Neuroinspired control strategies with applications to flapping flight

by

Michael Ray Dorothy

Dissertation

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Associate Professor Soon-Jo Chung, Chair and Director of Research
Professor Seth Hutchinson
Professor Petros Voulgaris
Assistant Professor Aditya Paranjape, Indian Institute of Technology, Bombay
Chris Kroninger, Army Research Laboratory

Abstract

This dissertation is centered on a theoretical, simulation, and experimental study of control strategies which are inspired by biological systems. Biological systems, along with sufficiently complicated engineered systems, often have many interacting degrees of freedom and need to excite large-displacement oscillations in order to locomote. Combining these factors can make high-level control design difficult. This thesis revolves around three different levels of abstraction, providing tools for analysis and design.

First, we consider central pattern generators (CPGs) to control flapping-flight dynamics. The key idea here is dimensional reduction - we want to convert complicated interactions of many degrees of freedom into a handful of parameters which have intuitive connections to the overall system behavior, leaving the control designer unconcerned with the details of particular motions. A rigorous mathematical and control theoretic framework to design complex three-dimensional wing motions is presented based on phase synchronization of nonlinear oscillators. In particular, we show that flapping-flying dynamics without a tail or traditional aerodynamic control surfaces can be effectively controlled by a reduced set of central pattern generator parameters that generate phase-synchronized or symmetry-breaking oscillatory motions of two main wings. Furthermore, by using a Hopf bifurcation, we show that tailless aircraft (inspired by bats) alternating between flapping and gliding can be effectively stabilized by smooth wing motions driven by the central pattern generator network. Results of numerical simulation with a full six-degree-of-freedom flight dynamic model validate the effectiveness of the proposed neurobiologically inspired control approach.

Further, we present experimental micro aerial vehicle (MAV) research with low-frequency flapping and articulated wing gliding. The importance of phase difference control via an abstract mathematical model of central pattern generators is confirmed with a robotic bat on a 3-DOF pendulum platform. An aerodynamic model for the robotic bat based on the complex wing kinematics is presented. Closed loop experiments show that control dimension reduction is achievable - unstable longitudinal modes are stabilized and controlled using only two control parameters. A transition of flight modes, from flapping to gliding and vice-versa, is demonstrated within the CPG control scheme.

The second major thrust is inspired by this idea that mode switching is useful. Many bats and birds adopt a mixed strategy of flapping and gliding to provide agility when necessary and to increase overall efficiency. This work explores dwell time constraints on switched systems with multiple, possibly disparate invariant limit sets. We show that, under suitable conditions, trajectories globally converge to a superset of the limit sets and then remain in a second, larger superset. We show the effectiveness of the dwell-time conditions by using examples of nonlinear switching limit cycles from our work on flapping flight.

This level of abstraction has been found to be useful in many ways, but it also produces its own challenges. For example, we discuss death of oscillation which can occur for many limit-cycle controllers and the difficulty in incorporating fast, high-displacement reflex feedback. This leads us to our third major thrust - considering biologically realistic neuron circuits instead of a limit cycle abstraction. Biological neuron circuits are incredibly diverse in practice, giving us a convincing rationale that they can aid us in our quest for flexibility. Nevertheless, that flexibility provides its own challenges. It is not currently known how most biological neuron circuits work, and little work exists that connects the principles of a neuron circuit to the principles of control theory.

We begin the process of trying to bridge this gap by considering the simplest of classical controllers, PD control. We propose a simple two-neuron, two-synapse circuit based on the concept that synapses provide attenuation and a delay. We present a simulation-based method of analysis, including a smoothing algorithm, a steady-state response curve, and a system identification procedure for capturing differentiation.

There will never be One True Control Method that will solve all problems. Nature's solution to a diversity of systems and situations is equally diverse. This will inspire many strategies and require a multitude of analysis tools. This thesis is my contribution of a few.

To all the people who put up with me... and even some of those who didn't

Contents

Chapter 1

Introduction

Engineered flapping flight holds promise for creating biomimetic micro aerial vehicles (MAVs) flying in low Reynolds number regimes (Re$< 10^5$) where rigid fixed wings drop substantially in aerodynamic performance. MAVs are typically classified as having maximum dimensions of 15 cm and flying at a nominal speed of 1–20 m/s in tight urban environments [1, 2]. Although natural flyers such as bats, birds, and insects have captured the imaginations of scientists and engineers for centuries, the maneuvering characteristics of unmanned aerial vehicles (UAVs) are nowhere near the agility and efficiency of animal flight [3–5]. Such highly maneuverable MAVs will make paradigm-shifting advances in monitoring of critical infrastructure such as power grids, bridges, and borders, as well as in intelligence, surveillance, and reconnaissance applications.

The central theme of this dissertation is to investigate the control and synchronization of coupled nonlinear oscillators, inspired by central pattern generators (CPGs) found in animal spinal cords, along with the complimentary peripheral nervous system, to control biomimetic flapping flight (see Figure 1.1). An engineered CPG network, which ensures the stability and robust adaptation of motion, can signifi-

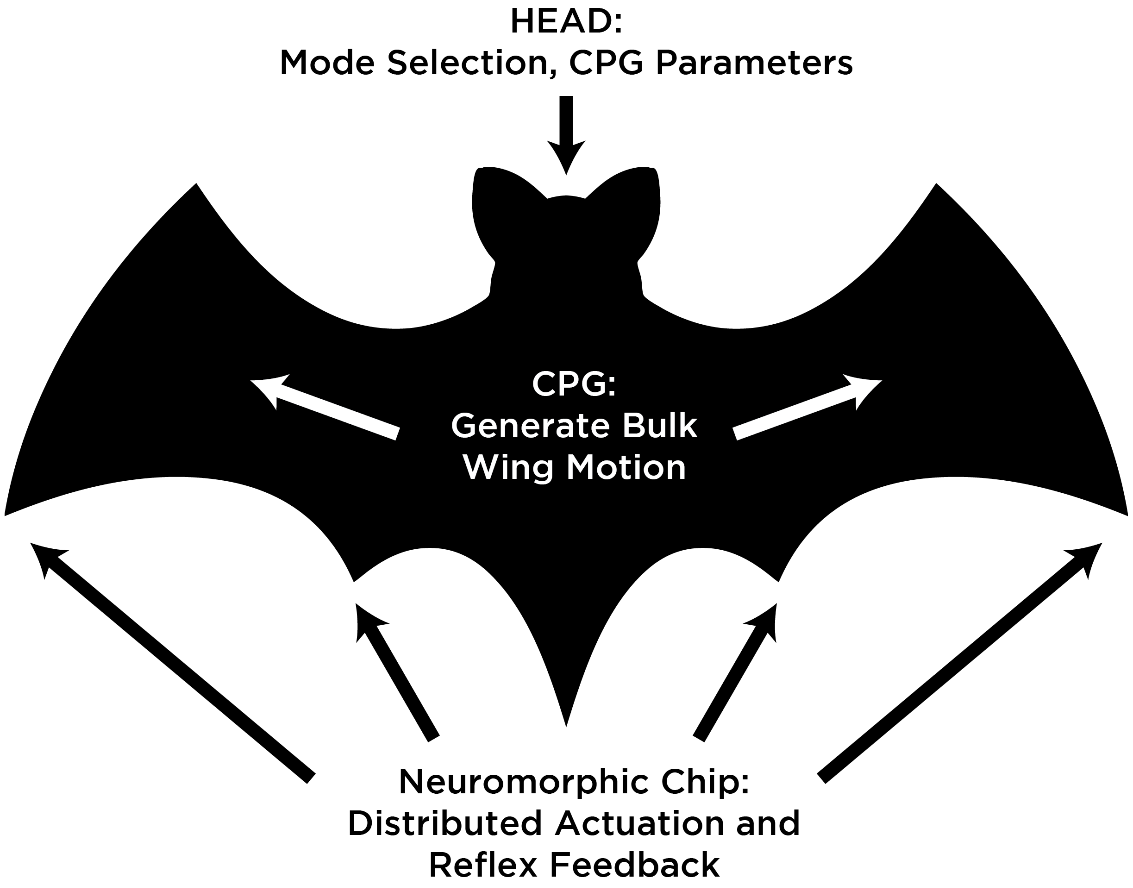


Figure 1.1: An overview of the three key components in this dissertation.

cantly reduce the complexity associated with flapping flight. Unique to this research approach is the potential to reverse-engineer the key mechanisms of highly adaptive and robust rhythmic pattern modulations of flapping flight by integrating the neurobiological principles with the rigorous mathematical tools borrowed from nonlinear synchronization theory and flight dynamics and controls.

Previous robotic flapping flyers and their control design consider one or two degrees of freedom in the wings [6–18]. However, even insects like the dragonfly (*Anax parthenope*) are reported to have complex three-dimensional movements by actively controlling flapping and twisting of four independent wings [3].

Furthermore, prior studies in flapping flight [1–3, 5, 17–27] assumed a very simple sinusoidal function for each joint to generate flapping oscillations, without deliberating on how multiple limbs (or their nervous systems) are connected and actuated to follow such a time-varying reference trajectory. However, as shall be seen later in this work, the use of sinusoidal functions to generate the oscillatory motions of the wings does not permit stable and agile flapping flying maneuvers especially with time-varying oscillation frequency ($\omega(t)$) and amplitude. Experimental results using high speed cameras have shown that the flapping motions in bats and birds are more complicated than perfect sinusoidal [3, 28] with a fixed amplitude. In order to bridge this gap, this work aims to establish a novel adaptive CPG-based control theory for flapping flight through neuromechanical modeling, nonlinear control and synchronization, numerical simulation, and experimentation.

In addition to pure flapping, birds switch between gliding and flapping - gliding when energy is plentiful and required maneuvering is minimal; flapping to gain energy or perform aggressive corrections in course. Our selection of Hopf oscillators to represent neuronal oscillators in Section 2.2 has a single bifurcation parameter which we use for rapid inhibition of oscillation (i.e., changing from a flapping mode to a gliding mode). This is a switching parameter.

Analyzing such switched systems has been a topic of high interest for a few decades [29]. One extremely common assumption among most of the work in the field is that each subsystem has a unique, stable equilibrium and that this equilibrium is common among all subsystems. This assumption was first challenged in a work by Alpcan and Başar, considering the possibility of having subsystems with equilibria not in common [30]. We further this thrust by incorporating the possibility of limit cycles or multiple equilibria in each individual subsystem.

The main tool that we preserve from the bulk of the switched systems literature is the dwell time. This is a constraint on how quickly the system can switch between modes. We provide a computation of dwell time for systems potentially containing a complicated set of equilibrium and non-equilibrium steady-states. In particular, we show that trajectories globally converge to a superset of the limit sets and then remain in a second, larger superset. We show the effectiveness of the dwell-time conditions by using examples of switching limit cycles from our work on flapping flight.

Next, we apply this theory in simulation and experimentation. We provide a full dynamic model, give 6-DOF simulation results, and detail a 8-DOF RoboBat supported by a compound pendulum which tests open loop and closed loop performance of phase difference control using coupled abstracted neuronal oscillators. This confirms the idea that dimensional reduction can be performed, allowing the control designer to use a small number of parameters which are intuitively connected to the vehicle dynamics while ignoring the complicated details of the wing motion. We stabilize the bat in conditions that are passively unstable, and we exert control in order to change the general location of the non-equilibrium steady-state.

Finally, in response to challenges we will find with the abstracted neuronal oscillators, we turn our attention to developing a simple control scheme for a spike/bursting neural network which can be implemented in analog hardware. Small vehicles (in particular, MAVs) have minimal size, weight, and power budgets. Analog hardware has promise to deliver efficient control systems for these vehicles [31,32], but encoding complex behavior patterns without traditional digital computing paradigms remains a challenge. Biological neuronal systems may solve this problem using a hybrid digital/analog strategy, but a full conceptual framework is yet unknown. Our goal is to develop the basics of a control theoretic framework which retains sufficiently rich behavior to someday reproduce the complexity and efficiency of biological control systems. Rather than being a purely scientific endeavor, we have made choices which also make hardware implementation feasible in the near or medium term.

We use a neuron model from Izhikevich [33] and a synapse model from Rabinovich [34]. Using the intuition that synapses cause attenuation and delay, we design a simple circuit intended to behave like a PD-controller. It consists of only two neurons, which are mutually excitatory. They are conceived of as being antagonistic motoneurons, common in biological systems involving antagonistic muscles, but are not immediately recognizable for engineered systems. Making assumptions on input/output signals allows us to make this connection.

Then, we provide a method of analysis for such neuron controllers. We hypothesize that differentiation is being performed instantaneousness in analog, and so present a delay-free smoothing algorithm that makes system identification more accurate. We compute steady-state behavior and use numerical system identification tools to analyze the derivative component. Finally, we test these tools by comparing the output of a simple pendulum system when driven by the neuron circuit and the identified model PD controller.

Additional literature review will be presented in the chapter introductions.

1.1 Organization

The organization and flow of this dissertation is shown in Figure 1.2. In Chapter 2, we prove phase synchronization characteristics for networks of coupled Hopf oscillators. In addition to the benefits of this scheme, we describe some challenges which drive the remainder of the dissertation. In Chapter 3, we present the general framework of switched systems. Further, we prove stability characteristics in terms of dwell time, entry sets, and no-escape sets for a class of switched systems with suitable Lyapunov functions. In Chapter 4, we validate the switching CPG scheme in both simulation

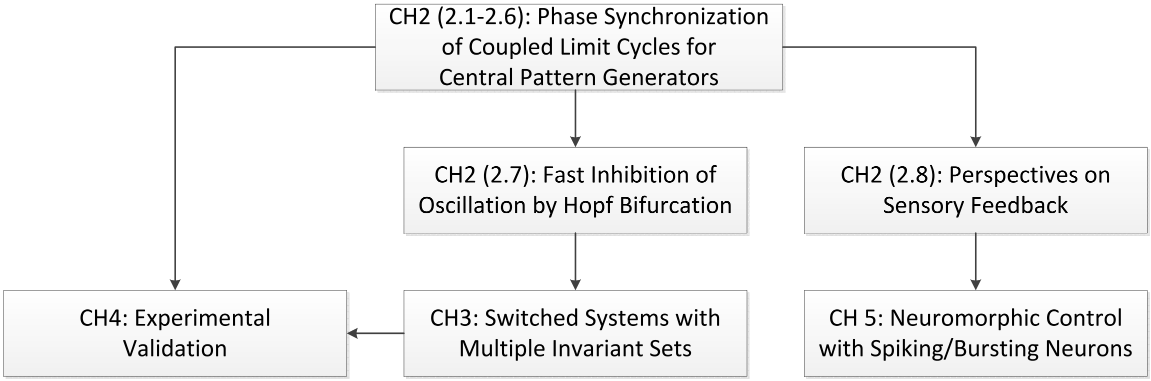

CH2 (2.1-2.6): Phase Synchronization of Coupled Limit Cycles for Central Pattern Generators
CH2 (2.7): Fast Inhibition of Oscillation by Hopf Bifurcation
CH2 (2.8): Perspectives on Sensory Feedback
CH4: Experimental Validation
CH3: Switched Systems with Multiple Invariant Sets
CH 5: Neuromorphic Control with Spiking/Bursting Neurons

Figure 1.2: This diagram shows the organization and flow of the dissertation.

and experimentation. In Chapter 5, we explore a simple PD-like network built from realistic spiking neurons and their associated synapses.

Chapter 2

Networks of Coupled Oscillators

2.1 Fundamentals of Limit Cycle Control Inspired by Neuroscience

Hooper [35] defines the central pattern generators of animals as neural networks that can endogenously (i.e., without rhythmic sensory or central brain input) produce coordinated patterns of rhythmic outputs. The self-sustained nature of CPGs is believed to reduce the computation burden of the brain. As illustrated in Fig. 2.1, the central controller, similar to the brain of an animal, is expected to stabilize the vehicle dynamics by commanding a reduced number of variables such as the frequency and phase difference of the oscillators instead of directly controlling multiple joints. The existence of CPGs has been confirmed by biologists [35–43]. Experiments with limbed vertebrates have shown that individual limbs can produce rhythmic movements endogenously [35, 44]. Such empirical data have been interpreted as evidence that each limb has its own CPGs that can behave in a self-sustained way. However, sensory feedback is also known to play a crucial role in altering motor patterns [35,45] to cope

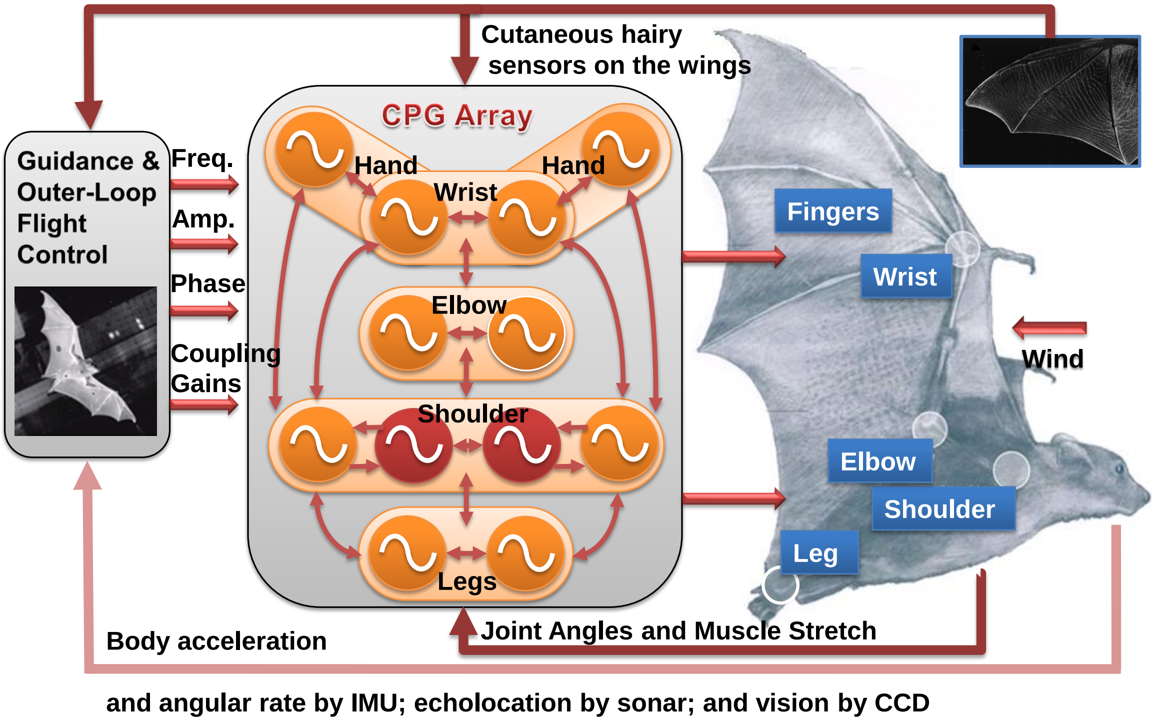


Figure 2.1: Hierarchical control structures with the main controller and the CPG network. The outer-loop flight control modulates the rhythmic patterns (frequency, amplitude, phase lag, coupling gains) of the CPG network, without the need for directly controlling a multitude of joints.

with environmental perturbations. Incorporation of simple sensory feedback into the CPG model has been presented in [46] for a turtle robot.

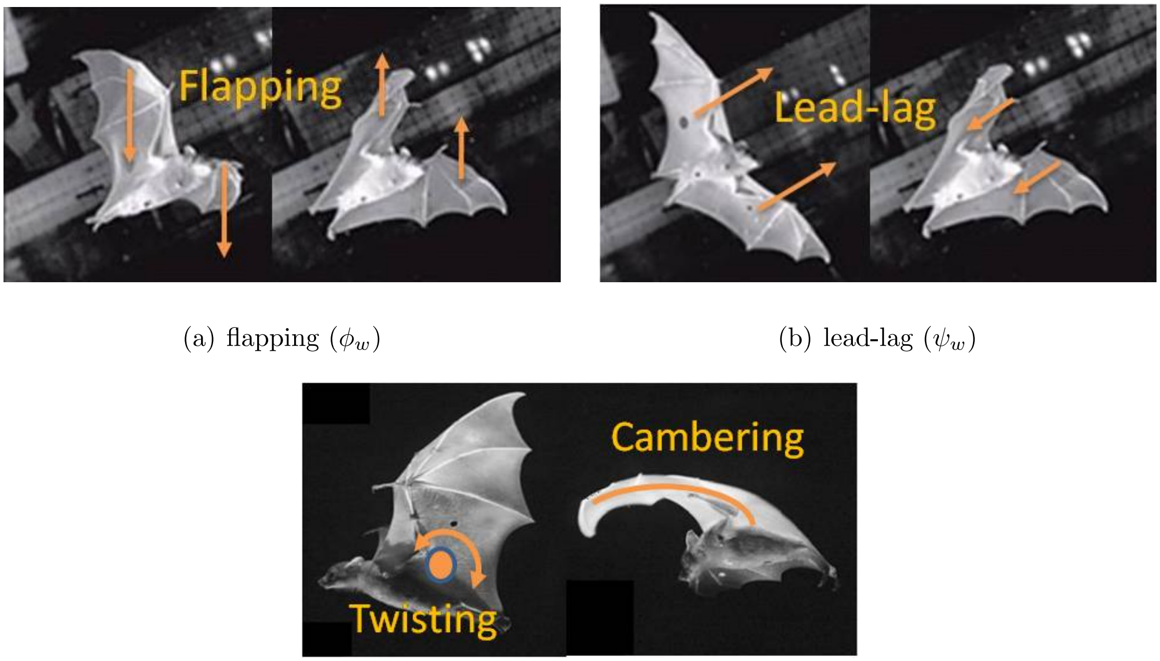
The most popular animal model for CPGs has been the lamprey, a primitive eel-like fish [47]. While the robotics community eagerly embraced the concept of CPG models for swimming or walking robots [46, 48–50], this work reports the first CPG-based control for flapping flight. The use of nonlinear oscillators for insect flapping flight has also been suggested by some biologists [24, 27].

While unsteady aerodynamics of flapping flight in low Reynolds number regimes has been extensively studied through numerical [2, 11, 20, 25, 26, 51, 52] and experimental studies [1, 9, 12, 19, 21, 28], one of the most interesting and least understood aspects of bio-inspired flapping flight is how to precisely control and synchronize a large number of interacting limbs and joints that generate complex three-dimensional oscillatory movements of the wings governed by unsteady aerodynamic forces. In this work, we focus on three stereotyped motion primitives to define the three dimensional movements of wings: main flapping (stroke) motion (Fig. 2.2a), lead-lag motion (Fig. 2.2b), and wing pitch twisting (Fig. 2.2c). Studying how to produce such synchronized wing motions is expected to shed light on the key characteristics of animal flapping flyers.

2.2 Robust and Adaptive Flapping Pattern Generation by CPGs

Our neurobiologically inspired approach begins by deriving an effective mathematical model of CPGs based on coupled nonlinear limit cycle dynamics. Once neurons form reciprocally inhibiting relations, they oscillate and spike periodically. An abstract mathematical model of complicated neuron models can be obtained by coupled nonlinear limit cycles that essentially exhibit the rhythmic behaviors of coupled neuronal networks. In the field of nonlinear dynamics, a limit cycle is defined as an isolated closed trajectory that exhibits self-sustained oscillation [53,54]. If *stable*, small perturbations (initial conditions) will be forgotten and the trajectories will converge to the limit cycle. This superior robustness makes a limit cycle an ideal simplified dynamic model of CPGs.

In the present work, we use the following limit-cycle model called the Hopf oscil-



(a) flapping (ϕ_w)

(b) lead-lag (ψ_w)

(c) pitch (θ_w) and cambering

Figure 2.2: Basic wing movements of bats (pictures from [21]). Except for cambering, birds exhibit similar wing movements. Twisting (pitching) changes the effective angle of attack while cambering changes the aerodynamic efficiency. The fingers and hind legs control the tension of the flexible membrane wings, which distinguish bats from birds [5].

lator, named after the supercritical Hopf bifurcation model with $\sigma = 1$:

$$\frac{d}{dt}\begin{pmatrix} u-a \\ v \end{pmatrix} = \begin{bmatrix} -\lambda\left(\frac{(u-a)^2+v^2}{\rho^2} - \sigma\right) & -\omega(t) \\ \omega(t) & -\lambda\left(\frac{(u-a)^2+v^2}{\rho^2} - \sigma\right) \end{bmatrix}\begin{pmatrix} u-a \\ v \end{pmatrix} + \mathbf{u}(t) \tag{2.1}$$

$$\text{Equivalently,}\quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x};\rho;\sigma) + \mathbf{u}(t), \quad \text{with}\quad \mathbf{x} = (u-a, v)^T$$

where the $\lambda > 0$ denotes the convergence rate to the symmetric limit circle of the radius $\rho > 0$ and $\mathbf{u}(t)$ is an external or coupling input. For a single Hopf oscillator with $\mathbf{u}(t) = 0$, a Lyapunov function $V = \left(\frac{(u-a)^2+v^2}{\rho^2} - 1\right)^2$ can be used to prove global asymptotic stability to the circular limit cycle. Also, the bifurcation parameter σ can change 1 to -1 such that $\left(\frac{(u-a)^2+v^2}{\rho^2}+1\right)$. This would change the stable limit cycle dynamics to the dynamics with a globally stable equilibrium point at the bias "a" (see [53]). Such a change can be used to turn the flapping oscillatory motion to the gliding mode, as shall be seen in Section 4.2. We assume $\sigma = 1$ unless noted otherwise. For coupled Hopf oscillators, the stability proof is much more involved and discussed in Section 2.3.

Also, the possibly time-varying parameter $\omega(t) > 0$ determines the oscillation frequency of the limit cycle. A time-varying $a(t)$ sets the bias to the limit cycle such that it converges to $u(t) = \rho\cos(\omega t + \delta) + a$ and $v(t) = \rho\sin(\omega t + \delta)$ on a circle. This bias "a" does not change the results of the stability proof. The output variable to generate the desired oscillatory motion of each joint is the first state u from the Hopf oscillator model in Eq. (2.1).

The Hopf oscillator has been a popular dynamic model of the engineered CPG arrays (e.g., see the salamander robot [49,55] and the turtle robot [46]). The stability of coupled Hopf oscillators has been extensively investigated in [46, 56]. One nice property of the Hopf oscillator in Eq. (2.1) is that its limit cycle is a symmetric circle as opposed to Van der Pol [36] or Rayleigh oscillators [53]:

$$\mathbf{f}(\mathbf{R}(\Delta)\mathbf{x};\rho;\sigma) = \mathbf{R}(\Delta)\mathbf{f}(\mathbf{x};\rho;\sigma) \quad \mathbf{R}(\Delta) = \begin{bmatrix} \cos\Delta & -\sin\Delta \\ \sin\Delta & \cos\Delta \end{bmatrix} \tag{2.2}$$

where $\mathbf{R}(\Delta) \in \mathcal{SO}(2)$ is a 2D rotational transformation such that $\mathbf{R}(-\Delta) = \mathbf{R}^{-1}(\Delta) = \mathbf{R}^T(\Delta)$. Also, its scaling factor can be expressed as

$$\mathbf{f}(g\mathbf{x};\rho;\sigma) = g\mathbf{f}(\mathbf{x};\rho/g;\sigma). \tag{2.3}$$

As shall been seen later, this property is exploited in the stability proof of phase synchronization.

