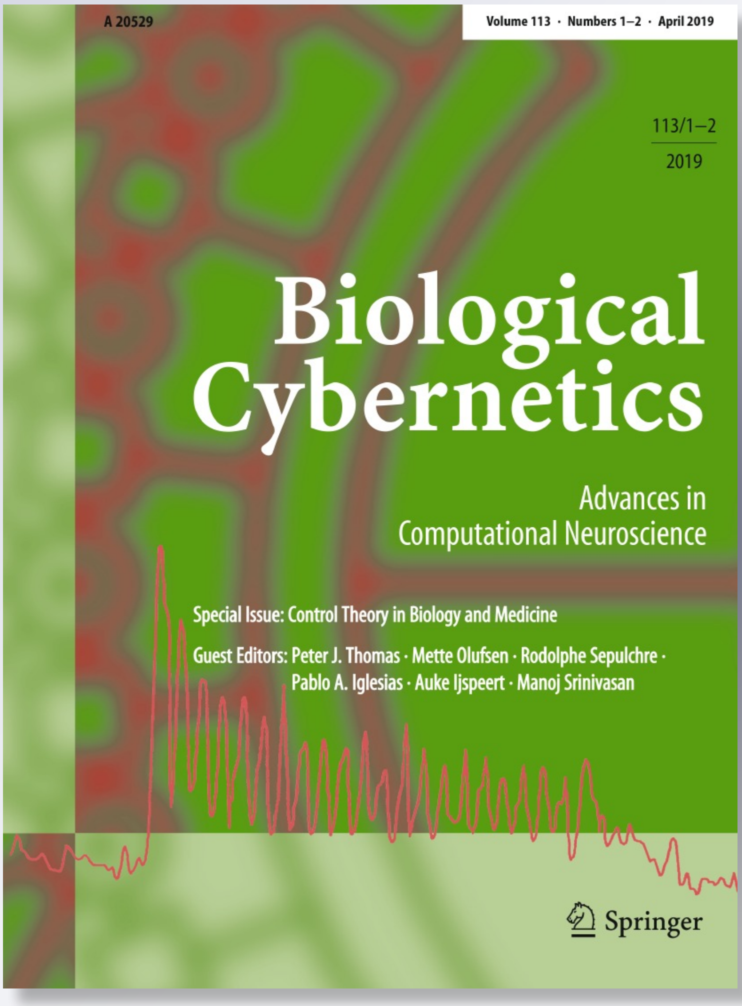
Spiking networks as efficient distributed controllers

Fuqiang Huang & ShiNung Ching

Biological Cybernetics
Advances in Computational
Neuroscience

ISSN 0340-1200
Volume 113
Combined 1-2

Biol Cybern (2019) 113:179-190
DOI 10.1007/s00422-018-0769-7



A 20529

Volume 113 · Numbers 1–2 · April 2019

113/1–2
2019

Biological Cybernetics

Advances in Computational Neuroscience

Special Issue: Control Theory in Biology and Medicine

Guest Editors: Peter J. Thomas · Mette Olufsen · Rodolphe Sepulchre · Pablo A. Iglesias · Auke Ijspeert · Manoj Srinivasan

Springer

Biological Cybernetics (2019) 113:179–190
https://doi.org/10.1007/s00422-018-0769-7

Biological Cybernetics

ORIGINAL ARTICLE

Spiking networks as efficient distributed controllers

Fuqiang Huang[1] · **ShiNung Ching**[1]

Received: 14 February 2018 / Accepted: 19 June 2018 / Published online: 27 June 2018

Abstract

In the brain, networks of neurons produce activity that is decoded into perceptions and actions. How the dynamics of neural networks support this decoding is a major scientific question. That is, while we understand the basic mechanisms by which neurons produce activity in the form of spikes, whether these dynamics reflect an overlying functional objective is not understood. In this paper, we examine neuronal dynamics from a first-principles control-theoretic viewpoint. Specifically, we postulate an objective wherein neuronal spiking activity is decoded into a control signal that subsequently drives a linear system. Then, using a recently proposed principle from theoretical neuroscience, we optimize the production of spikes so that the linear system in question achieves reference tracking. It turns out that such optimization leads to a recurrent network architecture wherein each neuron possess integrative dynamics. The network amounts to an efficient, distributed event-based controller where each neuron (node) produces a spike if doing so improves tracking performance. Moreover, the dynamics provide inherent robustness properties, so that if some neurons fail, others will compensate by increasing their activity so that the tracking objective is met.

Keywords Spiking networks · Neural networks · Decoding · Event-based control

1 Introduction

It is often argued that brain networks constitute controllers of unparalleled capability, such as in the ways that we as humans are able to manipulate our limbs and execute fine motor trajectories with ease [1]. Consequently, using brain-like networks for the purposes of control has been an appealing approach for several decades [2–4]. Indeed, inspired by neuroanatomy, artificial neural networks (ANNs) have been proposed and successfully used to solve a variety of control tasks such as system identification and learning problems [5–8]. Classically, the goal of an ANN is to realize approximate dynamic programming or adaptive control objectives by learning the weights of connections between neurons in order to minimize a prescribed error function. The neurons in an ANN produce graded activity through a static linear or nonlinear function of their inputs, scaled by these weights.

Despite the success of ANNs in certain contexts, much is left to be desired in the use of networks for control purposes. Indeed, we know that actual neurons in the brain possess often complex dynamics and produce activity primarily in the form of action potentials, or spikes, as opposed to the graded (i.e., real-valued) activity of ANNs. Understanding how such spiking activity is decoded into usable signals is one of the most prevalent questions in theoretical neuroscience [9,10].

Motivated by this question, our goal in this paper is to synthesize a spiking neural network whose activity can be decoded into a useful control signal. To do so, we leverage the idea from computational neuroscience of predictive coding [11,12], which posits that neurons act in a way that best reduces the future uncertainty of extrinsic inputs and latent variables. While often carried out in a statistical framework, recent work [13,14] has used predictive coding to build deterministic spiking networks that approximate dynamical

---

Communicated by Rodolphe Sepulchre.

ShiNung Ching holds a Career Award at the Scientific Interface from the Burroughs-Wellcome Fund. This work was partially supported by AFOSR 15RT0189, NSF ECCS 1509342 and NSF CMMI 1537015, from the US Air Force Office of Scientific Research and the US National Science Foundation, respectively.

This article belongs to the Special Issue on *Control Theory in Biology and Medicine*. It derived from a workshop at the Mathematical Biosciences Institute, Ohio State University, Columbus, OH, USA.

✉ Fuqiang Huang
fuqiang@wustl.edu

[1] The Department of Electrical and Systems Engineering, Washington University, St. Louis, MO 63130, USA

systems. Here, we use the ideas in [13,14] to elaborate on our previous work [15] in considering not system approximation, but rather *system control*. The approach is based on the following theoretical suppositions:

- Neurons fire spikes at certain times t_i, $i \in \mathbb{Z}^+$ resulting in a spike train of the form:

  $$o(t) = \sum_i \delta(t - t_i),$$

  where $\delta(\cdot)$ is a Dirac delta function.
- Spike trains of the neural network are converted into a real-valued control signal via

  $$u(t) = h(o(\cdot))$$

  where $h(\cdot)$ is a *decoder* that may in general depend on the current and past state of the spike train.
- The goal of the network is to emit spikes with timing t_i such that $u(t)$ produces a desired control objective in a known plant model.

In other words, we ask: if spikes are converted to a control signal via $h(\cdot)$, then *when* and *how* should those spikes be produced? Should neurons fire many spikes, densely in time; or should only a few neurons produce a small number of spikes, in a more sparse, efficient manner? As intuition might suggest, the answer to these questions will depend on the specific assumptions regarding the form of $h(\cdot)$, as well as the system being controlled.

