ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

BIOROB
EPFL Biorobotics Laboratory

SEMESTER PROJECT

Implementation of oscillator models and tools for analyzing coupled oscillators in FARMS

Author:
Savyaraj Ravindra Deshmukh

Supervisors:
Shravan Tata Ramalingasetty
Jonathan Arreguit

Professor:
Prof. Auke Jan Ijspeert

Jan, 2020

Contents

CHAPTER 1

INTRODUCTION

Locomotion, an important phenomenon for both animals and robots, has been an extensive area of research for several years through a biological as well as technological perspective. Animals are capable of performing immensely diverse movements with outstanding agility, accuracy and efficiency. Even the most advanced robots to date have a limited range of movements compared to animals. If these biological systems are to be emulated as closely as possible, a comprehensive understanding of biological mechanisms is necessary. In particular, the neuro-mechanical basis for locomotion can pave a way towards many significant applications in robotics.

Just as any biological system, neural system responsible for locomotion is a complex system composed of different sub-parts of the body exhibiting co-ordinated behavior. The role of external sensory stimuli from environment and unexpected perturbations make this system even more complicated. Locomotion in many vertebrates is hypothesized to be carried out by the neural circuits in the spinal cord in a distributed manner. These circuits receive primitive input commands from brain that dictate the higher level behavior such as speed or gait type. They are also modulated by the sensory feedback from peripheral nervous system as well as the neighbouring neural circuits through synaptic connections. Hence, it is very challenging to tackle the system as a whole.

Instead, many approaches begin the analysis by decoupling each of these parts as a separate block. FARMS (Framework for Animals and Robots Modeling and Simulation) is a framework created in the Biorobotics Laboratory (Biorob) to generate, integrate, analyze and visualize neural network models[1]. This project focuses on building tools for creating and analyzing neural control systems in FARMS with the following goals:

- Develop neural network controllers with biological models of neurons using the Neural Engineering Framework (NEF) by Eliasmith and Anderson [2004]
- Build analytical tools to study these networks using nonlinear dynamical systems analysis
- Demonstrate above tools using some key examples

The report is organized as follows:

1. Introduce the Neural Engineering Framework and theory relevant for locomotion
2. Demonstrate the framework through an example of lamprey swimming
3. Study bifurcation analysis tools for nonlinear dynamical systems
4. Present analysis of half center oscillator mechanism using numerical continuation methods

[1] repository link: https://gitlab.com/farmsim (please contact the creators if access is required)

CHAPTER 2

NEURAL ENGINEERING FRAMEWORK

2.1 NEURAL ENGINEERING

Here, we adapt the framework introduced by Eliasmith and Anderson [2004]. The eventual purpose is to have a generalized methodology to build and analyze neural networks as a control system that can perform a desired dynamical behavior.

The main concern for studying neural systems from biological as well as engineering perspective is of *representation*, internal state of the system that relates to external influences. For example, if we consider the task of voluntary arm movements in humans, the arm position, arm velocity, target position etc. are representations in the motor cortex of brain. These variables can be represented in the neural system as individual spikes (timing code), average rate of spikes (rate code), population activity of neurons (population code) or a combination of these codes. In the upcoming discussion, we limit ourselves to an rate based encoding, although spike based encoding is also possible in this framework. Similarly, we need to determine the procedure to *retrieve* the represented variables from the neural system. This is called *representational decoding*.

Another crucial aspect of a neural system is the class of dynamics that it can represent. If this framework is to be used for understanding locomotion, it must have appropriate means of integrating temporal dynamics of represented variables. We assume the building blocks of a dynamical system of neurons to be determined by physical neuron models that constitute the population. We represent these relations through a control theoretic description of the dynamical system.

Above discussion can be encapsulated in the following principles.

2.2 THREE PRINCIPLES

1. Neural representations are defined by the combination of nonlinear encoding (as determined by the neuron behavior) and weighted linear decoding
2. Transformations of neural representations are functions of variables that are represented by neural populations
3. Neural dynamics are characterized by considering neural representations as control theoretic state variables

Although neural transformations have significant implications on the applicability of neural systems, we focus our attention on dynamical transformations, which is highly relevant for modelling of locomotion.

2.3 SCALAR REPRESENTATION

Let us consider representation of a scalar variable for now. We assume the variable of interest x to be rescaled in range $[-1, 1]$.

Principle 1

As stated earlier, the input variable x is encoded in the rate response of a neuron. We model the neuron as a leaky integrate-and-fire neuron with following parameters:

$$\begin{aligned} \tau &- \text{time constant} \\ \tau^{ref} &- \text{absolute refractory time period} \\ J^{th} &- \text{threshold current} \\ \alpha &- \text{scaling variable} \\ J^{bias} &- \text{bias current} \end{aligned}$$

The firing rate a_i of each neuron is given by

$$\begin{aligned} a_i(x) &= G_i[J_i(x)] \\ &= \begin{cases} \frac{1}{\tau_i^{ref} - \tau_i (1 - \frac{J_i^{th}}{J_i(x)})}, & \text{if } J_i(x) > J_i^{th} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2.1}$$

where,

$$J_i(x) = \alpha_i x + J_i^{bias} \tag{2.2}$$

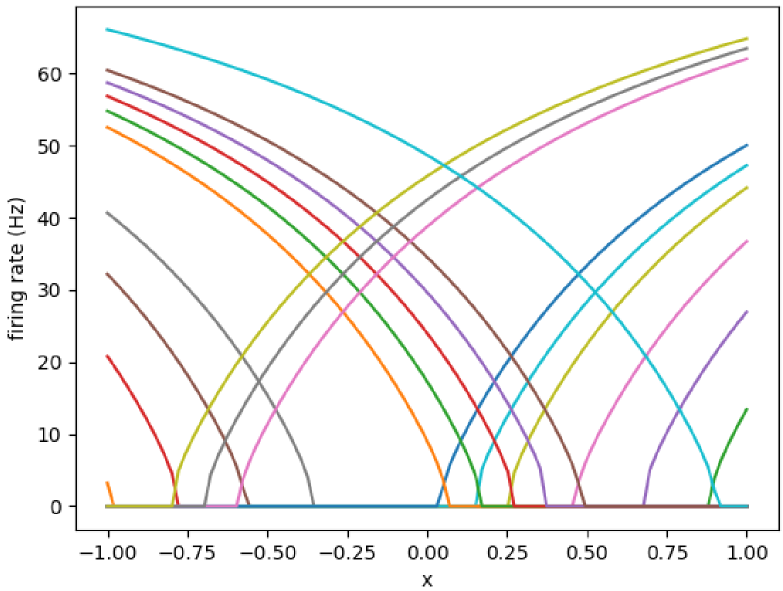


FIGURE 2.1
Tuning curves for a random distribution of 20 leaky integrate-and-fire (LIF) neurons

To determine the decoding, we estimate our variable as

$$\hat{x} = \sum_{i=1}^{N} a_i(x)\phi_i \tag{2.3}$$

where ϕ_i is the decoder for neuron i.

Now, since we know the variable we want to represent, we can optimize the values of decoders to minimize the error

$$E = \frac{1}{2}\int_{-1}^{1}[x - \sum_{i=1}^{N} a_i(x)\phi_i]^2 dx \tag{2.4}$$

Since we start off with a linear decoding, the optimization is simply a least squares regression over ϕ_i's, and the optimal decoders are given by

$$\boldsymbol{\phi} = \boldsymbol{\Gamma}^{-1}\boldsymbol{\Upsilon} \tag{2.5}$$

where,

$$\boldsymbol{\Gamma}_{ij} = \langle a_i(x)a_j(x)\rangle_x$$

and

$$\boldsymbol{\Upsilon}_j = \langle a_j(x)x\rangle_x$$

($\langle . \rangle_x$ denotes integral over x)

