2}}$$

$$= \frac{1 - \frac{v_{th}}{k}}{1 + \frac{1}{2k}}$$

$$\implies \xi_{spike} = -ln\left(\frac{1 - \frac{v_{th}}{k}}{1 + \frac{1}{2k}}\right)$$

$$\implies \frac{1}{\xi_{spike}} = -\frac{1}{ln\left(\frac{1 - \frac{v_{th}}{k}}{1 + \frac{1}{2k}}\right)}$$

$$= \frac{1}{ln\left(1 + \frac{1}{2k}\right) - ln\left(1 - \frac{v_{th}}{k}\right)},$$

which determines the frequency at which the LIF neuron spikes. Denote this frequency as a function of driving strength k by $\phi(k)$:

$$\phi(k) \triangleq \frac{1}{ln\left(1 + \frac{1}{2k}\right) - ln\left(1 - \frac{v_{th}}{k}\right)}. \tag{3.2}$$

The network will encode the constant driving force by spiking at a fixed rate determined by equation (3.2). Figure (2) shows a plot of equation (3.2) along with numerically computed spike rates for a simulated network driven with constant drive strength k. Similar to membrane voltage, the resulting PSC and readout dynamics are reduced to one neuron periodically spiking:

$$\dot{\rho_1} = -\rho_1 + \Omega_1$$

$$\implies \dot{\hat{x}} = -\Delta_1 \rho_1 + \Delta_1 \Omega_1$$

$$= -\hat{x} + \mathcal{U}_1 \Omega_1.$$

3. The spike train Ω_1 is a periodic sequence of impulses spaced in time by $\frac{1}{\phi(k)}$. Hence $\Omega_1(\xi) = \sum_{l=0}^{\infty} \delta\left(\xi - \frac{l}{\phi(k)}\right)$. The network estimate therefore has dynamics

$$\dot{\hat{x}} = -\hat{x} + \mathcal{U}_1 \sum_{l=0}^{\infty} \delta\left(\xi - \frac{l}{\phi(k)}\right). \tag{3.3}$$

The target dynamical system is

$$\dot{x} = -x + k\mathcal{U}_1$$
$$x(0) = \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix},$$

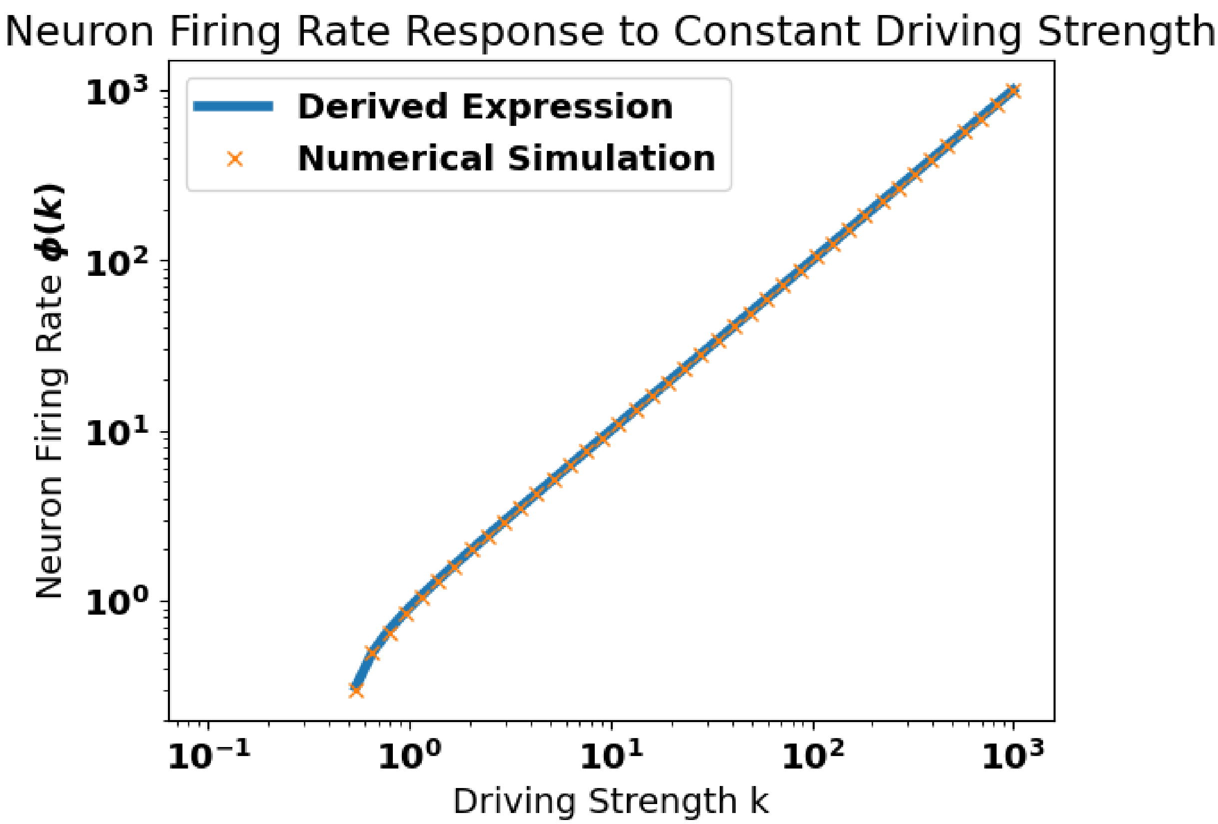


Figure 2: A log-log plot of equation (3.2) alongside the rates measured from numerical simulations. The simulation parameters are described at the beginning of this section, with the decoder matrix D chosen to be the first d rows of the N x N identity matrix. This ensures the singular values $S_j = 1$ as assumed in the derivation. The rate was measured as the number of spike resets divided by the duration of the simulation.

which has a stable fixed point at

$$x = k\mathcal{U}_1. \tag{3.4}$$

4. Equation (3.3) implies that the network estimate $\hat{x}$ will decay until the first spike ξ_1^1 occurs:

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

At this instant, the vector $\mathcal{U}_1$ is added to the network estimate.

$$\hat{x}(\frac{1}{\phi(k)}) = x(0)e^{-\frac{1}{\phi(k)}} + \mathcal{U}_1.$$

Decay again occurs until the next spike

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

$$= \left(x(0)e^{-\frac{1}{\phi(k)}} + \mathcal{U}_1\right)e^{-(\xi - \frac{1}{\phi(k)})}, \quad \frac{1}{\phi(k)} \le \xi < \frac{2}{\phi(k)}$$

$$\implies \hat{x}(\frac{2}{\phi(k)}) = \left(x(0)e^{-\frac{1}{\phi(k)}} + \mathcal{U}_1\right)e^{-(\frac{1}{\phi(k)})} + \mathcal{U}_1$$

$$= x(0)e^{-\frac{2}{\phi(k)}} + \mathcal{U}_1 e^{-\frac{1}{\phi(k)}} + \mathcal{U}_1.$$