As will be shown, when we assume that the decoder and plant both have linear dynamics, the answers to the above questions can be analytically obtained through an optimization problem whose solution is realized by endowing each neuron with drift-diffusion dynamics and a fixed spiking threshold that is analogous to the classical integrate-and-fire model neuron [16]. That is, rather than starting *a priori* with a prespecified neuronal network and learning its connections, we start with a control objective and synthesize the network and its dynamics in one step. Indeed, there are several other comparable ideas in theoretical neuroscience and machine learning within which our approach should be placed in context. In particular, the concepts of reservoir computing [8] and the neural engineering framework [17] both highlight the use of recurrent networks for a variety of objectives including control [18]. Such strategies work by training a set of decoding weights (analogous to learning the function $h(\cdot)$) that read activity from a *fixed* network with usually random architecture. The relationship between $h(\cdot)$ and the 'best' such architecture/dynamics is not considered, and this is the main focus of our work. Further, almost always the dynamics of such networks are formulated in discrete time, versus the overt continuous-time formulation we consider herein.

From a purely control-theoretic standpoint, our results can be interpreted in the context of a type of control problem over networks. Unique in our problem is that we explicitly deduce the particular connection motifs and dynamics of each node for a particular control objective goal. That is, whereas much effort has been directed at understanding how to elicit prescribed dynamics and patterns *of* networks (e.g., [19]), our work takes the direction of examining control *by* networks, finding the network dynamics that are most useful for generic control tasks. In this regard, our network can be viewed as an efficient event-based controller [20] where each neuron (node) remains silent unless producing a spike leads to a reduction in feedback error (or, more generally, an error-based cost function). The neurons in the network constitute a set of 'event detectors', whose spikes indicate a deviation in desired cost. The spikes are decoded into control signals that compensate for these deviations.

The remainder of this paper is organized as follows: Sect. 2 presents the background and preliminaries related to the optimization problem, which leads to the derivation of the spiking network in Sect. 3; analysis and performance characterization of such spiking network is shown in Sect. 4; conclusions and discussions are made in Sect. 5.

2 Preliminaries and formulation

The formulation of the problem is similar to that in our prior work [15], the major details of which are included here.

2.1 Spike train and firing rate

We consider spiking neural networks (SNNs) wherein each neuron produces activity in the form of instantaneous spikes, modeled as Dirac delta functions [21]. Thus, the kth neuron in the network emits a spike train $o_k(t)$, where

$$o_k(t) = \sum_{i \in \{1,2,\ldots\}} \delta\left(t - t_k^i\right),$$

where t_k^i denotes time of the ith spike from the kth neuron. The collection of spike trains over the network of N neurons is denoted $o(t) \in \mathbb{R}^N$, where $o(t) = (o_1(t), \ldots, o_N(t))$. It is very important to note that we do not yet assume any dynamics associated with the generation of spikes, only that they occur as instantaneous events.

As is conventional in neuroscience [21], we define a firing *rate* variable $r(t)$ (spikes per unit time) by low-pass filtering the spike trains via

$$\dot{r}(t) = -\lambda_d r(t) + \lambda_d o(t), \tag{1}$$

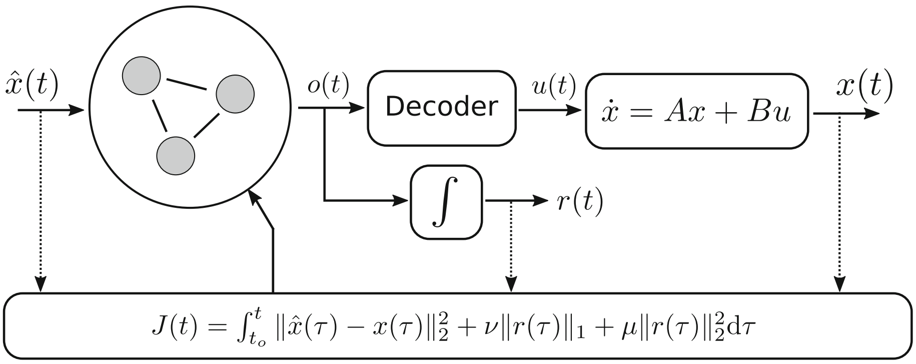


Fig. 1 Problem schema. We aim to study the circumstances in which a spiking neural network (SNN) can control a linear system through its decoded activity. To do so, we optimize the timing of the spiking outputs $o(t)$ in order to minimize a tracking objective, thereby, in effect, optimizing the dynamics of the SNN itself

where λ_d determines the (receding horizon) kernel over which rates are computed.

2.2 Linear control assumption

We postulate a putative function for the SNN wherein spikes are decoded for the purpose of controlling a linear time-invariant system of the standard form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2}$$

with system states $x(t) \in \mathbb{R}^n$ and external input $u(t) \in \mathbb{R}^m$, while $A \in \mathbb{R}^{n\times n}$ is the state matrix and $B \in \mathbb{R}^{n\times m}$ is the input matrix.

We proceed to study the control of such a system, under the tacit assumption that the pair (A, B) is controllable, which means that an external input $u(t)$ is able to steer the internal state between any two points within a finite time interval [22].

2.3 Decoding

As described in the introduction, the control itself hinges on a dynamical transformation of the spike trains $o(t)$ to the input signal $u(t)$. This transformation is termed the decoder and is elaborated on below. Endowed with a decoder, we consider the schema illustrated in Fig. 1, wherein the SNN takes the reference command $\hat{x}(t)$ as an input and produces spikes so that $x(t)$ follows $\hat{x}(t)$.

We assume that the spiking activity of the SNN is decoded into a signal $u(t) \in \mathbb{R}^m$ via a rate-instantaneous-type decoder

$$\begin{aligned} u(t) &= \mathrm{e}^{-\lambda_d t} u(t_0) + \Gamma \int_0^t \mathrm{e}^{-\lambda_d (t-\tau)} o(\tau) \mathrm{d}\tau + \Omega o(t) \\ &= \frac{1}{\lambda_d} \Gamma r(t) + \Omega o(t), \end{aligned} \tag{3}$$

with rate decoding weight $\Gamma \in \mathbb{R}^{m\times N}$ and instantaneous decoding weight $\Omega \in \mathbb{R}^{m\times N}$, where

$$\Gamma = [\Gamma_1\, \Gamma_2 \cdots \Gamma_N],$$
$$\Omega = [\Omega_1\, \Omega_2 \cdots \Omega_N],$$

in which $\Gamma_k, \Omega_k \in \mathbb{R}^m$ are decoding vectors associated with each neuron. These weights govern how the spiking activity of each neuron propagate to the inputs of the system to be controlled.

In this work, for a given neuron k, the jth elements of vectors Γ_k and Ω_k will have the same sign, i.e.,

$$\mathrm{sgn}(\Gamma_{jk}) = \mathrm{sgn}(\Omega_{jk}), \quad \text{where } j = 1, 2, \ldots, m,$$

so that jth neuron acts on the decoded signal with the same valence on both instantaneous and rate-based timescales.

The formulation of our decoder (3) and application to a control problem (2) deviates in a subtle but important way from the original work of [13,14]. Specifically, our decoder overtly contains two timescales and the resultant signal is not simply readout, but used toward a dynamical effector. As we will see, this shift in formulation allows for a feedback error signal to be used directly by the derived spike-generating network. Indeed, the dynamics in (3) encapsulates the two dominant theories regarding how spiking activity is decoded in the brain. Specifically, the $u(t)$ has terms that capture: (i) *rate decoding*, wherein the decoded signal depends proportionally on the firing rate; and (ii) *instantaneous decoding*, wherein the decoded signal depends proportionally on the exact timing of spikes via direct dependence on $o(t)$. As we will show herein, it turns out that both decoding principles confer distinct advantages in terms of $u(t)$ being usable as a control signal.

2.4 Optimization strategy

We proceed by formulating a control objective that will reward small errors while penalizing excessive neural activation:

$$J(t) = \int_{t_o}^{t} \|\hat{x}(\tau) - x(\tau)\|_2^2 + \nu\|r(\tau)\|_1 + \mu\|r(\tau)\|_2^2 \mathrm{d}\tau. \tag{4}$$

Two regularizing terms are present in this function. The penalty on the ℓ_1 norm of the firing rate, $\|r(\tau)\|_1 = \sum_i |r_i(\tau)| = \sum_i r_i(\tau)$ (for positive firing rates), is intended to prohibit the network from using too many spikes to execute the task, while the ℓ_2 norm, $\|r(\tau)\|_2^2 = r^{\mathrm{T}}(\tau)r(\tau)$, is introduced to distribute spiking throughout the population.