In summary, the representation can be characterized by following equations, namely encoding,

$$a_i(x) = G_i[\alpha_i x + J^{bias}]$$

and decoding,

$$\hat{x} = \sum_i a_i(x)\phi_i$$

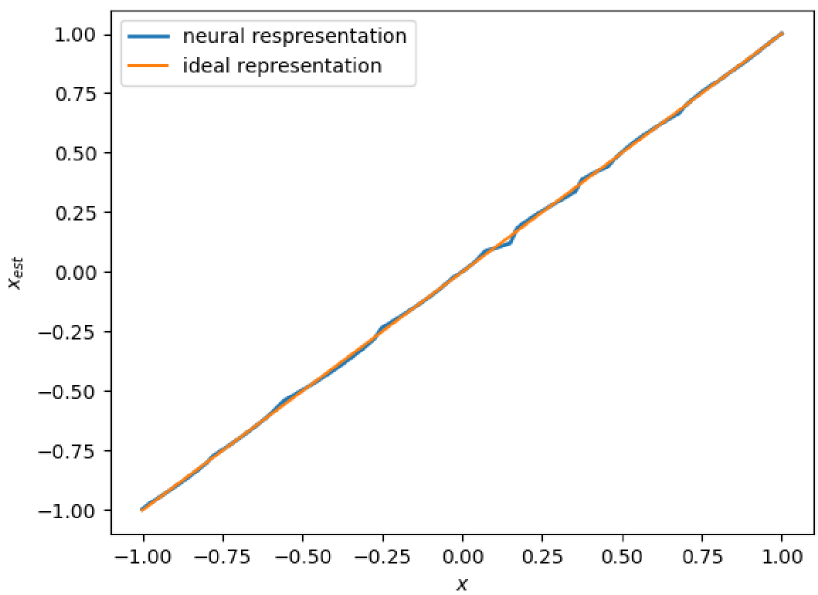


FIGURE 2.2
Scalar representation with 10 LIF neurons

2.4 Vector representation

A multidimensional vector can be represented in a neuron as

$$J_i(x) = \alpha_i \langle \tilde{\phi}_i x \rangle_n + J_i^{bias} \tag{2.6}$$

where $\tilde{\phi}_i$ are the preferred direction vectors of size n. This represents the direction of input vectors for which the neuron responds most strongly. As evident in figure 2.3a, it is an extension of scalar tuning curve to n dimensions. Even in the scalar case, the preferred direction vectors can be understood as $+1$ or -1 for "ON" or "OFF" neurons, which are the only two possible directions for a one dimensional space. The decoding is equivalent to the scalar case,

$$\hat{\boldsymbol{x}} = \sum_{i=1}^{N} a_i(\boldsymbol{x}) \boldsymbol{\phi}_i$$

where $\boldsymbol{\phi}_i$'s are the decoding vectors. Again, the procedure to find optimal decoders is similar,

$$\boldsymbol{\phi} = \boldsymbol{\Gamma}^{-1} \boldsymbol{\Upsilon}$$

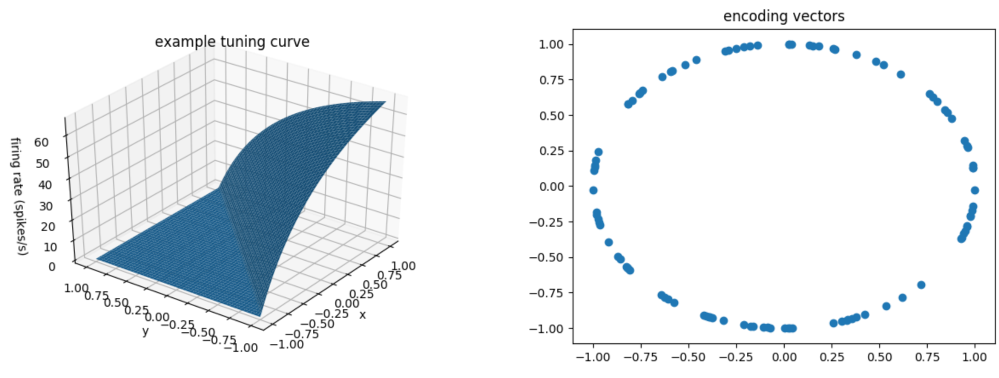
where,

$$\boldsymbol{\Gamma}_{ij} = \langle a_i(\boldsymbol{x}) a_j(\boldsymbol{x}) \rangle_{\boldsymbol{x}}$$

and

$$\boldsymbol{\Upsilon}_j = \langle a_j(\boldsymbol{x}) \boldsymbol{x} \rangle_{\boldsymbol{x}}$$

($\langle . \rangle_{\boldsymbol{x}}$ is an integral over n dimensional vector in this case)



(A) (B)

Figure 2.3
(A) Tuning curve of LIF neuron encoding a 2 dimensional vector, prefered direction $[1, -1]$, (B) Randomly distributed preferred direction vectors on a unit circle

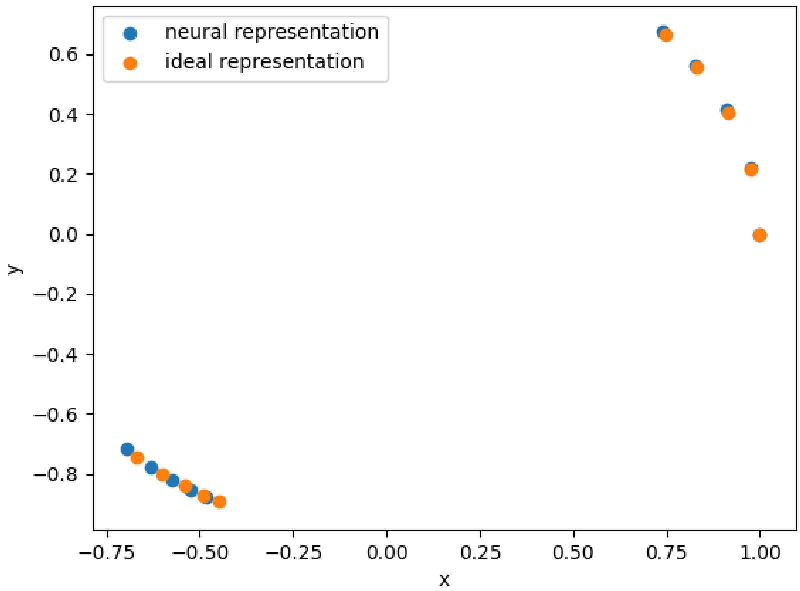


FIGURE 2.4
Representing 10 two-dimensional vectors with 20 LIF neurons

2.5 DYNAMICAL TRANSFORMATIONS

Principle 3

So far we have only considered static representations of variables in the neural populations. But to incorporate dynamics, we need to consider neuron as a dynamical system.

Let us consider for now the problem of dynamical transformation in neural populations to be described by a linear time invariant control system given by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + D(u(t) \end{aligned} \tag{2.7}$$

Just as in the static representation, we assume this system to be encoded in the neural population through the membrane potential of each neuron. But instead of using saturated activity given by each neuron, we model the time dynamics through a leaky integrator given by

$$\dot{V}_i = \frac{1}{\tau_i}(I_{ext,i} - V_i + \sum_j w_{ij} a_j(V_j)) \tag{2.8}$$

where a_j is the neural output of j'th neuron, given by

$$\begin{aligned} J_j(V_j) &= \alpha_j V_j + J_j^{bias} \\ a_j &= \frac{1}{\tau_j^{ref} - \tau_j(1 - \frac{J_j^{th}}{J_j})} \end{aligned}$$

This can be denoted by the block diagram in figure 2.5, where

$$h(s) = \frac{1}{1 + s\tau} \tag{2.9}$$

which is the synaptic filter represented in the frequency domain.

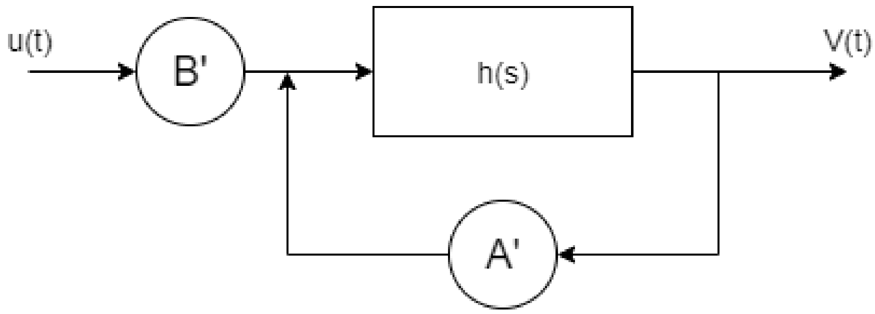


FIGURE 2.5
Generic block diagram for a leaky integrator neuron dynamics

Dynamics in figure 2.5 can be written as[1]

$$x(t) = h(t) * [A'x(t) + B'u(t)] \tag{2.10}$$

(* is the convolution operator)
Or in frequency domain

$$x(s) = h(s)[A'x(s) + B'u(s)] \tag{2.11}$$

But to use this system as a standard control theoretic model, we need

$$sx(s) = Ax(s) + Bu(s) \tag{2.12}$$

which is equation 2.7 written in the frequency domain [2]

Thus, neural dynamics can be rearranged as

$$x(s) = \frac{1}{1 + s\tau}[A'x(s) + B'u(s)]$$

So,

$$sx(s) = \frac{(A' - I)}{\tau}x(s) + \frac{B'}{\tau}u(s)$$

Comparing the two equations, we get

$$\begin{aligned} A' &= \tau A + I, \\ B' &= \tau B \end{aligned} \tag{2.13}$$

Using these matrices, for a given standard control system, we can now determine the equivalent neural system.

As usual, the encoding and decoding equations are

$$\begin{aligned} a_i(t) &= G[\alpha_i \langle x(t)\tilde{\phi_i}\rangle + J_i^{bias}] \\ x(t) &= \sum_i a_i(x(t))\phi_i \end{aligned} \tag{2.14}$$

[1] note that the inputs from other neurons can be incorporated in recurrent dynamics A' as explained later

[2] we assume the output variable y to be our representation variable and omit the matrices C and D since this neural population is assumed to be followed by further encodings and we encorporate these matrices in the subsequent population encoding

Using the convolution operator,

$$a_i(t) = G[\alpha_i \langle h(t) * [A'x(t) + B'u(t)]\tilde{\phi}_i \rangle + J_i^{bias}] \tag{2.15}$$

Replacing $x(t)$ with the neural outputs,

$$\begin{aligned} &= G[\alpha_i \langle h(t) * [A' \sum_j a_j(t)\phi_j + B'u(t)]\tilde{\phi}_i \rangle + J_i^{bias}] \\ &= G[h(t) * [\sum_j w_{ij} a_j(t) + B'\tilde{\phi}_i u(t)] + J_i^{bias}] \end{aligned} \tag{2.16}$$

where

$$w_{ij} = \alpha_i A' \phi_j \tilde{\phi}_i \tag{2.17}$$

Thus, the recurrent dynamics in A' can be embedded in the synaptic weights of the population, given by the equation above.