The third spike more clearly shows the recursive behavior

$$\hat{x}(\frac{3}{\phi(k)}) = \left[x(0)e^{-\frac{2}{\phi(k)}} + \mathcal{U}_1 e^{-\frac{1}{\phi(k)}} + \mathcal{U}_1\right]e^{-\frac{1}{\phi(k)}} + \mathcal{U}_1$$

$$= x(0)e^{-\frac{3}{\phi(k)}} + \mathcal{U}_1 e^{-\frac{2}{\phi(k)}} + \mathcal{U}_1 e^{-\frac{1}{\phi(k)}} + \mathcal{U}_1$$

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

$$\hat{x}(\frac{n}{\phi(k)}) = \sum_{l=0}^{n-1} \mathcal{U}_1 e^{-\frac{l}{\phi(k)}}$$

$$= \mathcal{U}_1 \frac{1 - e^{-\frac{n}{\phi(k)}}}{1 - e^{-\frac{1}{\phi(k)}}}.$$

For sufficiently large n, this converges to

$$\hat{x}(\xi_1^n) = \frac{\mathcal{U}_1}{1 - e^{-\frac{1}{\phi(k)}}}. \tag{3.5}$$

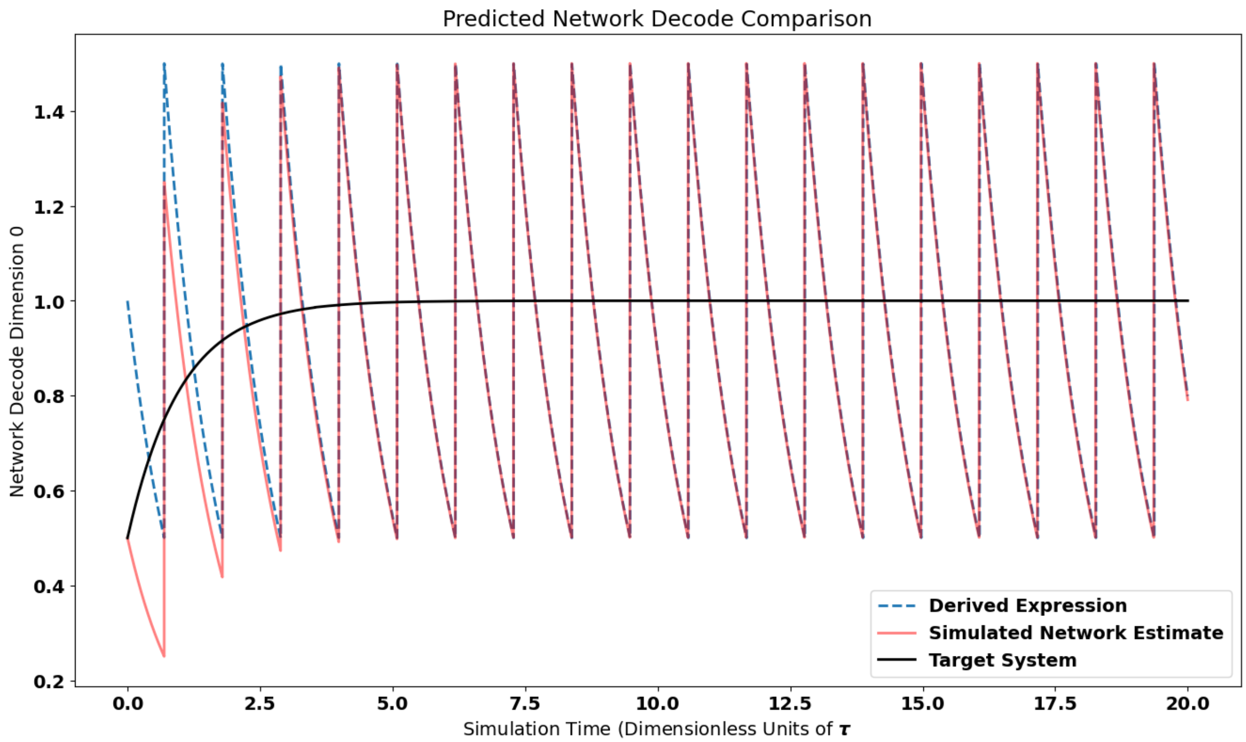


Figure 3: Comparison of the derived long-term network estimate equation (3.6) to numerical simulation. Parameters are the same as the previous figure, with $k = 1$.

5. The preceding argument states that after a transient interval, the network estimate at any spike time ξ_1^n is given by equation (3.5). As shown in figure (3), this convergence occurs after roughly 5 spikes for the case $k = 1$.

   We know from equation (3.3) that the estimate will decay exponentially from this value over an interval $\frac{1}{\phi(k)}$ until a spike returns it returns to the initial value. Thus the network estimate between two consecutive spikes is given by

$$\hat{x}(\xi) = \frac{\mathcal{U}_1}{1 - e^{-\frac{1}{\phi(k)}}} e^{-(\xi - \xi_1^n)}, \quad 0 \leq \xi - \xi_1^n < \frac{1}{\phi(k)}.$$

   Combine this expression with equation (3.5), we have an explicit expression for the long-term behavior of the network estimate given by

$$\hat{x}(\xi) = \frac{\mathcal{U}_1}{1 - e^{-\frac{1}{\phi(k)}}} e^{-(\xi - \xi_1^1) \mod \frac{1}{\phi(k)}}, \tag{3.6}$$

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

6. Assume the true system dynamics have settled to their fixed point $x = k\mathcal{U}_1$. From equation (3.6) the network estimate $\hat{x}$ and therefore error $e = x - \hat{x}$ is a periodic function of ξ with period $\frac{1}{\phi(k)}$. 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

$$RMSE_{spike} \triangleq \sqrt{\phi(k) \int_0^{\frac{1}{\phi(k)}} ||e(\tau)||^2 \, \mathrm{d}\tau}. \tag{3.7}$$

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

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

   Therefore the integral is

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

   The per-spike RMSE of the network estimate as a function of drive strength k is therefore

$$RMSE_{spike}(k) = \sqrt{k^2 + \phi(k) \left[\frac{1 - e^{-\frac{2}{\phi(k)}}}{2\left(1 - e^{-\frac{1}{\phi(k)}}\right)^2} - 2k \right]}. \tag{3.8}$$