Finally, we introduce the schema based on which we will optimize the dynamics by which neurons produce spikes. Specifically, as noted above, we adopt the 'greedy spiking premise' introduced in [14] so that the objective of neurons is to minimize the cost function $J(t)$. Thus, we enforce the policy that the neuron k fires a spike at time t_k^i only if doing so decreases the value of $J(t)$, i.e., with a slight abuse of notation,

$$J((t_k^i + \epsilon)|o_k(t_k^i) \text{ spikes}) < J((t_k^i + \epsilon)|o_k(t_k^i) = 0), \tag{5}$$

where $0 < \epsilon \ll \lambda_d$. It is important to emphasize that the optimization will take place over the spike times $\{t_k^i\}_{k\in\{1,\ldots,N\},i\in\mathcal{N}}$, and not over the decoding parameters Γ or Ω, which are assumed known *a priori*.

The policy in (5) is the central premise from which we will deduce our ensuing spiking network.

3 Derivation of the spiking network

3.1 The emergent network dynamics are linear with fixed spiking threshold

The above optimization problem results in a threshold spiking rule, which is our first and key result. Namely, we introduce a latent state variable $v_k(t)$, termed 'voltage,' for each neuron. If $v_k(t)$ for the kth neuron exceeds a fixed threshold $\bar{v}_k$ at time t, then this neuron fires a spike to decrease the value of the objective function (4). More formally, we state:

Proposition 1 *The spike times that greedily minimize (4) according to assumption (5) are realized through the threshold criteria*

$$v_k(t) > \bar{v}_k, \tag{6}$$

subject to the neuronal voltage relation

$$v_k(t) \equiv \Omega_k^{\mathrm{T}} B^T \left(\hat{x}(t) - x(t)\right) - \mu\lambda_d \bar{e}_k^{\mathrm{T}} r(t), \tag{7}$$

$$\bar{v}_k \equiv \frac{\Omega_k^{\mathrm{T}} B^T B \Omega_k + \nu\lambda_d + \mu\lambda_d^2}{2}, \tag{8}$$

where $\bar{e}_k \in \mathrm{R}^N$ is the standard entry vector with zero elements except for the kth row.

Proof See "Appendix A". □

The above equations, in fact, specify a *network* of interconnected dynamical spiking neurons. To see this, we denote the membrane voltage vector for all of neurons as

$$V(t) = (v_1(t), \ldots, v_N(t)), \tag{9}$$

then we can deduce dynamics for the network of neurons as:

$$\dot{V}(t) = \Omega^{\mathrm{T}} B^T A e(t) + \Omega^{\mathrm{T}} B^T c(t) + W^s r(t) + W^f o(t) \tag{10}$$

where $e(t) = \hat{x}(t) - x(t)$ is the feedback error, $c(t) = \dot{\hat{x}}(t) - A\hat{x}(t)$ is the feedforward signal and the weight matrix of connections among neurons are

$$W^s = -\frac{1}{\lambda_d}\Omega^{\mathrm{T}} B^T B \Gamma + \mu\lambda_d^2 I, \tag{11}$$

$$W^f = -\Omega^{\mathrm{T}} B^T B \Omega - \mu\lambda_d^2 I, \tag{12}$$

where I is the identity matrix. Several observations regarding the emergent spiking dynamics are notable. The emergent connectivity dynamics are separated into two parts: a slow, rate-driven component mediated by W^s (slow connectivity), and an instantaneous, spike-driven component mediated by W^f (fast connectivity). Note that these terms capture, in essence, the network connectivity (i.e., how a neuron is affected by spikes of other neurons). Since the diagonal elements are nonzero, these terms also capture the internal dynamics of each neuron.

Both of the regularization parameters ν and μ affect the threshold of $\bar{v}_k$, making it harder for the kth neuron to fire. Further, μ appears in W^f and, specifically, causes the value of $v_k(t)$ to instantaneously decrease following a firing event (i.e., in essence, a 'reset' in the latent variable $v_k(t)$). This prevents dense, repetitive spiking.

To enable tracking, the spiking network must be sensitive to both the desired state trajectory $\hat{x}(t)$ and its derivative $\dot{\hat{x}}(t)$ (i.e., to create $c(t)$). This dependence is consistent with the notion of an instantaneous spike encoder, which has been shown to be optimal from the perspective of signal reconstruction [23].

Finally, there is a duality in the parameters of the system to be controlled, and their manifestation in the SNN. Specifically, the state matrix A appears as, effectively, an input gain on the feedback error $e(t)$ and feedforward signal $c(t)$ on the SNN, whereas the input matrix B, modifies the internal connectivity among neurons. However, the number of neurons N depends solely on the dimensions of the decoding weights Γ and Ω, and not on the dimension of the system to be controlled.

The use of the feedback error $e(t)$ represents a deviation from our prior results in [15], as well as the original predictive coding network of [13,14]. This is because in the control formulation herein, the signal to be tracked $\hat{x}$ and the 'output' signal x both evolve under (2), so that the incremental optimization in (5) involves the same set of forward dynamics (see also “Appendix A”). In a non-control formulation, the signal x is simply the decoded activity, which has no dynamical relation to the tracking reference. Consequently, by allowing explicit incorporation of an error signal, we obviate the need to perform pseudo-inversion of the forward dynamics of the system, a potentially error-inducing step.

3.2 Output feedback

The above result is readily extended in the case of output feedback

$$y = Cx,$$

where $y(t) \in \mathrm{R}^l$ is the output signal and $C \in \mathrm{R}^{l\times n}$ is the output matrix, based on the revised objective function

$$J(t) = \int_{t_o}^{\mathrm{T}} \|\hat{y}(\tau) - y(\tau)\|_2^2 + \nu\|r(\tau)\|_1 + \mu\|r(\tau)\|_2^2 \mathrm{d}\tau. \tag{13}$$

The extended spiking rule becomes

$$v_k(t) > \bar{v}_k, \tag{14}$$

with

$$v_k(t) \equiv \Omega_k^{\mathrm{T}} B^T C^{\mathrm{T}} \left(\hat{y}(t) - y(t)\right) - \mu\lambda_d \bar{e}_k^{\mathrm{T}} r(t), \tag{15}$$

$$\bar{v}_k \equiv \frac{\Omega_k^{\mathrm{T}} B^T C^{\mathrm{T}} C B \Omega_k + \nu\lambda_d + \mu\lambda_d^2}{2}, \tag{16}$$

while the neural dynamics are

$$\begin{aligned}\dot{V}(t) = {}& \Omega^{\mathrm{T}} B^T C^{\mathrm{T}} C A e(t) + \Omega^{\mathrm{T}} B^T C^{\mathrm{T}} C c(t) \\ & + W^s r(t) + W^f o(t),\end{aligned} \tag{17}$$

where

$$W^s = -\frac{1}{\lambda_d}\Omega^{\mathrm{T}} B^T C^{\mathrm{T}} C B \Gamma + \mu\lambda_d^2 I, \tag{18}$$

$$W^f = -\Omega^{\mathrm{T}} B^T C^{\mathrm{T}} C B \Omega - \mu\lambda_d^2 I. \tag{19}$$

These equations are derived in a similar way as in “Appendix A”.

3.3 Adding a self-decay leads to exact integrate-and-fire dynamics

Note that the emergent dynamics (10) are similar to those of the classical integrate-and-fire (IF) neuron model [16], a well-known formal biological neuron model. As in [14], to allow the neuronal dynamics to match the classical IF model, one needs to add two additional terms to the dynamics of the latent variable $V(t)$: (i) a decay/leak action that forces the voltage to decay to zero in the absence of input; and (ii) a noise term that promotes stochastic firing, i.e.,

$$\begin{aligned}\dot{V}(t) = {}& -\lambda_V V(t) + \Omega^{\mathrm{T}} B^T A e(t) + \Omega^{\mathrm{T}} B^T c(t) \\ & + W^s r(t) + W^f o(t) + \sigma_v w(t)\end{aligned} \tag{20}$$

where, λ_V is the decay constant and $w(t)$ is a white noise process.

Note that the self-decay and noise term are introduced to match the neuron dynamics with the IF model and do not arise from the direct greedy optimization solution. However, the noise term is very important to promote realistic stochastic firing, which we will discuss later.