2.2.1 Key Advantage of CPG-based Control: Reduced Dimensionality and Bandwidth Requirement

The CPGs in animal spinal cords are known to relieve the computation burden of locomotion in the brain [35, 47]. Similarly, one significant advantage of CPG-based control over conventional control approaches is that CPG-based control reduces the dimensionality and bandwidth of signals required from the main controller to its actuators. As shown in Figure 2.1, the main outer-loop flight controller needs to command only the reduced number of CPG parameters (e.g., frequency, phase lag, and coupling gains) and much less frequently, instead of directly commanding time-specific reference signals for all the degrees of freedom in the wings and the body.

Combining feedback control with model-based reinforcement learning [57] is particularly attractive for control of agile aerospace vehicles, due to the superior robustness and adaptability. Unfortunately, online learning control is subject to the curse of dimensionality, exacerbated by a multitude of joints in the wings. In contrast, the learning-based controller using CPGs needs to adapt only the reduced dimensional CPG parameters. Such a model reduction approach for flight control has not been exploited in the literature. The reduced dimensionality of the CPG-based approach (i.e., controlling the reduced CPG parameters instead of all relevant degrees of freedom) makes learning-based adaptive flight control more practical.

2.2.2 Key Advantage of Hopf-based Control: Adaptive Pattern Modulation

Birds and bats modulate the CPG parameters (frequency, phase difference, and amplitude) for the flapping, twisting, lead-lag, cambering, and flexing of the wings during their flight, as a function of flight speed [3,58,59] and flight modes (e.g., turning, cruising, hovering, preying, and perching). High-speed film analyses [3, 58] reveal that the flapping angle and frequency are largest at zero forward speed or in hovering flight, and decrease with increasing flight speed V (e.g., $\propto V^{-0.277}$ for some bats [58]). Such time-varying CPG parameters, shown in Figure 2.1, will change the shape, size, and flexing of the wings, which constitute the morphological flight parameters [5]. Prior studies in flapping flight, although true in steady flight, assume that there is a constant or very narrow range of optimal frequency or amplitude [1, 2, 6, 9, 16, 18, 20, 25, 26, 52]. Agile vehicles with multiple flight modes may require a large envelope or discontinuous parameter changes. Typical sinusoidal signals can be modulated well for continuous and discontinuous changes in frequency, but discontinuous changes in amplitude or bias would require a low-pass filter, for which tuning may be burdensome. The Hopf oscillator guarantees continuous transitions for any time-varying input of these parameters, without any tuning. This same behavior extends to the ability to handle any initial conditions and the ability to reject disturbances.

2.2.3 Key Advantage of Hopf-based Control: Symmetric and Symmetry-Breaking Oscillation

Bats exhibit complex wing flapping motions generated by their multijointed and compliant wings, resulting in a closed orbit quite different from a symmetric circle or ellipse of a sinusoidal function. One aim of the neurobiological approach to engineered flapping flight is to produce the analytical model of a wing beat oscillator that matches empirical data [21,28,58,60]. While the benefits of nonlinear limit cycles for CPG models are articulated above, deriving an effective CPG model for engineered flapping flight has been largely an open problem (e.g., limit cycle dynamics, network topology, and how to integrate input and feedback signals). The key research issues include how to ensure the amplitude or phase synchronization of multiple coupled CPG oscillators and how to opportunistically break the symmetry of the oscillators for performing maneuvering of agile flapping flight. Unique to this Hopf formulation, as opposed to modulated sine, is the ability to set and control phase differences. Observations of birds have found these phase differences to be key in performing maneuvers [61], and we will later see that they can be useful for vehicle stability. First, we present how to construct stable coupled oscillators in the next section.

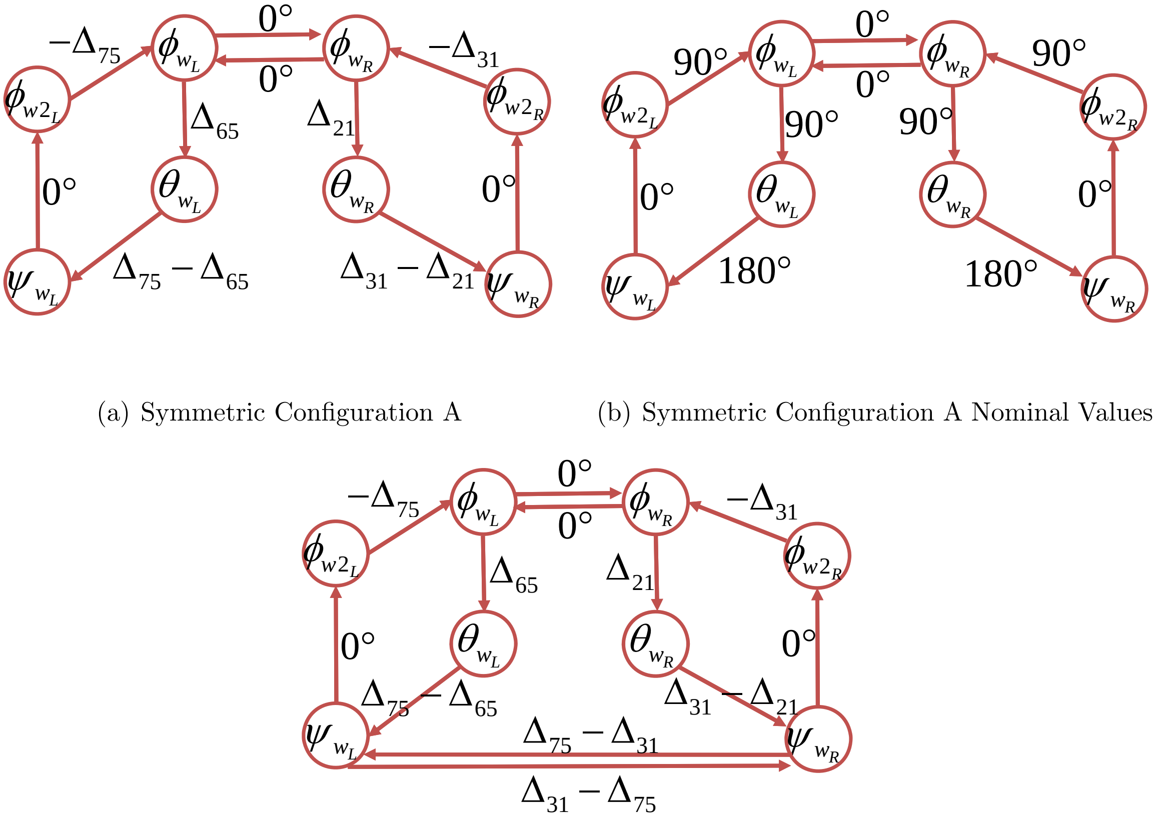
2.3 Almost Global Exponential Synchronization of CPG Oscillators

Synchronization means an exact match of the scaled amplitude or the frequency in this work. Hence, phase synchronization permits different actuators to oscillate at the same frequency but with a prescribed phase lag. However, a sinusoidal function is not adequate to create the complex coupling and synchronization between various joints and limbs. Hence, the use of coupled nonlinear oscillators in this work provides a feasible solution to construct complex synchronized motions of multiple wing joints. In essence, each CPG dynamic model in Eq. (2.1) is responsible for generating the limiting oscillatory behavior of a corresponding joint, and the diffusive coupling among CPGs reinforces phase synchronization. For example, the flapping angle has roughly a 90-degree phase difference with the pitching joint to maintain the positive angle of attack (see the actual data from birds in [3]). The oscillators are connected through diffusive couplings, and the i-th Hopf oscillator can be rewritten with a diffusive coupling with the phase-rotated neighbor.

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i; \rho_i) - k \sum_{j \in \mathcal{N}_i}^{m_i} \left(\mathbf{x}_i - \frac{\rho_i}{\rho_j} \mathbf{R}(\Delta_{ij}) \mathbf{x}_j \right) \tag{2.4}$$

where the Hopf oscillator dynamics $\mathbf{f}(\mathbf{x}_i; \rho_i)$ with $\sigma = 1$ is defined in Eq. (2.1), $\mathcal{N}_i$ denotes the set that contains only the local neighbors of the i-th Hopf oscillator, and m_i is the number of the neighbors. The 2×2 matrix $\mathbf{R}(\Delta_{ij})$ is a 2-D rotational transformation of the phase difference Δ_{ij} between the i-th and j-th oscillators. The positive (or negative) Δ_{ij} indicates how much phase the i-th member leads (or lags) from the j-th member and $\Delta_{ij} = -\Delta_{ji}$. The positive scalar k denotes the coupling gain.

We construct as many degrees of freedom as needed to more accurately model the joints of the wings, but let us focus on the key three flapping motions defined in Fig. 2.2, namely flapping angle ϕ_w, wing pitch (twisting) angle θ_w, and wing lead-lag angle ψ_w. Additionally, we assume that there is a second flapping joint ϕ_{w2} in the wing that can reduce the drag in the upstroke by folding the wings toward the body.



(a) Symmetric Configuration A

(b) Symmetric Configuration A Nominal Values

(c) Symmetric Configuration B

Figure 2.3: Graph configurations of the coupled Hopf oscillators on balanced graphs. Many other configurations are permitted in this paper and the unidirectional couplings can be replaced by the bi-directional couplings. The numbers next to the arrows indicate the phase shift Δ_{ij} from the i-th member to the j-th member while Figure b shows the nominal values of the phase shift from the symmetric wing configuration such that $\Delta_{21} = \Delta_{65}$=90 deg. and $\Delta_{31} = \Delta_{75} = -90$ deg. Such phase shifts define flight modes (wing movement gaits). Figure c shows an alternative configuration with additional coupling between the left and right wings.

Then, we can construct the whole state vector of the coupled oscillator such as

$$\{\mathbf{x}\} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \mathbf{x}_6 \\ \mathbf{x}_7 \\ \mathbf{x}_8 \end{pmatrix} = \begin{pmatrix} (u_1 - a_1, v_1)^T \\ (u_2 - a_2, v_2)^T \\ (u_3 - a_3, v_3)^T \\ (u_4 - a_4, v_4)^T \\ (u_5 - a_5, v_5)^T \\ (u_6 - a_6, v_6)^T \\ (u_7 - a_7, v_7)^T \\ (u_8 - a_8, v_8)^T \end{pmatrix} = \begin{pmatrix} (\phi_{w_R} - a_1, v_1)^T \\ (\theta_{w_R} - a_2, v_2)^T \\ (\psi_{w_R} - a_3, v_3)^T \\ (\phi_{w2_R} - a_4, v_4)^T \\ (\phi_{w_L} - a_5, v_5)^T \\ (\theta_{w_L} - a_6, v_6)^T \\ (\psi_{w_L} - a_7, v_7)^T \\ (\phi_{w2_L} - a_8, v_8)^T \end{pmatrix} \tag{2.5}$$

Note that $\mathbf{x}_i$ here represents the shifted Hopf oscillator vector such that $\mathbf{x}_i = (u_i - a_i, v_i)^T$ as seen in Eq. (2.1), where $a_i(t)$ is the center of oscillation. For example, if we need a 10-degree offset for the main flapping stroke angle ϕ_w, then we can set $a_1 = a_5 = 10$ deg. so that the flapping stroke angle oscillate around 10 degrees.

For stability analysis, we need to construct fully coupled dynamics of the augmented state vector $\{\mathbf{x}\}$.

$$\{\dot{\mathbf{x}}\} = [\mathbf{f}(\{\mathbf{x}\}; \rho)] - k\mathbf{G}\{\mathbf{x}\} \tag{2.6}$$

where $[\mathbf{f}(\{\mathbf{x}\}; \rho)] = [\mathbf{f}(\mathbf{x}_1; \rho_1); \mathbf{f}(\mathbf{x}_2; \rho_2); \cdots; \mathbf{f}(\mathbf{x}_n; \rho_n)]$. The $2n \times 2n$ matrix $\mathbf{G}$ is a Laplacian matrix with phase shifts $\mathbf{R}(\Delta_{ij})$ constructed from Eq. (2.4).

The coupling topology and phase shift between each oscillators are reflected in the $\mathbf{G}$ matrix. Such phase shifts along with the bifurcation parameter σ can be used to define different flight modes, similar to walking gaits. Numerous configurations are possible as long as they are on balanced graphs [62] and we can choose either a bidirectional or a uni-directional coupling between the oscillators. Some configurations considered in this paper are shown in Fig. 2.3. The numbers next to the arrows indicate the phase shift Δ_{ij}, hence $\Delta_{ij} > 0$ indicates how much phase the i-th member leads. Since the graphs in Figure 2.3 are on balanced graphs, the number of input ports equal the number of output ports. Further, all the phase shifts (Δ_{ij}) along one cycle should add up to a modulo of 2π. Figure 2.3b shows the nominal values of the phase shift from the symmetric wing configuration such that $\Delta_{21} = \Delta_{65} = 90$ deg. and $\Delta_{31} = \Delta_{75} = -90$ deg. The empirical data suggest that the pitching angle (θ_w) has approximately a 90-degree phase lag with the flapping angle (ϕ_w), which agrees with the aerodynamically optimal value [3,12]. For hovering flight, Dickison [12], using his Robofly testbed and numerical simulations, found that increasing the phase difference value Δ_{21} to 90 deg $+\delta$ further contributed to enhancing the lift generation, which is explained by the wake capture and rotational circulation lift mechanism. Hence, the ability to control Δ_{21} allows us to investigate the optimal value of the phase difference. In addition, the nominal value of $\Delta_{31} = -90$ deg, the phase difference between the flapping stroke angle and lead-lag angle will results an elliptical orbit of the wing. On the other hand, by having two difference phase differences for the left and right wings, we can investigate how symmetric-breaking wing rotations contribute the agile turning of flapping flight. Furthermore, by having an independent control of the phase difference Δ_{31} and Δ_{75}, we can investigate another symmetry-breaking impact of the differential delay in the lead-lag motion. Such differential phases can be used to stabilize the flapping flying dynamics.

The $\mathbf{G}$ matrix in Eq. (2.6) for Fig. 2.3a can be found as

$$\begin{bmatrix} 2\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \frac{\rho_1}{\rho_4}\mathbf{R}(\Delta_{31}) & -\frac{\rho_1}{\rho_5}\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\frac{\rho_2}{\rho_1}\mathbf{R}(\Delta_{21}) & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\frac{\rho_3}{\rho_2}\mathbf{R}(\Delta_{31} - \Delta_{21}) & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\frac{\rho_4}{\rho_3}\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\frac{\rho_5}{\rho_1}\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \frac{\rho_5}{\rho_8}\mathbf{R}(\Delta_{75}) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{\rho_6}{\rho_5}\mathbf{R}(\Delta_{65}) & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{\rho_7}{\rho_6}\mathbf{R}(\Delta_{75} - \Delta_{65}) & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{\rho_8}{\rho_7}\mathbf{I}_2 & \mathbf{I}_2 \end{bmatrix}, \qquad (2.7)$$

where often the radii (the amplitude of the oscillation from the bias a_i) are symmetric such that $\rho_1 = \rho_2$, $\rho_2 = \rho_6$, $\rho_4 = \rho_7$, and $\rho_5 = \rho_8$, although the difference of the maximum amplitude of each oscillation can be used to generate side forces or turning (rolling or yawing) moments.

The proof of phase synchronization boils down to finding the condition on k by which the flow-invariant synchronized state [56], constructed from $\mathbf{G}\{\mathbf{x}\} = \mathbf{0}$, is globally stable. In fact, by using contraction theory [56, 63], we can prove global exponential synchronization of the coupled Hopf oscillators. We first introduce the main theorem of contraction theory

Theorem 1. *For the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$, if there exists a uniformly positive definite metric, $\mathbf{M}(\mathbf{x}, t) = \boldsymbol{\Theta}(\mathbf{x}, t)^T\boldsymbol{\Theta}(\mathbf{x}, t)$, where $\boldsymbol{\Theta}$ is some smooth coordinate transformation of the virtual displacement, $\delta\mathbf{z} = \boldsymbol{\Theta}\delta\mathbf{x}$, such that the associated generalized Jacobian, F is uniformly negative definite, i.e., $\exists\ell > 0$ such that*

$$\mathbf{F} = \left(\dot{\boldsymbol{\Theta}}(\mathbf{x}, t) + \boldsymbol{\Theta}(\mathbf{x}, t)\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\boldsymbol{\Theta}(\mathbf{x}, t)^{-1} \leq -\ell\mathbf{I}, \tag{2.8}$$

then all system trajectories converge globally to a single trajectory exponentially fast regardless of the initial conditions, with a global exponential convergence rate of the largest eigenvalues of the symmetric part of $\mathbf{F}$.

Such a system is said to be contracting.

Proof. The proof is given in [63] by computing $\frac{d}{dt}\delta\mathbf{z}^T\delta\mathbf{z} = 2\delta\mathbf{z}^T\mathbf{F}\delta\mathbf{z}$. □

The synchronized flow-invariant subspace for the configuration in Fig 2.3a is defined by $\mathbf{G}\{\mathbf{x}\} = \mathbf{0}$ such that

$$\begin{aligned}\mathcal{M}(\{\mathbf{x}\}) \Longleftrightarrow & \mathbf{x}_1 = \frac{\rho_1}{\rho_2}\mathbf{R}(\Delta_{12})\mathbf{x}_2 = \frac{\rho_1}{\rho_3}\mathbf{R}(\Delta_{13})\mathbf{x}_3 = \frac{\rho_1}{\rho_4}\mathbf{R}(\Delta_{13})\mathbf{x}_4 \\ & = \frac{\rho_1}{\rho_5}\mathbf{x}_5 = \frac{\rho_1}{\rho_6}\mathbf{R}(\Delta_{56})\mathbf{x}_6 = \frac{\rho_1}{\rho_7}\mathbf{R}(\Delta_{57})\mathbf{x}_7 = \frac{\rho_1}{\rho_8}\mathbf{R}(\Delta_{57})\mathbf{x}_8 \end{aligned} \tag{2.9}$$

where we used $\Delta_{ij} = -\Delta_{ji}$.

The flow invariant subspace $\mathcal{M}$ in Eq. (2.9) can be re-written with respect to the first state vector $\mathbf{x}_1 = \mathbf{z}_1$ such that

$$\mathcal{M}(\{\mathbf{x}\}) \Longleftrightarrow \mathbf{z}_1 = \mathbf{z}_2 = \cdots = \mathbf{z}_n, \quad \{\mathbf{z}\} = \mathbf{T}(\Delta_{ij}, \rho_i)\{\mathbf{x}\} \tag{2.10}$$

where $\{\mathbf{z}\} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n)^T$ and $\mathbf{z}_1 = \mathbf{x}_1$, $\mathbf{z}_2 = \frac{\rho_1}{\rho_2}\mathbf{R}(\Delta_{12})\mathbf{x}_2$, $\mathbf{z}_3 = \frac{\rho_1}{\rho_3}\mathbf{R}(\Delta_{13})\mathbf{x}_3$ and so on. For example, the $\mathbf{T}$ matrix for the configuration in Fig. 2.3a is given as

$$\begin{aligned}&\mathbf{T}(\Delta_{ij}, \rho_i) = \\ &\mathrm{diag}\left(\mathbf{I}_2, \frac{\rho_1}{\rho_2}\mathbf{R}(\Delta_{12}), \frac{\rho_1}{\rho_3}\mathbf{R}(\Delta_{13}), \frac{\rho_1}{\rho_4}\mathbf{R}(\Delta_{13}), \frac{\rho_1}{\rho_5}\mathbf{I}_2, \frac{\rho_1}{\rho_6}\mathbf{R}(\Delta_{56}), \frac{\rho_1}{\rho_7}\mathbf{R}(\Delta_{57}), \frac{\rho_1}{\rho_8}\mathbf{R}(\Delta_{57})\right)\end{aligned} \tag{2.11}$$

Then, we present the main theorem of this section.

Theorem 2. *If the following condition is met, any initial condition* $\{\mathbf{x}\}$ *of the coupled Hopf oscillators in Eq. (2.4) and Eq. (2.6) on a balanced graph converges to the flow-invariant synchronized state* $\mathcal{M}$ *exponentially fast.*

$$k\lambda_{min}\left(\mathbf{V}^T(\mathbf{L}+\mathbf{L}^T)\mathbf{V}/2\right) > \lambda \tag{2.12}$$

where λ *is the convergence rate of the Hopf oscillator in Eq. (2.1),* $\lambda_{min}\left(\mathbf{V}^T(\mathbf{L}+\mathbf{L}^T)\mathbf{V}/2\right)$ *denotes the minimum eigenvalue, and* $\mathbf{L}$ *is the Laplacian matrix constructed from the* *balanced graph such that* $\mathbf{G} = \mathbf{T}^{-1}\mathbf{L}\mathbf{T}$ *with* $\mathbf{T}$ *defined from Eq. (2.10). In addition, the real orthonormal* $2n \times 2(n-1)$ *matrix* $\mathbf{V}$ *is constructed from the orthonormal eigenvectors of* $(\mathbf{L} + \mathbf{L}^T)/2$ *other than the ones vector* $\mathbf{1} = (\mathbf{I}_2; \mathbf{I}_2; \cdots; \mathbf{I}_2)$ *such that* $\mathbf{V}\mathbf{V}^T + \mathbf{1}\mathbf{1}^T/n = \mathbf{I}_{2n}$*.*

Proof. The proof can be obtained based on [56]. The proof here is simpler than [46] in the sense that we derive the Laplacian matrix and orthonormal flow-invariant matrix that are independent of the rotational angles. Consider the orthonormal space $\mathbf{V}$, constructed from the orthornomal eigenvectors of the symmetric part of $\mathbf{L}$ (see [62]). Then, the global exponential convergence to the flow-invariant synchronized state $\mathcal{M}$ is equivalent to

$$\mathbf{V}^T\{\mathbf{z}\} \rightarrow 0, \quad \text{globally and exponentially} \tag{2.13}$$

By pre-multiplying Eq. (2.6) by $\mathbf{T}^{-1}$ and using $\mathbf{T}\{\mathbf{x}\} = \{\mathbf{z}\}$ and $\mathbf{G} = \mathbf{T}^{-1}\mathbf{L}\mathbf{T}$, we can obtain

$$\{\dot{\mathbf{z}}\} = \mathbf{T}\left[\mathbf{f}(\{\mathbf{x}\}; \rho)\right] - k\mathbf{L}\{\mathbf{z}\} \tag{2.14}$$

where the CPG network in the example in Fig. 2.3a is on a balanced graph such that

$$\mathbf{L} = \begin{bmatrix} 2\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & -\mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & -\mathbf{I}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_2 & \mathbf{I}_2 \end{bmatrix} \tag{2.15}$$

In other words, we transformed the $\mathbf{G}$ matrix to the conventional graph Laplacian matrix $\mathbf{L}$.

Since $\mathbf{T}\left[\mathbf{f}(\{\mathbf{x}\};\rho)\right] = \mathbf{T}\left[\mathbf{f}(\mathbf{T}^{-1}\{\mathbf{z}\};\rho)\right]$, we can find

$$\begin{aligned}\mathbf{T}\left[\mathbf{f}(\{\mathbf{x}\};\rho)\right] &= \left[\frac{\rho_1}{\rho_i}\mathbf{R}(-\Delta_{1j})\mathbf{f}(\mathbf{x}_i;\rho_i)\right] = \left[\frac{\rho_1}{\rho_i}\mathbf{R}(-\Delta_{1j})\mathbf{f}(\frac{\rho_i}{\rho_1}\mathbf{R}(\Delta_{1j})\mathbf{z}_i;\rho_i)\right] \\ &= \left[\mathbf{f}(\mathbf{z}_i;\rho_1)\right] = \left[\mathbf{f}(\mathbf{z}_1;\rho_1);\mathbf{f}(\mathbf{z}_2;\rho_1);\cdots;\mathbf{f}(\mathbf{z}_n;\rho_1)\right]\end{aligned} \tag{2.16}$$

where we used $\mathbf{f}(\mathbf{R}(\Delta)\mathbf{x}) = \mathbf{R}(\Delta)\mathbf{f}(\mathbf{x})$ and $\mathbf{f}(g\mathbf{x};\rho) = g\mathbf{f}(\mathbf{x};\rho/g)$ from Eq. (2.2) and Eq. (2.3). The radius of the final augmented Hopf oscillators in Eq. (2.16) is identical to ρ_1.

By premultiplying $\mathbf{V}^T$ and substituting $\{\mathbf{z}\} = \mathbf{V}\mathbf{V}^T\{\mathbf{z}\} + \mathbf{1}\mathbf{1}^T\{\mathbf{z}\}$ result in

$$\mathbf{V}^T\{\dot{\mathbf{z}}\} = \mathbf{V}^T\left[\mathbf{f}(\mathbf{V}\mathbf{V}^T\{\mathbf{z}\} + \mathbf{1}\mathbf{1}^T/n\{\mathbf{z}\};\rho_1)\right] - k\mathbf{V}^T\mathbf{L}\mathbf{V}\mathbf{V}^T\{\mathbf{z}\} \tag{2.17}$$

where we used $\mathbf{L}\mathbf{1}\mathbf{1}^T = 0$.

We can construct the following virtual dynamics of $\mathbf{y}$ from the preceding equation

$$\dot{\mathbf{y}} = \mathbf{V}^T\left[\mathbf{f}(\mathbf{V}\mathbf{y} + \mathbf{1}\mathbf{1}^T/n\{\mathbf{z}\};\rho_1)\right] - k\mathbf{V}^T\mathbf{L}\mathbf{V}\mathbf{y} \tag{2.18}$$

which has $\mathbf{y} = \mathbf{V}^T\{\mathbf{z}\}$ and $\mathbf{y} = \mathbf{0}$ has two particular solutions.

The virtual system Eq. (2.18) is contracting (globally and exponentially stable) for $\mathbf{V}^T\left[\mathbf{f}\right]\mathbf{V} - k\mathbf{V}^T(\mathbf{L}+\mathbf{L}^T)\mathbf{V}/2 < 0$ by Theorem 1. This condition is equivalent to $k\lambda_{min}\left(\mathbf{V}^T(\mathbf{L}+\mathbf{L}^T)\mathbf{V}/2\right) > \lambda$, since the maximum eigenvalue of $\lambda_{max}(\mathbf{V}^T\left[\mathbf{f}\right]\mathbf{V}) \leq \lambda$. For the example in Fig. 2.3a, this condition corresponds to $k > \lambda/0.198$.

The same proof works for an arbitrary CPG network on balanced graph that has $\mathbf{V}^T(\mathbf{L}+\mathbf{L}^T)\mathbf{V}/2 > 0$. For undirected graphs (all the connections are bi-directional),

L automatically becomes a balanced symmetric matrix. □

In conclusion, Theorem 2 can be used to find the proper coupling strength k to exponentially and almost globally stabilize the coupled Hopf oscillators given in Eq. (2.4). Sometimes, the condition for k in Theorem 2 might be too conservative especially if the desired λ is large.