A simple example of the above framework is the neural integrator, which plays a central role in controlling eye positions (Eliasmith and Anderson [2004]). These networks convert a velocity signal for eye into a displacement signal. Although tuning curves for these populations are linear over a wide range, we can generalize them with LIF neurons for biologically plausible model. Experiments have shown that most of the neurons are active when eye position is centered ($x = 0$). Thus, we can assume the neural tuning curves to be similar to introduced earlier in figure 2.1.

For a given control signal $u(t)$, we simply need

$$\dot{x}(t) = u(t)$$

which implies $A = 0$ and $B = 1$.

This is the standard description of the system. Recalling equation 2.13, we get the neural description as

$$\begin{aligned} A' &= 1, \\ B' &= \tau \end{aligned} \tag{2.18}$$

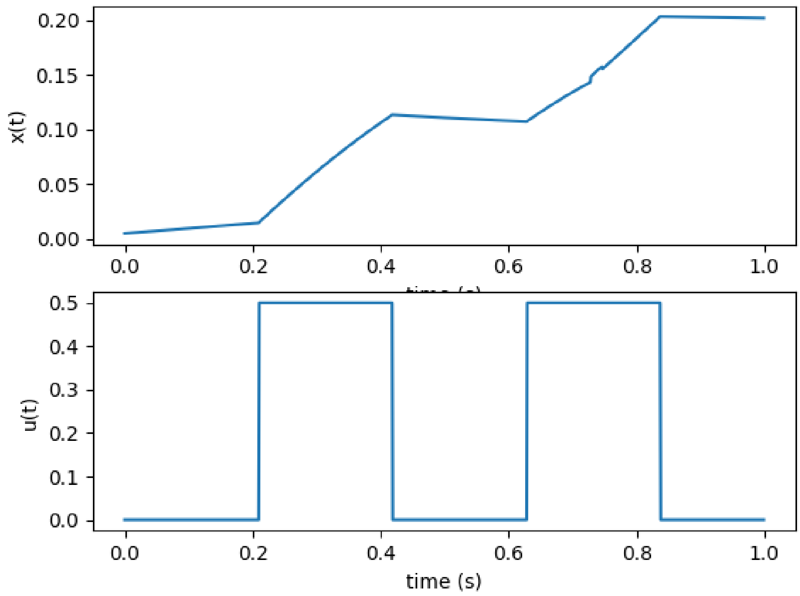


FIGURE 2.6
Integration of a pulse train with 20 LIF neurons, $\tau = 100\ ms$

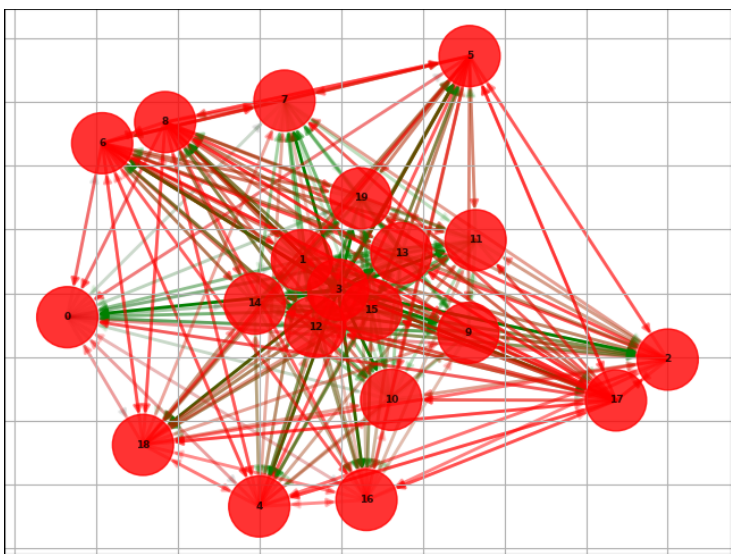


FIGURE 2.7
Network schematic for neural integrator with 20 LIF neurons. Green and red arrows denote excitatory and inhibitory synaptic weights respectively

2.5.1 ATTRACTOR NETWORKS

In dynamical systems theory, attractor networks can be characterized by a specific steady state dynamical behavior. In the state space of neural network, a point is *stable* if the system around its neighborhood converges to it over time. Thus, an *attractor* is a set of such dynamically stable points. For example, the neural integrator can be thought of exhibiting $\dot{\boldsymbol{x}} = \mathbf{0}$ in absence of an input. This is a very simplified version of

$$\dot{x} = Ax + Bu$$

Another class of networks relevant for locomotion is *cyclic attractor*, where the network dynamics show periodic behavior. For a linear dynamical system, periodic behavior can be obtained by

$$A = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \tag{2.19}$$

which is a simple harmonic oscillator with angular frequency ω.

This completes a brief summary of the Neural Engineering framework useful for dynamical characterization of neural networks. Next chapter is focused on the relevant features for locomotion - Central Pattern Generator (CPG).

CHAPTER 3

CPG ARCHITECTURE

3.1 LAMPREY LOCOMOTION

The lamprey originates from a group of animals that diverged from the main vertebrate evolutionary line at very early stages (around 450 million years ago). But over the years, it hasn't changed significantly, which makes it a prototype vertebrate. Also, its brain and spinal cord exhibit all the key vertebrate features, but contain an order of degree fewer nerve cells. Moreover, its brainstem and spinal cord can be maintained in vitro over a period of several days, with spontaneous or induced activity in the locomotor CPGs (Grillner et al. [1995]). These properties make it an ideal experimental platform for neuroscience and locomotion studies.

Studies have revealed that locomotion in vertebrates is mainly achieved through neural circuits present in the spincal cord, referred as Central Pattern Generators (CPGs). These are distributed neural networks that can generate rhythmic behaviour without any phasic input (refer to Ijspeert [2008] for an overview of CPGs in animal and robot locomotion). The level of activity in these CPGs is controlled from descending drives from the lower brainstem, and higher level of stimulation leads to faster locomotion.

So far, there have been many efforts to model CPG's through a system of coupled oscillators. The approach is to assume each CPG activity as a single oscillator, and the interconnections can be modelled to fit expected behavior. Using the framework we have so far, this abstraction can be omitted to have a population of neurons that represent an oscillator. Thus, the CPG segment activity is represented by a group of neurons, and the interactions can be translated into neural connections. The overall methodology is explained in what follows. We begin with following set of assumptions about the lamprey locomotion:

1. The spinal cord is composed of individual segments of neural population
2. Each segment is capable of generating oscillatory behavior in isolation
3. Inter-segmental connectivity is limited to local interactions
4. The wavelength of swimming wave is equal to the lamprey length
5. The inter-segmental phase lags are independent of swimming frequency

Through various experiments (Grillner et al. [1995]), it has been confirmed that the swimming pattern of lamprey follows a travelling wave to propel it forward.

$$x(z,t) = Asin(\omega t - kz) \tag{3.1}$$

3.1.1 Oscillator design

We begin with description of a single segment. Let us go back to the harmonic oscillator in the previous example (equation 2.19). A general solution can be written as

$$x(t) = A \begin{bmatrix} sin(\omega t + \phi) \\ cos(\omega t + \phi) \end{bmatrix} \tag{3.2}$$

To represent a two-dimensional vector in a neural population, we need to define suitable parametrization of the variables,

$$y = \sqrt{1 - x^2} \tag{3.3}$$

This constrains the possible dynamics to a unit circle, which is the *cyclic attractor* (Figure 3.1).