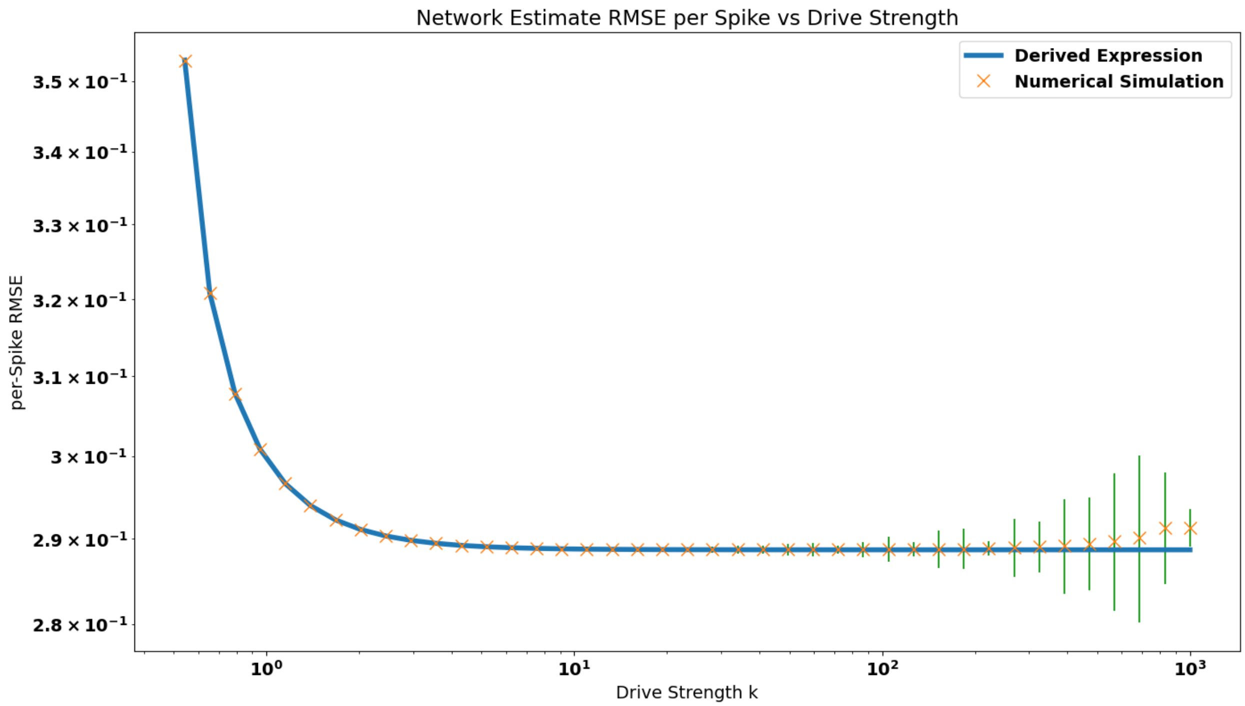
   To write the RMSE explicitly as a function of firing rate $\phi(k)$, we invert equation (3.2) to obtain

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

Substitute this for k to obtain

$$
\begin{aligned}
RMSE_{spike}(\phi) &= \sqrt{k(\phi)^2 + \phi \left[\frac{1 - e^{-\frac{2}{\phi}}}{2\left(1 - e^{-\frac{1}{\phi}}\right)^2} - 2k(\phi) \right]} \\
&= \sqrt{\left(\frac{v_{th} + \frac{e^{-\frac{1}{\phi}}}{2}}{1 - e^{-\frac{1}{\phi}}} \right)^2 + \phi \left[\frac{1 - e^{-\frac{2}{\phi}}}{2\left(1 - e^{-\frac{1}{\phi}}\right)^2} - 2\,\frac{v_{th} + \frac{e^{-\frac{1}{\phi}}}{2}}{1 - e^{-\frac{1}{\phi}}} \right]}.
\end{aligned}
\tag{3.9}
$$

Equations (3.8) and (3.9) are plotted in figure (4). 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.



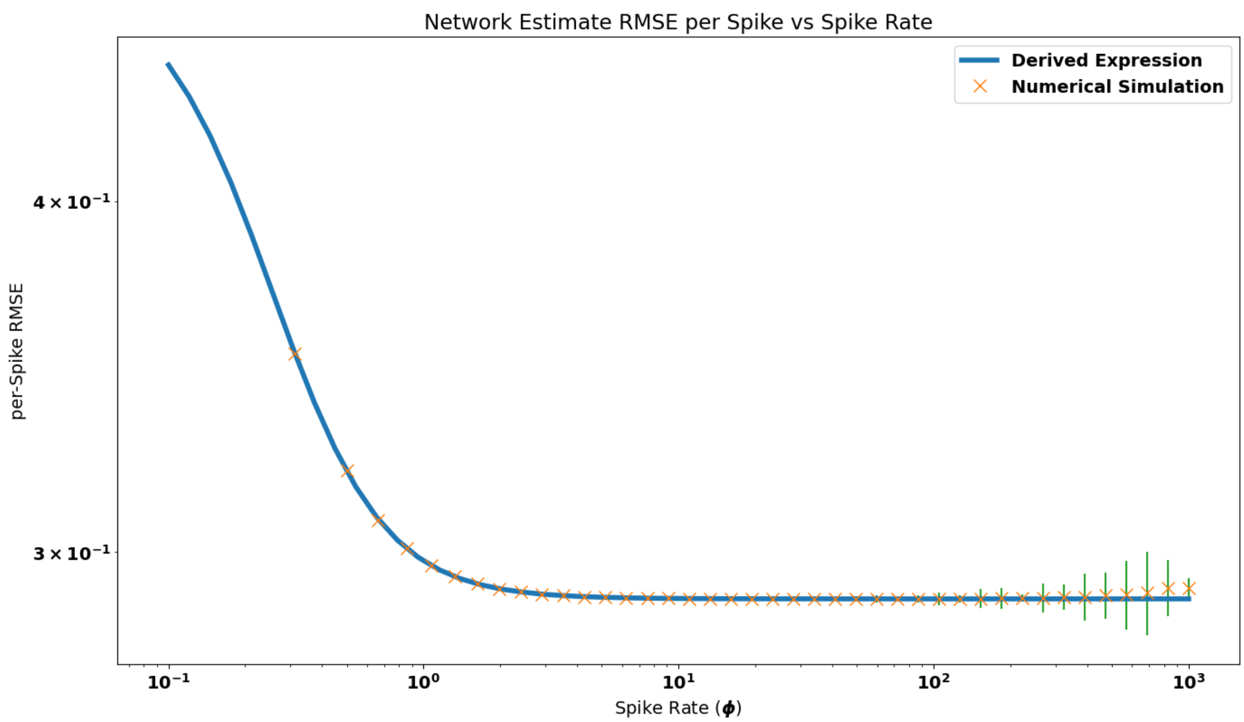


Figure 4: ***Top:*** A log-log plot of equation (3.8). ***Bottom:*** A log-log plot of equation (3.9). ***Both:*** Each simulated data point is the RMSE averaged over all inter-spike intervals in a simulation of length $T = 80\tau_s$ at a constant (in time) drive strength. Between simulations, the spike 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: $R\hat{M}SE = \sqrt{\hat{\phi} \sum_{\tau \text{ between spikes}} e(\xi)^T e(\xi)\, d\xi}$. The increase in standard deviation is due to finite approximation error from numerical integration.

4 Analysis: RMSE vs Spike Rate for a Fixed Dynamical System

Here we derive the RMSE of a signal representing a given dynamical system with a varying spike rate. The RMSE is computed over a constant interval of time for a fixed target system while the spike rate is varied.