2.4 Boundedness of Hopf-Kuromoto Oscillator for any k

In this section, we demonstrate that the oscillators always approach or enter a closed n-disk, for any k. Since ρ_i are selectable, this disk is not the standard n-disk. Set

$$D_i = \{(x_i, y_i) \in \mathbb{R}^2 | x_i^2 + y_i^2 \leq \rho_i\}. \tag{2.19}$$

Then, the closed n-disk in question is

$$D = \prod_{i=1}^{n} D_i. \tag{2.20}$$

Consider arbitrary initial conditions. If any oscillators are outside their disk, they have $\frac{r_i}{\rho_i} > 1$. Consider the oscillator with maximum $\frac{r_i}{\rho_i}$. Its r dynamics are

$$\dot{r}_i = -\lambda r_i (r_i^2/{\rho_i}^2 - 1) - k\rho_i \sum_{j \in \mathcal{N}_i} \left(\frac{r_i}{\rho_i} - \frac{r_j}{\rho_j} \cos(\theta_i - \theta_j - \phi_{ij}) \right). \tag{2.21}$$

The first term (the self-gain) is negative if $\frac{r_i}{\rho_i} > 1$. The second term is negative or zero if the oscillator has maximum $\frac{r_i}{\rho_i}$. Therefore, the oscillator with maximum $\frac{r_i}{\rho_i}$ has $\dot{r} < 0$ at least until $\frac{r_i}{\rho_i} = 1$, i.e., the maximum oscillator approaches or enters its disk. Therefore, the oscillator network approaches or enters the n-disk. Since initial conditions are arbitrary, this $(\frac{r_i}{\rho_i})_{\max}$ will never increase.

2.5 Death of Oscillation

The Hopf oscillator is a 2D generalization of the Kuromoto oscillator. Synchronization attempts in Kuromoto oscillators are thwarted by "Splay states" [64]. While the 2D generalization allows smooth amplitude modulation, it still succumbs to Splay states. We will highlight a generalized version of Splay states, hereafter referred to only as death of oscillation.

In Section 2.3, we proved that trajectories of the coupled system converge to the flow-invariant manifold defined by $\mathbf{G}\{\mathbf{x}\} = \mathbf{0}$. The key insight is to recognize that when this condition has been satisfied, the oscillators are uncoupled. Thus, the final invariant set is an intersection of the set defined by $\mathbf{G}\{\mathbf{x}\} = \mathbf{0}$ and the ω-limit set of the uncoupled systems. There are *two* components to this intersection: the synchronized limit cycle *and* the origin. Unfortunately, while the origin is completely unstable in the uncoupled system, it is not necessarily completely unstable in the coupled system. In fact, in a restricted case, we can prove that if the contraction condition is met, the origin has a stable manifold.

Proposition 3. *Consider two identical n-dimensional oscillators, each with at least one completely unstable equilibria. Couple them in the form*

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{f}(\mathbf{x}_1, t) + \mathbf{u}(\mathbf{x}_2) - \mathbf{u}(\mathbf{x}_1) \\ \dot{\mathbf{x}}_2 = \mathbf{f}(\mathbf{x}_2, t) + \mathbf{u}(\mathbf{x}_1) - \mathbf{u}(\mathbf{x}_2). \end{cases} \tag{2.22}$$

Following [65], assume $\mathbf{h} = \mathbf{f} - 2\mathbf{u}$ is contracting. Then, even though $\mathbf{x}_1$ and $\mathbf{x}_2$

exponentially tend to one another, each equilibria has a stable manifold of some dimension.

Proof. Consider WLOG a single equilibria located at the origin. Denote $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big|_0 = A$ and $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\big|_0 = B$. Consider the augmented system $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$. Denote the symmetric part of

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\bigg|_0 = \begin{bmatrix} A - B & B \\ B & A - B \end{bmatrix} \tag{2.23}$$

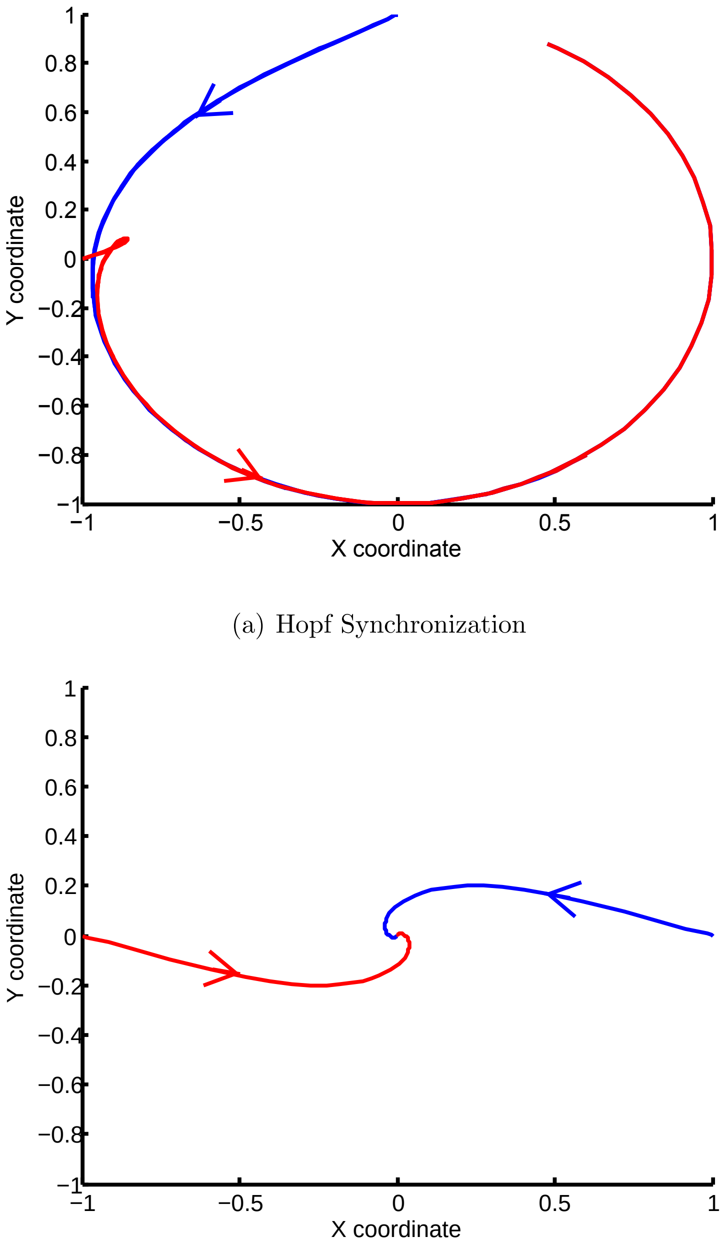
by

$$J = \begin{bmatrix} A_s - 2B_s & 0 \\ 0 & A_s - 2B_s \end{bmatrix} + \begin{bmatrix} B_s & B_s \\ B_s & B_s \end{bmatrix} = H + P. \tag{2.24}$$

Notice that P has rank at most n, and thus has at least n zero eigenvalues. Additionally, all the eigenvalues of H are negative by the assumption of $\mathbf{h}$ contracting. According the Weyl's theorem, at least n eigenvalues of J must be less than zero. Therefore, the linearized augmented system at the origin cannot be positive definite. □

While this is easily generalized to systems with all-to-all coupling, we are unsure whether this impossibility theorem can be fully-generalized to other coupling schemes such as the diffusive coupling used in the majority of this work. Nevertheless, death of oscillation remains an important challenge to providing a complete solution to limit cycle driven plants. Figure 2.4 shows an example of the Hopf-Kuromoto oscillator. We see the desired CPG behavior as well as the possibility of death of oscillation. Figure 2.5 shows an example using the Van der Pol oscillator exhibiting the same type of possibilities.

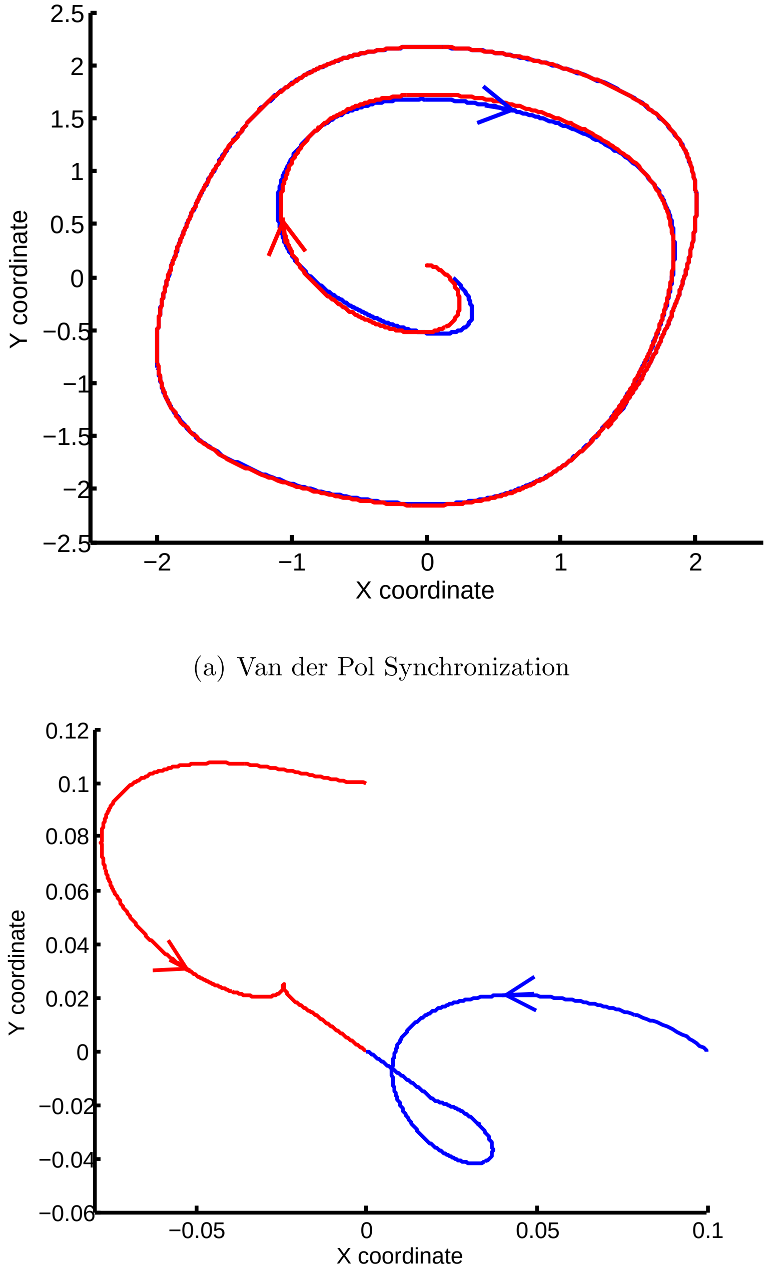
Taken together, we have shown that any k will result in bounded solutions and that suitably-large k will always track the desired, oscillatory trajectory or collapse



(a) Hopf Synchronization

(b) Hopf Death of Oscillation

Figure 2.4: Figure (a) shows the desired behavior of a network of two Hopf-Kuromoto oscillators and Figure (b) shows the death of oscillation behavior for some initial conditions.



(a) Van der Pol Synchronization

(b) Van der Pol Death of Oscillation

Figure 2.5: Figure (a) shows the desired behavior of a network of two Van der Pol oscillators and Figure (b) shows the death of oscillation behavior for some initial conditions.

to the origin. In physical systems, noise or disturbances can play a role in moving the trajectory off the (relatively) low-dimensional stable manifold going to the origin and onto the manifold converging to desired oscillatory trajectory.

2.6 Nonlinear Synchronization Manifold

While we can't rule out death of oscillation, we can generalize our theory to oscillators with a nonlinear synchronization manifold. Unlike the Hopf oscillator we've been focusing on, many important oscillators do not have a linear synchronization manifold. Examples include the Van der Pol oscillator, EL Repressilator [66], or Hopf-Kuromoto oscillators with time- or space-varying phase differences. Computing the nonlinear synchronization manifold for particular oscillators could be a project on its own. If we do an analytical description of the nonlinear synchronization manifold, we can proceed directly.

Theorem 4. *Consider, in $\mathbb{R}^n$, the deterministic system*

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, t\right). \tag{2.25}$$

Assume that there exists a flow-invariant manifold $\mathcal{M}$ (i.e. an embedded submanifold $\mathcal{M} \subset \mathbb{R}^n$ such that $\forall t : \mathbf{f}\left(\mathcal{M}, t\right) \subset T\mathcal{M}$), which implies that any trajectory starting in $\mathcal{M}$ remains in $\mathcal{M}$. If this flow-invariant manifold can be described as a level set of a map of manifolds (i.e. $\mathbf{x} \in \mathcal{M} \Leftrightarrow \mathbf{g}\left(\mathbf{x}\right) = 0$) and if the auxiliary system

$$\dot{\mathbf{y}} = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right) \mathbf{f}\left(\mathbf{h}\left(\mathbf{y}, \mathbf{x}\right)\right) \tag{2.26}$$

is contracting with

$$\mathbf{h}\left(\mathbf{g}\left(\mathbf{x}\right),\mathbf{x}\right) = id_x \tag{2.27}$$

and

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)\mathbf{f}\left(\mathbf{h}\left(0,\mathbf{x}\right)\right) = 0, \tag{2.28}$$

then all trajectories of system 2.25 will exponentially converge to $\mathcal{M}$.

Proof. Both $\mathbf{y} = \mathbf{g}\left(\mathbf{x}\right)$ and $\mathbf{y} = \mathbf{0}$ are particular solutions to system 2.26. If system 2.26 is contracting with respect to $\mathbf{y}$, then all its solutions converge exponentially to a single trajectory, which implies in particular that $\mathbf{g}\left(\mathbf{x}\right)$ converges exponentially to $\mathbf{0}$. □

2.7 Fast Inhibition of Oscillation by Hopf Bifurcation

As stated earlier, we can rapidly inhibit the oscillatory motion of the coupled Hopf oscillators in Eq. (2.4) by exploiting the bifurcation property of the Hopf oscillator model. In other words, changing the $\sigma = 1$ in Eq. (2.1) to $\sigma = -1$ would rapidly convert the limit cycle dynamics to exponentially stable dynamics converging to the origin such that $u \to a$ and $v \to 0$. This single bifurcation parameter (σ) can be used to switch the flapping flight mode to the gliding or soaring mode without dramatically changing the CPG oscillator network. We analyze such switched systems from a high-level perspective in Section 3. Simulation results that alternate between two different flight modes are presented in Section4.2 and experimental results are presented in Sections 4.4.2 and 4.4.3.

Theorem 5. *For any positive gain $k > 0$, any initial condition $\{\mathbf{x}\}$ of the coupled*

Hopf model with $\sigma = -1$ given in Eq. (2.4) converges to the origin ($\{\mathbf{x}\} \rightarrow \mathbf{0}$) such that $u_i \rightarrow a_i$ and $v_i \rightarrow 0$ for all $i = 1, \cdots, n$. The oscillation frequency ω need not change to zero.

Proof. It is straightforward to show that $\sigma = -1$ will make the uncoupled Hopf oscillator in Eq. (2.1) exponentially stable dynamics for any (u, v) except the shifted origin $(a, 0)$ since the symmetric part of the Jacobian $\mathbf{F}$ in Eq. (2.8) is now strictly negative definite regardless of any ω. Thus, any positive k will lead to exponentially synchronizing dynamics that tend exponentially to the origin and this can be shown similar to the proof of Theorem 2. □

We can also turn the limit cycle dynamics to the dynamics with a stable equilibrium by changing the coupling gains, as described as fast inhibition in [62]. However, the method using bifurcation is superior in the sense that we can keep the original coupling gains and Laplacian matrices for alternating flight modes. It should be noted that changing ω to zero would also result in no reciprocal flapping motion, however the converged steady-state value depends on the initial conditions, whereas $\sigma = -1$ would lead to convergence to the same value $(a, 0)$.

2.8 Perspectives on Sensory Feedback

An important area of research is incorporating sensory feedback to modulate oscillators [46,59,67]. Currently, most approaches only consider smooth, small disturbances from the environment in the form of phase lag and amplitude reduction. Biological sensory feedback pathways are far more complicated. Rossignol et al. [68] expressed the complication by saying, "The more we dig into the details of these sensorimotor interactions, the more it seems improbable that they should work so smoothly, but they do." This type of flexibility is difficult to account for with abstracted neuronal oscillators. For a simple example, the mammalian trip reflex during walking is large, rapid, and highly state-dependent. One approach is to ignore the fast, yet continuous nature of neuronal feedback pathways and instead create a set of new states in a state machine that are activated by a discrete trip sensor [69]. Rather than focusing on piecemeal solutions for particular applications, we will turn our attention to more biologically-accurate neuron models in Chapter 5, which will come with built-in flexibility.

2.9 Chapter Summary

We investigated the hypothesis that the phase control and synchronization of coupled nonlinear oscillators, inspired by central pattern generators (CPGs) found in animal spinal cords, can effectively produce and control stable flapping flight patterns and can be used to stabilize the flapping flying vehicle dynamics. An engineered CPG network, which ensures the stability and robust adaptation of motion, can significantly reduce the complexity associated with engineered flapping flight.

We made general remarks for methods which analyze phase synchronization of coupled oscillators. Not only did we provide a synchronization proof for Hopf oscillator networks, but we showed that through the generation of a stable manifold, incremental stability implied by contraction theory is compatible with the possibility of death of oscillation. In addition, we provided a non-contraction method to ensure boundedness in all cases as well as a contraction method for oscillating systems without a linear synchronization manifold.

Central to the agile flight of natural flyers is the ability to execute complex synchronized three-dimensional motions of the wings. In this chapter, we introduced a mathematical and control-theoretic framework of CPG control theory that enables such synchronized wing maneuvers. Because of the oscillatory nature of flapping flight, it is important to have a control law which allows for smooth changes in flapping frequency and other oscillation parameters. We showed that the central controller, similar to the brain of an animal, can stabilize the vehicle dynamics by commanding a reduced number of control variables such as the frequency and phase difference of the oscillators instead of directly controlling multiple joints. Such a CPG-based method allows for stable and rapid changes in flapping parameters such as wing pitch and the lead-lag angle. We will validate these results using simulation and experimentation in Chapter 4.

Chapter 3

Switched Systems with Multiple Invariant Sets

Bifurcations have been of interest to dynamical systems theory for decades. However, most control strategies view such behavior as damaging and try to mitigate it [70]. Relatively less work actively inserts bifurcations as part of a control strategy as we proposed in Chapter 2. Another possible application is walking robots. Fig. 3.1 shows a hypothetical switching pattern for a walking robot application utilizing central pattern generation. A guidance/navigation engineer may design limit cycle subsystems for walking and jumping modes (shown as a Hopf oscillator and a Van der Pol oscillator), while utilizing steady-state control strategies for static balancing or tasks requiring fine motor control.

Mode-switching also implicates a large body of literature on switched systems [29]. Many works on stability of switched systems assumes that all subsystems have a common equilibrium point. [71–73] consider weak Lyapunov functions in the style of LaSalle for a common equilibrium. [74] considers equilibrium location changes, but holds the vector field constant. They connect the result to averaging theory. [75] con-

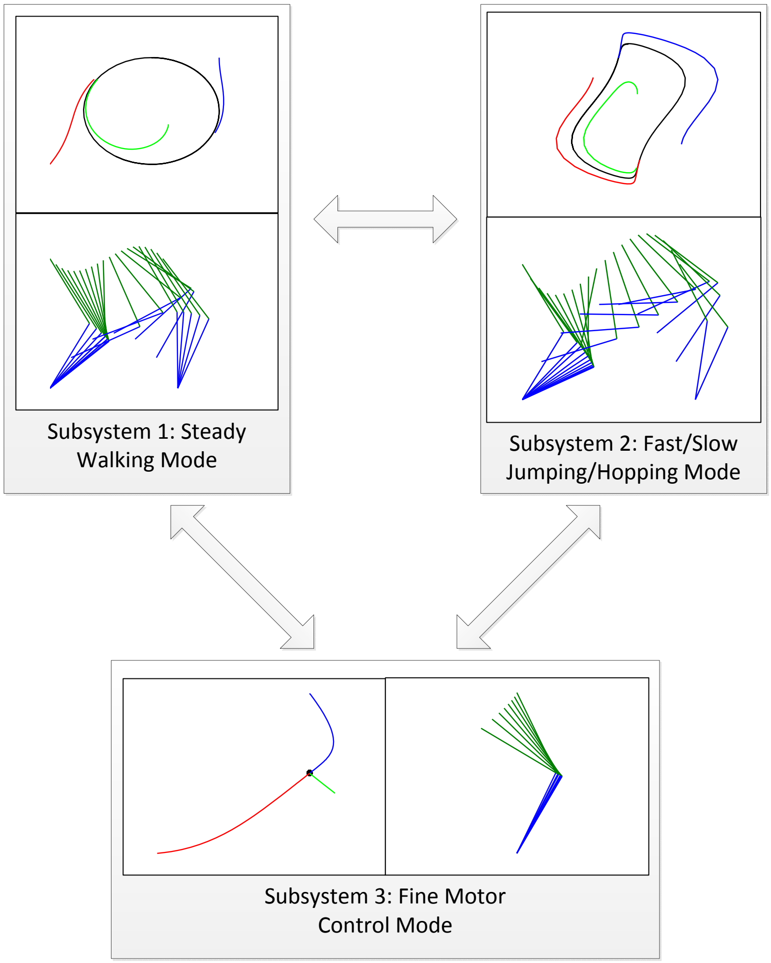


Figure 3.1: Schematic of mode switching with non-equilibrium limit sets.

siders practical stability of affine systems with multiple distinct equilibria. Alpcan and Başar investigated dwell time criteria for nonlinear globally exponentially stable subsystems which could have differing equilibria [30]. Such systems have no single globally attractive equilibrium point. The authors of [30] reported an explicit construction of the dwell time and a conservative invariant set. This chapter is inspired by that work and is a generalization of it. We generalize their result to switched systems where each subsystem may have multiple invariant sets. We pursue a similar dwell time strategy in order to provide spatial bounds for the switched system. The resulting construction is slightly more complicated, as we consider $\dot{V}$ in order to isolate the invariant sets rather than using the Lyapunov function alone.

Systems with bifurcation often contain multiple ω-limit sets which cannot be globally exponentially stable. Instead, results such as LaSalle's invariant set theorem [76] allow us to analyze asymptotic stability of this larger class of systems. LaSalle's theorem and much of the switched systems literature are both Lyapunov-based, and we will make use of Lyapunov functions to define all the relevant sets. The benefit to relying on Lyapunov functions is that this requires no special structure on the subsystems' entire vector fields. The tradeoff is that we fail to exploit any special structure the subsystems may possess and the result relies on being able to find suitable Lyapunov functions.

Section 3.1 provides background assumptions and definitions. Section 3.2 begins by reconsidering existing results. Sections 3.2.3 through 3.2.5 present two methods to accomplish the goal. Choice of a particular method will depend on specific situations and design constraints. Section 3.3 shows a numerical example.

3.1 Preliminaries and Definitions

Consider a set of continuous-time dynamical systems defined by

$$\dot{\mathbf{x}} = \mathbf{f}_p\left(\mathbf{x}\right), \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $p \in \mathcal{P}$, with some index set $\mathcal{P} = \{p_1, p_2, ..., p_{\max}\}$. A piecewise constant switching signal $\sigma : [0, \infty) \to \mathcal{P}$ specifies the active subsystem at each time. Assume, for ease of analysis, that $\mathbf{f}_p$ are each continuous with continuous first partials. Together, (3.1), the index set, and the switching signal define a switched system.

Only some systems admit stability results for arbitrary switching signals, so we will consider a constraint on how quickly the switching signal can make consecutive switches.

Definition Consider a switched system with switching times $\{t_1, t_2, ...\}$. It is said to have *dwell time* τ if $t_{i+1} - t_i \geq \tau \ \ \forall i \in \mathbb{N}$.

Next, we review and introduce some important subsets of $\mathbb{R}^n$. We have not yet provided strict assumptions on Lyapunov-like functions. At present, it is enough to assume that each subsystem has a (possibly different) $\mathcal{C}^1$ Lyapunov-like function, which is bounded above and below on every bounded subset of $\mathbb{R}^n$. Furthermore, assume that each is radially unbounded ($V_p(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$). This ensures that every sublevel set describes a compact region. We assume for the remainder of this paper that the minimum value of each V_p is zero. Define

$$\mathcal{G}_p = \{\mathbf{x} \in \mathbb{R}^n | V_p(\mathbf{x}) = 0\} \tag{3.2}$$

as the set which attains the minimum value of V_p. Let κ be a positive constant and define

$$\mathcal{N}_p\left(\kappa\right) = \left\{\mathbf{x} \in \mathbb{R}^n | V_p\left(\mathbf{x}\right) \leq \kappa\right\}, \tag{3.3}$$

a closed κ-neighborhood of $\mathcal{G}_p$. For the purposes of Theorem 6, $\mathcal{N}_p\left(\kappa\right)$ is connected, but it is not necessarily connected in the remainder of the paper (See Figure 3.4). Let the union over $\mathcal{P}$ be

$$\mathcal{N}(\kappa) = \bigcup_{p \in \mathcal{P}} \mathcal{N}_p(\kappa). \tag{3.4}$$

Additionally, we define a superset, $\mathcal{M}(\kappa)$, in a series of steps with

$$\alpha_p\left(\kappa\right) = \max_{\mathbf{x} \in \mathcal{N}(\kappa)} V_p\left(\mathbf{x}\right), \tag{3.5}$$

and

$$\mathcal{M}_p\left(\kappa\right) = \left\{\mathbf{x} \in \mathbb{R}^n : V_p\left(\mathbf{x}\right) \leq \alpha_p\left(\kappa\right)\right\}. \tag{3.6}$$

Finally, we create a closed union of closed sublevel sets,

$$\mathcal{M}\left(\kappa\right) = \bigcup_{p \in \mathcal{P}} \mathcal{M}_p. \tag{3.7}$$

Notice that the dependence on κ carries through once we use it in $\mathcal{N}(\kappa)$. For the purposes of Theorem 6, $\mathcal{M}$ is a connected superset of $\mathcal{N}$. Theorem 8 will introduce a different notion which is not necessarily connected. Figure 3.2 provides a one-dimensional example to help visualize these sets.

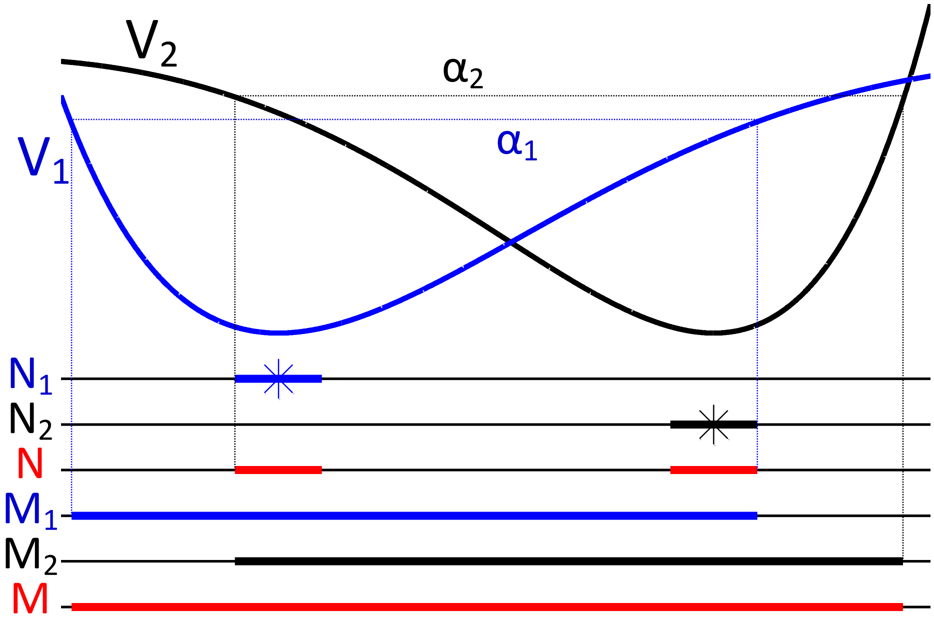


Figure 3.2: Qualitative example of how $\mathcal{N}$ and $\mathcal{M}$ are built for a switched system consisting of two subsystems, each with a single equilibrium, but at different locations.