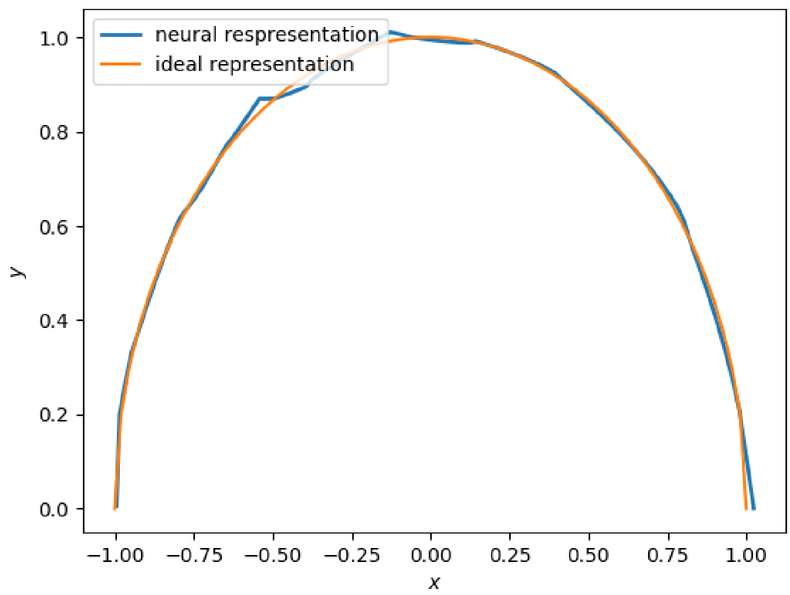


Figure 3.1
Parametrization of two-dimensional vector for an harmonic oscillator with LIF neurons

We also need to define the equivalent neural system dynamics (Equation 2.13)

$$\begin{aligned} A' &= \tau A + I \\ &= \begin{bmatrix} 1 & \tau\omega \\ -\tau\omega & 1 \end{bmatrix} \end{aligned} \tag{3.4}$$

Finally, coupling weights are given by equation 2.17. Figure 3.2 shows the resultant behavior, given by the usual linear decoding

$$\hat{\boldsymbol{x}} = \sum_{i=1}^{N} a_i(\boldsymbol{x})\boldsymbol{\phi}_i$$

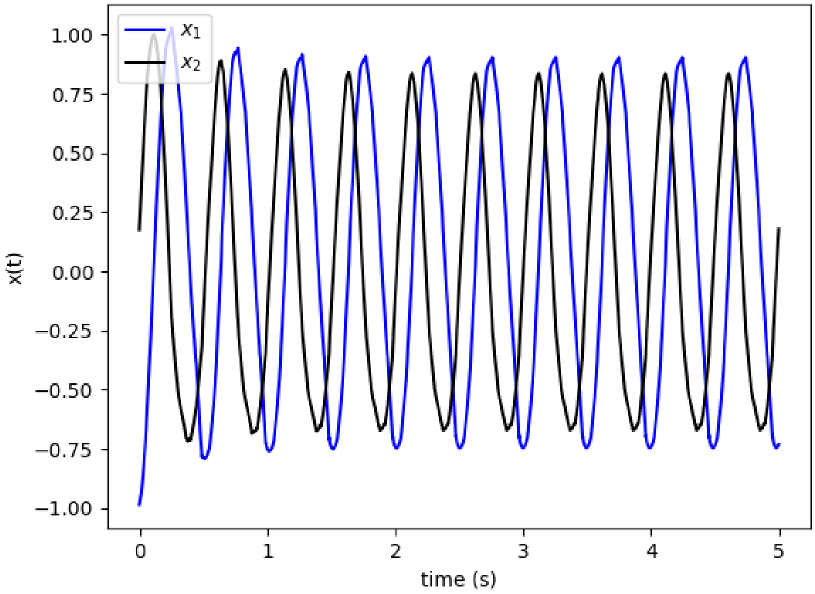


FIGURE 3.2
A single harmonic oscillator represented by 50 LIF neurons. The two variables oscillate with $freq = 2$ Hz and $\frac{\pi}{2}$ lag as expected

3.1.2 COUPLING AND SYNCHRONIZATION

To synchronize two harmonic oscillators, we can simply couple them with a linear term

$$x_i(t) = \begin{bmatrix} 0 & \omega \\ -\omega & -g \end{bmatrix} x_i(t) + \begin{bmatrix} 0 & 0 \\ 0 & g \end{bmatrix} x_j(t) \qquad i, j \in \{1, 2\}, \quad i \neq j \tag{3.5}$$

where g is the coupling strength. For $g < \omega$, we can prove that two oscillators synchronize asymptotically[1]

To represent this coupling in neural connections, we note that each variable x_i can be represented by the neural activity as

$$x_i(t) = \sum_{k=1}^{N_i} a_k^i(t)\phi_k^i \tag{3.6}$$

which can be used to determine synaptic weights for a given coupling matrix G (equation 2.17)

$$w_{kl}^{ij} = \alpha_k^i G \phi_l^j \tilde{\phi}_k^i \tag{3.7}$$

[1]The difference dynamics are given by $\dot{x}_1 - \dot{x}_2 = \begin{bmatrix} 0 & \omega \\ -\omega & -2g \end{bmatrix} (x_1 - x_2)$ leading to eigenvalues $\lambda_{1,2} = -g \pm \sqrt{g^2 - \omega^2}$ where $Re(\lambda_{1,2}) < 0$ for $0 < g < \omega$

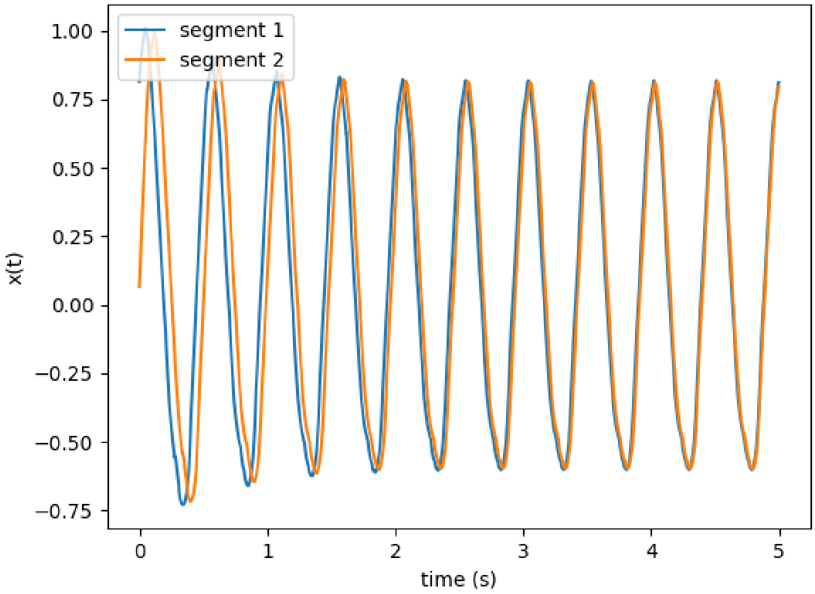


FIGURE 3.3
Coupling two oscillators represented by 50 LIF neurons, $g = \omega/20$

3.1.3 OSCILLATOR CHAIN AND TRAVELLING WAVE GENERATION

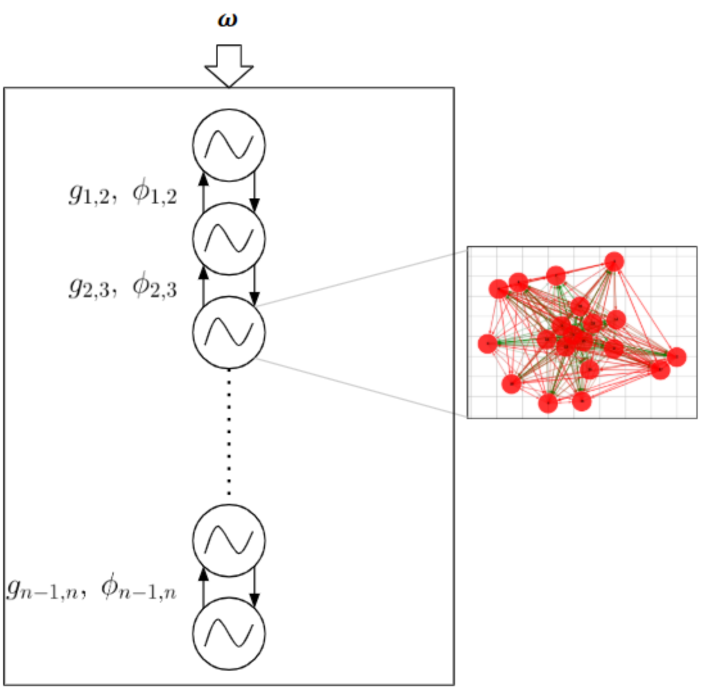


FIGURE 3.4
CPG schematic of n oscillators with neighbour coupling. Each oscillator is modelled by a neural population with harmonic oscillator encoding

The equation for travelling wave (equation 3.1) suggests that we can model the CPG with chain of oscillators which are coupled through a constant phase lag. The coupling equation 3.5 can be modified in this case as

$$\begin{aligned} x_i(t) &= \begin{bmatrix} 0 & \omega \\ -\omega & -g \end{bmatrix} x_i(t) + G_{ij}x_j(t) \\ G_{ij} &= \begin{bmatrix} 0 & 0 \\ -gsin(\phi_{ij}) & gcos(\phi_{ij}) \end{bmatrix} \end{aligned} \tag{3.8}$$

where ϕ_{ij} is the desired phase lag between oscillators i and j. From our set of assumptions in the beginning, 4 suggests that the phase lags in consecutive oscillators should be

$$\phi_{ij} = \frac{2\pi}{number\ of\ segments} \qquad \text{if} \quad i \leftrightarrow j \tag{3.9}$$

Figure 3.5 shows the output of 8 segments with 50 LIF neurons. It exhibits a travelling wave with exactly one wavelength along the body length as expected.