1. Our system is described by

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

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

$$c(\xi) = \mathcal{U}_1$$

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

$$N = 4,$$

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

2. With the given initial conditions, $v_j = 0$ for $j \neq 1$ for all ξ. From equations (2.7) and (2.8), the dynamics simplify to

$$\dot{v_1} = \Lambda_1 v_1 + (\Lambda_1 + 1)\rho_1 + S_1 \mathcal{U}_1^T \mathcal{U}_1 - \Omega_1.$$

It is clear that $\Lambda_1 = -1$ so that

$$\dot{v_1} = -v_1 + S_1 - \Omega_1, \tag{4.1}$$

which is a form of the well-known Leaky Integrate-and-Fire (LIF) model.

3. Assuming a spike has occurred at $v_{th} = \frac{1}{2}$, the voltage has just been reset so that $v_1(0) = -\frac{1}{2}$. Until the next spike, the neuron's trajectory is integrated as

$$v(\xi) = S_1 - e^{-\xi}(S_1 + \frac{1}{2}).$$

The spike occurs at $v(\xi_{spike}) = v_{th}$ or

$$v_{th} = S_1 - e^{-\xi}(S_1 + \frac{1}{2})$$

$$\implies \frac{S_1 - v_{th}}{S_1 + \frac{1}{2}} = e^{-\xi_{spike}}$$

$$\implies \xi_{spike} = ln(S_1 + \frac{1}{2}) - ln(S_1 - v_{th}).$$

The inverse of the preceding expression gives the firing rate of the neuron,

$$\phi(S_1) \triangleq \frac{1}{ln(1+\frac{1}{2\,S_1}) - ln(1-\frac{v_{th}}{S_1})}. \tag{4.2}$$

Equation (4.2) describes the neuron's firing rate as a function of the decode matrix D's singular values. Thus for a given target dynamical system, the decode matrix D determines the neuron's firing rates.

4. From equations (2.9), (2.2), and (2.11),

$$\dot{\rho} = -\rho + \Omega$$

$$\implies \dot{\hat{x}} = -\hat{x} + \Delta\Omega$$

$$\implies \dot{\hat{x}} = -\hat{x} + S_1^{-1}\mathcal{U}_1 \sum_{l=0}^{\infty} \delta\left(\xi - \frac{l}{\phi}\right), \tag{4.3}$$

where the last equality follows from the periodicity of the LIF neuron firing at rate ϕ.

We solve equation (4.3) and inductively derive an explicit expression for its asymptotic behavior in time. Note that equation (4.3) implies that the network estimate $\hat{x}$ will decay until the first spike ξ_1^1 occurs:

$$\hat{x}(\xi) = x(0)e^{-\xi}, \quad 0 \leq \xi < \xi_1^1.$$

At this instant, the scaled vector $S_1^{-1}\mathcal{U}_1$ is added to the network estimate,

$$\hat{x}(\xi_1^1) = x(0)e^{-\xi_1^1} + S_1^{-1}\mathcal{U}_1.$$

Decay again occurs until the next spike

$$\hat{x}(\xi) = \hat{x}(\xi_1^1)e^{-(\xi-\xi_1^1)},$$

$$= \left(x(0)e^{-\xi_1^1} + S_1^{-1}\mathcal{U}_1\right)e^{-(\xi-\xi_1^1)}, \quad 0 \leq \xi - \xi_1^1 < \frac{1}{\phi}$$

$$\implies \hat{x}(\xi_1^1 + \frac{1}{\phi}) = \left(x(0)e^{-\xi_1^1} + S_1^{-1}\mathcal{U}_1\right)e^{-\frac{1}{\phi}} + S_1^{-1}\mathcal{U}_1$$

$$= x(0)e^{-\left(\xi_1^1+\frac{1}{\phi}\right)} + S_1^{-1}\mathcal{U}_1 e^{-\frac{1}{\phi}} + S_1^{-1}\mathcal{U}_1.$$

The third spike more clearly shows the recursive behavior

$$\hat{x}(\xi_1^1 + \frac{2}{\phi}) = \left[x(0)e^{-\left(\xi_1^1+\frac{1}{\phi}\right)} + S_1^{-1}\mathcal{U}_1 e^{-\frac{1}{\phi}} + S_1^{-1}\mathcal{U}_1.\right]e^{-\frac{1}{\phi}} + S_1^{-1}\mathcal{U}_1$$

$$= x(0)e^{-\left(\xi_1^1+\frac{2}{\phi}\right)} + S_1^{-1}\mathcal{U}_1 e^{-\frac{2}{\phi}} + S_1^{-1}\mathcal{U}_1 e^{-\frac{1}{\phi}} + S_1^{-1}\mathcal{U}_1.$$

Let us consider the n^{th} spike sufficiently far from $\xi = 0$ such that the transient term $x(0)e^{-\left(\xi_1^1+\frac{n-1}{\phi}\right)}$ can be neglected. This leads to the expression

$$\hat{x}\left(\xi_1^1 + \frac{n}{\phi}\right) = \sum_{l=0}^{n-1} S_1^{-1}\mathcal{U}_1 e^{-\frac{l}{\phi}}$$

$$= S_1^{-1}\mathcal{U}_1 \frac{1 - e^{-\frac{n}{\phi}}}{1 - e^{-\frac{1}{\phi}}}.$$

For sufficiently large n, this converges to

$$\hat{x}\left(\xi_1^1 + \xi_1^n\right) = \frac{S_1^{-1}\mathcal{U}_1}{1 - e^{-\frac{1}{\phi}}}.$$

Between two spikes, the dynamics are exponential decay

$$\hat{x}\left(\xi\right) = \frac{S_1^{-1}\mathcal{U}_1}{1 - e^{-\frac{1}{\phi}}} e^{-\left(\xi - \xi_1^n\right)}, \quad 0 \leq \xi - \xi_1^n < \frac{1}{\phi},$$

so that the long term network estimate is

$$\hat{x}\left(\xi\right) = \frac{S_1^{-1}\mathcal{U}_1}{1 - e^{-\frac{1}{\phi}}} e^{-\left(\xi - \xi_1^1\right) \mod \frac{1}{\phi}}.$$

Applying equation (4.2),

$$e^{-\frac{1}{\phi}} = e^{ln\left(1-\frac{v_{th}}{S_1}\right)-ln\left(1+\frac{1}{2\,S_1}\right)}$$

$$= \frac{1-\frac{v_{th}}{S_1}}{1+\frac{1}{2\,S_1}}$$

$$\implies 1-e^{-\frac{1}{\phi}} = 1-\frac{1-\frac{v_{th}}{S_1}}{1+\frac{1}{2\,S_1}}$$

$$= \frac{1+\frac{1}{2\,S_1}-1+\frac{v_{th}}{S_1}}{1+\frac{1}{2\,S_1}}$$

$$= \frac{\frac{1}{S_1}\left(\frac{1}{2}+v_{th}\right)}{1+\frac{1}{2\,S_1}}$$

$$= \frac{\frac{1}{2}+v_{th}}{S_1+\frac{1}{2}}s$$

$$\implies \frac{S_1^{-1}}{1-e^{-\frac{1}{\phi}}} = \frac{1}{S_1}\frac{S_1+\frac{1}{2}}{\frac{1}{2}+v_{th}}$$

$$= \frac{1+\frac{1}{2\,S_1}}{\frac{1}{2}+v_{th}}$$

$$= 1+\frac{1}{2\,S_1},$$

where the last equality uses the fact that $v_{th} = \frac{1}{2}$ from equation (2.12). The network estimate is therefore

$$\hat{x}(\xi) = \left(1+\frac{1}{2\,S_1}\right)e^{-\ \left(\xi-\xi_1^1\right) \mod \frac{1}{\phi}}\,\mathcal{U}_1 \tag{4.4}$$

Equation (4.4) is plotted in figure (5). The trajectories converge indefinitely at $\tau \simeq 5$.