3.2 Stability Results

We first restructure the result in [30] slightly to add clarity and to better facilitate the generalization presented here.

3.2.1 Unique Equilibrium Case

Theorem 6 ([30])**.** *Consider a family of systems defined by (3.1), each with a single, globally exponentially stable equilibrium, denoted* $\mathbf{x}_p^*$*. Suppose that the exponential decay rate of each Lyapunov function, as described in*

$$\dot{V}_p(\mathbf{x}) \leq -\varepsilon V_p(\mathbf{x}),\, \forall \mathbf{x} \in \mathbb{R}^n, \forall p \in \mathcal{P}, \tag{3.8}$$

is at least $\varepsilon > 0$.

Furthermore, given a positive constant κ, define the sets as in Section 3.1 and assume $\mu(\kappa) \in (1, \infty)$ such that

$$\frac{V_r(\mathbf{x})}{V_q(\mathbf{x})} \leq \mu(\kappa), \forall q, r \in \mathcal{P} \ \ \forall \mathbf{x} \in \mathbb{R}^n \setminus \mathcal{N}(\kappa). \tag{3.9}$$

Then, for every switching signal with dwell time

$$\tau > \frac{\log \mu(\kappa)}{\varepsilon}, \tag{3.10}$$

(i) There exists a time T such that $\mathbf{x}(T^-) \in \mathcal{N}_{\sigma(T^-)}(\kappa)$, and

(ii) For any time $\bar{t}$ such that $\mathbf{x}(\bar{t}) \in \mathcal{N}_{\sigma(\bar{t})}(\kappa)$, $\mathbf{x}(t) \in \mathcal{M}(\kappa)$ for all $t \geq \bar{t}$.

Remark 1. *Not all choices of V_p will give a finite value of $\mu(\kappa)$. Typical polynomial constructions for Lyapunov functions must have the same polynomial order. Scaling or stretching Lyapunov functions may be useful, but there will be implications for both the spatial parameter μ and the temporal parameter ε. We will will revisit the idea of scaling briefly in Sections 3.2.3-3.2.4.*

Proof. We will only provide a sketch of the relevant features. The proof proceeds in two parts:

(i) Consider a finite time interval $[t_0, T]$ with corresponding switching times $t_1 < t_2 < \cdots < t_{n_\sigma}$, where n_σ is the number of switches inside the interval. Between switches, $\sigma(t)$ is constant. If the trajectory enters $\mathcal{N}_{\sigma(t)}$ (not just $\mathcal{N}$), the result is trivial. Otherwise, the behavior of $V_{\sigma(t)}(\mathbf{x}(t))$ between switches satisfies (3.8). Denote the limit from the right/left as superscript $+/-$, respectively. Then,

$$V_{\sigma(t_{i+1}^-)}(\mathbf{x}(t_{i+1}^-)) \leq e^{-\varepsilon(t_{i+1}-t_i)} V_{\sigma(t_i^+)}(\mathbf{x}(t_i^+)). \tag{3.11}$$

At switches,

$$V_{\sigma(t_i^+)}(\mathbf{x}(t_i^+)) \leq \mu(\kappa) V_{\sigma(t_i^-)}(\mathbf{x}(t_i^-)) \tag{3.12}$$

holds. We can iterate on i to obtain

$$V_{\sigma(T^-)}(\mathbf{x}(T^-) \leq e^{((\log \mu(\kappa)/\tau)-\varepsilon)(T-t_0)} V_{\sigma(t_0)}(\mathbf{x}(t_0)). \tag{3.13}$$

Importantly, under the dwell time condition, (3.10), this implies that by taking T suitably large, we can make $V_{\sigma(T^-)}(\mathbf{x}(T^-))$ arbitrarily small. Thus, $\mathbf{x}(T^-) \in \mathcal{N}_{\sigma(T^-)}(\kappa)$. This proof is existential, not constructive. We cannot calculate a particular time T for any particular problem.

(ii) The second part of the proof shows that after a switch at time t_i, the dwell time is sufficiently large to force the trajectory back into $\mathcal{N}_{\sigma(t)}(\kappa)$ before a subsequent switch at time t_{i+1}. Furthermore, the trajectory cannot escape $\mathcal{M}(\kappa)$ in that interval.

□

Details are available in [30]. The proof presented above is different from [30] in one important way - we specify that the trajectory enters $\mathcal{N}_{\sigma(t)}(\kappa)$ rather than $\mathcal{N}(\kappa)$. In fact, it is an error to do the latter. The trajectory may pass through $\mathcal{N}(\kappa) \setminus \mathcal{N}_{\sigma(t)}(\kappa)$ and then switch after it has emerged, which may cause it to exit $\mathcal{M}(\kappa)$. There is nothing special about $\mathcal{N}(\kappa) \setminus \mathcal{N}_{\sigma(t)}(\kappa)$ while $\sigma(t)$ remains constant. It may be aesthetically displeasing to have the entry set change in time along with $\sigma(t)$, but we must do this, because the time-independent formulation is false. We will demonstrate with an example.

Example This example is a slight variation of Example 2 from [30]. Choose

$$\dot{\mathbf{x}}_p = A\mathbf{x}_p + b_p, \tag{3.14}$$

but with

$$A = \begin{bmatrix} -1 & -10 \\ 10 & -1 \end{bmatrix} \tag{3.15}$$

$$b_1 = \begin{bmatrix} 10 \\ 1 \end{bmatrix}, b_2 = \begin{bmatrix} -1 \\ 10 \end{bmatrix}, b_3 = \begin{bmatrix} 1 \\ -10 \end{bmatrix}.$$

We are able to use the same Lyapunov functions, $V_1(\mathbf{x}) := x_1^2 + (x_2 - 1)^2$, $V_2(\mathbf{x}) := (x_1 + 1)^2 + x_2^2$, and $V_3(\mathbf{x}) := (x_1 - 1)^2 + x_2^2$. One can check that τ can be the same as in [30], but that is not important here. Consider the trajectory shown in Figure 3.3. The red circles show $\mathcal{N}(\kappa)$; the black circles show $\mathcal{M}(\kappa)$. Subsystem 1 is active at the start, and the trajectory is shown in blue. $\mathcal{N}_1(\kappa)$ is the red circle centered around $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$. The trajectory passes through $\mathcal{N}_3(\kappa)$, which is the red circle centered around $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$. This is entering $\mathcal{N}(\kappa)$, but not $\mathcal{N}_{\sigma(t)}(\kappa)$. Now, notice that we could have selected initial conditions to make the entry into $\mathcal{N}_3(\kappa)$ occur at arbitrarily large time. This allows us to place a switch anywhere along the trajectory, regardless of τ (this is a "free switch" that we will see again in Section 3.2.5). In this case, we switched to subsystem 2 at the location where the trajectory changes to green. It exits $\mathcal{M}(\kappa)$ shortly thereafter.

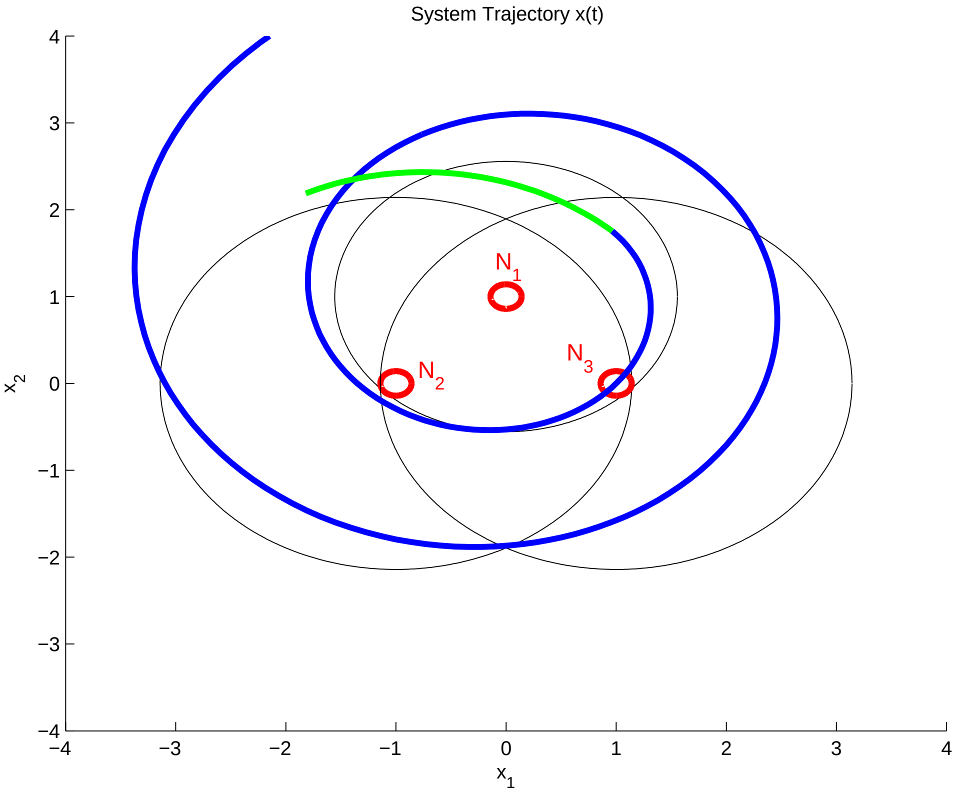


Figure 3.3: Example trajectory demonstrating the need for time-dependent $\mathcal{N}_{\sigma(t)}(\kappa)$.

3.2.2 Problem Statement: Switching Systems having Multiple Invariant Sets

Alpcan and Başar considered subsystems, each having a globally exponentially stable equilibrium point [30]. We will relax this condition to allow for systems with multiple invariant sets. Consider a switched system with C^1 functions $V_p : \mathbb{R}^n \to \mathbb{R}$ bounded on every bounded subset of $\mathbb{R}^n$ such that

$$\dot{V}_p(\mathbf{x}) \leq 0, \, \forall \mathbf{x} \in \mathbb{R}^n, \tag{3.16}$$

and $V_p(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$. Denote

$$E_p = \left\{ \mathbf{x} \in \mathbb{R}^n | \dot{V}(\mathbf{x}) = 0 \right\}. \tag{3.17}$$

Our problem is as follows. Is there a dwell time condition that suffices for ultimate boundedness?

Two challenges are immediately apparent. First, if $\dot{V}$ vanishes outside of $\mathcal{N}_p\left(\kappa\right)$, there is not a strictly positive Lyapunov decay rate outside of $\mathcal{N}_p\left(\kappa\right)$, so it is unclear how long to wait between switches. Second, even in the absence of switching, the active system may never enter $\mathcal{N}_p\left(\kappa\right)$ (E_p may not be contained in $\mathcal{N}_p\left(\kappa\right)$; see Figure 3.4). The following three subsections describe two methods for overcoming these challenges.

3.2.3 Intermediate Solution: Expand Entry Neighborhood

The simplest idea is to realize that $\mathcal{N}_p\left(\kappa\right)$ grows in size as we increase κ. Assuming that E_p is bounded, we can pick κ large enough so that $E_p \subset \mathcal{N}_p\left(\kappa\right)$. Even so, not all Lyapunov functions satisfying (3.16) will decay exponentially on $\mathbb{R}^n \setminus \mathcal{N}_p\left(\kappa\right)$. For example, consider a single one-dimensional subsystem $\dot{x} = -\arctan(x)$ with $V = x^2$. Nevertheless, the following lemma is useful:

Lemma 1. *Consider a switched system with Lyapunov functions $\mathcal{V}_p$ satisfying $\dot{\mathcal{V}}_p \leq 0$ with $\mathcal{V}_p(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$ and E_p bounded. Then, for sublevel Lyapunov sets $\mathcal{N}_p(\kappa)$ such that $E_p \subset Int(\mathcal{N}_p(\kappa))$ (where Int denotes the interior) and any $\varepsilon \in \mathcal{R}$, there exists a different set of continuous, radially unbounded Lyapunov functions V_p such that,*

1. *$V_p(\mathbf{x}) = 0$ for $\mathbf{x} \in \mathcal{N}_p$, and $V_p(\mathbf{x}) > 0$ for $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{N}_p$.*
2. *$V_p(\mathbf{x}(t+t_0)) \leq e^{-\varepsilon t} V_p(\mathbf{x}(t_0))$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{N}_p$ and $t, t_0 \in \mathbb{R}$.*

Proof. Bhatia [77] constructed a unique continuous function $s(\mathbf{x})$ on $\mathbb{R}^n \setminus \mathcal{N}_p$ such that $s(\mathbf{x}(t+t_0)) = s(\mathbf{x}(t_0)) - t$ and $s(\mathbf{x}) \to 0$ as $\mathbf{x} \to \mathcal{N}_p$. Then, we can select any constant ε and set

$$V_p(\mathbf{x}) = \begin{cases} 0 & \text{for } \mathbf{x} \in \mathcal{N}_p \\ e^{\varepsilon s(\mathbf{x})} & \text{for } \mathbf{x} \in \mathbb{R}^n \setminus \mathcal{N}_p. \end{cases} \tag{3.18}$$

Hence, on $\mathbb{R}^n \setminus \mathcal{N}_p$,

$$V_p(\mathbf{x}(t+t_0)) = e^{\varepsilon s(\mathbf{x}(t+t_0))} = e^{\varepsilon(s(\mathbf{x}(t_0))-t)} = e^{-\varepsilon t} V_p(\mathbf{x}(t_0)). \tag{3.19}$$

□

This means that for suitably large κ, there exist V_p that decay exponentially outside $\mathcal{N}_p(\kappa)$. A simple constant shift can patch the original Lyapunov function on $\mathcal{N}_p(\kappa)$ with the construction of Lemma 1 outside $\mathcal{N}_p(\kappa)$ at $\partial\mathcal{N}_p$ (the boundary of $\mathcal{N}_p$), and the mere continuity of V_p outside of $\mathcal{N}_p$ does not harm any essential parts of the proof. (Note that performing a constant shift outside $\mathcal{N}_p(\kappa)$ will scale ε by α_p, but we could just perform the construction again with a larger ε to correct for this). However, relying on the construction of Lemma 1 may not allow for (3.9) to hold, and we must assume that we can find a set of Lyapunov functions which satisfy both the exponential decay property and the μ property, (3.9). It is left as an open problem to determine if it is generally possible to construct a set of Lyapunov functions that satisfy both conditions for any switched system.

Now, the conditions required for Theorem 6 hold (noting that is is not essential to have exponential Lyapunov decay within $\mathcal{N}_p$).

Remark 2. *It is useful to notice that our set definitions do not require a single value of κ for all subsystems. If our subsystems have qualitatively different invariant sets, it may be very damaging to require a single κ. Instead, we can choose a set, $\boldsymbol{\kappa} = \{\kappa_p\}$, of values and restate all our set definitions as functions of $\boldsymbol{\kappa}$.*

Making κ_p very large is problematic in two ways. First, the purpose of defining a large neighborhood is to neglect the troubling areas (by this, we mean that we do not know much about the trajectory inside $\mathcal{N}(\boldsymbol{\kappa})$). However, with large κ_p, we may be cutting out substantial portions of our state space. Secondly, it may lead to larger dwell time or larger $\mathcal{M}(\boldsymbol{\kappa})$. In the next two subsections, we describe a method to more tightly tailor our strategy.

3.2.4 First Main Result: Tightly Tailored Entry Set

Since the troubling regions are just those where $\dot{V}_p \leq 0$ is closest to zero, we will now define a smaller set containing these regions. Choose a set $\boldsymbol{\delta} = \{\delta_{p_1}, \delta_{p_2}, ..., \delta_{p_{\max}}\}$ with $\delta_{p_i} > 0$, and define the set

$$\mathcal{H}_p(\delta_p) = \left\{ \mathbf{x} \in \mathbb{R}^n : \dot{V}_p(\mathbf{x}) > -\delta_p \right\}, \tag{3.20}$$

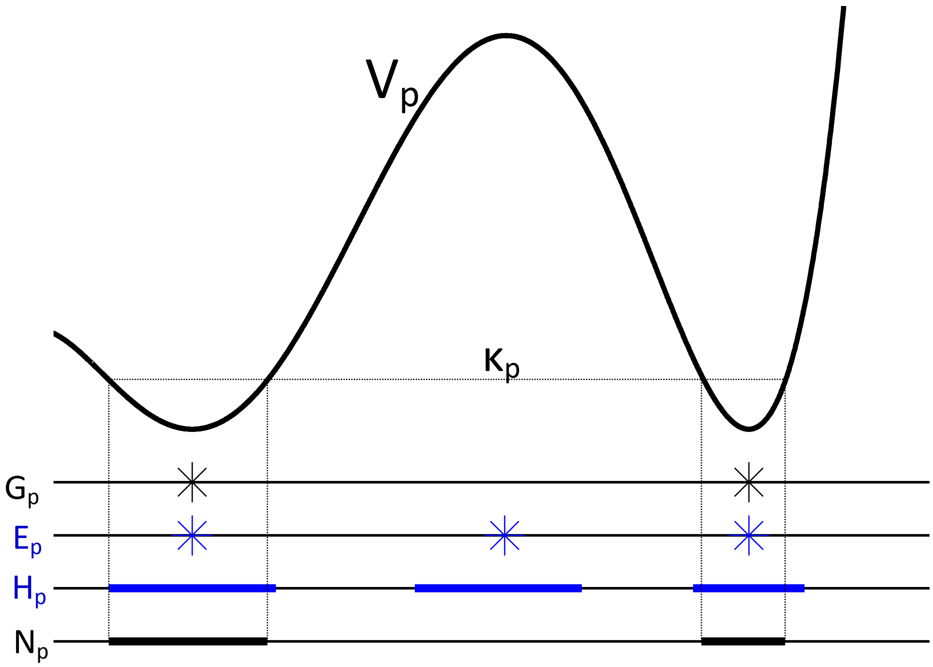


Figure 3.4: Qualitative example of tighter tailoring. The subsystem has two stable equilibria and a single unstable equilibrium. Notice that $\mathcal{N}_p \subset \mathcal{H}_p$.

so that $E_p \subset \mathcal{H}_p$. As usual, we can also define $\mathcal{H}(\boldsymbol{\delta}) = \bigcup\limits_{p \in \mathcal{P}} \mathcal{H}_p$. Figure 3.4 provides a one-dimensional example to help visualize how these sets are constructed. Similar to before, not all V_p satisfying (3.16) decay exponentially outside $\mathcal{H}_p(\delta_p)$. Define

$$\gamma_p(\delta_p) = \max_{\mathbf{x} \in \mathcal{H}_p} V_p(\mathbf{x}). \tag{3.21}$$

Furthermore, set

$$\mathcal{L}_p(\delta_p) = \{\mathbf{x} \in \mathbb{R}^n : V_p(\mathbf{x}) \leq \gamma_p(\delta_p)\}. \tag{3.22}$$

Since $\mathcal{L}_p$ is a sublevel Lyapunov set, it is compact. From compactness, $\dot{V}_p$ attains a minimum on $\mathcal{L}_p$, while V_p attains a finite maximum. Thus, on $\mathcal{L}_p \setminus \mathcal{H}_p$, an exponential decay rate can be computed, while we can use Lemma 1 outside $\mathcal{L}_p$. Again, if necessary, a simple constant shift can patch the two functions together at $\partial\mathcal{L}_p$.

While the construction in Lemma 1 only gives $s(\mathbf{x})$ continuous in the multi-dimensional sense, it is clearly directionally differentiable along the subsystem trajectories. Thus, writing $\dot{V}$ is sound notation. Putting it all together, we can compute ε such that

$$\dot{V}_p\left(\mathbf{x}\right) \leq -\varepsilon V_p\left(\mathbf{x}\right), \forall \mathbf{x} \in \mathbb{R}^n \setminus \mathcal{H}_p\left(\delta_p\right), \forall p \in \mathcal{P}. \tag{3.23}$$

Unfortunately, $\mathcal{H}_p$ may be disconnected, and it is not necessarily invariant even in the absence of switching. We will engage these problems directly in Sec. 3.2.5, but for now, we can proceed directly to a simple theorem demonstrating the usefulness of embedding $\mathcal{N}$ inside $\mathcal{H}$.

Theorem 7. *Consider a family of systems defined by (3.1), each with a radially unbounded Lyapunov-like function that satisfies (3.16). Assume $\mathcal{H}_p$, E_p bounded and $\mathcal{G}_p \subset \mathcal{H}_p(\boldsymbol{\delta})$. Compute $\varepsilon(\boldsymbol{\delta})$ so that (3.23) is satisfied and assume*

$$\frac{V_r\left(\mathbf{x}\right)}{V_q\left(\mathbf{x}\right)} \leq \mu(\boldsymbol{\delta}), \forall q, r \in \mathcal{P}, \ \ \forall \mathbf{x} \in \mathbb{R}^n \setminus \mathcal{H}\left(\boldsymbol{\delta}\right) \tag{3.24}$$

holds for finite $\mu(\boldsymbol{\delta})$. Furthermore, compute $\kappa_p > 0$ such that $\mathcal{G}_p \subset \mathcal{N}_p(\boldsymbol{\kappa}) \subseteq \mathcal{H}_p(\boldsymbol{\delta})$. Then, for every switching signal with dwell time $\tau > \frac{\log \mu(\boldsymbol{\delta})}{\varepsilon}$, there exists a time T such that $\mathbf{x}(T^-) \in \mathcal{H}_{\sigma(T^-)}\left(\boldsymbol{\delta}\right)$.

Proof. Notice that all of the necessary assumptions are valid on $\mathbb{R}^n \setminus \mathcal{H}_{\sigma(t)}(\boldsymbol{\delta})$. By taking suitably large T, we can make $V_{\sigma(T^-)}(\mathbf{x}(T^-))$ arbitrarily small. Therefore, either $\mathbf{x}(T^-) \in \mathcal{N}_{\sigma(T^-)}(\boldsymbol{\kappa}) \subseteq \mathcal{H}_{\sigma(T^-)}\left(\boldsymbol{\delta}\right)$ *or* the trajectory enters $\mathcal{H}_{\sigma(t)}\left(\boldsymbol{\delta}\right)$ somewhere else before that time. □

Remark 3. *In [30], κ was a single tuning parameter. In Section 3.2.3, $\boldsymbol{\kappa} = \{\kappa_p\}$*

was introduced as a possible set of tuning parameters. Now, per Theorem 7, the set of tuning parameters is $\boldsymbol{\delta} = \{\delta_p\}$, and $\boldsymbol{\kappa}$ is computed as a consequence of our selection of $\boldsymbol{\delta}$. Since nearly every parameter/set which follows is dependent on $\boldsymbol{\delta}$, we will often omit explicit dependence in favor of readability.

3.2.5 Second Main Result: No-Escape Set

This section assumes the trajectory has entered $\mathcal{H}_{\sigma(t)}(\boldsymbol{\delta})$ at some time and proceeds to build the relevant no-escape set, which will be denoted $\mathcal{M}_p$.

The primary problem is that $\mathcal{H}_p$ is not necessarily an invariant set even for periods of time when $\sigma(t)$ constant is constant (i.e., no switching). For example, a subsystem may contain a locally unstable equilibrium. With δ_p small, $\mathcal{H}$ is certainly not an invariant set. One way this can be problematic is that we can get a free extra switch.

Example Consider two one-dimensional subsystems, $\dot{x}_1 = -x_1$ and $\dot{x}_2 = x_2 - x_2^3$, with $V_2 = (x-1)^2(x+1)^2$ and small δ_2. Assume a nonzero initial condition with the first subsystem being active. The trajectory can become arbitrarily close to zero before switching to the second subsystem. While the second subsystem is active ($\sigma(t)$ is constant), the trajectory can clearly leave $\mathcal{H}_2$. Furthermore, since the second subsystem started arbitrarily close to the origin, it can take an arbitrarily long time to exit $\mathcal{H}_2$. Thus, no finite dwell time can prevent at least one switch from being possible outside of $\mathcal{H}$.

There are two ways to compute a dwell time and an associated spatial bound, but we need a few more definitions first. Set the usual $\gamma(\boldsymbol{\delta}) = \max\limits_{p\in\mathcal{P}} \gamma_p(\delta_p)$ and $\mathcal{L}(\boldsymbol{\delta}) = \bigcup\limits_{p\in\mathcal{P}} \mathcal{L}_p$. Compute

$$\xi_p(\boldsymbol{\delta}) = \max_{\mathbf{x}\in\mathcal{L}(\boldsymbol{\delta})} V_p(\mathbf{x}), \tag{3.25}$$

and set $\xi(\boldsymbol{\delta}) = \max_{p\in\mathcal{P}} \ \xi_p(\boldsymbol{\delta})$.

Theorem 8. *Consider a family of systems defined by (3.1), each with a radially unbounded Lyapunov-like function that satisfies (3.16). Assume $\mathcal{H}_p$, E_p bounded and $\mathcal{G}_p \subset \mathcal{H}_p(\boldsymbol{\delta})$. Compute $\varepsilon(\boldsymbol{\delta})$ so that (3.23) holds and assume*

$$\frac{V_r\left(\mathbf{x}\right)}{V_q\left(\mathbf{x}\right)} \le \mu(\boldsymbol{\delta}), \forall q, r \in \mathcal{P} \ \ \forall \mathbf{x} \in \mathbb{R}^n \setminus \mathcal{H}\left(\boldsymbol{\delta}\right) \tag{3.26}$$

holds for finite $\mu(\boldsymbol{\delta})$. Furthermore, compute $\kappa_p > 0$ such that $\mathcal{G}_p \subset \mathcal{N}_p(\boldsymbol{\kappa}) \subseteq \mathcal{H}_p(\boldsymbol{\delta})$. Set $\kappa = \min_{p\in\mathcal{P}} \kappa_p$. Then, for every switching signal with dwell time

$$\tau > \frac{\log \frac{\xi(\boldsymbol{\delta})}{\kappa}}{\varepsilon}, \tag{3.27}$$

for every $\bar{t}$ such that $\mathbf{x} \in \mathcal{H}_{\sigma(\bar{t})}$, $\mathbf{x}(t) \in \mathcal{M}\left(\boldsymbol{\delta}\right) = \bigcup_{p\in\mathcal{P}} \mathcal{M}_p$ for all $t \ge \bar{t}$, where

$$\mathcal{M}_p\left(\boldsymbol{\delta}\right) = \left\{\mathbf{x} \in \mathbb{R}^n : V_p\left(\mathbf{x}\right) \le \xi_p\right\}. \tag{3.28}$$

Proof. Consider the following sequence of times. Assume a switch occurs or the system is started at t_0. Assume further that there is a time $t_0 \le t_{\text{enter}}$ at which the trajectory enters $\mathcal{H}_{\sigma(t_0^+)}(\boldsymbol{\delta})$. There may or may not be a time at which the trajectory exits $\mathcal{H}_{\sigma(t_0^+)}(\boldsymbol{\delta})$, which we will denote $t_{\text{enter}} \le t_{\text{exit}}$. If there is, $t_{\text{exit}} - t_{\text{enter}}$ could possibly be arbitrarily large. Therefore, the second switching time t_1 may be shortly after t_{exit} while the trajectory is outside of $\mathcal{H}_{\sigma(t_0^+)}(\boldsymbol{\delta})$ regardless of the dwell time.