4 Analysis and performance characterization

We proceed to analyze the key dynamical features of the spiking network and characterize necessary conditions for the network to achieve the desired control performance.

4.1 Fast versus slow network interactions and ensuing control performance

The network exhibits coupling on two timescales: fast coupling mediated by W^f and slow coupling mediated by W^s. These timescales are related to the two timescales of decoding presented in (3). We now highlight the different functional roles of these timescales.

4.1.1 Fast interactions for immediate error reduction and feasibility

Our derivation highlights an important insight regarding the putative decoder (3) for the purpose of control, namely that the instantaneous term is necessary in order to ensure that command tracking is possible. Indeed, without instantaneous decoding, the spiking condition in (6) becomes

$$-\mu\lambda_d \bar{e}_k^{\mathrm{T}} r(t) > \frac{\nu\lambda_d + \mu\lambda_d^2}{2}. \tag{21}$$

Two observations can be immediately made: (i) the control error $x(t) - \hat{x}(t)$ no longer appears in this condition, and, (ii) more bluntly, this condition can never be satisfied, since $r(t)$ is always nonnegative and the right hand side is positive. *Thus, under the greedy policy* (5), *a spike can never decrease the cost in this decoding scenario (i.e., with rate decoding only).*

This result can be interpreted, since the decoder and the system to be controlled both involve integration of their afferent inputs. Without instantaneous decoding, if a neuron spikes, a delta function will be added to the spike trains $o(t)$ and the decoded signal $u(t)$ will exhibit a jump discontinuity. Consequently, $x(t)$ can produce, at most, a trapezoidal change over the 'greedy' horizon ϵ.

4.1.2 Slow interactions for efficient tracking

Unlike fast interactions, which are pertinent to the feasibility of the control problem, slow interactions can improve performance through more efficient use of spikes, since each spike is able to have a lasting effect on other neurons.

To illustrate this effect, we consider a simple example of tracking in a linear system with $A = -10$, $B = 1$ and compare the ensuing network performance achieved by a network of 100 neurons for $|\Gamma_k| = 0$ (Fig. 2a) and $|\Gamma_k| = 200$ (Fig. 2b). The parameters for the spiking network are chosen as $\lambda_V = 0$Hz, $\sigma_v = 50$ $\mu = 0.03$, $\nu = 0.3$, $\lambda_d = 10$Hz, $\Omega_k = 0.5$ for $k = 1 \ldots 50$, $\Omega_k = -0.5$ for $k = 51 \ldots 100$, while Γ_k has the same sign as Ω_k.

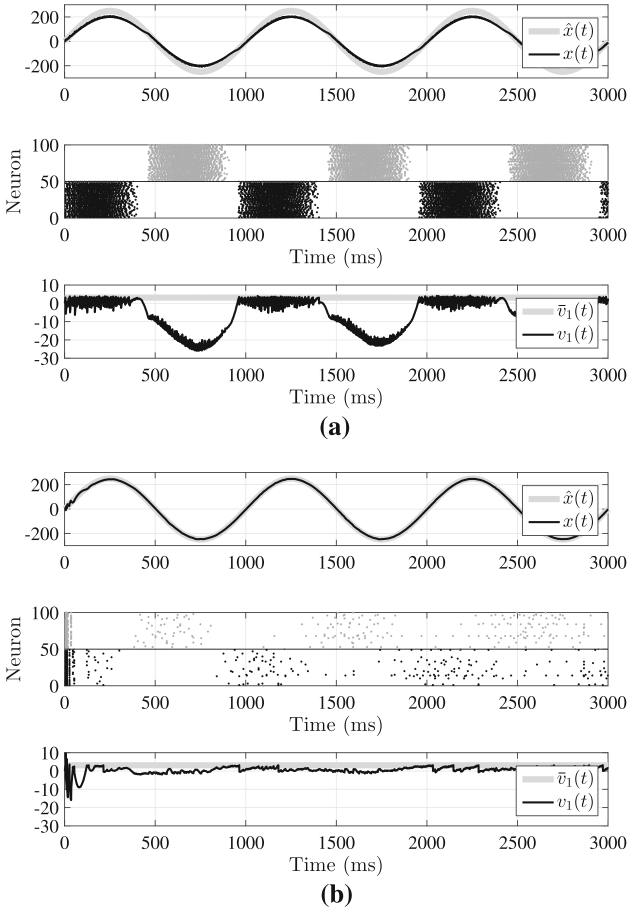


Fig. 2 Performance of the proposed spiking network **a** without and **b** with rate decoding. The first panels of **a** and **b** demonstrate the tracking performance; the spiking events, i.e., the spiking activity across the population are shown in the second panels while the membrane voltage for the first neuron and its threshold are plotted in the third panels. **a** $|\Gamma_k| = 0$, **b** $|\Gamma_k| = 200$

By comparing Fig. 2a, b, we note substantially sparser and more random spiking outputs are observed when rate decoding is used. In addition to the sparser firing, we also note that the use of rate decoding also leads to generally better tracking performance. This increase in performance can be qualitatively understood since each neuron is able to have a longer lasting effect on the control signal. In fact, lower firing rates $r(t)$ are inextricably associated with low errors via (7). This rationale for this stems from the fact that the threshold $\bar{v}_k$ in (8) is independent of the rate decoding weight Γ. Thus, if the dynamics of the network produce a decrease in $r(t)$ (as is the case when Γ is utilized), then the resultant error must also decrease, else the membrane potentials $v_k(t)$ incur a contradictory increase above threshold.

4.2 Network size and robustness to neuronal failure

To demonstrate the effect of network size N, we fix the weight of rate decoding $|\Gamma_k| = 200$, change the number of neurons and generate corresponding step responses.

As shown in Fig. 3, with a greater number of neurons, the control performance appears to increase in terms of both the bandwidth and damping associated with a nominal step response. This is perhaps expected, since more neurons are able to contribute to the decoded signal. The drawback here is, of course, that greater 'effort' is used from the standpoint of computational burden.

A more important property of the spiking network is its robustness to neuron failure, i.e., its performance when subsets of neurons randomly cease spiking. Indeed, the connectivity within the network serves as a feedback mechanism, wherein neurons compensate for each other by up- or down-regulating their firing.

We show this robustness by disabling some neurons during certain time intervals (300–1200 and 1800–2700 ms). As shown in Fig. 4a, when only instantaneous decoding is used and some neurons are disabled, the tracking performance deteriorates because of insufficient neurons for spiking, while in Fig. 4b, the rate-instantaneous decoding is implemented, the system is still able to track the desired trajectory well, showing robustness to disabled neurons. This result is not difficult to understand, as mentioned above, with rate decoding, the spiking distribution is sparse, hence only a few neurons are needed to ensure good tracking.

4.3 Homogeneous versus heterogeneous decoding weights

In the above cases, we considered single-input scenarios wherein we used instantaneous decoding weights Ω_k that are

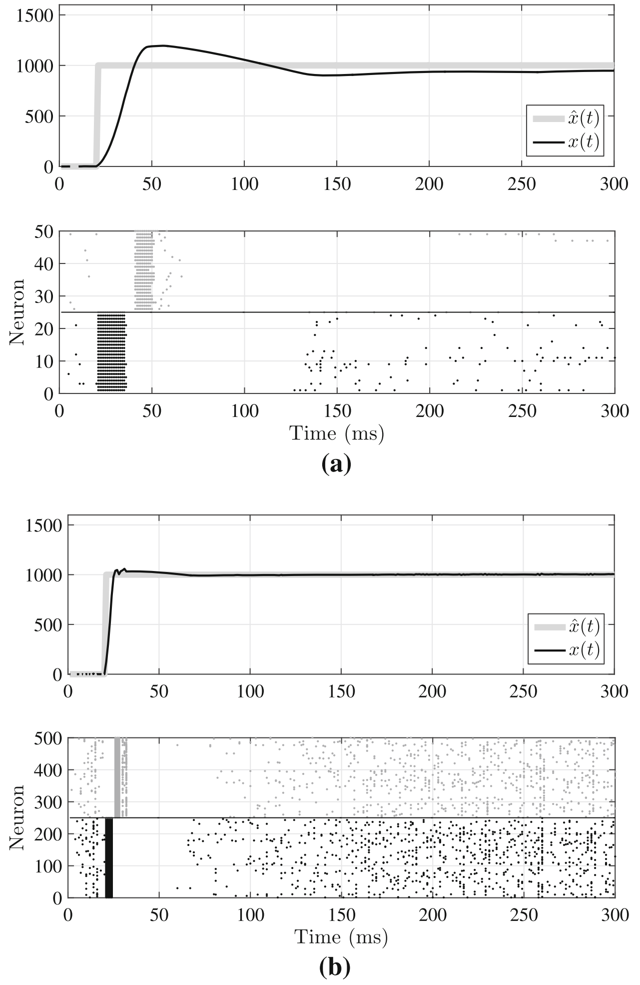


Fig. 3 Performance of the proposed spiking network for different numbers of neurons. **a** $N = 50$, **b** $N = 500$