This example demonstrates the applicability of Neural Engineering Framework to design neural controllers. In particular, a Central Pattern Generator can be designed through individual oscillators made of neural populations, with a well defined methodology for translating the oscillator coupling in abstract space to synaptic connections in the neural system. Although this example was limited to linear control system, the nonlinear dynamical models of neurons promise to be a useful candidate for a more general CPG design through this framework.

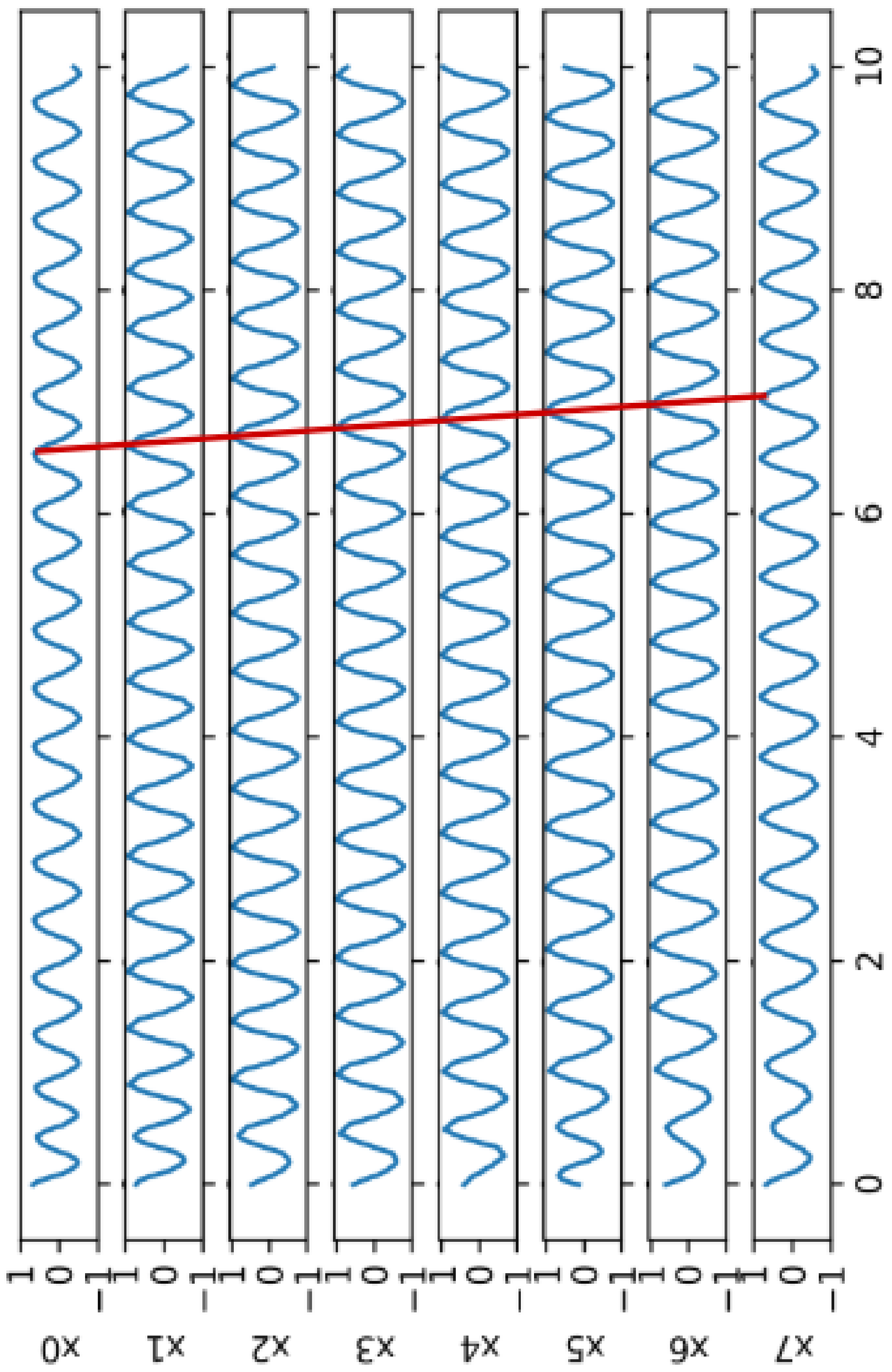


FIGURE 3.5
Travelling wave in 8 locally coupled oscillators represented by neural population of 50 LIF neurons each

CHAPTER 4

BIFURCATION ANALYSIS

In this chapter, the neural mechanisms for oscillations and locomotion is probed through an analytical point of view. The section is organized as follows:

1. Overview of some standard two dimensional neuron models, which are also implemented in FARMS framework
2. Overview of nonlinear analysis tools for dynamical systems
3. A brief introduction to numerical continuation methods
4. Analysis of a well known mechanism for locomotion - half center oscillator, through COCO continuation toolbox (Dankowicz and Schilder [2013])
5. Extensions and limitations of continuation methods to higher dimensional parameter spaces

4.1 IMPLEMENTATION OF NEURON MODELS IN FARMS

1. **Morris Lecar Model**
   This is a two dimensional simplified version of the Hodgkin-Huxley neuron. It was first developed to reproduce the oscillatory behaviors in particular type of muscle fibers subject to change in ion channel conductance (Morris and Lecar [1981]).
   Here we consider a variation useful for nonlinear analysis given by Liu et al. [2014]

$$
\begin{aligned}
C \frac{\mathrm{d}V}{\mathrm{d}t} =& I_{\text{stim}} - g_{\text{fast}}\, m_{\infty}(V)\,(V - E_{\text{Na}}) - g_{\text{slow}}\, w\,(V - E_{\text{k}}) \\
& - g_{\text{leak}}\,(V - E_{\text{leak}}) \\
\frac{\mathrm{d}w}{\mathrm{d}t} =& \phi_w \frac{w_{\infty}(V) - w}{\tau_w(V)}
\end{aligned}
\tag{4.1}
$$

   V is the fast activation variable and w is the slow recovery variable. E represents the equilibrium potential and g_{fast}, g_{slow} and g_{leak} are the conductances of corrosponding fast, slow and leak currents respectively. The steady state activation functions are given by

$$\begin{aligned} m_\infty(V) &= 0.5\left[1+\tanh\left(\frac{V-\beta_m}{\gamma_m}\right)\right] \\ w_\infty(V) &= 0.5\left[1+\tanh\left(\frac{V-\beta_w}{\gamma_w}\right)\right] \\ \tau_w(V) &= \frac{1}{\cosh\left(\frac{V-\beta_w}{2\gamma_w}\right)} \end{aligned} \tag{4.2}$$

Since the system is two dimensional, phase plane analysis is easily tractable. Thus, it finds many applications in nonlinear analysis of neural dynamics for computational neuroscience (Lecar [2007]).

2. **Matsuoka Model**
This model was developed by Matsuoka [1985] to attempt modelling stable oscillatory behaviors that are observed in biological systems.

The dynamics for two Matsuoka neurons with mutual inhibition is given by:

$$\begin{aligned} \tau\frac{\mathrm{d}}{\mathrm{d}t}V_i(t) + V_i(t) &= c - ay_j(t) - bw_i(t) \\ (i, j &= 1, 2; j \neq i) \\ T\frac{\mathrm{d}}{\mathrm{d}t}w_i(t) + \nu_i w_i(t) &= y_i(t) \\ y_i(t) &= g\left(V_i(t) - \theta_i\right) \end{aligned} \tag{4.3}$$

$g(.)$ is a piecewise linear function $g(x) = \max\{0, x\}$ which represents the threshold property of neurons. ν_i variable is used to capture the adaptive behavior observed in real neurons and plays a crucial role in generating stable limit cycles.

It is observed that g has a linear behavior in a limited sense, such that $g(kx) = kg(x)$ which simplifies the analytical treatment (covered in detail by Matsuoka [2011]). Since then, this model has been widely used to model central pattern generators (Kimura et al. [1999], Taga et al. [1991])

3. **Fitzhugh Nagumo model**
This is a two dimensional simplified model of neurons modelled by FitzHugh [1961] and Nagumo et al. [1962]. It closely resembles to the Van der Pol oscillator with a forcing input. The dynamics are:

$$\begin{aligned} \dot{V} &= V - \tfrac{V^3}{3} - w + I_{\text{ext}} \\ \tau\dot{w} &= V + a - bw \end{aligned} \tag{4.4}$$

As in Morris Lecar model, V here is a fast activation variable and w is a slow recovery variable. This model is used in the following section to study bifurcation analysis tools.