The proof will proceed in two parts. First, we will show that the trajectory does not leave $\mathcal{M}$ before a third switching time, called t_2. Then, we will compute a dwell time so that the trajectory must re-enter $\mathcal{H}_{\sigma(t)}$ before t_2. By induction, the trajectory will never leave $\mathcal{M}$.

(i) We will show that the trajectory never leaves $\mathcal{M}$ on $[t_{\text{enter}}, t_1]$, then on $[t_1, t_2]$.

Since $\mathbf{x}(t_{\text{enter}}) \in \mathcal{H}_{\sigma(t_0)} \subset \mathcal{M}_{\sigma(t_0)}$ and $\mathcal{M}_{\sigma(t_0)}$ is a sublevel Lyapunov set, the trajectory cannot leave $\mathcal{M}_{\sigma(t_0)}$ on $[t_{\text{enter}}, t_1]$ since no switching occurs on this interval. Next, notice

$$\mathbf{x}(t_1) \in \mathcal{L}_{\sigma(t_1^-)} \subset \mathcal{L} \subset \mathcal{M}_{\sigma(t_1^+)}. \tag{3.29}$$

Thus, the trajectory cannot leave $\mathcal{M}_{\sigma(t_1^+)}$ on $[t_1, t_2]$.

(ii) We will show that the trajectory must re-enter $\mathcal{H}_{\sigma(t)}$ before t_2. Define $\Delta = t_2 - t_1 \geq \tau$, giving

$$V_{\sigma(t_2^-)}(\mathbf{x}(t_2^-)) \leq e^{-\varepsilon\Delta} V_{\sigma(t_1^+)}(\mathbf{x}(t_1^+)) \leq e^{-\varepsilon\Delta} \xi_{\sigma(t_1^+)} \leq e^{-\varepsilon\Delta} \xi. \tag{3.30}$$

The dwell time given by (3.27) gives that $V_{\sigma(t_2^-)}(\mathbf{x})(t_2^-) \leq \kappa$. Thus, the trajectory enters $\mathcal{N}_{\sigma(t_1^+)}$ *or* reenters $\mathcal{H}_{\sigma(t)}$ elsewhere before time t_2.

□

Corollary 1. *Taken together, Theorems 7 and 8 provide a dwell time which guarantees finite time entry into $\mathcal{H}_{\sigma(t)}$ and the invariance of $\mathcal{M}$ thereafter. Given a particular problem, if we want to apply both theorems, we should set $\tau > \frac{\log \max\left(\mu, \frac{\xi}{\kappa}\right)}{\varepsilon}$.*

Section 3.3 will show examples that have $\frac{\xi}{\kappa} > \mu$, but it is trivial to construct cases with very large μ. Next, we construct some lower bounds for μ which are helpful for determining that the bounds in these theorems are tighter than some other possible bounds. Define

$$\eta_p\left(\boldsymbol{\delta}\right) = \max_{\mathbf{x} \in \mathcal{H}} V_p(\mathbf{x}), \tag{3.31}$$

and the following proposition will be useful.

Proposition 9. *Given the preceding set definitions, $\mu\gamma \geq \xi_p \geq \eta_p \; \forall p$.*

Proof. Since $\mathcal{H}_p \subset \mathcal{L}_p$, $\mathcal{H} \subset \mathcal{L}$, and $\xi_p \geq \eta_p$.

Without loss of generality, consider $\mathbf{x} \in \mathcal{L}$ such that $V_1(\mathbf{x}) = \xi_1$. There are three cases:

Case 1. *Suppose $\mathbf{x} \in \mathcal{H}_1$. Then, $\xi_1 = \gamma_1$. Since $\gamma \geq \gamma_1$ and $\mu \geq 1$, $\mu\gamma \geq \xi_1$.*

Case 2. *Suppose $\mathbf{x} \in \mathcal{H} \setminus \mathcal{H}_1$. Then, $\xi_1 = \eta_1$. Since ξ_1 is a maximum of V_1 over $\mathcal{L} \supset \mathcal{H}$ and H is open, $V_1(\mathbf{x})$ must be a local maximum. But, if it is a local maximum, $\nabla V_1(\mathbf{x}) = 0$ and $\dot{V}_1(\mathbf{x}) = 0$. Thus, $\mathbf{x} \in \mathcal{H}_1$, and the problem is reduced to Case 1.*

Case 3. *Suppose $\mathbf{x} \in \mathcal{L} \setminus \mathcal{H}$. Then, (3.26) holds, and*

$$\mu \geq \frac{V_1(\mathbf{x})}{V_p(\mathbf{x})} \; \forall p. \tag{3.32}$$

Then, for some p, $\mu\gamma \geq \mu\gamma_p \geq \mu V_p(\mathbf{x}) \geq V_1(\mathbf{x}) = \eta_1 = \xi_1$.

□

Notice that in the ideal case, with each subsystem consisting only of a single exponentially stable equilibrium, $\gamma = \kappa$, $\mathcal{L}_p = \mathcal{H}_p = \mathcal{N}_p$, and $\xi = \alpha$ can be attained. In the case they are attained, $\tau = \frac{\log \max\left(\mu, \frac{\xi}{\kappa}\right)}{\varepsilon} = \frac{\log \max\left(\mu, \frac{\xi}{\gamma}\right)}{\varepsilon} = \frac{\log \mu}{\varepsilon}$, (3.28) collapses to (3.6), and we can consider this a true generalization of the result in [30]. In fact, one can notice that both examples in [30] satisfy the equality $\mu\kappa = \xi$.

Remark 4. *Using similar techniques as those in Theorem 8, one can find a bound for τ and $\mathcal{M}$ that looks like $\min(\max(\mu\gamma, \eta_p), \xi_p)$. This expression reduces to ξ_p given Proposition 9.*

Remark 5. *We chose to define the dwell time as a constant across all subsystems, but allowed the spatial calculation to proceed with regard to individual subsystems. This* *is consistent with the idea that applications will utilize a single, subsystem invariant, dwell time. Since only the final union matters for the no-escape set, it is of no importance that the individual pieces were built with different constants determining sublevel sets.*

3.3 Examples

We provide two numerical examples to illustrate the main results of this chapter. In particular, we want to demonstrate that tight tailoring can provide better results than expanding the entry neighborhood.

3.3.1 One-Dimensional Example

Consider a family of systems defined by

$$f_1(x) = -x(x-1)^2,\ f_2(x) = -(x-1). \tag{3.33}$$

Select $V_1(x) = x^2$ and $V_2(x) = (x-1)^2$ and $\delta = [0.05; 0.1]$. Then, $\mathcal{H}_1 = [-0.139, 0.197] \cup [0.803, 1.139]$, $\mathcal{H}_2 = [0.776, 1.224]$, $\mu = 67.28$, and $\varepsilon = 1.29$. We can set $\kappa = [0.0193; 0.050]$, giving $\mathcal{N}_1 = [-0.139, 0.139] \subset \mathcal{H}_1$ and $\mathcal{N}_2 = [0.776, 1.224] \subset \mathcal{H}_2$. Then, we can compute $\gamma = [1.297; 0.050]$, $\mathcal{L}_1 = [-1.139, 1.139]$, $\mathcal{L}_2 = [0.776, 1.224]$, and $\xi = [1.497; 4.575]$. Finally, this results in $\tau = 4.240$ and $\mathcal{M} = [-1.224, 3.139]$. The size of $\mathcal{M}$ is 4.362.

We compare the large-κ method with two metrics. First, we select κ so that τ is the same as the tight-tailoring method (we make the methods temporally-equivalent) and compare the size of $\mathcal{M}$. Second, we select κ so that the size of $\mathcal{M}$ is the same as the tight-tailoring method (we make the methods spatially-equivalent) and compare the required dwell time. In order to make $\tau = 4.240$, we must set $\kappa = [2.271; 0.257]$. This results in $\mu = 8.837$, $\varepsilon = 0.514$, and $\mathcal{M} = [-1.507, 3.507]$, which has size 5.014. For the second case, we can make $\mathcal{M} = [-1.181, 3.181]$ by setting $\kappa = [1.395; 0.0328]$. The result is that $\mu = 42.48$, $\varepsilon = 0.0657$, and $\tau = 57.08$. In each case, tight tailoring shows better performance.

3.3.2 Two-Dimensional Limit Cycle Example

Consider a family of systems defined by

$$\begin{aligned}
\mathbf{f}_1\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{bmatrix} -\left(x^2+y^2-1\right) & -1 \\ 1 & -\left(x^2+y^2-1\right) \end{bmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \\
\mathbf{f}_2\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{bmatrix} -\left(x^2+y^2+1\right) & -1 \\ 1 & -\left(x^2+y^2+1\right) \end{bmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \\
\mathbf{f}_3\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 2 \\ -2 \end{pmatrix}.
\end{aligned} \tag{3.34}$$

This is a simplified version of (2.1), a central pattern generator used for control of swimming [78] or flying robots [79]. Subsystem 1 has an unstable equilibrium at the origin and a stable limit cycle of unit radius centered around the origin. Subsystem 2 has a stable equilibrium at the origin. Subsystem 3 has a stable equilibrium at $\begin{bmatrix} 2 & 0 \end{bmatrix}^T$. Take the corresponding Lyapunov functions to be $V_1(\mathbf{x}) = (x^2+y^2-1)^2$, $V_2(\mathbf{x}) = (x^2+y^2)^2$, and $V_3(\mathbf{x}) = ((x-2)^2+y^2)^2$, respectively. If we choose $\boldsymbol{\delta} = [0.2, 0.1, 0.1]$, then $\epsilon = 0.22$, $\mu = 2215$, $\boldsymbol{\kappa} = [0.042, 0.022, 0.025]$, $\boldsymbol{\gamma} = [1.0, 0.22, 0.25]$, $\boldsymbol{\xi} = [23, 33, 136]$, and $\tau = 38.9$. Figures 3.5, 3.6, and 3.7 show the resulting sets. In fact, many of the resulting parameters (chief among them dwell time and size of

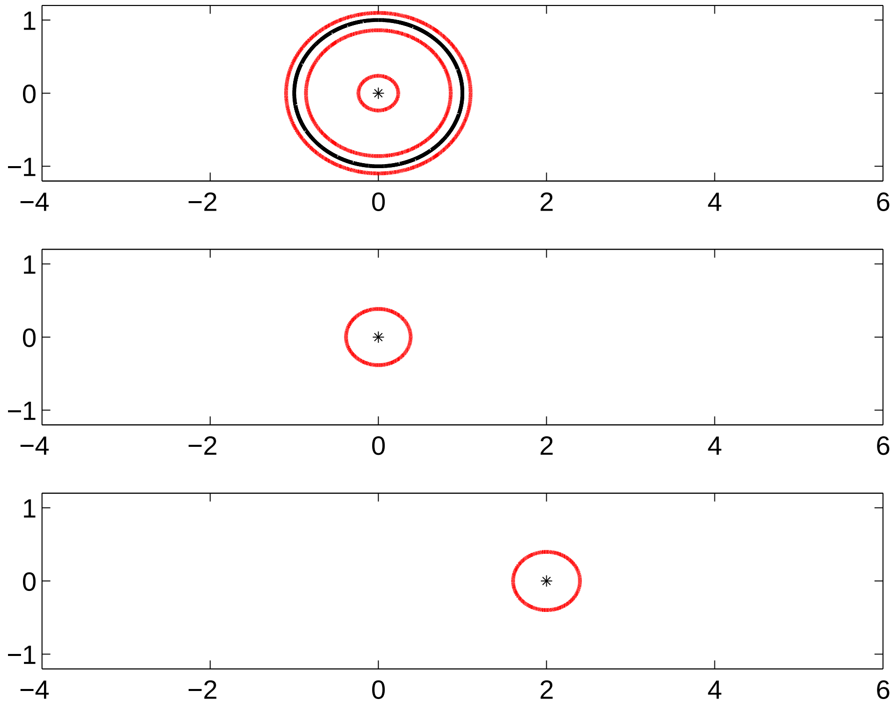


Figure 3.5: The sets E (black) and $\mathcal{H}$ (red) for each subsystem.

$\mathcal{M}$) are determined through nontrivial relations. Optimization and tradeoffs between spatial performance and temporal performance is problem-specific.

3.4 Chapter Summary

The ideas here have an eye toward bringing rigor to complicated flapping fliers, which often switch modes for energy and agility concerns.

We derived a stability result for switched systems which are constructed from subsystems which possibly contain multiple invariant sets. This amounted to a generalization and refinement of the argument presented in [30] and is in the spirit of dwell time methods for switched systems. This result can be applied to a larger class of dynamical systems than those in [30], including those which contain bifurcations. We

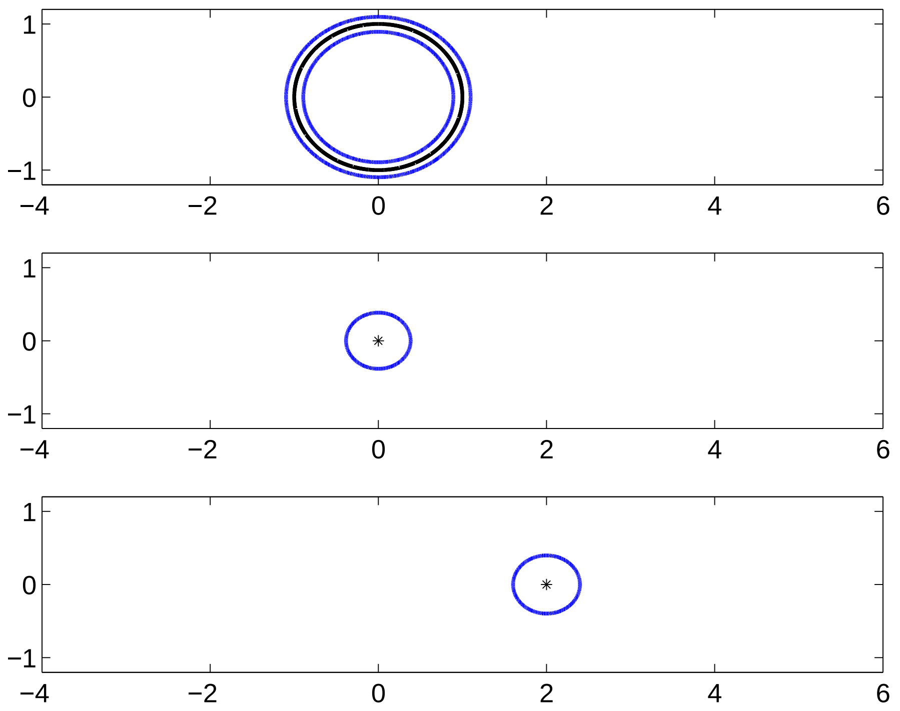


Figure 3.6: The sets $\mathcal{G}$ (black) and $\mathcal{N}$ (blue) for each subsystem. The example is suitably well-behaved so that $E = \mathcal{G}$ and $\mathcal{H} = \mathcal{N}$ except for the unstable origin and its surrounding neighborhood when $p = 1$.

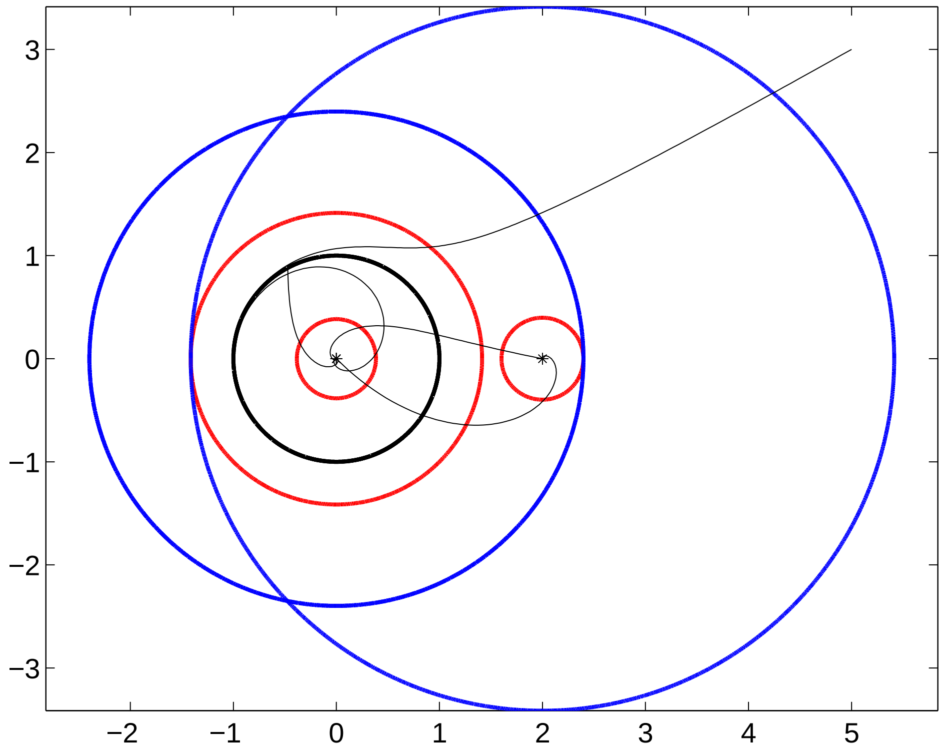


Figure 3.7: The sets E_p (black), $\mathcal{L}$ (red), and $\mathcal{M}$ (blue). The skinny black line is an example trajectory.

provide a set of tuning parameters, $\boldsymbol{\delta}$, which cause interrelated and problem-specific effects on performance.

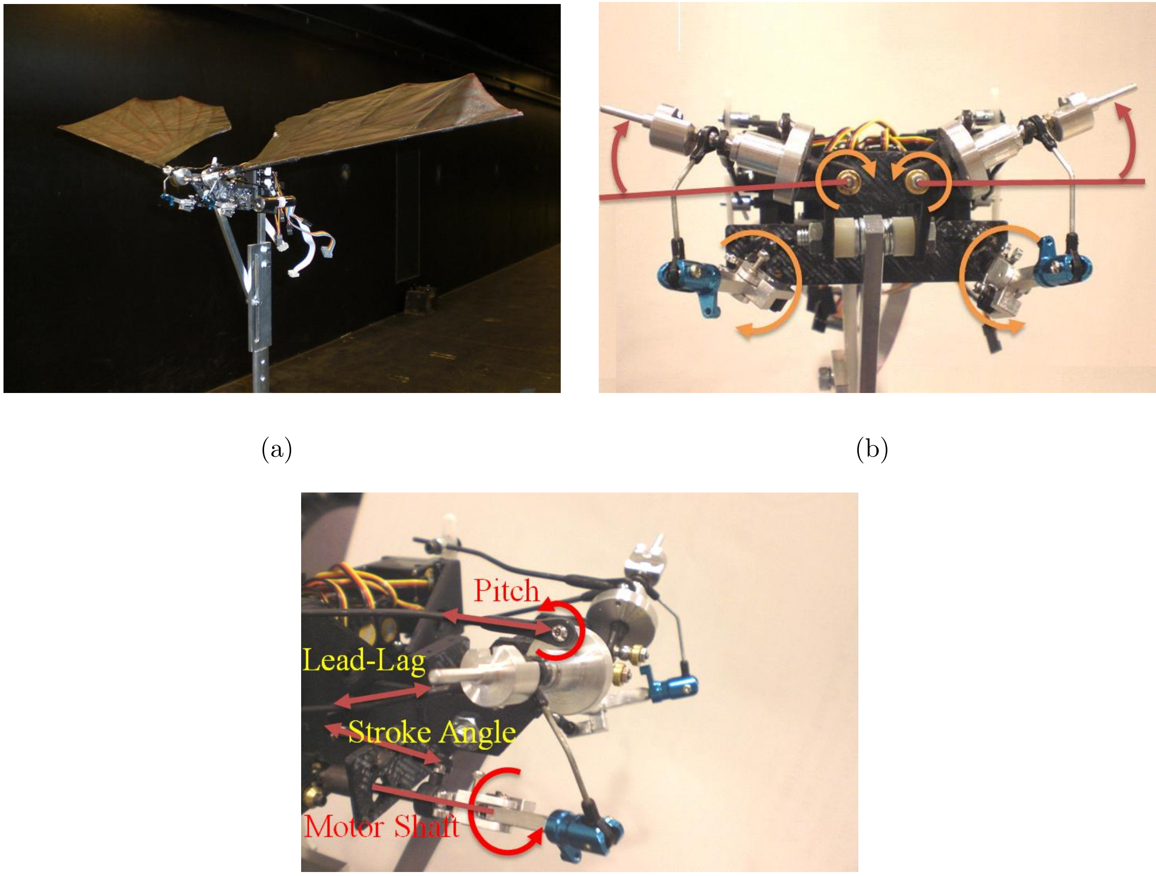
Chapter 4

Flapping Flight Simulation and Experiment

4.1 Wing Kinematics, Aerodynamic Forces, and Vehicle Dynamics

We derive a three dimensional model of the wing kinematics in this section. The wing kinematic model supports both flexible and rigid wings. Also, we present the 6-DOF dynamic equations of motion of flapping flight that can be used to validate the coupled wing control driven by CPG. The aim of this section is to illustrate that the effective angle of attack varies as a function of the wing span distance as well as the flapping (stroke) angular rate and it can be effectively controlled by the synchronized pitching (wing rotation) control.

We present a realistic modeling that encompasses a tilted stroke angle, the lead-lag motion, and the relative body velocity, in addition to the stroke and pitch angles. In deriving these equations, the actual control degrees-of-freedom of the robotic bat



(a)

(b)

(c)

Figure 4.1: A 8-DOF robotic bat (with 10 control variables) MAV [80]

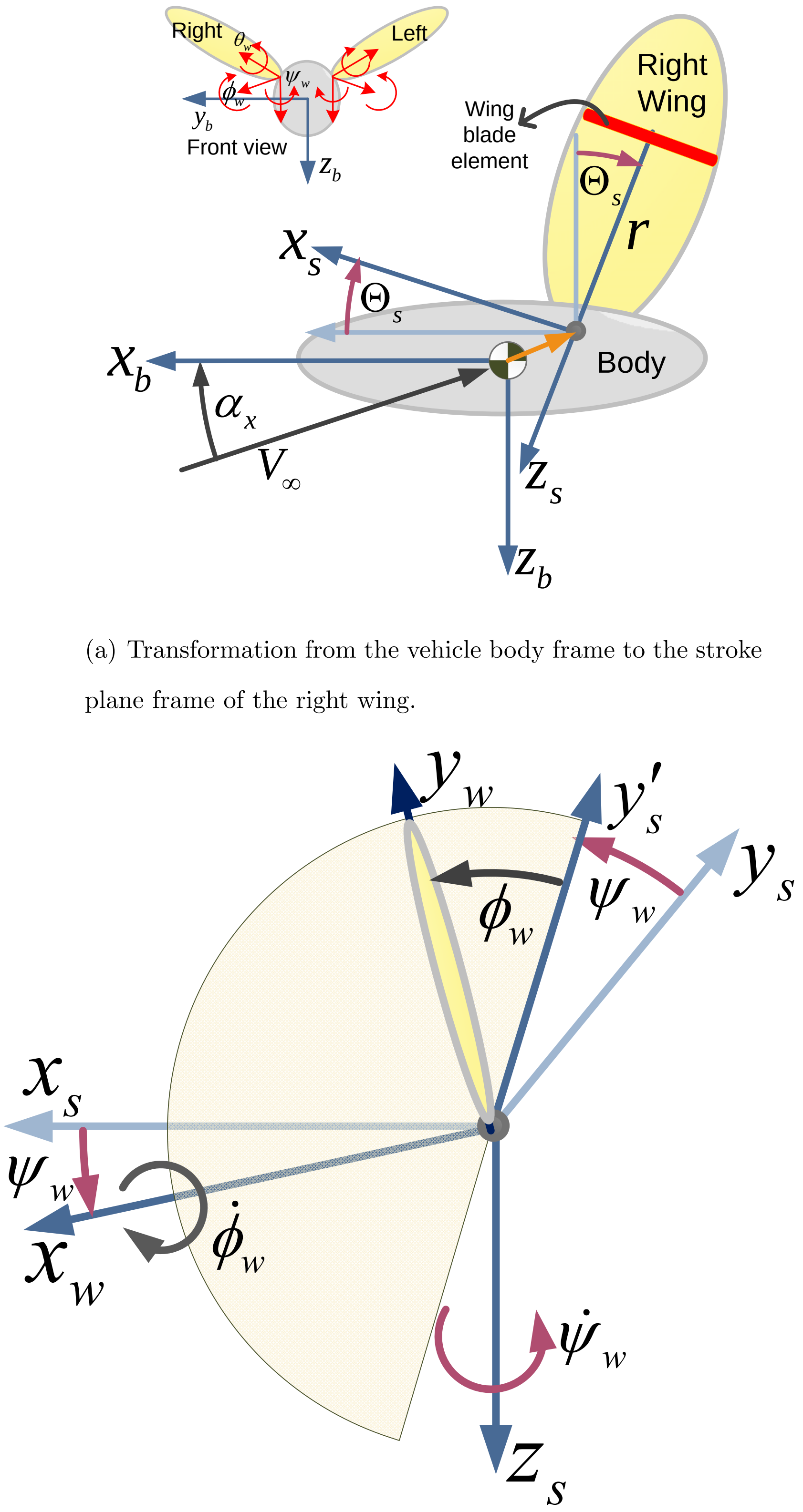
MAV testbed shown in Fig. 4.1 are considered.

Figure 4.2a shows a side view of the flapping flying MAV with the body frame $\mathbf{x}_b = (x_b, y_b, z_b)^T$ and the stroke-plane frame $\mathbf{x}_s = (x_s, y_s, z_s)^T$ of the right wing. Similar to the robotic bat in Fig. 4.1, we assume that each wing has flapping (ϕ_w), pitch (θ_w), and lead-lag (ψ_w) control. We develop only the equations of the right wing since the similar expressions for the left wing can straightforwardly follow. The center of the stroke-plane frame is located at (d_x, d_y, d_z), and it is tilted by the inclination angle $\Theta_s(t)$, which can be a function of time and the forward velocity. Without the lead-lag motion, the axes y_s and z_s define the stroke plane. Hence, the transformation between these coordinate axes can be given by

$$\mathbf{x}_b = \mathbf{T}_{bs}(\Theta_s)\mathbf{x}_s + (d_x, d_y, d_z)^T, \quad \text{where} \quad \mathbf{T}_{bs}(\Theta_s) = \begin{bmatrix} \cos\Theta_s & 0 & \sin\Theta_s \\ 0 & 1 & 0 \\ -\sin\Theta_s & 0 & \cos\Theta_s \end{bmatrix} \tag{4.1}$$

where in this chapter $\mathbf{T}_{bs}$ denotes the transformation from $\mathbf{x}_s$ to $\mathbf{x}_b$, whereas $\mathbf{T}_{sb} = \mathbf{T}_{bs}^T$ would correspond to the transformation from $\mathbf{x}_b$ to $\mathbf{x}_s$.
For a hovering insect, the stroke plane is almost horizontal (i.e., $\Theta_s = 90$ deg in our coordinate definition in Fig 4.2a), resulting in forward and backward reciprocating motions. This is the assumption used for some prior work [6, 7, 12, 51]. In contrast, the stroke angle of birds and bats varies as a function of flight speed; at a low speed, the angle is almost horizontal ($\Theta_s = 90$ deg) and it approaches $\Theta_s = 0$ deg as the flight speed increases.