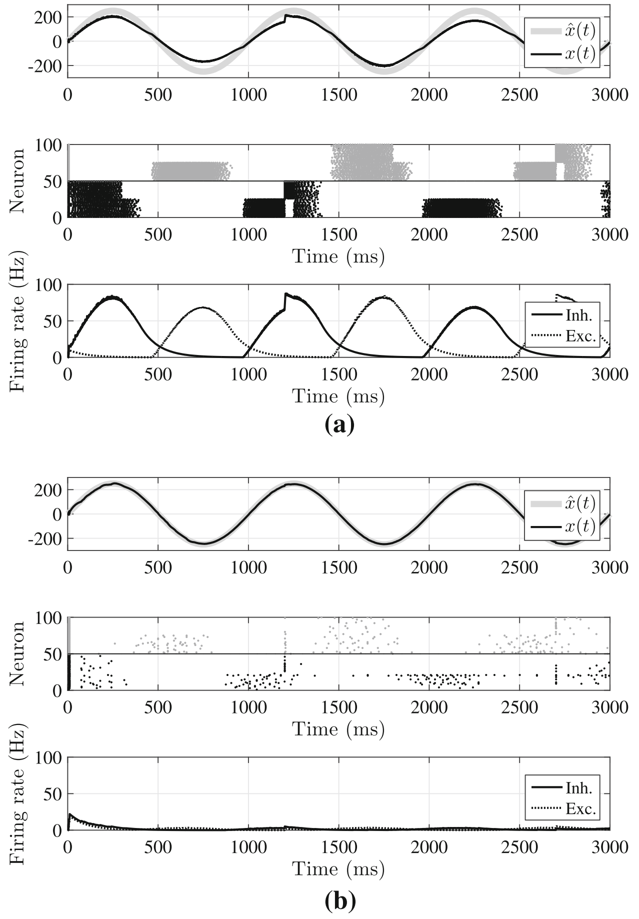


Fig. 4 Robustness of the proposed network to neuronal failure. Spikes in the red shaded regions are manually suppressed. The internal dynamics of the network result in other neurons producing compensatory spikes, mitigating the effect on tracking. **a** $|\Gamma_k| = 0$, **b** $|\Gamma_k| = 200$

homogeneous over k. However, in a multi-input system, heterogeneous weights are necessary. The reason lies in Eqs. (3), (6)–(8). Specifically, if all $|\Omega_k|$'s are the same, each neuron responds in the same way to the error, so that there is no way to disassociate error in different states. However, if these weights are heterogeneous, neurons will have a sensitivity to error in different states. Thus, as a whole, the network will act to reduce error across all controlled states. This argument is conceptually similar to that of observability of the error with respect to Ω.

To illustrate this notion, we compare step responses of a dual-input–dual-output system,

$$\begin{bmatrix}\dot{v}_1\\ \dot{v}_2\end{bmatrix} = \begin{bmatrix}-1 & 0\\ 0 & -20\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix},$$

controlled by homogeneous and heterogeneous decoding, respectively. The parameters for the spiking network are chosen as $N = 100$, $\lambda_V = 0\text{Hz}$, $\sigma_v = 50$, $\mu = 0.03$, $\nu = 0.3$, $\lambda_d = 10\text{Hz}$, $|\Gamma_k| = 200$.

In Fig. 5a, we set $\Omega_k = [0.5\ 0.5]^{\mathrm{T}}$ for $k = 1 \ldots 50$, $\Omega_k = [-0.5\ -0.5]^{\mathrm{T}}$ for $k = 51 \ldots 100$ and we see that both states cannot stay in the static states while the control signals in the second chart seems exactly the same. In contrast, when $\Omega \sim \mathcal{N}(0, 0.5^2)$—the normal distribution with 0 mean and 0.5 standard deviation, both states are able to track respective set points (Fig. 5b).

5 Discussion and conclusions

In this work, we have proposed a dynamical, recurrently connected spiking network whose activity can be decoded for the purposes of controlling a linear system in terms of a tracking objective. The network amounts to an event-based, distributed control strategy that enjoys robustness to failures of particular nodes (neurons). However, the design does rely

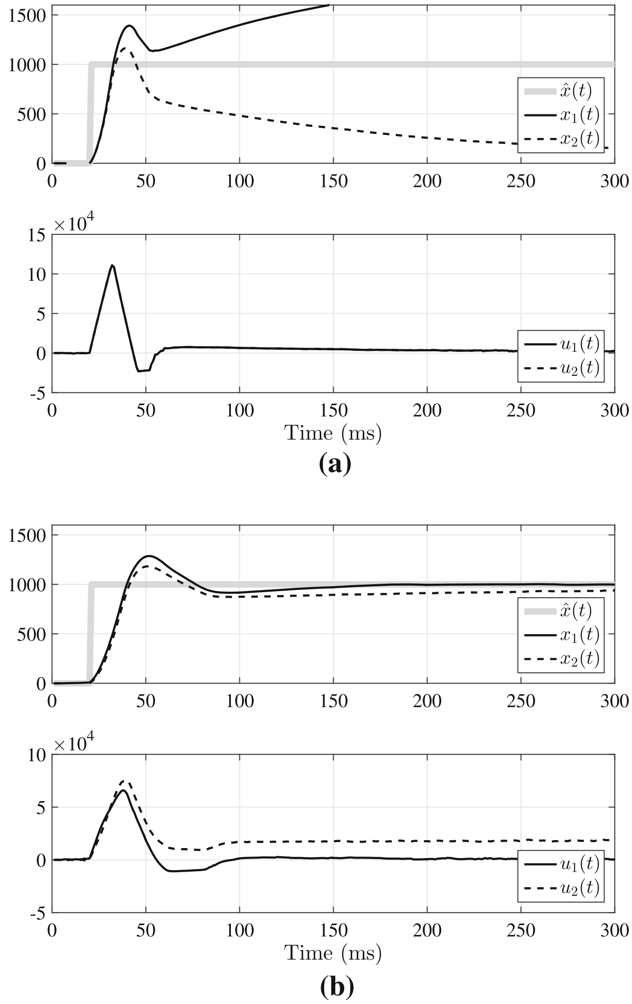


Fig. 5 In a multiple-input setting, the decoding weights must be heterogeneous. Here, performance is shown for **a** homogeneous versus **b** heterogeneous decoding

on knowledge of the system matrices A and B, which are embedded in the emergent network dynamics.

5.1 Limitations of the greedy spiking assumption: feasibility

As shown in Sect. 4.1, the greedy optimization premise underlying our network necessitates the use of instantaneous decoding. This assumption plays out in terms of another, more direct limitation: that the proposed network can only be used to control state variables that are directly actuated. Intuitively, this is because for the greedy assumption to work, a spike must be able to produce a nonzero instantaneous derivative in the error variable. In terms of the derived network, this manifests in the term $B^{\mathrm{T}}(\hat{x} - x)$ in (7). Thus, the kth neuron will only produce a spike in response to error that can propagate through B^{T} (i.e., variables that are directly actuated).

One way to formalize this notion is to restrict ourselves to the output feedback situation, wherein the objective function is specified in terms of y (13). In this case, a necessary condition for feasibility is simply:

$$B^{\mathrm{T}} C^{\mathrm{T}} \neq 0. \tag{22}$$

In fact, the greedy cost reduction policy means that even if the system is fully controllable and observable, we still cannot steer all of the states if $B^{\mathrm{T}} C^{\mathrm{T}} = 0$.