4.2 BIFURCATION ANALYSIS

As we move from linear one dimensional leaky integrator model of neurons towards the nonlinear four dimensional Hodgkin-Huxley model, the dynamics become more and more involved. The picture becomes even more obscure when multiple neurons are coupled through excitatory or inhibitory connections. Thus, synthesis of large scale networks capable of specific tasks often requires standard optimization algorithms to obtain the relevant parameters. For example, Ijspeert [2001] develop CPG for a simulated salamander using leaky integrator neural network, where the parameters are determined using genetic algorithm. The network is capable of generating both swimming and walking gaits with controllable speed and direction as the tonic excitation to the network is varied. This fact is important from biological perspective since experiments in fictive locomotion have shown that descending drive is the primary modulation factor in such class of animals (Ijspeert [2008])

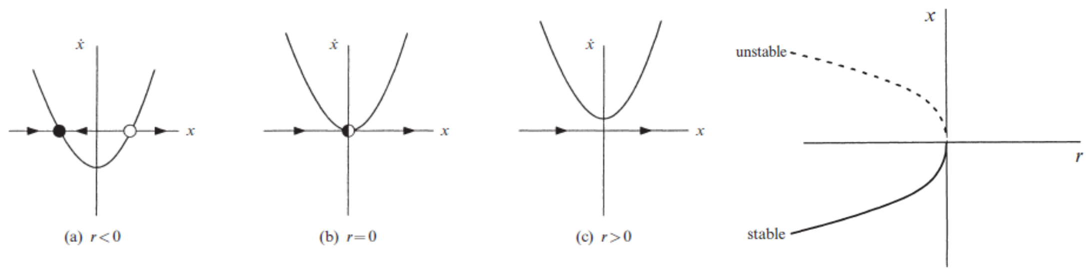
But one of the limitations of such algorithms is that the parameters are obtained without using any information about the known dynamical model, hence they may not be sufficient to give any insights about mechanisms that give rise to the observed behavior.

If we consider the particular case of locomotion, the primary concern is to obtain stable limit cycles in the network. A limit cycle corresponds to periodic behavior of a dynamical system in the state space such that the system asymptotically converges or diverges from this cycle. If a system in stable limit cycle is slightly perturbed (within the attraction basin*), it again converges back to the cycle. Such behavior is naturally found in many living things, which tend to follow a certain behavior despite slight disturbances from the environment.

In the following sections, an alternate approach is studied - *bifurcation analysis*. Bifurcation occurs when changes in a system parameter causes the topological changes of trajectories in phase space. Consider a simple example of one dimensional system given by (Strogatz [2018])

$$\dot{x} = r + x^2 \tag{4.5}$$

where r is the control parameter. Depending on the values of r, three types of behaviors are observed, as shown in figure 4.1a.



(A) (B)

FIGURE 4.1
(A) System behavior at different r values. black and white points are stable and unstable fixed points respectively (B) Bifurcation diagram

For $r < 0$, we have two fixed points of the system, one stable and one unstable. As r approaches 0, the two fixed points move towards each other. Once r becomes positive, there are no fixed points in the system. This type of bifurcation is called a saddle-node bifurcation.

Such analysis can be very informative for studying general form of dynamical systems that exhibit topologically similar behavior. For instance, if we focus on stable oscillatory dynamics, we can limit to only a subset of all possible bifurcations in the dynamical system. One such example is subcritical hopf bifurcation, which is analyzed in detail through following sections.

4.3 Bifurcation analysis tools - review

Here, some of the well known tools for bifurcation analysis and continuation methods are summarized.

- **AUTO**, Doedel [1984]
  - One of the earliest packages for such analysis
  - The continuation algorithms are still used for the new packages being developed
  - Developed in low level languages like FORTRAN
- **MATCONT**, Dhooge et al. [2008]
  - Very well established MATLAB package
- **CEPAGE**, Lodi et al. [2017]
  - Developed particularly for analysis of **CPG networks**
  - Phase response curve (PRC) is used to perform continuation analysis - faster than exact computation (Novičenko and Pyragas [2012])
  - Limited to weakly coupled oscillators
- **DYVIPAC**, Nguyen et al. [2015]
  - Parameter sampling from uniform distribution
  - Local stability analysis from the lyapunov exponents
  - **Visualization using Parallel Coordinates (PC)**

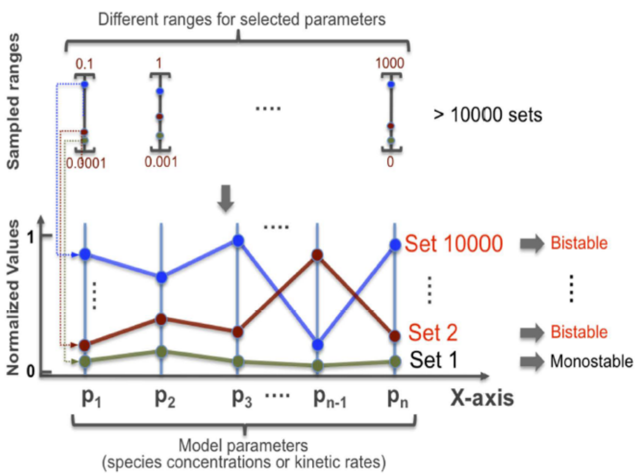


Figure 4.2
Example of a PC plot in DYVIPAC

4.4 CONTINUATION METHODS

Continuation methods are used to solve N-dimensional nonlinear equations of the form

$$F(x) = 0, \qquad F : \mathbf{R}^N \to \mathbf{R}^N \tag{4.6}$$

where F is a smooth mapping of x
If there is any approximate knowledge of the solution x_0 and nonsingular approximation of jacobian $A = F'(x_0)$, any Newton-type alogrithms such as

$$x_{i+1} := x_i - A_i^{-1} F(x_i), \quad i = 0, 1, \ldots \tag{4.7}$$

can be applied. But if the initial guess is not close enough to the equilibrium, which is often the case with high dimensional systems, this method seldom works.
Instead, we consider a new parametrization of the original equation (Allgower and Georg [2003])
$H : R^N \times R \to R^N$ such that

$$H(x, 1) = G(x), \quad H(x, 0) = F(x) \tag{4.8}$$

where $G : \mathbf{R}^N \to \mathbf{R}^N$ is a smooth map having **known** zero points and H is also smooth. Then the task is modified to tracing an implicitly defined curve $c(s) \in H^{-1}(0)$ from a starting point $(x_1, 0)$ to solution point $(x_0, 1)$.This is the basic idea behind a numerical continuation algorithm. If the jacobian instead turns out to be singular, then x_0 is a bifurcation point of the system, giving rise to multiple solution branches of $F = 0$.

For example, following steps consist of pseudo-arclength continuation algorithm (Henderson [2007])

1. Find initial point x_0 on the solution curve of $F(x) = 0$
2. If the jacobian estimate is non-singular, use tangent of the solution curve to estimate new initial guess
3. Use iterative methods such as Newton's method to reach the solution manifold
4. Singularities in the Jacobian indicate bifurcation points
5. Type of bifurcation can be characterized by special functions which change sign at a bifurcation point

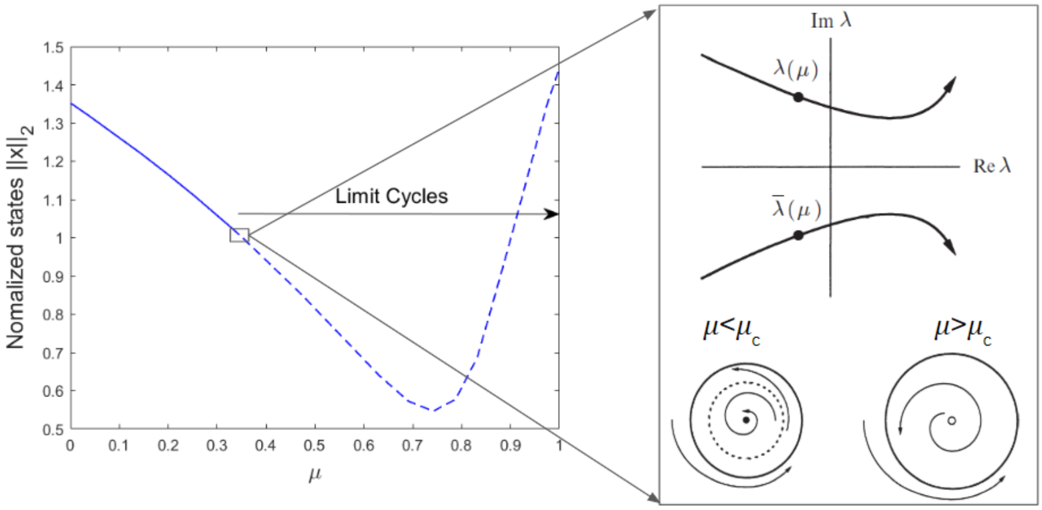


FIGURE 4.3
subcritical hopf bifurcation in a fitzhugh nagumo oscillator (continuation performed using COCO toolbox)

CHAPTER 5

HALF CENTER OSCILLATOR

Here, we study half center oscillator mechanism for locomotion generation using bifurcation analysis. The half center oscillator is one of the earliest attempts to model neural control of locomotion with a Central Pattern Generator (Brown [1914]). Numerous experiments (ex. Grillner et al. [1995]) indicate that rhythmic pattern of alternating bursts of flexor and extensor activities is produced by two symmetrically organized excitatory neural populations that drive alternating activity of flexor and extensor motoneurons and reciprocally inhibit each other via inhibitory interneurons. (detailed analysis of this mechanism can be found in Daun et al. [2009]). We use this mechanism as a testing bed for bifurcation analysis using numerical continuation methods.