If there is no lead-lag motion, the additional transformation for a wing stroke angle ϕ_w would complete all the required transformation between the body frame and the wing frame. However, a nonzero lead-lag angle further complicates the wing kinematics. Choosing the rotational axes for flapping, lead-lag, and pitch depends on the actual hardware setup and actuators, and our choice is influenced by the robotic bat MAV shown in Fig. 4.1 (see Section 4.4). In contrast with Azuma's derivation in [3] where the stroke angle $\Theta_s(t)$ is dependent on the $\phi_w(t)$ and the lead-lag angle $\psi_w(t)$, our $\Theta_s(t)$ is an independent control variable. Our decision is based on the observation that $\Theta_s(t)$ can be an important control variable for efficient engineered flapping flight. Further, this kind of actuator mechanism is easier to implement and control. As shown in Fig. 4.2b, the lead-lag angle is defined by the rotation about the z_s axis- the z-axis in the stroke plane frame. In contrast with the fixed angle rotation



(a) Transformation from the vehicle body frame to the stroke plane frame of the right wing.

(b) Transformation from the stroke plane frame to the wing frame .

Figure 4.2: Schematic of the 3D wing motions

in [3], then we rotate about the new x-axis to obtain the wing frame $\mathbf{x}_w$. For both wings, the positive direction of ψ_w is the forward direction, while the positive stroke angle ϕ_w indicates an upstroke motion. This sign convention does not agree with the original positive direction of rotation for the right wing, so extra care should be taken to determine the correct angular transformation matrices.

For the right wing, the transformation between the stroke plane frame ($\mathbf{x}_s$) and the wing frame ($\mathbf{x}_w$) can be written as

$$\mathbf{x}_s = \mathbf{T}_{sw}(\phi_w, \psi_w)\mathbf{x}_w = \begin{bmatrix} \cos\psi_w & \sin\psi_w & 0 \\ -\sin\psi_w & \cos\psi_w & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_w & \sin\phi_w \\ 0 & -\sin\phi_w & \cos\phi_w \end{bmatrix} \mathbf{x}_w \tag{4.2}$$

In order to compute the local lift and drag of a blade element with width dr and wingspan coordinate $r \in [0, R]$, we need to transform the velocities in body coordinates to the incident velocities in the rotated wing frame. For example, consider the free-stream forward speed V_∞ with the body angle of attack α_x and the side-slip angle α_y. Note that α_y is commonly denoted by β in the aerospace community, but in this chapter β denotes the direction of the relative wind of a blade element. Then, the free-stream velocity in the body frame can be written as

$$\mathbf{V}_b = (V_\infty \cos\alpha_y \cos\alpha_x, V_\infty \sin\alpha_y, V_\infty \cos\alpha_y \sin\alpha_x)^T + \mathbf{v}_i + \mathbf{v}_E \tag{4.3}$$

where $\mathbf{v}_i$ and $\mathbf{v}_E$ denote the induced velocity vector and the wind velocity vector respectively. In other words, in the absence of $\mathbf{v}_i$ and $\mathbf{v}_E$, the vector $\mathbf{V}_b$ equals the velocity of the vehicle in the body frame. Let us assume that α_x and α_y include the effects of the induced velocity and $\mathbf{v}_E$ is small.

Then, the free-velocity vector $\mathbf{V}_b$ in the body frame can be transformed to the wing frame. In addition, we can also compute the additional velocity on the wing frame induced from the body angular rate $\mathbf{\Omega}_b = (p, q, r)^T$ and the offset distance $\mathbf{d} = (d_x, d_y, d_z)^T$ of the stroke plane frame (see Fig. 4.2a). By adding these two terms, we can obtain

$$\mathbf{V}_b^w = \mathbf{T}_{ws}(\phi_w, \psi_w)\mathbf{T}_{sb}(\Theta_s)(\mathbf{V}_b + \mathbf{\Omega}_b \times \mathbf{d}) \tag{4.4}$$

In order to compute the rotational velocity on the wing frame produced by the flapping ϕ_w and lead-lag ψ_w motions, as well as a relatively slower stroke angle change $\Theta_s(t)$, it is more convenient to construct the angular rate vector in the stroke plane frame as follows

$$\mathbf{\Omega}_{tot} = \mathbf{T}_{sb}(\Theta_s)\mathbf{\Omega}_b + \begin{pmatrix} -\cos\psi_w\dot{\phi}_w \\ \sin\psi_w\dot{\phi}_w + \dot{\Theta}_s \\ -\dot{\psi}_w \end{pmatrix} \tag{4.5}$$

Then, we can compute the induced rotational velocity from the wing motions of the blade element dr

$$\mathbf{V}_{rot}^w = (\mathbf{T}_{ws}(\phi_w, \psi_w)\mathbf{\Omega}_{tot}) \times \left(\begin{pmatrix} 0 \\ r \\ 0 \end{pmatrix} + \begin{pmatrix} x_w(r) \\ y_w(r) \\ z_w(r) \end{pmatrix} \right) + \begin{pmatrix} \dot{x}_w(r) \\ \dot{y}_w(r) \\ \dot{z}_w(r) \end{pmatrix} \tag{4.6}$$

where $x_w(r)$, $y_w(r)$ and $z_w(r)$ are the deformation of the blade element due to aeroelastic deformation or active cambering control that can be found in bat flight. Hence, the derivations in this section can be used for flexible wing models, although the $C_L(\alpha)$ and $C_D(\alpha)$ functions should be corrected for such cambered wing shapes.

By adding $\mathbf{V}_b^w$ in Eq. (4.4) and $\mathbf{V}_{rot}^w$ in Eq. (4.6), we can obtain the total velocity of the wind at the blade element, located at r on the wing span axis,

$$\mathbf{V}_w = \begin{pmatrix} V_{wx} \\ V_{wy} \\ V_{wz} \end{pmatrix} = \mathbf{T}_{ws}(\phi_w, \psi_w)\mathbf{T}_{sb}(\Theta_s)(\mathbf{V}_b + \mathbf{\Omega}_b \times \mathbf{d}) \\ + (\mathbf{T}_{ws}(\phi_w, \psi_w)\mathbf{\Omega}_{tot}) \times \left(\begin{pmatrix} 0 \\ r \\ 0 \end{pmatrix} + \begin{pmatrix} x_w(r) \\ y_w(r) \\ z_w(r) \end{pmatrix} \right) + \begin{pmatrix} \dot{x}_w(r) \\ \dot{y}_w(r) \\ \dot{z}_w(r) \end{pmatrix} \tag{4.7}$$

A similar expressions can be obtained for the left wing. Also, the preceding derivations can straightforwardly be extended to account for the second joint (elbow) if each wing has one.

Now, we can obtain the local effective angle of attack α_w of the blade element to determine aerodynamic forces and torque. Let us assume that the deformation of a rigid wing is negligible and there is no active cambering control. Also, the contribution from the body angular rate $\mathbf{\Omega}_b$ is small. Then, Eq. (4.7) reduces to

$$\begin{pmatrix} V_{wx} \\ V_{wy} \\ V_{wz} \end{pmatrix} = \mathbf{T}_{ws}(\phi_w, \psi_w)\mathbf{T}_{sb}(\Theta_s)\mathbf{V}_b + \left(\mathbf{T}_{ws}(\phi_w, \psi_w) \begin{pmatrix} -\cos\psi_w \dot{\phi}_w \\ \sin\psi_w \dot{\phi}_w + \dot{\Theta}_s \\ -\dot{\psi}_w \end{pmatrix} \right) \times \begin{pmatrix} 0 \\ r \\ 0 \end{pmatrix} \tag{4.8}$$

Then, we can obtain the local incident angle β_w (measured clockwise), the angle of attack α_w, and the speed of the wind V_r on the blade element on the right wing as follows

$$\begin{aligned}
\beta_w(r,t) &= \tan^{-1}\frac{-V_{wz}}{V_{wx}}, \quad \alpha_w(r,t) = \theta_w(t) - \beta_w(r,t) \\
V_r^2(r,t) &= \sqrt{V_{wx}^2 + V_{wz}^2}
\end{aligned} \tag{4.9}$$

where we neglected the flow along the wing span V_{wy} and the wing rotation $\theta_w(t)$ controller can be properly designed to yield a positive angle of attack for both upstroke and downstroke motions.

If the MAV were flying with zero flight path angle and $\Theta_s = 0$, we can obtain

$$\beta_w(r,t) = \tan^{-1}\frac{r\dot{\phi}_w}{V_\infty} = \tan^{-1}\frac{2rk_r}{c} \quad \text{and } k_r = \frac{\dot{\phi}_w c}{2V_\infty} \tag{4.10}$$

where the reduced frequency k_r compares the velocity by the wing flapping motion with the forward speed, thereby indicating the degree of unsteady aerodynamics (if $k_r > 0.05$, unsteady effects dominate). We can see that the sign of β_w is consistent with the positive direction of the flapping (stroke) angle ϕ_w since the downstroke $\dot{\phi}_w < 0$ leads to the negative flow angle $\beta_w < 0$.

Once we obtain the local effective angle of attack α_w, we can proceed to obtain the aerodynamic forces of the blade element by evaluating the lift and drag coefficients, $C_L(\alpha_w)$ and $C_D(\alpha_w)$. Flapping flight, typically within a low Reynolds number regime (Re $< 10^5$), is governed by unsteady aerodynamics characterized by large-scale vortex structures. It is understood that the main lift enhancement mechanism of flapping flight is governed by (1) the leading edge vortex (LEV) that leads to delayed stall at a very high angle of attack, (2) the rotational circulation lift, and (3) wake capture that generate aerodynamic forces during flapping angle reversals [12]. In particular, Dickinson's series of papers [12, 51], by cross-validating the numerical computation and experimentation using the Robofly, shows that a quasi-steady aerodynamic model predicts the aerodynamic coefficients reasonably well. While Computational Fluid Dynamics (CFD) is much too computationally burdensome to justify its use in control design, this quasi-steady approximation method can be verified and improved by the experimental set-up such as the robotic bat described in [80].

The seminal paper by Dickinson [12] used a hovering pair of wings without a forward speed as follows

$$\begin{aligned} C_L(\alpha_w) &= 0.225 + 1.58\sin(2.13\alpha_w - 7.2\,\text{deg}) \\ C_D(\alpha_w) &= 1.92 - 1.55\cos(2.04\alpha_w - 9.82\,\text{deg}) \end{aligned} \tag{4.11}$$

It should be noted that Dickinson's robotfly's setup used a horizontal stroke plane, as typically seen in insect flight, whereas we assume a 90-deg stroke plane angle. The angle α_w for a general flapping wing is time-varying, as described in this section. A recent paper [51] considers a nonzero-forward speed and the coefficients of Eq. (4.11) can be modified to become functions of the reduced frequency (k_r).

From the quasi-steady approximation, we can compute the lift and drag forces acting on the blade element with width dr as follows.

$$dL = \frac{1}{2}\rho C_L\left(\alpha_w(r,t)\right)c(r)V_r^2(r,t)dr, \quad dD = \frac{1}{2}\rho C_D\left(\alpha_w(r,t)\right)c(r)V_r^2(r,t)dr \tag{4.12}$$

In addition, Ellington [52] derived the wing circulation $\Gamma_r = \pi\dot{\alpha}c^2(3/4 - \hat{x}_0)$ based on the Kutta-Joukowski condition. This quasi-steady approximation for the rotational lift can be written as

$$dL_{rot} = \frac{1}{2}\rho\left(2\pi(\frac{3}{4} - \hat{x}_0)\right)c^2(r)V_r(r,t)\dot{\alpha}_w dr \tag{4.13}$$

where $\hat{x}_0$ is the location of the pitch axis along the mean chord length. Also, $\dot{\alpha}_w$ can be computed from Eq. (4.9) and often approximated reasonably well by the angular rate of the wing pitch motion $\dot{\theta}_w$.

The total x and z directional forces of a single wing (either right or left) in the body frame are obtained as

$$\begin{aligned} F_{wz} &= \int_{r=0}^{R} dD \sin\beta_w - (dL + dL_{rot}) \cos\beta_w \\ F_{wx} &= \int_{r=0}^{R} -(dL + dL_{rot}) \sin\beta_w - dD \cos\beta_w, \end{aligned} \tag{4.14}$$

where the positive direction of z_b is downward as shown in Fig. 4.2a.

The F_{wx} and F_{wz} forces on the wing frame given in Eq. (4.14) can be transformed into the forces in the vehicle body frame:

$$\mathbf{F}_{\text{right}} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}_{\text{right}} = \mathbf{T}_{bs}(\Theta_s)\mathbf{T}_{sw}(\phi_w, \psi_w) \begin{pmatrix} F_{wx} \\ 0 \\ F_{wz} \end{pmatrix}_{\text{right}} \tag{4.15}$$

where we added the subscript *right* to indicate the right wing. A similar expression can be obtained for the left wing ($\mathbf{F}_{\text{left}}$). Each wing has different wing angular parameters such as ϕ_w, ψ_w, and θ_w, although the stroke plane angle Θ_s is the same for both wings.

In order to compute the rotational moments generated by the aerodynamic forces, we first calculate the position of the wing blade element with respect to the body frame

$$\mathbf{p}(r) = \mathbf{T}_{bs}(\Theta_s)\mathbf{T}_{sw}(\phi_w, \psi_w) \begin{pmatrix} 0 \\ r \\ 0 \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} \tag{4.16}$$

Then, we can compute the aerodynamic moments with respect to the c.g.

$$\begin{pmatrix} dM_x \\ dM_y \\ dM_z \end{pmatrix} = \mathbf{p}(r) \times \left(\mathbf{T}_{bs}(\Theta_s)\mathbf{T}_{sw}(\phi_w, \psi_w) \begin{pmatrix} -(dL + dL_{rot})\sin\beta_w - dD\cos\beta_w \\ 0 \\ dD\sin\beta_w - (dL + dL_{rot})\cos\beta_w \end{pmatrix} \right) + \begin{pmatrix} dM_{x0} \\ dM_{y0} \\ dM_{z0} \end{pmatrix} \tag{4.17}$$

$$\begin{pmatrix} dM_{x0} \\ dM_{y0} \\ dM_{z0} \end{pmatrix} = \mathbf{T}_{bs}(\Theta_s)\mathbf{T}_{sw}(\phi_w, \psi_w)\mathbf{T}_{\theta_w}(\theta_w)\frac{1}{2}\rho V_r^2 c(r)dr \begin{pmatrix} rc_{l0} \\ c(r)(c_{m0} + c_{m\alpha,w}\alpha_w) \\ rc_{n0} \end{pmatrix} \tag{4.18}$$

$$M_x = \int_{r=0}^{R} dM_x, \quad M_y = \int_{r=0}^{R} dM_y, \quad M_z = \int_{r=0}^{R} dM_z, \tag{4.19}$$

where dM_{x0}, dM_{x0}, and dM_{x0} denote the constant aerodynamic moments that include the moment at the mean aerodynamic center, computed by the moment coefficients c_{l0}, c_{m0}, $c_{\alpha,w}$, and c_{n0}. The transformation matrix $\mathbf{T}_{\theta_w}(\theta_w)$ rotates the wing frame about the y_w axis by the wing pitch rotation angle θ_w.

By combining all the forces and moments from the right wing and the left wing, we can derive 6-DOF equations of motion for the flapping flying MAV in the body frame, whose orientation with respect to the inertial frame is described by the Euler angles. We assume the mass and the moment of inertia of the wing compared to the body weight are negligible so that the c.g. remains fixed. Then, we can obtain the following set of equations. The translational motion of the c.g. of the flapping flying vehicle driven by the aerodynamic force terms in Eq. (4.15) can be expressed as

$$m\dot{\mathbf{V}}_b + m\mathbf{\Omega}_b \times \mathbf{V}_b = \mathbf{T}_{be}(\phi_b, \theta_b, \psi_b)\mathbf{F}_g + \mathbf{F}_{\text{right}} + \mathbf{F}_{\text{left}} + \mathbf{A} \tag{4.20}$$

where the Euler angular transformation matrix $\mathbf{T}_{be}(\phi_b, \theta_b, \psi_b)$ determines the orientation of the body frame with respect to the inertial frame.Each wing has different wing angular parameters such as ϕ_w, ψ_w, and θ_w, although the stroke plane angle Θ_s is the same for each wing.

The equations of rotational motion are driven by the aerodynamic moments $\mathbf{M}_{\text{right}}$ and $\mathbf{M}_{\text{left}}$ of each wing that can be obtained from Eq. (4.17)

$$\mathbf{I}_b\mathbf{\Omega}_b + \mathbf{\Omega}_b \times (\mathbf{I}_b\mathbf{\Omega}_b) = \mathbf{M}_{\text{right}} + \mathbf{M}_{\text{left}} + \mathbf{B} \tag{4.21}$$

The relationship between the body angular rate $\mathbf{\Omega}_b = (p, q, r)^T$ and the Euler angle vector $\mathbf{q}_b = (\phi_b, \theta_b, \psi_b)^T$ can be determined by [80]

$$\dot{\mathbf{q}}_b = \mathbf{Z}(\mathbf{q}_b)\mathbf{\Omega}_b \tag{4.22}$$

where any other orientation representations such as quaternions can be used in lieu of the Euler angles in the preceding equations. Also, any disturbance force and torque can be added to the equations.

4.2 CPG-based Flapping Flight Control and Simulation Results

The aim of this section is to show that CPG-based flight control can stabilize and control flapping flight dynamics given in Sec. 4.1 by commanding a reduced set of CPG parameters that generate the phase-synchronized or symmetry-breaking oscillatory motions of two main wings. In particular, we show that the dynamics can be controlled without using aerodynamic control surfaces such as ailerons, elevators,

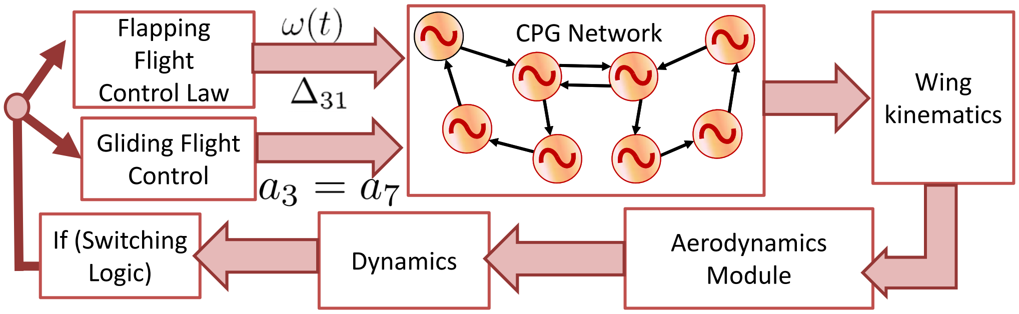


Figure 4.3: Block diagram of the simulation for two alternating flight modes: flapping and gliding.

rudders, and directional control of tail wings. This result can be extended to a more sophisticated nonlinear flight control law.

4.2.1 CPG-based Flapping Flight Control

The example presented in this section has three different flight modes: gliding, flapping, and turning. We do not consider the aerodynamics of the second joint and instead assume each wing to be a single rigid body. That is, in Eq. (4.7), we set $(x_w(r), y_w(r), z_w(r))^T = 0$ and assume the aerodynamic coefficients do not vary along the chord of the wing blade element.

This dynamic model with three dimensional wing kinematics is constructed in Matlab/Simulink, allowing us to demonstrate how stability can be obtained for flapping flight driven by a CPG network (see Fig. 4.3).

4.2.2 Gliding Mode Control

We produce gliding flight with no reciprocal flapping motion by setting the bifurcation parameter $\sigma = -1$ in Eq. (2.1). As discussed in Sec. 2.7, Setting $\omega(t) = 0$ without

$\sigma = -1$ will not ensure convergence to the controlled bias values ($u_i \rightarrow a_i$). This provides us with simple control of our wing by exploiting the bifurcation of Hopf oscillators, causing them to snap to a single non-oscillatory value corresponding to the bias. We can then control the lead lag motion (ψ_w), flapping angle (ϕ_w), and wing pitch angle (θ) by their bias parameters. A negative (positive) flapping angle or negative (positive) lead-lag angle can provide a pitch-down (pitch-up) moment due to drag or lift, respectively. We have therefore reduced control dimensionality to three actively controlled parameters: wing pitch, wing flapping angle, and lead-lag angle. In fact, depending on the physical characteristics of the specific vehicle, controlling only one of wing flapping angle or lead-lag angle could be sufficient for longitudinal gliding stability. For the network in Fig. 2.3a, we can use the following bias for the lead-lag angle of each wing:

$$a_3(t) = a_7(t) = -k_p\theta_b - k_d q - k_i \int_0^t \theta_b dt + \psi_{bias} \tag{4.23}$$

where k_p, k_d, and k_i are positive gains and ψ_{bias} is a constant. The simulation uses nonzero gains for PID control of flapping angle and lead-lag bias. Flapping angle seems most important, which is in keeping with previous study of the importance of drag-based stability [81], and the conclusion that birds glide more like tailless vehicles than conventional tailed aircraft. Only integral control is used for pitch bias, in order to obtain a constant wing angle of attack. Lateral modes with articulated wings warrant further attention.

4.2.3 Flapping Flight Control by Flapping Frequency:

Seemingly more difficult than stability of gliding flight is stability of flapping flight. We propose a novel control law unique to our CPG set-up which reduces control dimensionality to only four parameters. The first parameter is the oscillation frequency $\omega(t)$ of the coupled Hopf oscillators in Eq. (2.6). Flapping frequency, $\omega(t)$, correlates with increased lift and thrust. Those in turn, correlate with velocity of the body. For example, we can consider the following control law

$$\omega(t) = K \int_0^t \dot{\omega} dt = K \int_0^t \left(|V_{desired}| - |V_{actual}| \right) dt \tag{4.24}$$

and use the following corollary.

Corollary 2. *From the dynamic equation of the Hopf oscillator in Eq. (2.1), the time-varying $\omega(t)$ does not affect the synchronization stability proof for Theorem 2*

Proof. Since the symmetric part of $\mathbf{f}$ cancels the ω term and ω does not change the maximum eigenvalue of $\mathbf{V}^T \left[\mathbf{f}\right] \mathbf{V}$. The rest of the proof follows Theorem 2. □

4.2.4 Flapping Flight Control by Phase Differences and Symmetry Breaking:

Phase differences can be strong control parameters, but our oscillator stability proof in Theorem 2 assumes constant or relatively slowly varying phase differences. The error terms from the additional time-varying parameters other than $\omega(t)$ can be obtained by contraction analysis [63], which shows the boundedness of the synchronization error.

Corollary 3. *For time-varying phase differences $\Delta_{ij}(t)$, the synchronization of the rotated Hopf states $\{\mathbf{z}\}$ globally converges to the bounded error defined by $\mathbf{V}^T\mathbf{T}\dot{\mathbf{T}}^{-1}\{\mathbf{z}\}$.*

Proof. Recall the relationship between the original Hopf variables $\{\mathbf{x}\}$ and $\{\mathbf{z}\} = \mathbf{T}(\Delta_{ij}, \rho_i)\{\mathbf{x}\}$ in Eq. (2.10). Since the function $\dot{\mathbf{T}}(\Delta_{ij}, \rho_i)$ is nonzero, Eq. (2.14) becomes

$$\{\dot{\mathbf{z}}\} + \mathbf{T}\dot{\mathbf{T}}^{-1}\{\mathbf{z}\} = \mathbf{T}\left[\mathbf{f}(\{\mathbf{x}\};\rho)\right] - k\mathbf{L}\{\mathbf{z}\} \tag{4.25}$$

Consequently, the virtual system in Eq. (2.18) becomes

$$\dot{\mathbf{y}} = \mathbf{V}^T\left[\mathbf{f}(\mathbf{V}\mathbf{y} + \mathbf{1}\mathbf{1}^T/n\{\mathbf{z}\};\rho_1)\right] - k\mathbf{V}^T\mathbf{L}\mathbf{V}\mathbf{y} + \epsilon(t) \tag{4.26}$$

where the error term $\epsilon(t)$ comes from the nonzero time-derivative of the $\mathbf{T}$ matrix since some Δ_{ij} is time-varying.

$$\epsilon(t) = -\mathbf{V}^T\mathbf{T}\dot{\mathbf{T}}^{-1}\{\mathbf{z}\} \tag{4.27}$$

Hence, although the $\mathbf{y}$ system in Eq. (4.26) is contracting, the Hopf oscillators do not perfectly synchronize because $\mathbf{y} = \mathbf{0}$ is no longer the particular solution. By robust contraction analysis [63], where $P_1(t)$ defines represents a desired system trajectory and $P_2(t)$the actual system trajectory in a disturbed flow field given in Eq. (2.18) with the error term. Also, consider the distance $R(t)$ between two trajectories $P_1(t)$ and $P_1(t)$ such that

$$\dot{R}(t) + \ell R(t) \leq \|\epsilon(t)\| \tag{4.28}$$

where $\ell > 0$ is the contraction rate of the virtual system Eq. (4.26) such that $\ell = k\lambda_{min}(\mathbf{V}^T(\mathbf{L} + \mathbf{L}^T)/2\mathbf{V})) - \lambda$. Hence, the synchronization error converges to the ball of the radius $\|\epsilon(t)\|\ /\ell$. □

If we select graph configuration A in Fig. 2.3, we have four available phase differences. We use two degrees of freedom as follows. We do not break symmetry in lead-lag phase differences, setting

$$\Delta_{75} - \Delta_{65} = \Delta_{31} - \Delta_{21}. \tag{4.29}$$

We now control $\Delta_{31} - \Delta_{21}$ with PD control and body pitch as the input. We use a low pass filter on the derivative input. Not only is this effective in stabilizing the longitudinal motion, the nominal value can be used to select ascent or decent angle. In order to control pitch symmetry breaking with one parameter, consider Δ as the constant nominal value (90°) of Δ_{21} and Δ_{65}. Then, set

$$\Delta_{65} = \Delta + \delta, \quad \Delta_{21} = \Delta - \delta. \tag{4.30}$$

The phase difference between flapping and pitch is vital for lift and thrust generation. Therefore, this difference between the right and left wings causes roll and proverse yaw. This method of symmetry breaking was observed by Hedrick and Biewener [61].

The second method of symmetry breaking we use is lead-lag amplitude. We set this proportional to yaw rate to provide yaw rate damping. Due to the coupled nature of the motions, this causes yaw rate and roll angle to go to zero.