5.2 The role of noise

Another key issue in the proposed network is the role of noise, which we have modeled as additive white noise in (20). Such noise is not only needed to promote more realistic stochastic firing, but is also needed for the network to attain the desired performance. Indeed, the magnitude of the noise, parameterized by σ_V can have a substantial effect on this performance.

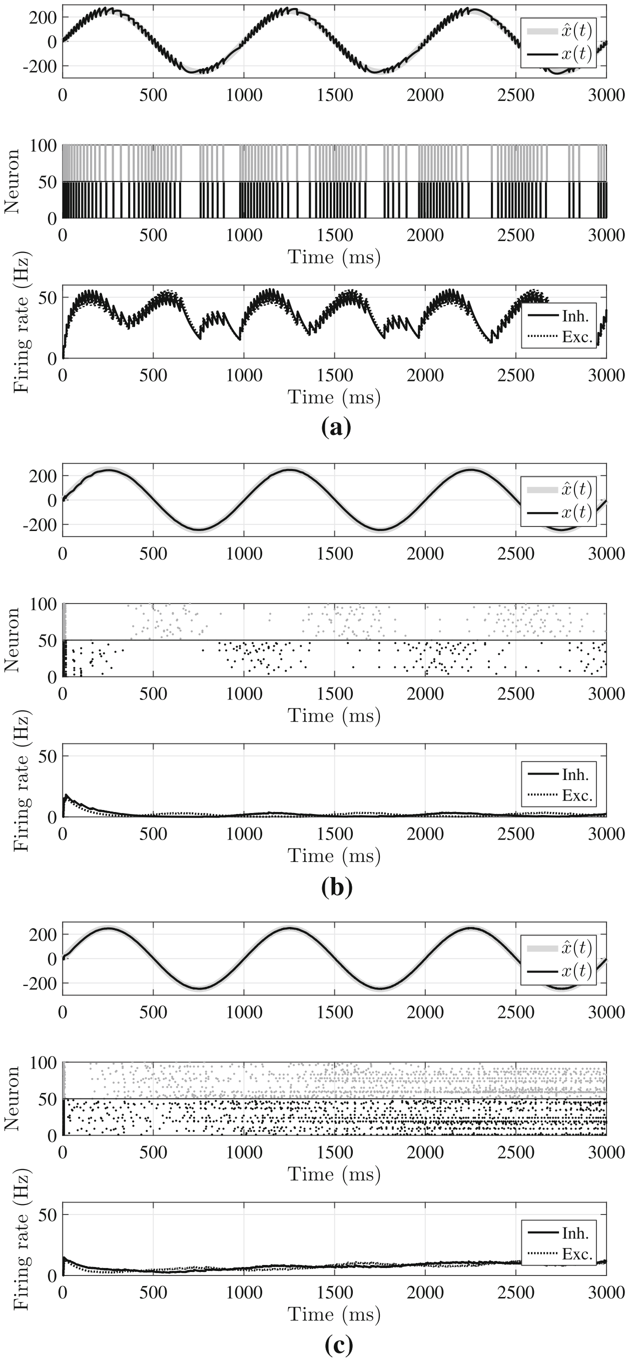
Note specifically that without noise, as shown in Fig. 6a all of the neurons with the same kernel will fire spikes together, since they will meet the same threshold at the same time according to (7) (assuming uniform initial conditions). This results in larger tracking error, non-random spiking distribution and larger firing rate, compared with Fig. 6b wherein a small amount of noise is added. However, if the noise magnitude is too large, as seen in Fig. 6c, the firing rate becomes aberrantly high and the tracking performance is also deteriorated. Hence, in this sense, there appears to be an 'optimal' σ_V, though ascertaining this value analytically is difficult. Thus, some amount of tuning is needed.

5.3 The effect of the leak term

We note that the leak term $\lambda_V V(t)$ is not an output of the optimization scheme, but rather a phenomenological addition that allows the neuronal dynamics match the classical IF model. The incorporation of the leak term deteriorates the control performance as shown in Fig. 7a, b.

5.4 Units and scaling of parameters

The parameters in (6)–(12) can be expressed in biologically meaningful units. For example, λ_d can be specified in Hz while the decoding weights $|\Gamma|$ and $|\Omega|$, determining the membrane voltage and the threshold, can be expressed as postsynaptic potential size in mV. It is important to note, however, that there is a tradeoff in the scale of these parameters in terms of the input gain matrix B and the size N of the network. In particular, with more neurons, the decoding weights can become smaller since more neurons are contributing to the control signal.

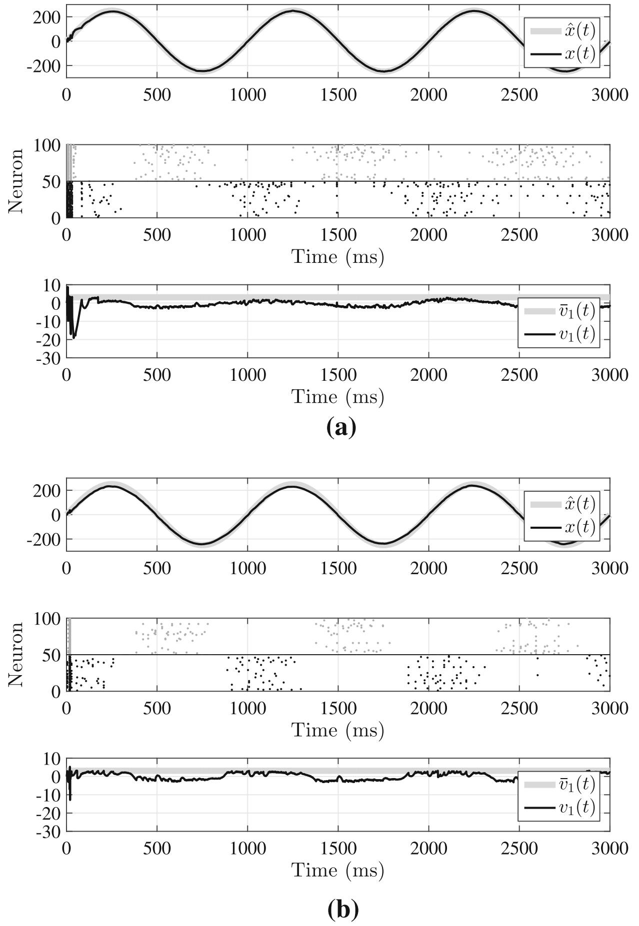


(a)

(b)

(c)

Fig. 6 The magnitude of the extrinsic noise σ_V is a key parameter. In this example, too little or too much noise leads to degradation in performance. **a** $\sigma_V = 0$, **b** $\sigma_V = 50$, **c** $\sigma_V = 200$



(a)

(b)

Fig. 7 Excessive leak λ_V deteriorates performance. **a** $\lambda_V = 0$, **b** $\lambda_V = 20$

5.5 Synaptic dynamics allow for spike transmission only

Note that in (10)–(11), interactions among neurons are mediated not only by the spike train $o(t)$, but also through the firing rate $r(t)$. However, the requirement that neurons transmit firing rates can be eliminated by endowing neurons with a synaptic state $S(t)$, which has the dynamics

$$\dot{S}(t) = -\lambda_d S(t) + \lambda_d W^s o(t), \tag{23}$$

With the dynamics (23), $S(t)$ is equivalent to $W^s r(t)$. In this way, the original dynamics can be rewritten as

$$\dot{V}(t) = \Omega^{\mathrm{T}} B^T A e(t) + \Omega^{\mathrm{T}} B^T c(t) + S(t) + W^f o(t).$$

5.6 Relationship to predictive coding

The control problem we consider is based on the notion of predictive neural coding [14], wherein the goal of the SNN is to mimic the input–output response of a linear system. Our formulation can be viewed as a generalization of this schema in three main ways: (i) we consider the possibility of spiking information being used instantaneously and in terms of rate; (ii) we consider the derived signal (i.e., $u(t)$) to be itself controlling a separate dynamical system. There is a potential semantic nuance here, since the system to be controlled could also be formulated as part of the 'decoder.' In fact, this nuance is what ultimately enables us to (iii) introduce explicit feedback error to the dynamics of neurons, which avoids the use of pseudo-inversion in obtaining the spiking dynamics [14]. It is worth noting that several features of our derived network do differ in interpretation from [14]. Indeed, the fast and slow interactions in our model arise directly from the formulation of our decoder, which has two timescales. As we showed in our results, when the goal of the network is system control (or, more specifically, error reduction), these two timescales are necessary to achieve performance. This is in contrast to the goal of emulating a system, wherein the timescales arise from forward integration of the system dynamics, coupled with the aforementioned pseudo-inversion. It is also important to note that our scheme cannot be reduced to emulating a previously designed dynamical controller (e.g., a classical PID-type scheme). Rather, our network works on the primitive goal of error reduction, with no prior assumption or specification on controller dynamics.