5.1 SYSTEM MODEL

To make the analysis simple, we limit ourselves to single neurons instead of a neural population. Here the half center oscillator is modelled using two mutually inhibitory Fitzhugh Nagumo models which are 2 dimensional simplified models which nevertheless produces most of the characteristic behavior of neurons with very intuitive parameters. We consider following neuron model:

$$\begin{aligned} \dot{V}_i &= V_i - \frac{V_i^{2}}{3} - w_i + I_{\text{ext}} + I_{in,i} \quad i \in \{1,2\} \\ \tau \dot{w}_i &= V_i + a - bw_i \end{aligned} \tag{5.1}$$

The coupling term is given by:

$$I_{in,i} = g * V_j \qquad i,j \in \{1,2\}, \quad i \neq j \tag{5.2}$$

The individual neurons are parametrized such that they cannot produce any rhythmic activity on their own. This is achieved by applying a tonic forcing (I_{ext}) to both the neurons.
The system schematic is shown in figure 5.1

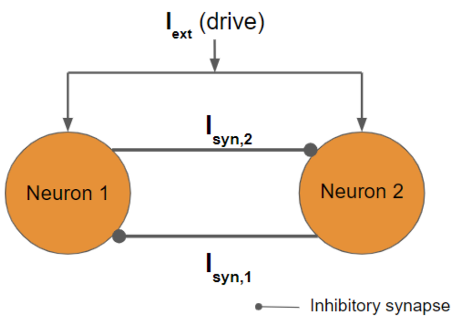


FIGURE 5.1

5.2 CONTINUATION ANALYSIS

We consider the neuron synaptic weights to be primary parameters since they play crucial role for rhythm generation. Continuation is performed in the COCO toolbox (Dankowicz and Schilder [2013]). Figure 5.2 shows the system behavior as inhibition between the two neurons is varied in the range $[0, 1]$. The solid and dashed line represent stable and unstable equilibrium points. As we increase g from 0, the fixed point changes its stability at $g = 0.329$ through a hopf bifurcation. The phase space looks topologically similar to figure 4.3, and a new limit cycle originates. Thus, for $g > 0.329$, the fixed point is no longer stable and system exhibits oscillatory behavior. We can call $g = 0.329$ as the threshold coupling strength since rhythm generation is not possible for lower coupling.

Further, using the hopf bifurcation point as initial guess, it is also possible to follow the bifurcation manifold along another parameter. We use the external drive (I) as second control parameter to perform hopf bifurcation continuation. Figure 5.3 shows the bifurcation manifold in two dimensional parameter space, I and g. It divides the space in two distinct dynamical regions, fixed point and limit cycle. To realize its application, let us assume that this rhythm generation model is used with a goal optimize some locomotion performance such as distance travelled or energy consumed during motion. Instead of blindly optimizing over whole parameter space, we can immediately constrain the parameters to values that lie in the limit cycle region. The computational efforts needed for numerical continuation are much lower since we simply use iterative algorithms to find a solution manifold. Hence, continuation analysis promises to be a powerful tool to characterize nonlinear dynamical systems.

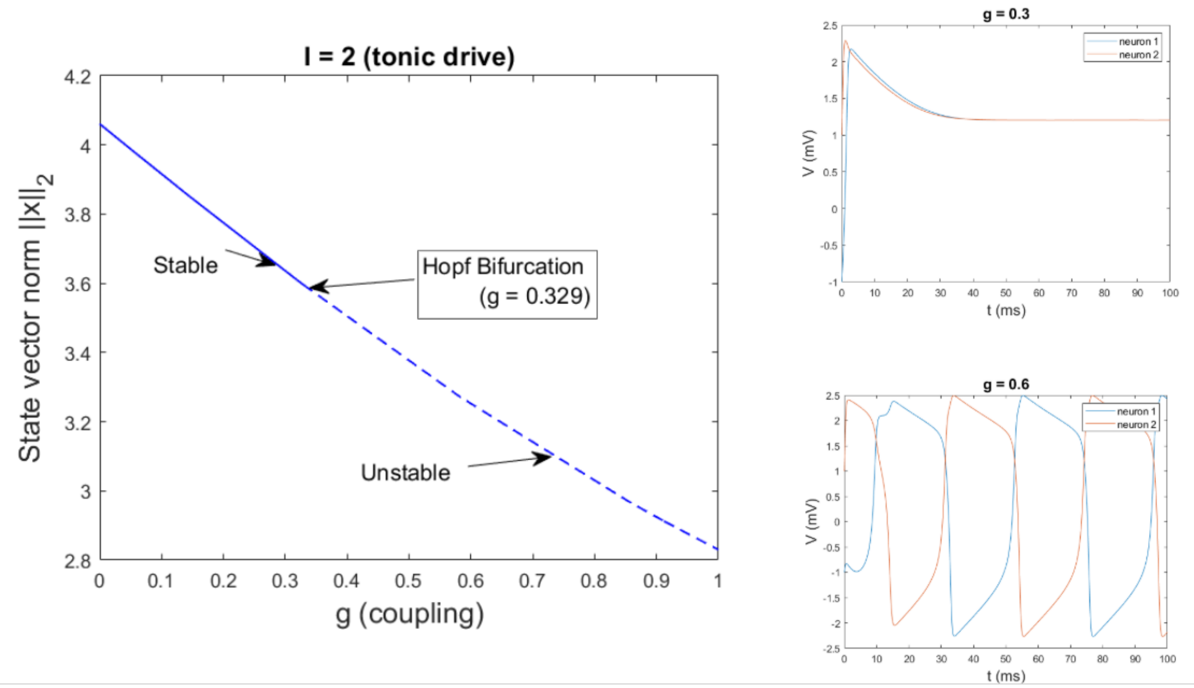


FIGURE 5.2
Equilibrium point continuation with coupling strength (g) as the control parameter and $I = 1$, $\tau = 80\ ms$, $a = 0.7$, $b = 0.8$

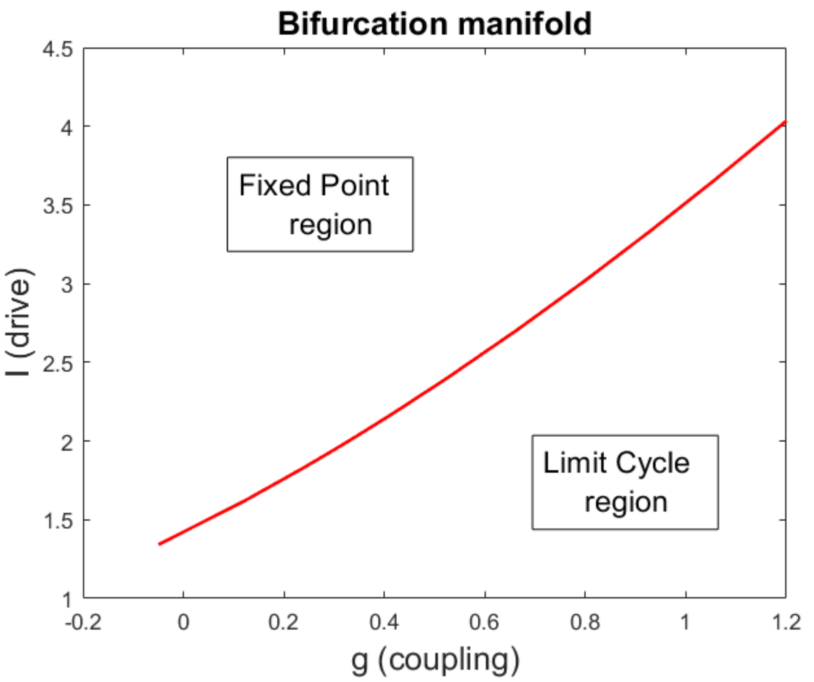


FIGURE 5.3
Bifurcation manifold for hopf bifurcation continuation with drive (I) and coupling strength (g) as control parameters

5.3 HIGH DIMENSIONAL CONTINUATION

For systems with a high dimensional parameter space, instead of continuation curves, we need to find a N dimensional manifold **M** in the parameter space.

$$F(\mathbf{u}, \lambda) = 0, \quad F : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n \tag{5.3}$$

A generic strategy is as follows (Henderson [2007])

1. Find a point u_i on the boundary of M_i
2. Build a neighborhood $\mathcal{N}_i$ of u_i
3. Use iterative methods such as Newton's method to project points in $\mathcal{N}_i$ on M_i
4. Singularities in the Jacobian indicate bifurcations which can be identified using special functions

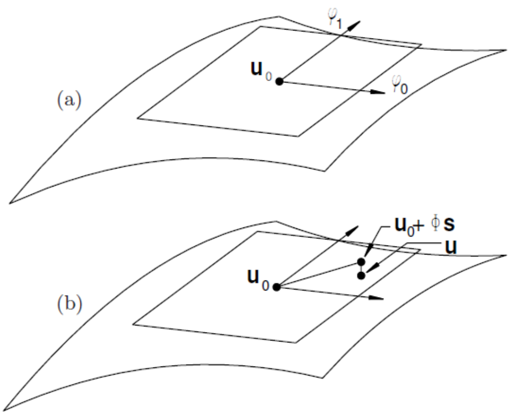


FIGURE 5.4
(a) Tangent space of the solution manifold M at u_0 (b) Projecting a point s in the tangent space orthonormally to obtain a point u in M