5. Suppose the systems have settled so that equation (4.4) holds. To compute the RMSE of the estimate, consider the interval between two successive spikes. The RMSE over this period is

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

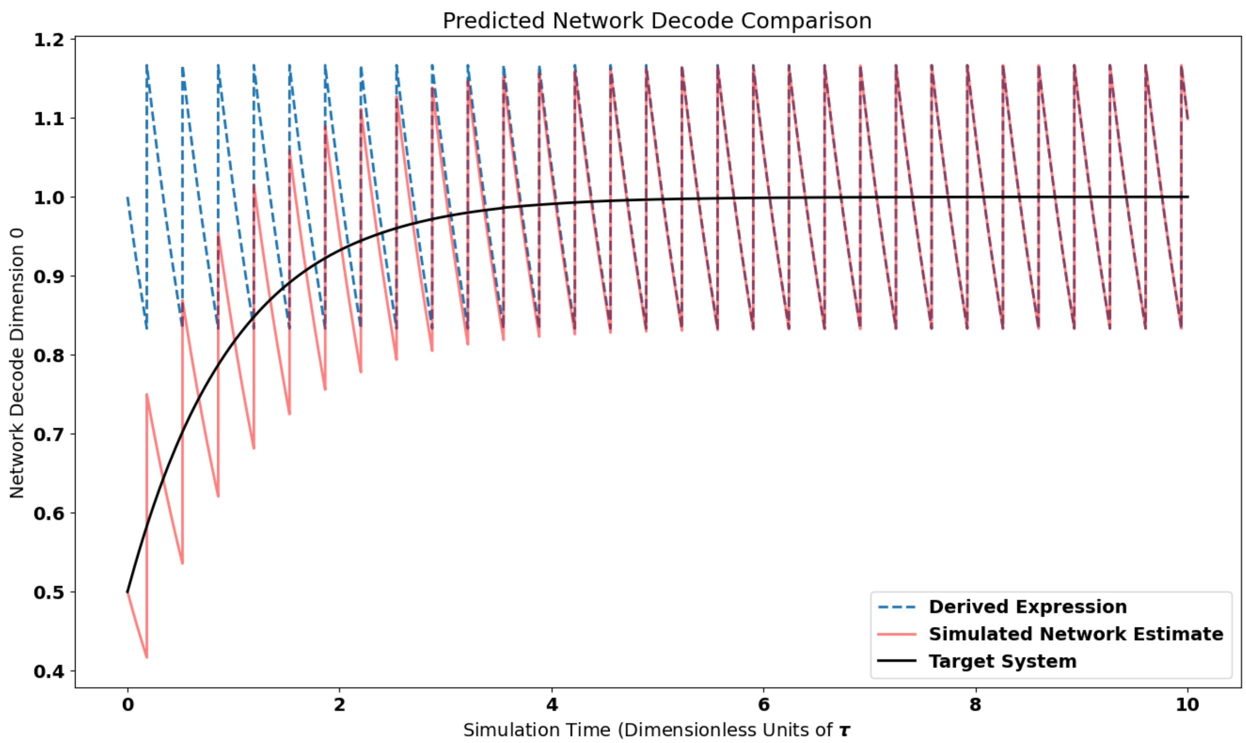


Figure 5: Comparison of the derived long-term network estimate equation (4.4) to numerical simulation. The simulation parameters are described at the beginning of this section, with the decoder matrix D chosen to be the first $d = 2$ rows of the N x N identity matrix, scaled by 3. This ensures the singular value $S_1 = 3$. Using the rate computed by equation (4.2), the derived estimate is computed then overlaid on the numerical simulation after offsetting time by the first spike arrival, ξ_1^1.

Note that the target dynamical system settles to a fixed point $x = \mathcal{U}_1$ so that

$$
\begin{aligned}
e(\tau) &= x(\tau) - \hat{x}(\tau) \\
&= \mathcal{U}_1 - \left(1 + \frac{1}{2\,S_1}\right) e^{-\tau}\,\mathcal{U}_1 \\
&= \mathcal{U}_1 \left[1 - e^{-\tau}\left(1 + \frac{1}{2\,S_1}\right)\right] \\
\implies e^T e\,(\tau) &= \left[1 - e^{-\tau}\left(1 + \frac{1}{2\,S_1}\right)\right]^2 \\
&= 1 - 2\,e^{-\tau}\left(1 + \frac{1}{2\,S_1}\right) + e^{-2\,\tau}\left(1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2}\right).
\end{aligned}
$$

The integral is therefore

$$
\begin{aligned}
\int_0^{\frac{1}{\phi}} e^T e(\tau)\,\mathrm{d}\tau &= \frac{1}{\phi} - 2\left(1 + \frac{1}{2\,S_1}\right)\frac{\frac{1}{2} + v_{th}}{S_1 + \frac{1}{2}} + \frac{1}{2}\left(1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2}\right)\left(1 - e^{\frac{2}{\phi}}\right) \\
&= \frac{1}{\phi} - 2\frac{1}{S_1}\left(S_1 + \frac{1}{2}\right)\frac{\frac{1}{2} + v_{th}}{S_1 + \frac{1}{2}} + \frac{1}{2}\left(1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2}\right)\left(1 - e^{\frac{2}{\phi}}\right) \\
&= \frac{1}{\phi} - \frac{1 + 2\,v_{th}}{S_1} + \frac{1}{2}\left(1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2}\right)\left(1 - e^{\frac{2}{\phi}}\right), \\
&= \frac{1}{\phi} - \frac{1 + 2\,v_{th}}{S_1} + \frac{1}{2}\frac{1}{S_1}\left(1 + \frac{1}{4\,S_1} + 2\,v_{th} - \frac{v_{th}^2}{S_1}\right) \\
&= \frac{1}{\phi} - \frac{1 + 2\,v_{th} - \frac{1}{S_1}\left(\frac{1}{4} - v_{th}^2\right)}{2\,S_1} \\
&= \frac{1}{\phi} - \frac{1}{S_1},
\end{aligned}
$$

where we have used the earlier result

$$
\frac{S_1^{-1}}{1 - e^{\frac{1}{\phi}}} = 1 + \frac{1}{2\,S_1},
$$

and

$$e^{-\frac{2}{\phi}} = \frac{\left(1 - \frac{v_{th}}{S_1}\right)^2}{\left(1 + \frac{1}{2S_1}\right)^2}$$

$$= \frac{1 - 2\,\frac{v_{th}}{S_1} + \frac{v_{th}^2}{S_1^2}}{1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2}}$$

$$\implies 1 - e^{-\frac{2}{\phi}} = \frac{1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2} - 1 + 2\,\frac{v_{th}}{S_1} - \frac{v_{th}^2}{S_1^2}}{1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2}}$$

$$= \frac{\frac{1}{S_1}\left(1 + \frac{1}{4\,S_1} + 2\,v_{th} - \frac{v_{th}^2}{S_1}\right)}{1 + \frac{1}{S_1} + \frac{1}{4\,S_1^2}}.$$