A Derivation of the spiking rule (6)–(12)

The methodology to derive the dynamics of the spiking network is based on the schema originally developed in [14]. Our derivation deviates insofar as we utilize the feedback error directly, consistent with our considering of a control rather than prediction objective.

We begin by quantifying the effect of any added spike on the overall cost. Assume that the kth neuron is silent at time t_s and there are no spikes since time t_s, then we can get the expression

$$\tilde{r}(t) = \mathrm{e}^{-\lambda_d(t-t_s)} r(t_s),$$

where $\tilde{r}(t)$ denotes the firing rate at time t assuming no spikes fired since time t_s.

If the kth neuron spikes at time t_s, then a delta function $\delta(t-t_s)$ is added to $o_k(t)$ resulting in

$$\begin{aligned} r(t) &= \mathrm{e}^{-\lambda_d(t-t_s)} r(t_s) + \int_{t_s}^{t} \mathrm{e}^{-\lambda_d(t-\tau)} \lambda_d o(\tau)\mathrm{d}\tau \\ &= \tilde{r}(t) + \int_{t_s}^{t} \mathrm{e}^{-\lambda_d(t-\tau)} \lambda_d \bar{e}_k \delta(\tau - t_s)\mathrm{d}\tau \\ &= \tilde{r}(t) + \mathrm{e}^{-\lambda_d(t-t_s)} \lambda_d \bar{e}_k. \end{aligned}$$

Define $\tilde{u}(t)$ and $\tilde{x}(t) = \mathrm{e}^{A(t-t_s)} x(t_s) + \int_{t_s}^{t} \mathrm{e}^{A(t-\tau)} B\tilde{u}(\tau)\mathrm{d}\tau$ as the decoded output and the system states when there is no spike since time t_s, then according to the relationship between $u(t)$ and $r(t)$, $o(t)$, i.e., Eq. (3), we have

$$\begin{aligned} u(t) &= \tilde{u}(t) + \frac{1}{\lambda_d} \Gamma \mathrm{e}^{-\lambda_d(t-t_s)} \lambda_d \bar{e}_k + \Omega_k o_k(t) \\ &= \tilde{u}(t) + \mathrm{e}^{-\lambda_d(t-t_s)} \Gamma_k + \Omega_k o_k(t), \end{aligned}$$

where Γ_k is the kth column of Γ while Ω_k is the kth column of Ω. Similarly, by Eq. (2), we obtain

$$\begin{aligned} x(t) &= \tilde{x}(t) + \int_{t_s}^{t} \mathrm{e}^{A(t-\tau)} B \mathrm{e}^{-\lambda_d(t-t_s)} \Gamma_k + B\Omega_k o_k(\tau)\mathrm{d}\tau \\ &= \tilde{x}(t) + \mathrm{e}^{-\lambda_d(t-t_s)} \left(\int_{t_s}^{t} \mathrm{e}^{\lambda_d(t-\tau)} \mathrm{e}^{A(t-\tau)} \mathrm{d}\tau \right) B\Gamma_k \\ &\quad + \int_{t_s}^{t} \mathrm{e}^{A(t-\tau)} B\Omega_k o_k(\tau)\mathrm{d}\tau \\ &= \tilde{x}(t) + \mathrm{e}^{-\lambda_d(t-t_s)} \left(\int_{0}^{t-t_s} \mathrm{e}^{(A+\lambda_d I)\zeta} \mathrm{d}\zeta \right) B\Gamma_k \\ &\quad + \mathrm{e}^{A(t-t_s)} B\Omega_k. \end{aligned}$$

In summary, when there is a new spike from the kth neuron at time t_s, the firing rate, decoded output and system states have sudden changes as

$$\begin{aligned} & r(t) \to r(t) + h(t-t_s)\lambda_d \bar{e}_k \\ & u(t) \to u(t) + h(t-t_s)\Gamma_k + \Omega_k o_k(t) \\ & x(t) \to x(t) + h(t-t_s)H(t-t_s)B\Gamma_k + \mathrm{e}^{A(t-t_s)} B\Omega_k, \end{aligned} \tag{24}$$

where

$$\begin{aligned} h(t) &= \mathrm{e}^{-\lambda_d t} \mathbf{1}(t) \\ H(t) &= \int_0^t \mathrm{e}^{(A+\lambda_d I)\zeta} \mathrm{d}\zeta, \end{aligned}$$

where $\mathbf{1}(t)$ denotes the unit Heaviside function. For convenience, from this point afterward, we will use h and H to denote $h(t-t_s)$ and $H(t-t_s)$, respectively.

With the above equations, the spiking assumption (5) can be translated into

$$\begin{aligned} & \int_{t_o}^{t_s+\epsilon} \|\hat{x} - x - hHB\Gamma_k - \mathrm{e}^{A(\tau-t_s)} B\Omega_k\|_2^2 \\ & \quad + \nu\|r + h\lambda_d \bar{e}_k\|_1 + \mu\|r + h\lambda_d \bar{e}_k\|_2^2 \mathrm{d}\tau \\ < & \int_{t_o}^{t_s+\epsilon} \|\hat{x} - x\|_2^2 + \nu\|r(\tau)\|_1 + \mu\|r(\tau)\|_2^2 \mathrm{d}\tau. \end{aligned}$$

With the definitions of ℓ_1 and ℓ_2 norms , we get

$$\begin{aligned} \int_{t_o}^{t_s+\epsilon} & - 2h\Gamma_k^{\mathrm{T}} B^{\mathrm{T}} H^{\mathrm{T}} \left(\hat{x} - x\right) + h^2 \Gamma_k^{\mathrm{T}} B^{\mathrm{T}} H^{\mathrm{T}} H B \Gamma_k \\ & - 2\Omega_k^{\mathrm{T}} B^{\mathrm{T}} \mathrm{e}^{A^{\mathrm{T}}(\tau-t_s)} \left(\hat{x} - x\right) \\ & + 2h\Gamma_k^{\mathrm{T}} B^{\mathrm{T}} H^{\mathrm{T}} \mathrm{e}^{A(t-t_s)} B\Omega_k \end{aligned}$$

$$
\begin{aligned}
&+ \Omega_k^{\mathrm{T}} B^{\mathrm{T}} \mathrm{e}^{A^{\mathrm{T}}(\tau-t_s)} \mathrm{e}^{A(\tau-t_s)} B\Omega_k \\
&+ \nu h\lambda_d + 2\mu h\lambda_d \bar{e}_k^{\mathrm{T}} r + \mu h^2 \lambda_d^2 \mathrm{d}\tau < 0.
\end{aligned}
$$

Note that $h(\tau - t_s) = \mathrm{e}^{-\lambda_d(\tau-t_s)} = 0$ and $\mathrm{e}^{A(\tau-t_s)} = 0$ for $\tau < t_s$, and rearrange the inequality to obtain

$$
\begin{aligned}
&\int_{t_s}^{t_s+\epsilon} 2h\Gamma_k^{\mathrm{T}} B^{\mathrm{T}} H^{\mathrm{T}} \left(\hat{x} - x\right) + 2\Omega_k^{\mathrm{T}} B^{\mathrm{T}} \mathrm{e}^{A^{\mathrm{T}}(\tau-t_s)} \left(\hat{x} - x\right) \\
&\quad - 2\mu h\lambda_d \bar{e}_k^{\mathrm{T}} r \mathrm{d}\tau \\
&> \int_{t_s}^{t_s+\epsilon} h^2 \Gamma_k^{\mathrm{T}} B^{\mathrm{T}} H^{\mathrm{T}} H B \Gamma_k + 2h\Gamma_k^{\mathrm{T}} B^{\mathrm{T}} H^{\mathrm{T}} \mathrm{e}^{A(t-t_s)} B\Omega_k \\
&\quad + \Omega_k^{\mathrm{T}} B^{\mathrm{T}} \mathrm{e}^{A^{\mathrm{T}}(\tau-t_s)} \mathrm{e}^{A(\tau-t_s)} B\Omega_k \\
&\quad + \nu h\lambda_d + \mu h^2 \lambda_d^2 \mathrm{d}\tau.
\end{aligned}
$$