4.2.5 Two Alternating Flight Modes for Altitude Control

Inspired by altitude stabilization of animal flight, we propose a switching logic between flapping mode and gliding mode. Requirements for switching use the current bifurcation parameter $\sigma \in (-1, 1)$ to determine what mode we are in, as well as altitude and velocity information to determine whether to switch mode. Recalling that

Table 4.1: Simulation parameters

$\rho_1 = \phi_{w,max}$=50°	$\rho_2 = \theta_{w,max}$=30°	$\rho_3 = \psi_{w,max}$=15°	m=0.3kg
$a_1 = a_5$=0	$a_2 = a_6$=0°	$a_3 = a_7$=-5°	$\mathbf{I}_b$= 0.0012*eye(3) kgm^2
k=50 or 0	λ= 10 or 50	c= 0.15 m	$c_{m0} = -0.5$
	R=0.32 m	Θ_s=20°	

the z-direction is positive downward, we set the test for gliding mode as

$$\begin{aligned} &\text{if } (\sigma = 1 \ \ \& \ \ z_b < -h_{\text{max,flap}} \ \ \& \ \ V_{bx} > V_{\text{x,max}}) \text{ or} \\ &\quad (\sigma = -1 \quad z_b < -h_{\text{min,glide}} \quad V_{bx} > V_{\text{x,min}}) \\ &\quad \text{then glide } (\sigma = -1) \text{ by the control law in Sec.4.2.2} \qquad (4.31) \\ &\text{else flap } (\sigma = 1) \text{ by the control law in Sec.4.2.3 and 4.2.4.} \end{aligned}$$

The switching logic ensures that we have sufficient altitude and forward velocity to glide, but will interrupt the constant ascending flapping flight with periods of gliding.

4.3 Simulation Results

The body states resulting from the control laws in Sec. 4.2.1 are shown in Fig. 4.5. The control parameters and the wing angles driven by the CPG network are shown in Fig. 4.6. The key simulation parameters are listed in Table 1 and are valid unless actively controlled in Sec. 4.2.1. The vehicle begins in a gliding mode, transitions to slowly ascending flapping flight, then executes a turn. Transition states are not well explored. As seen in the next paragraph, most transitions involve discontinuities which could be avoided in a more sophisticated control law. Most troublesome is the

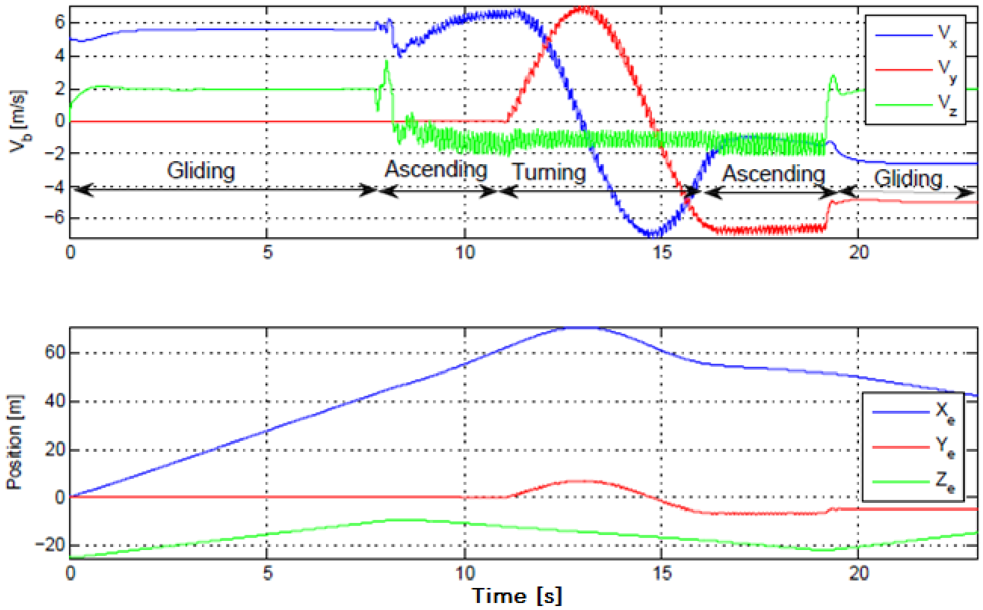


Figure 4.4: Velocity and position in the inertial frame of the two alternating flight modes, flapping and gliding.

transition from gliding to flapping, as the body angle of attack nears −90°. Remarkably, the flapping mode corrects this transitory mode and should be a testament to the potential of our overall scheme.

To turn, at the 10 second mark, we set $\delta = 3°$, accompanied by setting $\dot{\omega} = 0$, shifting the zero point and scaling the proportional feedback for lead-lag coupling, scaling the derivative feedback for lead-lag coupling, and turning off yaw damping due to lead-lag amplitude symmetry breaking. At the 15 second mark, we return $\delta = 0°$. What is interesting here is that we have symmetric flapping, and the vehicle settles into a nice banked turn. The bank angle, rate of turn, and qualitative characteristic (e.g. amplitude of body pitch oscillation) of the turn are interestingly linked to the scale and shift of the lead-lag coupling, but an exact correlation is not yet known. This must be further investigated and understood in order to implement a better nonlinear control law. Finally, at the 20 second mark, we return the original scaling to

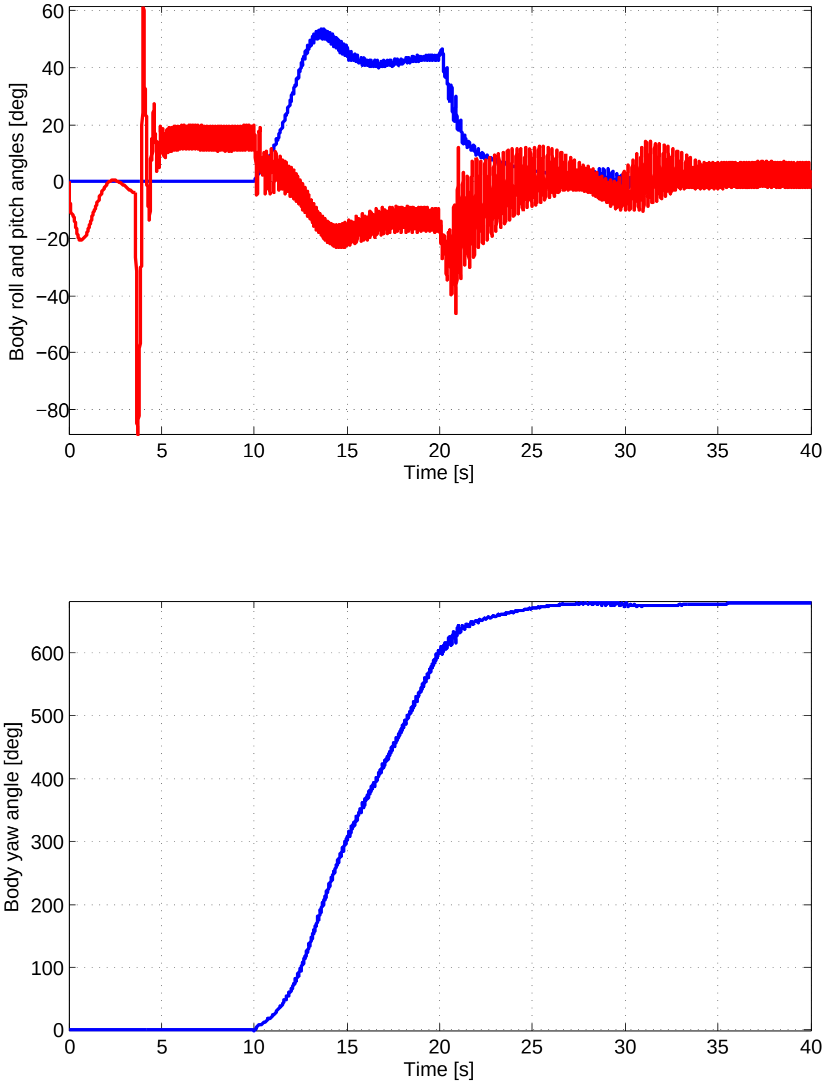
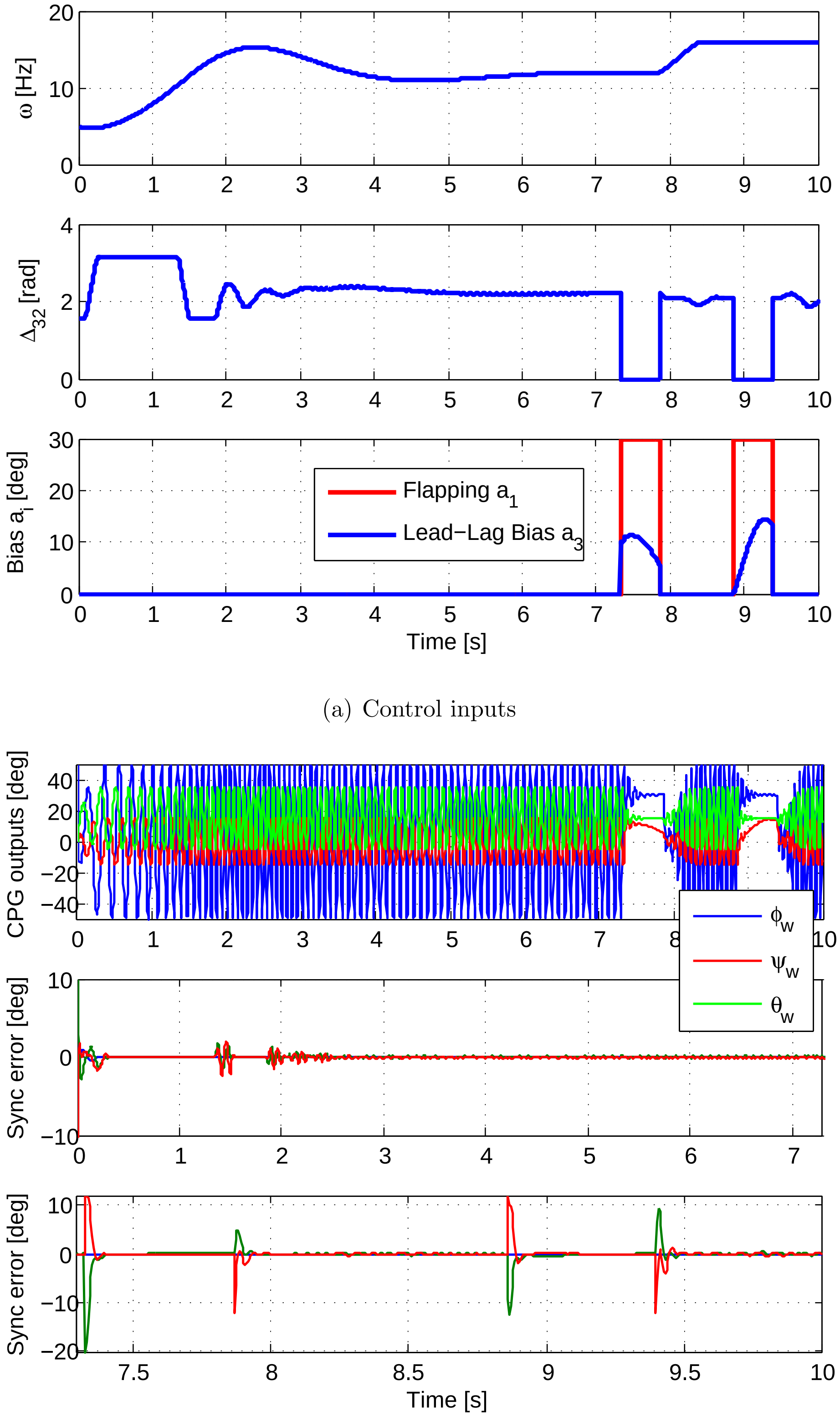


Figure 4.5: Euler Angles of the two alternating flight modes, flapping and gliding. In the top figure, the red curve is the pitch angle and the blue curve is the roll angle.



(a) Control inputs

(b) CPG array output and synchronization errors

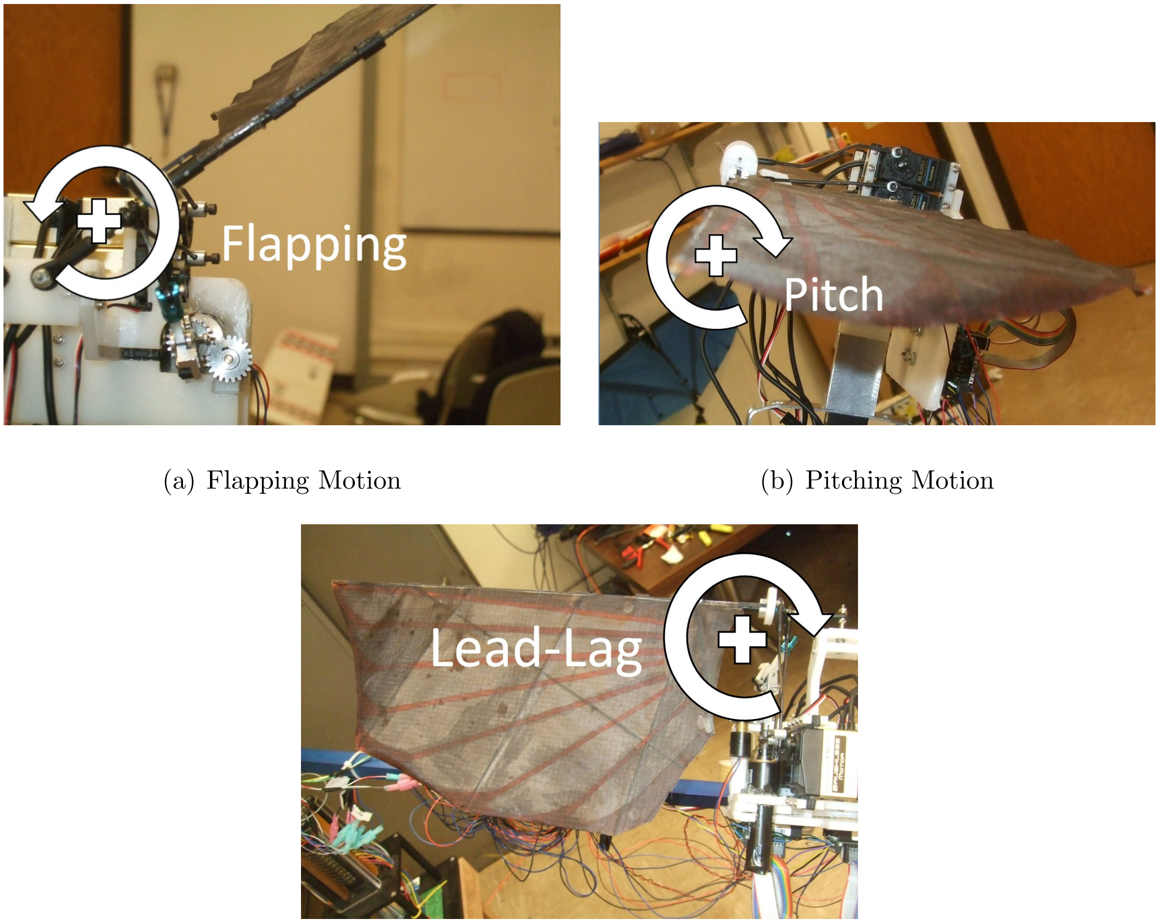
Figure 4.6: State vectors of the two alternating flight modes, flapping and gliding.

proportional feedback of lead-lag coupling, but select a new zero point for proportional feedback and also a new scaling for derivative feedback. At this point, we also turn on yaw damping by lead-lag amplitude symmetry breaking. The roll angle through the turn is about 40° and the average global yaw rate is between 90 and $100°/sec$. This is feasible in light of experimentation [61].

Figure 4.6b shows the resulting oscillatory behavior of the flapping (ϕ_w), pitch (θ_w), and lead-lag motion (ψ_w) commanded by the CPG network and highlights the effects of our changing control variables on CPG behavior. From arbitrary initial conditions, the CPG network synchronizes globally and exponentially, indicated by the synchronization errors defined as the first element of $(\mathbf{x}_i - R(\Delta_{ij})\rho_i/\rho_j\mathbf{x}_j)$– see Fig. 4.6b. When the phase difference Δ_{ij} is time-varying, there is a small residual error in the synchronization ($\leq \pm 0.2°$), but still effectively small due to Corollary 3. Otherwise, the synchronization errors tend exponentially to zero as predicted by Theorem 2.

4.4 Robobat Experiments

RoboBat is a highly controllable platform, modeled after the kinematics of a bat. Eight degrees of freedom are provided; three in each shoulder joint and two for the amplitude of flapping. Shoulder joints are also analogous to human shoulder joints, able to move forward, backwards, up, down, and can twist in both directions. Those motions correspond to lead-lag, flapping, and pitching respectively. These 8 degrees of freedom are combined with variable speed motors to allow for maximum flapping in control schemes. The flapping motion of the wings are independently powered by two 5 watt Maxon motors. Electronic controllers for the two Maxon motors allow for precise control of motor velocity and thus flapping frequency. All other degrees



(a) Flapping Motion

(b) Pitching Motion

(c) Lead-Lag Motion

Figure 4.7: Positive wing motion directions

of freedom are controlled with Futaba servos. Two US Digital absolute encoders are attached to the sides of the two motors and connected with gears in order to measure the absolute position of the wings. This position data is used to create a closed loop controller for the wings and allows them to synchronize to a desired signal from the CPG's.

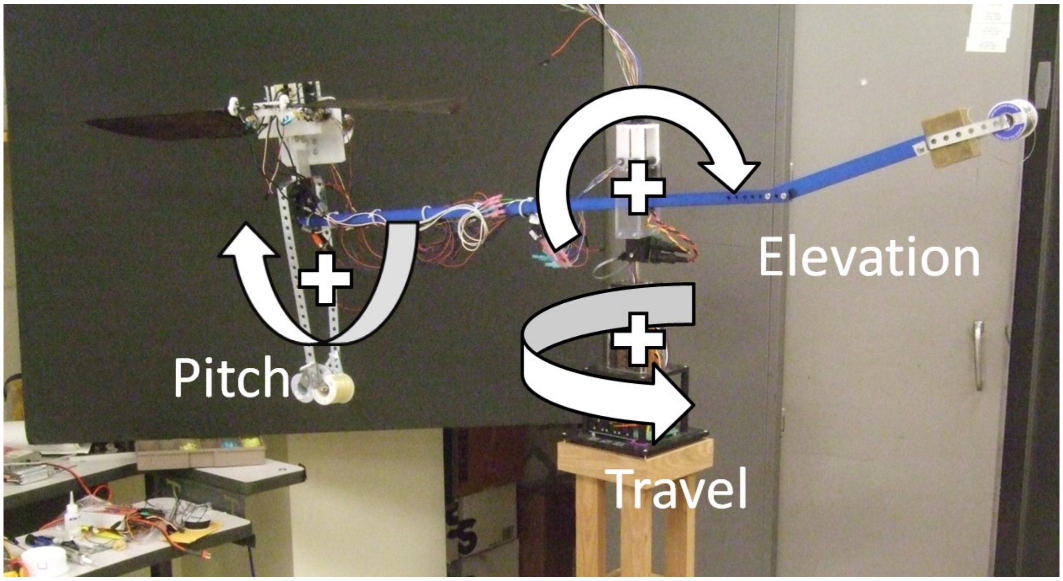


Figure 4.8: RoboBat Testbed

4.4.1 Kinematics of Pendulum System

We present the dynamic model of the current RobotBat, which is not intended to be a free flying platform. It is intended as a testbed for CPG control designs, experimental confirmation of unsteady aerodynamics, and experimental determination of optimal wing motions. The weight and power requirements have not been optimized for free flight. In order to test longitudinal control strategies, it has been attached to a Quanser pendulum platform, which provides encoder feedback signals that we can use for control. Figure 4.8 shows the three degrees of freedom: travel, elevation, and pitch (λ,ϵ,θ). For more information on the testbed, see [82].

Of note is the fact that the pitch rotation point is not near the center of gravity of the bat. To make experimentation feasible, we have affixed a counterweight on the pitch arm. By moving this counterweight or changing its mass, we can alter the natural stability of the pitch motion. One consequence of this scheme is that the pitch motion has an artificially high moment of inertia. Therefore, we expect that our moment-producing control schemes for a tailless vehicle will have even more effectiveness in a free flier. To move toward computations of actual forces and moments generated, we desire dynamic modeling of the pendulum set-up and the unsteady aerodynamics. If we define our generalized coordinates to be $[q_1, q_2, q_3] = [\epsilon, \theta, \lambda]$, then using Lagrange's equations, $\frac{d}{dt}\frac{\partial L(q,\dot{q})}{\partial \dot{q}} - \frac{\partial L(q,\dot{q})}{\partial q} = F$ and algebraic manipulations, we can transform the EOM to standard robot form [54].

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau. \tag{4.32}$$

The forces and moments on the right hand side can be found as a function of wing kinematics and an aerodynamic model, The generalized forces remain intact through the transformation to robot form, i.e., $F = \tau$, and are computed to be where L,T, and M are found later in the aerodynamic model. This formulation can then be applied to a free-flying MAV. Here, we present a refinement on the aerodynamic model of [79].

The pendulum rig consists of

1. A solid bar with mass M_p hinged at its center of gravity such that it can spin about the vertical axis (angle given by λ, positive counter-clockwise) and rotate upwards and downwards in the vertical plane (angle denoted by ϵ, positive downwards).

2. A compound pendulum mounted on one end of the bar consisting of two point masses: the robotic bat itself modelled as a point mass m_b and a variable mass m. The compound pendulum is free to swing in the plane normal to the bar, with the swing angle given by θ.

3. A counter-weight, m_w, located at the opposite end of the bar as the bat.

Three frames of reference can be defined for this system, given an inertial frame of reference I fixed to the Earth:

1. A frame B fixed to the compound pendulum with its origin at the suspension point. The frame B parallel to the aircraft body axis frame centered at the aircraft CG.

2. A frame P with its origin at the bar's hinge point such that under nominal conditions, the axes of P and B are parallel to each other.

3. A frame S constructed locally at every wing station for calculation the local wind velocity and the aerodynamic forces and moments.

The frame I is first rotated about the z-axis by an angle λ, followed by a rotation about the x-axis by ϵ to coincide with the P frame. Therefore, the following rotation matrix is obtained to transform the components of a vector from I to P:

$$\begin{aligned} R_{PI} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\epsilon & -\sin\epsilon \\ 0 & \sin\epsilon & \cos\epsilon \end{bmatrix} \begin{bmatrix} \cos\lambda & -\sin\lambda & 0 \\ \sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\lambda & -\sin\lambda & 0 \\ \cos\epsilon\sin\lambda & \cos\epsilon\cos\lambda & -\sin\epsilon \\ \sin\epsilon\sin\lambda & \sin\epsilon\cos\lambda & \cos\epsilon \end{bmatrix} \end{aligned} \tag{4.33}$$

The frame P is rotated about the y-axis to obtain frame B:

$$R_{BP} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{4.34}$$

The rotation matrix from B to S was covered in Section 4.1.

4.4.2 Open Loop Experiments

Experiments are conducted by commanding flapping frequency and phase differences while observing the bat's orientation and velocities from the encoder data. Using dSPACE's ControlDesk software, a GUI is created for direct interaction with the real time controller of the bat. Control variables can be changed and plotted in real time, and data is captured and saved to a MATLAB binary file.

As mentioned in Section 4.4.1, there is an offset between the center of the bat and the pitch rotational point. This creates a coupling between the dynamics of the second pendulum with the longitudinal dynamics of the bat. While we use this to our advantage to obtain stability states desired for testing phase difference control, it necessarily creates a large rotational moment of inertia that is many times that of an actual bat. Therefore, we expect the pitching moments from phase difference control to have less effect in this experimentation than in free flight of a low moment of inertia bat. Regardless, we hope to see pitch control via only flapping/lead-lag phase difference even in this set-up.

First, we consider steady-state behavior of pitching motion with respect to phase differences. Most variables were held constant and are given in Table 4.2. In the first experiment, the phase difference between flapping and lead-lag, Δ_{31}, was varied between 140 deg and 240 deg twice. The system was allowed to converge to a non-equilibrium steady state. Encoder data was captured for 20 seconds. Figure 4.9 shows the minimum, maximum, and average value over the 20 second period. As expected, between 180 and 240 deg, the forward velocity and elevation curves look very flat, but the pitch angle increased between 6 and 8 degrees. This corresponds with the idea that lift and thrust generation remained similar while only a control moment was created. It is postulated that a free flight system with a low pitching moment of

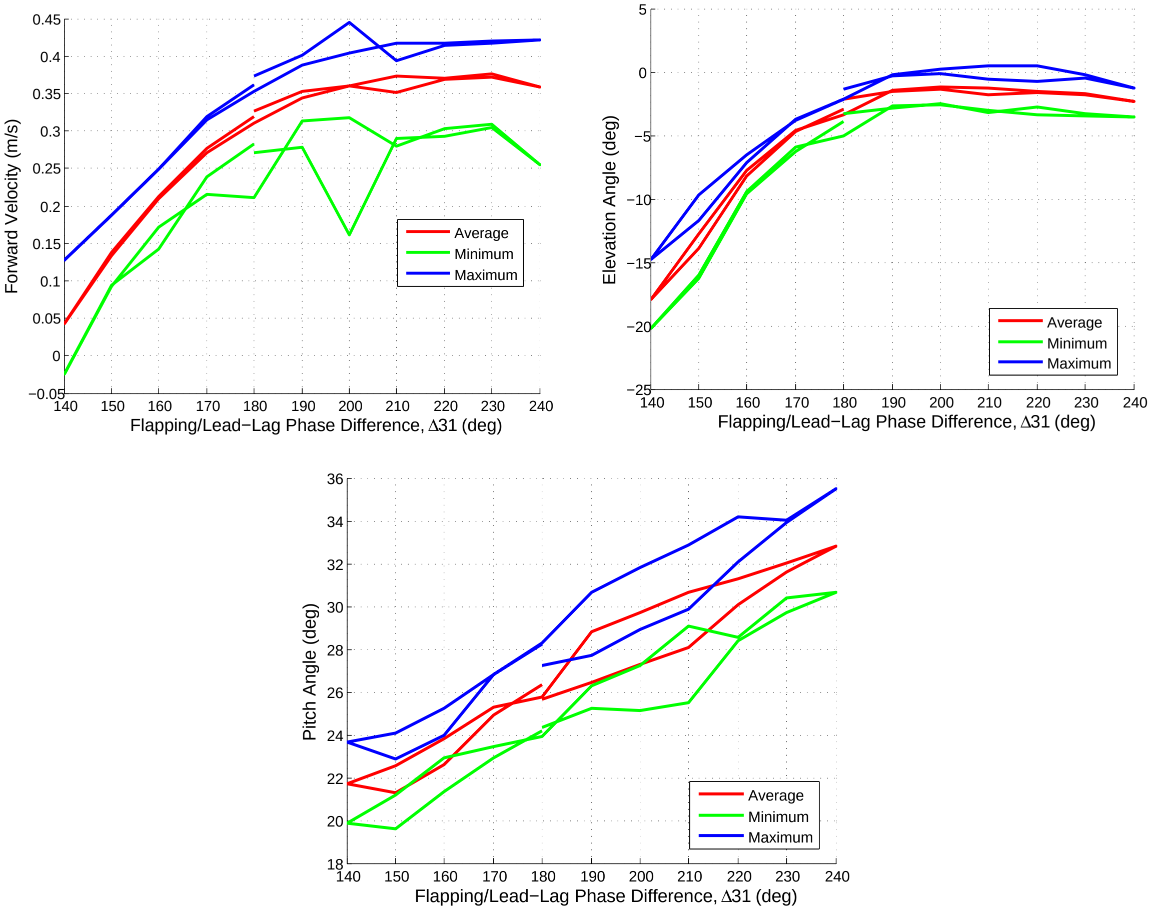


Figure 4.9: Open-loop pitch control via phase differences (2 Hz).

inertia would see much stronger pitching effects from such control. This is supported by physical intuition and the numerical simulations performed previously [79]. The lower range of phase differences, 140-180 deg, saw a large dropoff of thrust and lift generation. In fact, the bat came to a complete stop at one point with the phase difference at 140 deg. Therefore, we should not plan to use this range in control of flapping flight.