Consequently the per-spike RMSE of the network estimate is given by

$$RMSE_{spike}(s, \phi(s)) = \sqrt{1 - \frac{\phi}{S_1}}. \tag{4.5}$$

To write the above equation as a function of only ϕ, we invert equation (4.2) to obtain

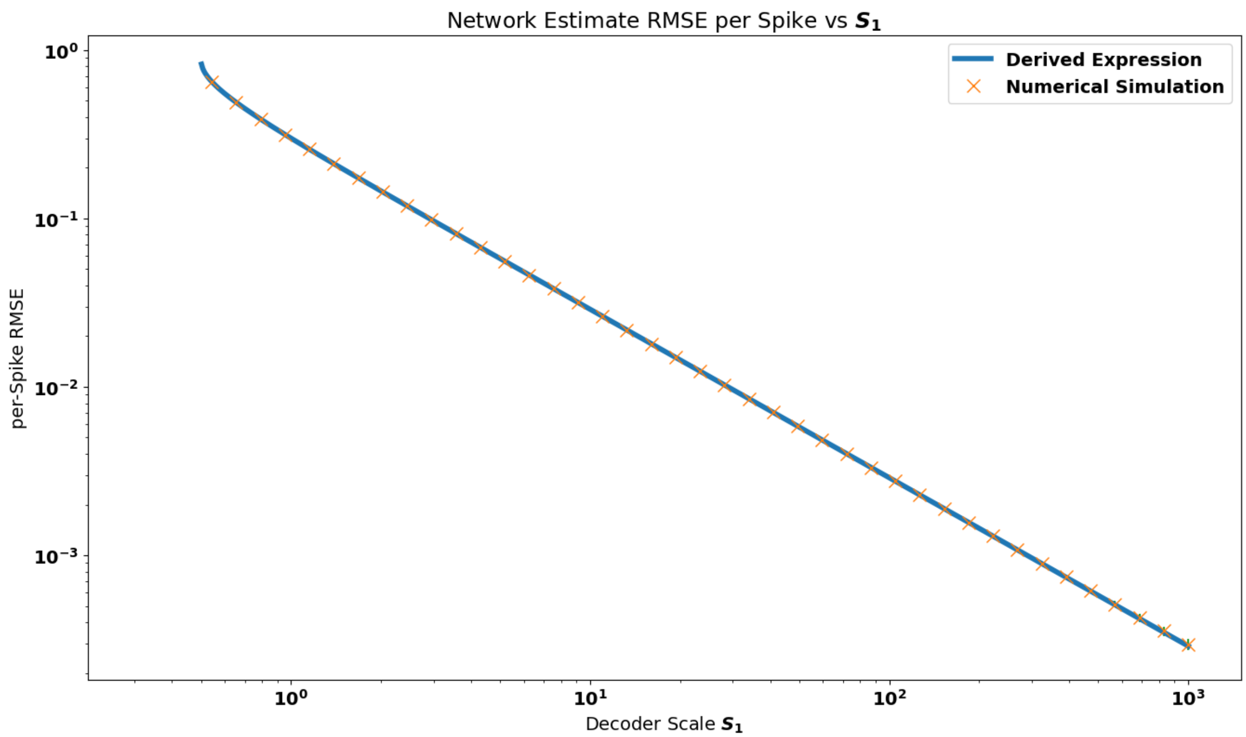
$$S_1(\phi) = \frac{v_{th} + \frac{e^{-\frac{1}{\phi}}}{2}}{1 - e^{-\frac{1}{\phi}}}$$

$$\implies RMSE_{spike} = \sqrt{1 - \phi\left(\frac{1 - e^{-\frac{1}{\phi}}}{v_{th} + \frac{1}{2}e^{-\frac{1}{\phi}}}\right)}$$

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

$$= \sqrt{1 - 2\phi\tanh\frac{1}{2\,\phi}}. \tag{4.6}$$

The preceding equation is plotted in figure (6). Above spikes rates of $\phi = 1$, the relationship is linearly decreasing on a logarithmic scale.