By examining $\epsilon \ll \lambda_d$ into the future, we can then approximate the integrands as constants so that (using $h(\tau - t_s) \approx 1$, $H(\tau - t_s) \approx 0$ and $\mathrm{e}^{A(\tau-t_s)} \approx I$ for $\tau - t_s \sim \epsilon$)

$$
\Omega_k^{\mathrm{T}} B^{\mathrm{T}} \left(\hat{x} - x\right) - \mu\lambda_d \bar{e}_k^{\mathrm{T}} r > \frac{\Omega_k^{\mathrm{T}} B^{\mathrm{T}} B\Omega_k + \nu\lambda_d + \mu\lambda_d^2}{2}.
$$

Defining

$$
\begin{aligned}
v_k(t) &\equiv \Omega_k^{\mathrm{T}} B^{\mathrm{T}} \left(\hat{x} - x\right) - \mu\lambda_d \bar{e}_k^{\mathrm{T}} r \\
\bar{v}_k &\equiv \frac{\Omega_k^{\mathrm{T}} B^{\mathrm{T}} B\Omega_k + \nu\lambda_d + \mu\lambda_d^2}{2},
\end{aligned}
$$

the spiking rule becomes

$$
v_k > \bar{v}_k.
$$

This implies that when $v_k(t)$ is larger than $\bar{v}_k$, the kth neuron fires a spike, thus decreasing the value of the cost function.

It now remains to deduce the differential form of the dynamics on the latent variable $v_k(t)$. With $V = (v_1, \ldots, v_N)$, we can write

$$
V(t) = \Omega^{\mathrm{T}} B^{\mathrm{T}} \left(\hat{x}(t) - x(t)\right) - \mu\lambda_d r(t). \tag{25}
$$

Let $e(t) = \hat{x}(t) - x(t)$ and take derivatives of Eq. (25), we could get that

$$
\dot{V}(t) = \Omega^{\mathrm{T}} B^{\mathrm{T}} \left(\dot{\hat{x}}(t) - \dot{x}(t)\right) - \mu\lambda_d \dot{r}(t).
$$

Note that $u(t) = \frac{1}{\lambda_d}\Gamma r(t) + \Omega o(t)$, and

$$
\begin{aligned}
&\dot{\hat{x}}(t) = A\hat{x}(t) + c(t) \\
&\dot{x}(t) = Ax(t) + Bu(t) = Ax(t) + \frac{1}{\lambda_d} B\Gamma r(t) + B\Omega o(t) \\
&\dot{r}(t) = -\lambda_d r(t) + \lambda_d o(t),
\end{aligned}
$$

then,

$$
\begin{aligned}
\dot{V}(t) &= \Omega^{\mathrm{T}} B^{\mathrm{T}} \left(\dot{\hat{x}}(t) - \dot{x}(t)\right) - \mu\lambda_d \dot{r}(t) \\
&= \Omega^{\mathrm{T}} B^{\mathrm{T}} \left(A\hat{x}(t) + c(t)\right) \\
&\quad - \Omega^{\mathrm{T}} B^{\mathrm{T}} \left(Ax(t) + \frac{1}{\lambda_d} B\Gamma r(t) + B\Omega o(t)\right) \\
&\quad - \mu\lambda_d \left(-\lambda_d r(t) + \lambda_d o(t)\right) \\
&= \Omega^{\mathrm{T}} B^{\mathrm{T}} Ae(t) + \Omega^{\mathrm{T}} B^{\mathrm{T}} c(t) \\
&\quad + \left(-\frac{1}{\lambda_d} \Omega^{\mathrm{T}} B^{\mathrm{T}} B\Gamma + \mu\lambda_d^2 I\right) r(t) \\
&\quad - \left(\Omega^{\mathrm{T}} B^{\mathrm{T}} B\Omega + \mu\lambda_d^2 I\right) o(t).
\end{aligned}
$$

This last step highlights the core difference in the network dynamics under the control objective versus the original predictive coding framework. Because $\dot{\hat{x}}$ and $\dot{x}$ are both subject to the same (linear) dynamics in our case, the feedback error $(\hat{x} - x)$ can be retained explicitly here.

With the definition in (11) and (12), the voltage differential equation can be finally written as

$$
\dot{V}(t) = \Omega^{\mathrm{T}} B^{\mathrm{T}} Ae(t) + \Omega^{\mathrm{T}} B^{\mathrm{T}} c(t) + W_1^s r(t) + W_1^f o(t).
$$

References

1. Todorov E, Jordan MI (2002) Optimal feedback control as a theory of motor coordination. Nat Neurosci 5(11):1226–1235
2. Paul JW (1989) Neural networks for control and system identification. In: Proceedings of the 28th IEEE conference on decision and control, 1989. IEEE, pp 260–265
3. Narendra KS, Parthasarathy K (1990) Identification and control of dynamical systems using neural networks. IEEE Trans Neural Netw 1(1):4–27
4. Eliasmith C, Anderson CH (2004) Neural engineering: computation, representation, and dynamics in neurobiological systems. MIT press, Cambridge
5. Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ (1992) Neural networks for control systems—a survey. Automatica 28(6):1083–1112
6. Miller WT, Werbos PJ, Sutton RS (1995) Neural networks for control. MIT press, Cambridge
7. Lewis FW, Jagannathan S, Yesildirak A (1998) Neural network control of robot manipulators and non-linear systems. CRC Press, Boca Raton
8. Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Comput Sci Rev 3(3):127–149
9. Bialek W, Rieke F, Van Steveninck De Ruyter RR, Warland D (1991) Reading a neural code. Science 252(5014):1854–1857
10. Abbott LF, DePasquale B, Memmesheimer R-M (2016) Building functional networks of spiking model neurons. Nat Neurosci 19(3):350–355
11. Rao RPN, Ballard DH (1999) Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nat Neurosci 2(1):79
12. Bastos AM, Usrey WM, Adams RA, Mangun GR, Fries P, Friston KJ (2012) Canonical microcircuits for predictive coding. Neuron 76(4):695–711

13. Boerlin M, Denève S (2011) Spike-based population coding and working memory. PLoS Comput Biol 7(2):e1001080
14. Martin B, Christian KM, Sophie D (2013) Predictive coding of dynamical variables in balanced spiking networks. PLoS Comput Biol 9(11):e1003258
15. Fuqiang H, James R, ShiNung C (2017) Optimizing the dynamics of spiking networks for decoding and control. In: American control conference (ACC), 2017. IEEE, pp 2792–2798
16. Abbott LF (1999) Lapicque's introduction of the integrate-and-fire model neuron (1907). Brain Res Bull 50(5–6):303–304
17. Bekolay T, Bergstra J, Hunsberger E, DeWolf T, Stewart TC, Rasmussen D, Choo X, Voelker A, Eliasmith C (2014) Nengo: a python tool for building large-scale functional brain models. Front Neuroinf 7:48
18. Waegeman T, Schrauwen B et al (2012) Feedback control by online learning an inverse model. IEEE Trans Neural Netw Learn Syst 23(10):1637–1648
19. Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233
20. Seuret A, Prieur C, Tarbouriech S, Zaccarian L (2016) Lq-based event-triggered controller co-design for saturated linear systems. Automatica 74:47–54
21. Peter D, Abbott LF (2001) Theoretical neuroscience, vol 10. MIT Press, Cambridge
22. Kalman RE (1963) Mathematical description of linear dynamical systems. J Soc Ind Appl Math Ser A Control 1(2):152–192
23. Johnson EC, Jones DL, Ratnam R (2016) A minimum-error, energy-constrained neural code is an instantaneous-rate code. J Comput Neurosci 40(2):193–206