Chapter 6

Conclusions and future work

The Neural Engineering Framework shows promising results towards building a generic methodology for locomotion controllers and CPGs. This framework was used to model a CPG in lamprey as a distributed network of oscillators. Through the framework,a building block of rhythm generation was created, which can be coupled to other blocks in multiple ways to build neural control models. It could successfully generate a travelling wave for lamprey locomotion. Still, there are some limitations to this framework:

- The neural population is modeled with a simple one dimensional model of leaky integrator. This limits all possible dynamic realizations through these networks. For example, analysis of half center oscillator shows that the rhythm generation is a highly nonlinear process, which cannot be naively translated into a linear dynamical system
- Current overview of NEF does not cover the nonlinear transformations possible in neural populations, which might be of significant value for some problems
- The single harmonic oscillator given in this framework is limited by its parameterization of sinosoidal functions, since it is preferable to have generic encoding depending on a given dynamical system to be represented

The limitations due to parametrization can be mitigated by including a fourier series encoding in the neural populations that best approximate the desired dynamics. To extend the linearity assumption, it might be useful to incorporate higher dimensional models of neurons in this framework. But this requires a careful design of relevant encoding and decoding processes that are possible with the models.

The numerical continuation methods are also useful tools for nonlinear systems analysis. These methods along with optimization tools can be of great help in the design process for neural controllers. Some limitations to these tools are:

- Extensions to higher dimensions -
  In the half center analysis example, a bifurcation manifold could be easily traced in a two dimensional parameter space, which is just a curve. But while designing more complex neural controllers, the parameter space is often very large. Here, we need to trace high dimensional manifolds instead of curves, which brings new intricacies into consideration, such as representation of the manifold. This question is very nontrivial in high dimensions since different representations can have significant effect on accuracy and convergence speeds of the algorithms. A strong knowledge of algebraic topology is needed to have a better understanding of such manifolds. Henderson [2007] gives good overview for manifold representation methods and continuation algorithms in higher dimensions
- Another factor to consider is the type of bifurcations encountered in high dimensional spaces. Although there has been an extensive study on different bifurcations and their properties in 2 or 3

dimensional systems, theoretical foundations for higher dimensions are fairly limited.

To overcome above issues, a thorough study on continuation and manifold representation might prove useful. Nevertheless, it proves to be a powerful approach for a deeper understanding of nonlinear dynamical systems.

BIBLIOGRAPHY

E. L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 2003. ISBN 978-0-89871-544-6. doi: 10.1137/1.9780898719154. URL https://epubs.siam.org/doi/book/10.1137/1.9780898719154.

T. G. Brown. On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *The Journal of Physiology*, 48(1):18–46, Mar. 1914. ISSN 0022-3751. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1420503/.

H. Dankowicz and F. Schilder. *Recipes for Continuation*. Computational Science & Engineering. Society for Industrial and Applied Mathematics, May 2013. ISBN 978-1-61197-256-6. doi: 10.1137/1.9781611972573. URL https://epubs.siam.org/doi/book/10.1137/1.9781611972573.

S. Daun, J. E. Rubin, and I. A. Rybak. Control of oscillation periods and phase durations in half-center central pattern generators: a comparative mechanistic analysis. *Journal of Computational Neuroscience*, 27(1):3, Jan. 2009. ISSN 1573-6873. doi: 10.1007/s10827-008-0124-4. URL https://doi.org/10.1007/s10827-008-0124-4.

A. Dhooge, W. Govaerts, Y. A. Kuznetsov, H. G. E. Meijer, and B. Sautois. New features of the software MatCont for bifurcation analysis of dynamical systems. *Mathematical and Computer Modelling of Dynamical Systems*, 14(2):147–175, Apr. 2008. ISSN 1387-3954. doi: 10.1080/13873950701742754. URL https://doi.org/10.1080/13873950701742754.

E. Doedel. The computer-aided bifurcation analysis of predator-prey models. *Journal of Mathematical Biology*, 20(1):1–14, Aug. 1984. ISSN 1432-1416. doi: 10.1007/BF00275858. URL https://doi.org/10.1007/BF00275858.

C. Eliasmith and C. H. Anderson. *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. MIT Press, Aug. 2004. ISBN 978-0-262-55060-4. Google-Books-ID: J6jz9s4kbfIC.

R. FitzHugh. Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical Journal*, 1(6):445–466, July 1961. ISSN 0006-3495. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1366333/.

S. Grillner, T. Deliagina, A. El Manira, R. H. Hill, G. N. Orlovsky, P. Wallén, Ekeberg, and A. Lansner. Neural networks that co-ordinate locomotion and body orientation in lamprey. *Trends in Neurosciences*, 18(6):270–279, Jan. 1995. ISSN 0166-2236. doi: 10.1016/0166-2236(95)80008-P. URL http://www.sciencedirect.com/science/article/pii/016622369580008P.

M. E. Henderson. Higher-Dimensional Continuation. In B. Krauskopf, H. M. Osinga, and J. Galán-Vioque, editors, *Numerical Continuation Methods for Dynamical Systems*, pages 77–115. Springer Netherlands, Dordrecht, 2007. ISBN 978-1-4020-6355-8 978-1-4020-6356-5. doi: 10.1007/978-1-4020-6356-5_3. URL http://link.springer.com/10.1007/978-1-4020-6356-5_3.

A. J. Ijspeert. A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, 84(5):331–348, Apr. 2001. ISSN 1432-0770. doi: 10.1007/s004220000211. URL https://doi.org/10.1007/s004220000211.

A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653, May 2008. ISSN 0893-6080. doi: 10.1016/j.neunet.2008.03.014. URL http://www.sciencedirect.com/science/article/pii/S0893608008000804.

H. Kimura, S. Akiyama, and K. Sakurama. Realization of Dynamic Walking and Running of the Quadruped Using Neural Oscillator. *Autonomous Robots*, 7(3):247–258, Nov. 1999. ISSN 1573-7527. doi: 10.1023/A:1008924521542. URL https://doi.org/10.1023/A:1008924521542.

H. Lecar. Morris-Lecar model. *Scholarpedia*, 2(10):1333, Oct. 2007. ISSN 1941-6016. doi: 10.4249/scholarpedia.1333. URL http://www.scholarpedia.org/article/Morris-Lecar_model.

C. Liu, X. Liu, and S. Liu. Bifurcation analysis of a Morris–Lecar neuron model. *Biological Cybernetics*, 108(1):75–84, Feb. 2014. ISSN 1432-0770. doi: 10.1007/s00422-013-0580-4. URL https://doi.org/10.1007/s00422-013-0580-4.

M. Lodi, A. Shilnikov, and M. Storace. CEPAGE: A toolbox for Central Pattern Generator analysis. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, May 2017. doi: 10.1109/ISCAS.2017.8050580. ISSN: 2379-447X.

K. Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological Cybernetics*, 52(6):367–376, Oct. 1985. ISSN 1432-0770. doi: 10.1007/BF00449593. URL https://doi.org/10.1007/BF00449593.

K. Matsuoka. Analysis of a neural oscillator. *Biological Cybernetics*, 104(4):297–304, May 2011. ISSN 1432-0770. doi: 10.1007/s00422-011-0432-z. URL https://doi.org/10.1007/s00422-011-0432-z.

C. Morris and H. Lecar. Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal*, 35(1):193–213, July 1981. ISSN 00063495. doi: 10.1016/S0006-3495(81)84782-0. URL https://linkinghub.elsevier.com/retrieve/pii/S0006349581847820.

J. Nagumo, S. Arimoto, and S. Yoshizawa. An Active Pulse Transmission Line Simulating Nerve Axon. *Proceedings of the IRE*, 50(10):2061–2070, Oct. 1962. ISSN 0096-8390, 2162-6634. doi: 10.1109/JRPROC.1962.288235.

L. K. Nguyen, A. Degasperi, P. Cotter, and B. N. Kholodenko. DYVIPAC: an integrated analysis and visualisation framework to probe multi-dimensional biological networks. *Scientific Reports*, 5, July 2015. ISSN 2045-2322. doi: 10.1038/srep12569. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4518224/.

V. Novičenko and K. Pyragas. Computation of phase response curves via a direct method adapted to infinitesimal perturbations. *Nonlinear Dynamics*, 67(1):517–526, Jan. 2012. ISSN 1573-269X. doi: 10.1007/s11071-011-0001-y. URL https://doi.org/10.1007/s11071-011-0001-y.

S. H. Strogatz. *Nonlinear Dynamics and Chaos : With Applications to Physics, Biology, Chemistry, and Engineering*. CRC Press, May 2018. ISBN 978-0-429-49256-3. doi: 10.1201/9780429492563. URL https://www.taylorfrancis.com/books/9780429492563.

G. Taga, Y. Yamaguchi, and H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65(3):147–159, July 1991. ISSN 0340-1200, 1432-0770. doi: 10.1007/BF00198086. URL `http://link.springer.com/10.1007/BF00198086`.