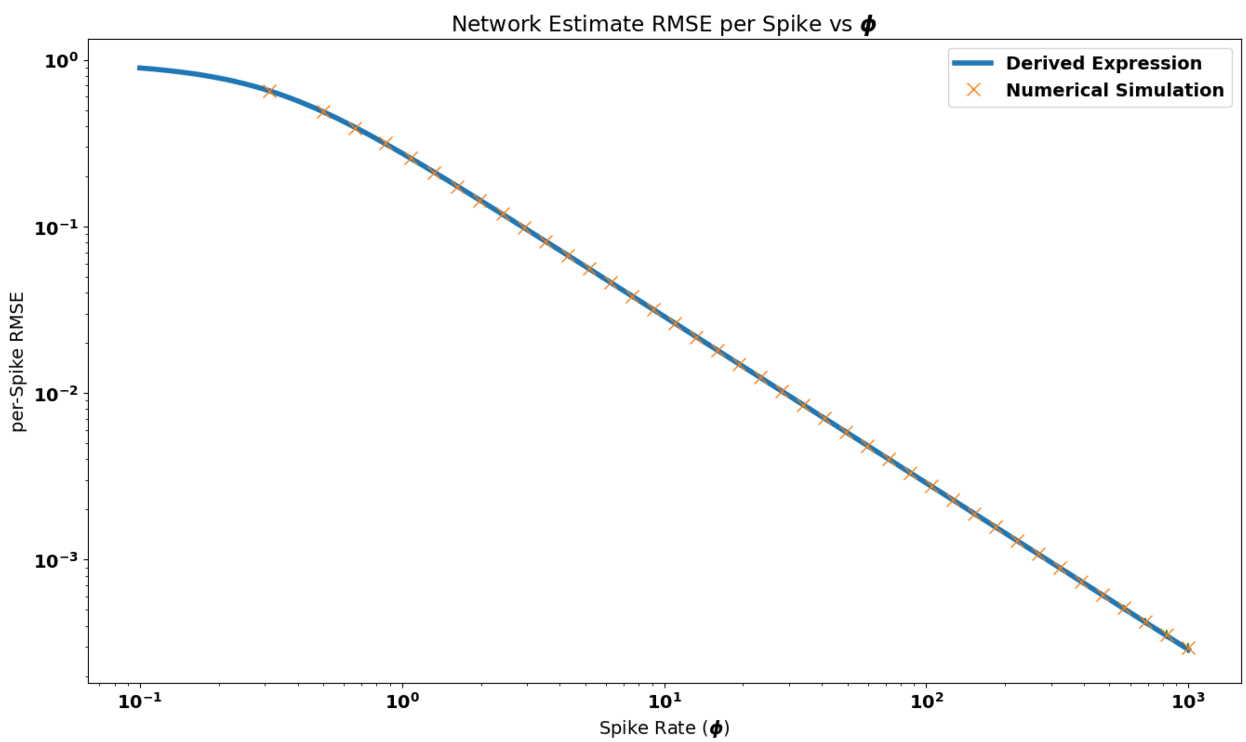


Figure 6: ***Top:*** A log-log plot of equation (4.5) compared with numerical measurements. ***Bottom:*** A log-log plot of equation (4.6). ***Both:*** Each simulated data point is the RMSE averaged over all inter-spike intervals in a simulation of length $T = 80\tau_s$ with $d\xi = 10^{-4}$. Green vertical lines visible towards the larger values are +/- 1 standard deviation over the number of inter-spike intervals in a given simulation. 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: $R\hat{M}SE = \sqrt{\hat{\phi} \sum_{\tau \text{ between spikes}} e(\xi)^T e(\xi) \, d\xi}$.

5 Analysis: Comparison with Predictive Coding Framework and Gap-Junction Coupling

Here we compare the self-coupled model with the predictive coding framework (PCF) as defined in Boerlin & Deneve, 2013. We demonstrate that an assumption in this model leads to a network estimate whose accuracy indefinitely diverges with time. The correction of this assumption leads to an intermittent model featuring membrane voltage coupling we loosely term gap-junction coupling. We show that the gap-junction model rectifies network estimation error introduced by the PCF. We then show that the corrected gap-junction model and the self-coupled mode produce identical results.

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 we are given in dimensionless form

   - 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

   $$o_j(\xi) \triangleq \sum_{k=0}^{\infty} \delta(\xi - \xi_j^k), \tag{5.1}$$

   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

   $$\dot{r}(\xi) \triangleq -r(\xi) + o(\xi). \tag{5.2}$$

The network estimate is defined as

$$\hat{x}(\xi) \triangleq Dr(\xi), \tag{5.3}$$

which gives rise to the network estimation error

$$e(\xi) \triangleq x(\xi) - \hat{x}(\xi). \tag{5.4}$$

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 uses regularization on the rate $r(\xi)$ for the sake of biological plausibility. At present we ignore this regularization and note that they can only increase the network estimation error e, the sole network objective. 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:

$$\begin{aligned} v_j &\triangleq d_j^T (x - \hat{x}) \\ \implies d_j^T e &= v_{th}, \end{aligned} \tag{5.5}$$

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

$$\dot{V} = -v + D^T(A+I)Dr + D^T Bc - D^T Do. \tag{5.6}$$

Equation (5.6) scales the spike train o_j by $d_j^T d_j$. Thus the spiking behavior is described by

$$\begin{aligned}
&v_{th} = \frac{d_j^T d_j}{2} \\
&\text{if } v_j > v_j^{th}, \\
&\text{then } v_j^{'} = v_j - d_j^T d_j \int \delta(\tau)\, d\tau, \\
&\text{and } r_j^{'} = r_j + \int \delta(\tau)\, d\tau.
\end{aligned} \tag{5.7}$$

Equations (5.6) and (5.7) specify the PCF model we compare against. Figure (7) shows simulations of the PCF model with the following parameters:

$$\begin{aligned}
A &= -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
c(\xi) &= 10 \begin{bmatrix} cos(\frac{\pi}{2}\xi) \\ sin(\frac{\pi}{2}\xi) \end{bmatrix} + 8 \\
D_{\mathrm{ij}} &\sim \mathcal{N}(0,1) \text{ Columns Normalized to Unit Length} \\
d\xi &= 10^{-5}, \\
N &= 32, \\
x(0) &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}.
\end{aligned} \tag{5.8}$$

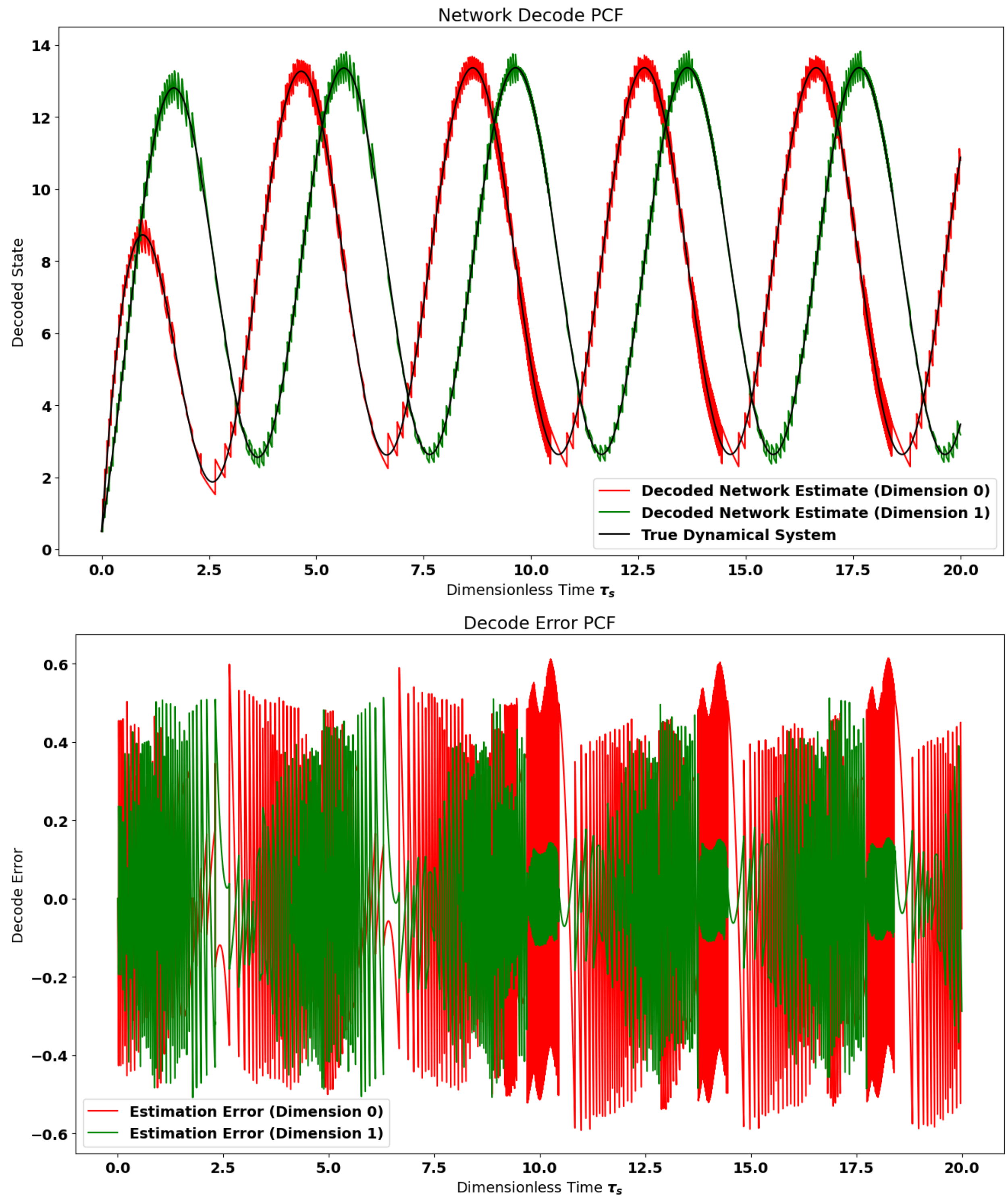


Figure 7: Simulation of PCF model given by equations (5.7) and (5.6). ***Top:*** Network estimate given by equation (5.3). ***Bottom:*** Estimation Error for PCF network from equation (5.4). The simulation parameters are given in equation (5.8). The numerical implementation is identical to that in section (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 (5.6) termed the gap-junction model.
The derivation is identical as the PCF until we derive the voltage dynamics.

$$\dot{V} = D^T Ax + D^T Bc + D^T Dr - D^T Do.$$

Instead of assuming $x = \hat{x}$, we apply the definition of voltage, equation (5.5) 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

$$\dot{V} = D^T A \left(D^{T\dagger} V + Dr \right) + D^T Dr + D^T Bc - D^T Do$$

$$\implies \dot{V} = D^T A D^{T\dagger} V + D^T (A + I) Dr + D^T Bc - D^T Do. \tag{5.9}$$

Equation (5.9) in conjunction with an identical spiking rule from PCF, equation (5.7) 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.

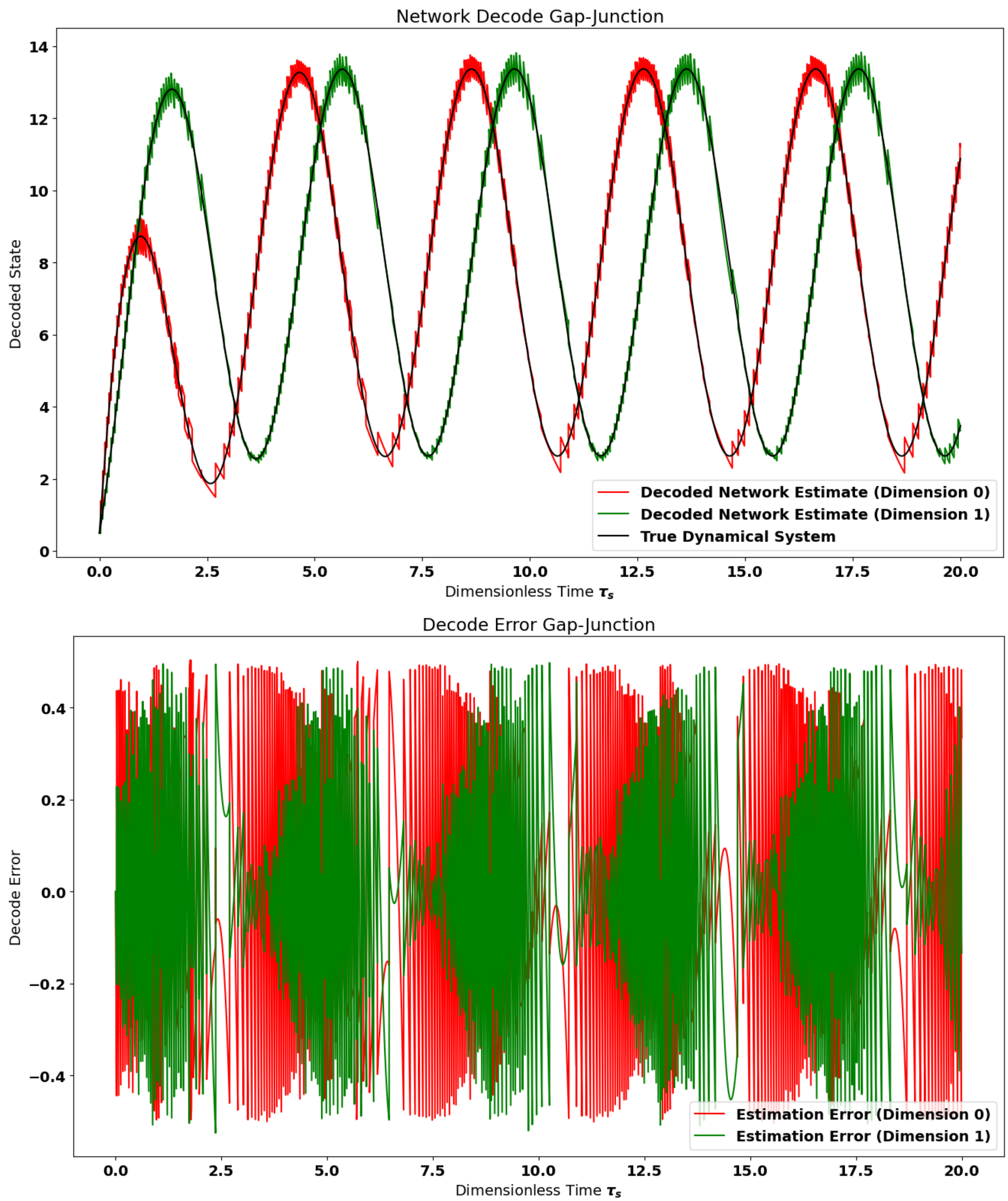


Figure 8: Simulation of the Gap-Junction model given by equations (5.7) and (5.9). ***Top:*** Network estimate given by equation (5.3). ***Bottom:*** Estimation Error for the Gap-Junction network from equation (5.4). The simulation parameters are the same as the previous figure. As with the PCF model, the network is only numerically stable if spikes are restricted to one per time step.

3. ***Spike Rate Laws:*** Here we explicitly solve the for the steady state behavior of the PCF and gap-junction models in response to a constant stimulus. We compute their per-spike RMSE and compare each with the self-coupled network model.

Let all 3 models have the same parameters as given by equation (5.8) with the exception that

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

and $x(0) = \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix}$. From equation (5.6), 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 (5.7) 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 (5.7) that the threshold voltage varies with $||d_{j_{max}}||^2$. With initial condition $v_{pcf}(0) = -\frac{||d||^2}{2}$, the neuron's trajectory is integrated as

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

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

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

$$\phi_{pcf} \left(d^T c \right) = \frac{1}{ln \left(d^T c + \frac{||d||^2}{2} \right) - ln \left(d^T c - \frac{||d||^2}{2} \right)} \tag{5.10}$$

A self-coupled neuron spike adds U_1 to its network estimate. Using an identical analysis to this case as done in section (4), we substitute d for U_1 to arrive at the steady state network estimate of the PCF network:

$$\hat{x}_{pcf}(\xi) = \left(1 + \frac{1}{2\, d^T c} \right) e^{- \ \left(\xi - \xi_1^1\right) \mod \frac{1}{\phi}} \ d.$$

wit