The second experiment repeated the same process at 2.5 Hz instead of 2 Hz. Only one sweep through the values was performed. The results are plotted in figure 4.10 and support the same conclusion as the first experiment. Additionally, they

	Flap Freq	Δ_{21}	Δ_{31}
Experiment 1	2 Hz	90 deg	Varied
Experiment 2	2.5 Hz	90 deg	Varied

Table 4.2: Experiment Parameters

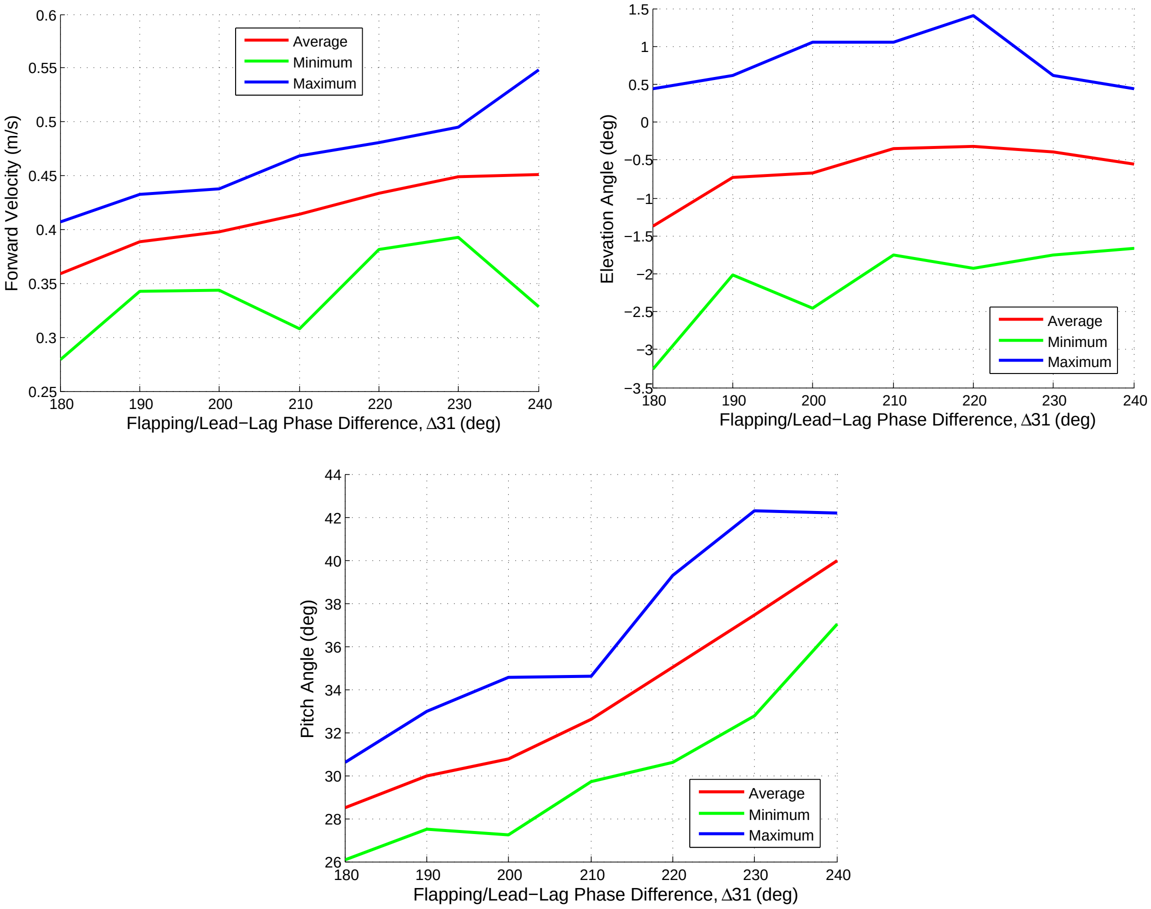


Figure 4.10: Open-loop pitch control via phase differences (2.5 Hz).

preliminarily confirm the postulate that flapping frequency can be used as strong control of forward velocity and altitude. Finally, note that we are not concerned about the fact that all the relevant body pitch angles are all around 20-40 degrees. Adjustment of CG location can set the trim state as desired while control moments

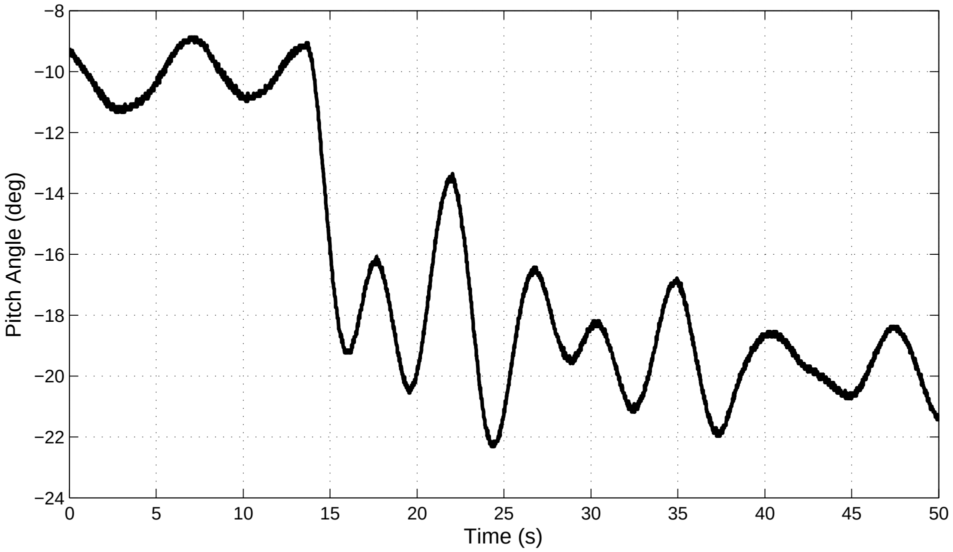


Figure 4.11: Experimental Results of Pitch Control at 2.5 Hz

are created from phase differences.

4.4.3 Closed Loop Experiments

We turn our attention to demonstrating that the CPG structure allows very simple top-level controllers to provide stability and control in closed loop.

Simple symmetric PID controllers were used for all experiments,

$$\Delta_{31} = \Delta_{75} = -5(\theta - \theta_d) - 0.5\dot{\theta} - 0.1 \int_0^t (\theta - \theta_d)\, dt, \tag{4.35}$$

where θ is the pitch angle and $\dot{\theta}$ is computed using the derivative filter, $\frac{\omega_{cf}^2 s}{s^2 + 2\zeta_f \omega_{cf} s + \omega_{cf}^2}$, with $\omega_{cf} = 40\pi$ and $\zeta_f = 0.9$. The saturation values were set so $\Delta_{31} \in [180°, 270°]$. Even though we are able to use simple PID controllers in the top level, the overall controller is very nonlinear due to the CPGs in Equation (2.4).

We begin experimentation at an open loop flapping frequency of 2.5 Hz. At this frequency, the open loop appeared stable. Figure 4.11 shows the response to

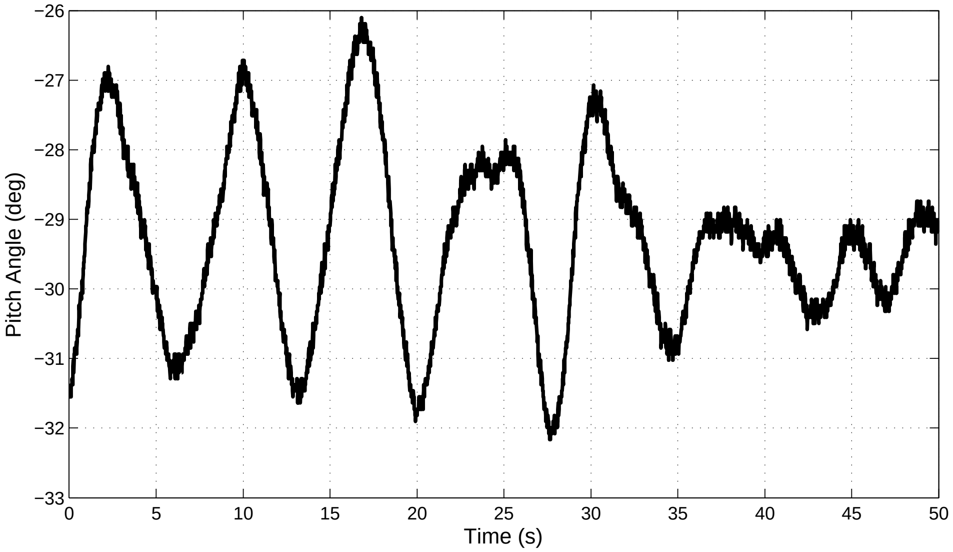


Figure 4.12: Experimental Results of Pitch Stability by Control at 3Hz. The system starts in open loop, which is unstable. At approximately 22 seconds, the closed-loop controller was turned on, stabilizing the system.

a change in desired body pitch from $-10°$ to $-20°$. Two notes should be kept in mind. First, the actual value of body pitch is affected by the precise position of the pitch counterweight. Therefore, it is not worrisome that the values are not exactly around zero or some other intuitively desired value. Second, at 2.5 Hz, the apparent maximum change of body pitch due to open loop control of Δ_{31} is around $10 - 12°$. This experiment demands a change of 10° and experiences saturation problems as it nears the final desired state.

Moving the frequency to 3 Hz caused instability in the open loop. Figure 4.12 shows that by activating the PID control of Δ_{31}, we can stabilize the unstable system. At this frequency, we also have appreciably more control authority. Figure 4.13 shows a commanded pitch change of 15°, which is easily obtained. We expect that at speeds typical of bat flight (2-3 m/s with frequencies of 7-10 Hz [21]) and pitch moment of inertias not inflated by the pendulum setup we will see even more control effectiveness.

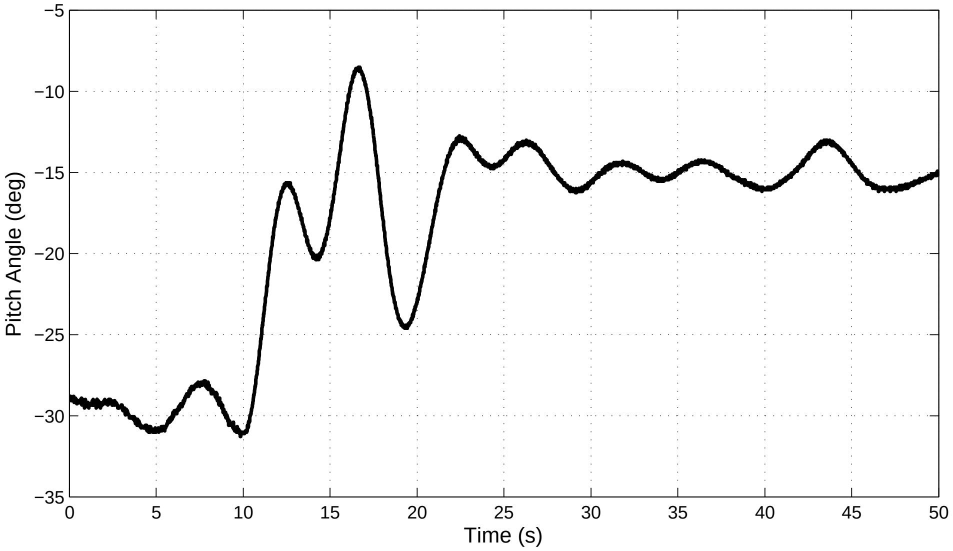


Figure 4.13: Experimental Results of Pitch Control at 3Hz

The numerical results in Section 4.3 support the idea that this control effectiveness will be higher.

Finally, we would like to feed back velocity into flapping frequency and demonstrate flight mode changes. Velocity control is achieved using a similarly simple PID controller on top of the CPG network. Importantly, RoboBat was able to transition to a glide after simply flipping the sign of the bifurcation parameter σ in Eq. 2.1. After dissipating energy in the glide, the transition back to flapping was again as simple as flipping the sign. Experimental results are found in Fig. 4.14. During the flapping phase, both pitch and velocity controllers are active (desired velocity: 0.5m/s, desired pitch: 35 degrees). During the glide phase, notice that rapid inhibition causes the CPG outputs to go to zero (the plotted CPG output is the normalized value of the flapping angle). At this time, the frequency and phase difference values are not meaningful, as the CPGs exhibit exponentially stable fixed point dynamics. During such a glide, control laws like those proposed in [83] could easily be used to control the bias signals of the Hopf oscillator network.

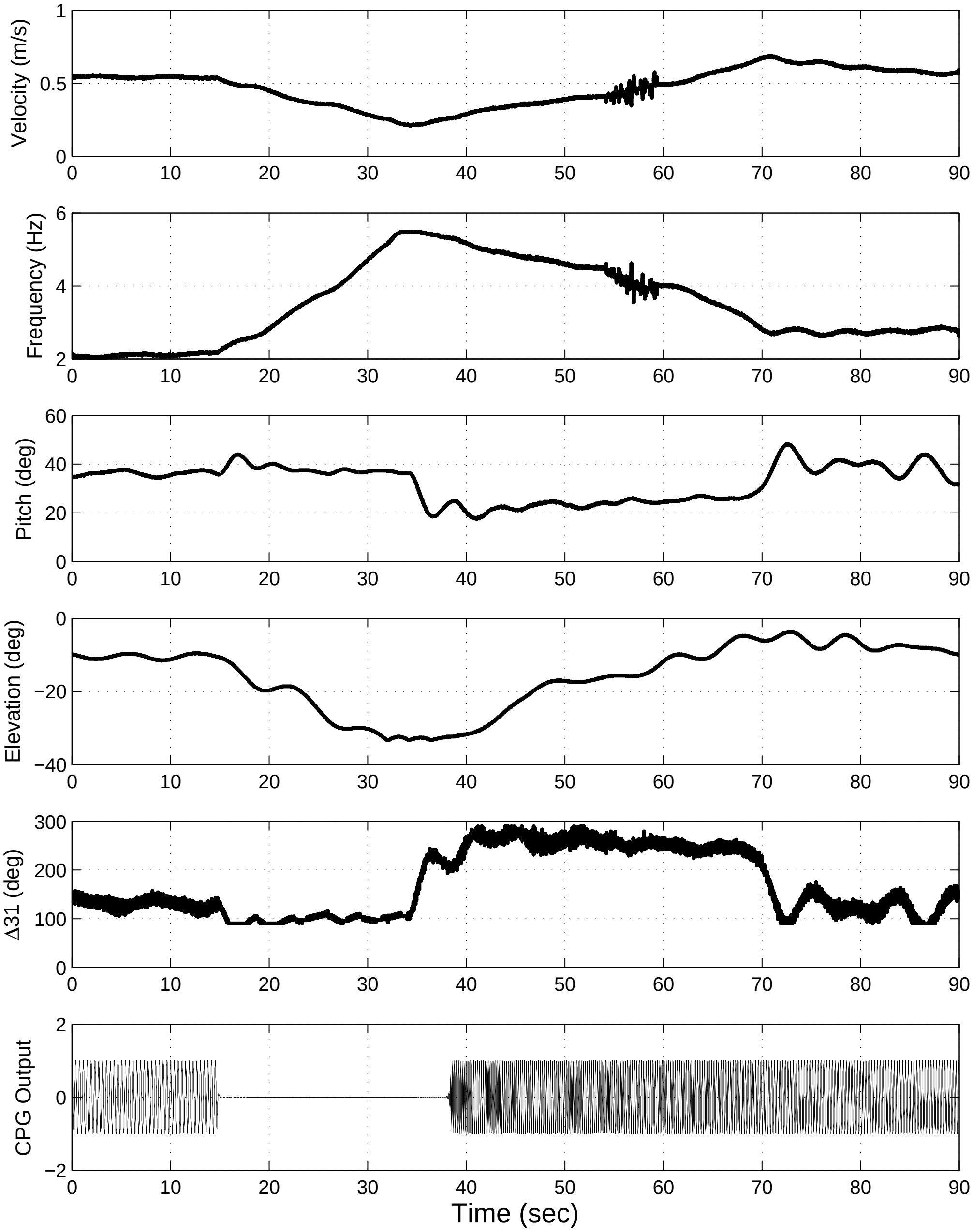


Figure 4.14: Experimental Results of Flight Mode Change with Pitch and Velocity Control

4.5 Chapter Summary

In order to show the effectiveness of the proposed CPG-based flapping flight control, we presented numerical simulation and experimentation by using a realistic vehicle model with three dimensional wing kinematics. The dynamic model and wing kinematic model of flapping flight represent the more complete and complex mechanical flapping system developed for wind tunnel testing. This dynamic model includes a tilted stroke plane angle, the lead-lag motion, and the relative body velocity, in addition to the flapping and pitch angles of each wing. We also showed that CPG-based flight control can stabilize and alternate two different flight modes of flapping and gliding flight by using synchronized and symmetry-breaking (phase difference) oscillatory motions of two main wings. This result is interesting in the sense that tailless flapping flight dynamics could be effectively controlled without using aerodynamic control surfaces. This result agrees with a prior claim of biologists that birds acts more like tailless aircraft.

Further, the 8-DOF robotic bat introduced in this chapter provides a useful contribution to the state of the art. Previous models have been very limited by mechanical simplicity to generating only sinusoidal waveforms. Our model allows not only allows experiments involving motions not used before such as lead-lag and pitch for each wing, but also for independent control of flapping frequency and amplitude. From the data collected by this testing, we were able to show that CPGs allow for the smooth transitions of control parameters, as well as generating useful wing trajectories.

We demonstrated the ability to stabilize and control longitudinal motions via CPGs with the RoboBat. As expected, the top-level controllers are of low dimension and can be made simply, because most of the hard work is done by the CPGs. Given the extent of mechanical coupling in the design, it is remarkable that such control was immediately as effective as it was. As mechanical design of actuators develops, we expect robotic fliers in free flight to be able to utilize the key feature of phase synchronization and control of phase differences in stability and control of body motions. The major problem of identifying a method of proving such stability analytically is still open. However, this chatper has demonstrated the result experimentally.

Chapter 5

Spiking Neuron Circuit for Simple Controller

5.1 Neuron and Synapse Models

Neuroscience has not yet solved the problems of cognition, memory, or information processing [84]. It has, however, discovered some potential tools which can be exploited for such purposes and highlighted some simple neural circuits which display interesting, complicated behaviors. We want to preserve as many of these features as possible, even if we don't yet know how to fully exploit them.

The two most basic building blocks of a neural network are neurons and synapses. Biological neurons are cells which have a membrane with low inherent ion permeability. A combination of ion pumps and channels regulates the concentration of ions inside and outside the membrane. An electric potential across the membrane is determined by these concentrations. The time history of ion concentrations results in a time history of the neuron's voltage. Ion channels respond to this voltage as well as chemical signals from synapses, opening to selectively allow ions to diffuse through the membrane. Different types of neurons have different sets of relevant ions and a variety of possible ion channel behaviors. The most basic of these responses is the action potential, first discovered as a fast inward sodium current followed by a slower outward potassium current. The resulting voltage behavior looks like a spike followed by a refractory period.

Mathematical models for neuron-like objects are abundant [85]. Some, like perceptron or integrate-and-fire models, have been developed primarily from simple computational or theoretic principles, without regard to biological realism. Others, like Hodgkin and Huxley's squid axon model, have a high degree of biological realism (modeling actual ion currents), but they can introduce computational complexity (e.g., Hodgkin-Huxley is a fourth-order differential equation). We chose to follow Izhikevich's lead in selecting a reduced-dimension model which retains the ability to produce much of the same *qualitative* behavior as biological neurons [86].

Neurons display very complex behaviors [87]. While we don't know exactly which qualitative behaviors are actually encoding information in biological systems, our perspective is that bursting behavior and spike latency are important for even very simple behavioral networks. For example, consider the classic half-cell oscillator [88]. It is the simplest known method of constructing a biologically-realistic central pattern generator (as opposed to the mathematical abstraction introduced in Chapter 2, which is believed to be necessary for a multitude of repeated, oscillatory behaviors exhibited by nearly every higher animal. Besides reciprocal inhibition, this network requires spike frequency adaptation (due to spike latency) and post-inhibitory rebound. Izhikevich's model may be the simplest model which exhibits these types of behavior [86].

For a network of multiple neuron, the mathematical form of Izhikevich's model for ith neuron is

$$\begin{aligned} C_i\dot{v_i} &= k_i(v_i - v_{r,i})(v_i - v_{t,i}) - u_i + I_i \qquad (5.1)\\ \dot{u_i} &= a_i(b_i(v_i - v_{r,i}) - u_i), \end{aligned}$$

with after-spike resetting

$$\text{if } v_i \geq +30mV, \text{ then} \begin{cases} v_i \leftarrow c_i \\ u_i \leftarrow u_i + d_i \end{cases} \qquad (5.2)$$

is a modified quadratic-and-fire model [33]. The main variable, v, is a voltage across the neuron membrane. There is a single slow variable, u, which may be capable of representing two or three known ion currents inherent in the neuron's internal dynamics. C and a are the major timescales, while c and d determine the reset (and beginning of the refractory period). Additional constants are v_r and v_t, the resting potential and threshold potential, respectively. The input current, I, is general and can incorporate multiple currents coming into the neuron. We will use the notation I_i to represent the sum of all currents going into the ith neuron. In biological experiments, a patch clamp can inject a current. Synapses or sensors (e.g., stretch receptors) can also produce a current in a neuron. These currents can cause an equilibrium bifurcation, initiating a burst. During the burst, the slow variable produces spike frequency adaptation. Finally, a limit cycle bifurcation can cause the burst to cease.

Though we've determined that this model is useful for our purposes, questions remain about the details of its dynamics. It's not clear whether the discrete reset is adding complexity or is benign according to Poincarè's construction of dynamics near a limit cycle. For example, Izhikevich claims the model can produce chaos [86], but it's not clear whether this is behavior due solely to parameter selection or if chaos is injected through the forcing function.

Our selection of Izhikevich's model was also greatly influenced by the fact that we can approximate its behavior with simple structures in complimentary metal-oxide semiconductor (CMOS) hardware [89]. Ongoing work is being done to simulate these electrical circuits in NGSPICE and prepare our ability to procure chips which reproduce neuron networks in terms of physical voltages and currents.

Synapses provide signaling between neurons. When an action potential reaches a synapse, it triggers vesicle fusion and release of quantized amounts of neurotransmitter into the synaptic cleft. On incredibly short timescales, the neurotransmitter diffuses across the cleft and binds to receptors on the post-synaptic neuron. These receptors trigger the opening of ion channels, allowing a current to flow in the post-synaptic neuron.

Like neurons, there is no shortage of mathematical models for synapse behavior [34, 90, 91]. Our selection criteria were driven by control theoretic concerns (namely delay and attenuation, detailed in Section 5.3) rather than biological realism. It is likewise unknown which features are most important for general synaptic coding, so future work may desire a model which reproduces quantized vesicle release more accurately. In the model we selected (by [34]), a sigmoid function governs the turning on of a synapse according to

$$S_{\infty,ij} = \begin{cases} \tanh((v_i - v_{th,i})/v_{slope,i}) & \text{for } v_i > v_{th,i} \\ 0 & \text{otherwise,} \end{cases} \tag{5.3}$$

where i is the neuron turning on the synapse and j is the neuron being influenced by the synapse, $v_{th,i}$ is the synapse threshold and $v_{slope,i}$ defines the sharpness of the smoothed activation function. Then, the percentage of usable neurotransmitter available in the cleft is described by

$$\frac{dS_{ij}}{dt} = \frac{S_{\infty,ij} - S_{ij}}{\tau_{ij}(1 - S_{\infty,ij})}, \tag{5.4}$$

having intrinsic activation timescale τ_{ij}. Finally, the current produced in the post-synaptic neuron is

$$I_{ij} = g_{ij}(t)S_{ij}(t)(v_j(t) - v_{rev,j}), \tag{5.5}$$

where g_{ij} is the synaptic conductance, $v_{rev,j}$ is the reversal voltage, and v_j is from Equation 5.1. Note that I_{ij} may be only one component of the overall input current to the jth neuron, I_j. Future work in self-tuning or learning networks can incorporate synaptic plasticity - an increase or decrease in g_{ij} according to the time-history of spiking among the synaptically connected neurons. For the remainder of this work, we will assume constant synapse conductance (i.e., g_{ij} is constant).

We believe these structures will provide us unparalleled design flexibility. We also expect these structures to be implementable in CMOS with a low-power footprint. When combined with memristor technology (a variable-resistance resistor to provide variable synapse weights), the potential is tantalizing.

5.2 Antagonistic Motoneurons

While we can reproduce neuronal/synaptic behavior in hardware, a robotics engineer has few options for biologically-inspired sensors or actuators. Instead, electromechanical devices still dominate the landscape. We must be able to incorporate inputs from them and encode outputs which are useful to them. This process will likely lack some degree of biological inspiration, due not only to the difference in devices, but also because models of biological devices are not yet fully known [92]. We will need to make some simplifying assumptions which may be relaxed in future work.

Our primary assumption will be antagonistic motoneurons. Biological motoneurons are antagonistic, but they also have antagonistic sensors/actuators (i.e., a biceps/triceps motoneuron pair has two stretch receptors and two muscle groups; even this is a simplification). One interesting consequence of biological antagonistic pairs is that coactivation can increase joint rigidity. Perhaps a clever mechanical design can provide this feature in robotic applications in the future.

Instead, we assume that the output of antagonistic motoneurons are merely summed (with a sign change indicating direction) before sent to a mechanical actuator. This is sensible from an engineering standpoint, as the motoneuron outputs are merely currents on a chip, which are easy enough to sum.

Likewise, sensing in biological systems is complicated. Even ignoring the many layers of potential feedback (distributed cutaneous sensors, equilibrioception, etc.), antagonistic muscle pairs have corresponding antagonistic low-level feedback in the form of stretch receptor pairs. Instead, rotary encoders are the preferred sensor for joint angle in robotics. We assume, for now, that these stretch receptor signals will be additive inverses of one another (i.e., if the biceps stretch receptor signals a stretch, the triceps stretch receptor signals a contraction of equal magnitude). Therefore, we merely send the error signal directly to one motoneuron while sending its additive inverse to the opposing motoneuron.

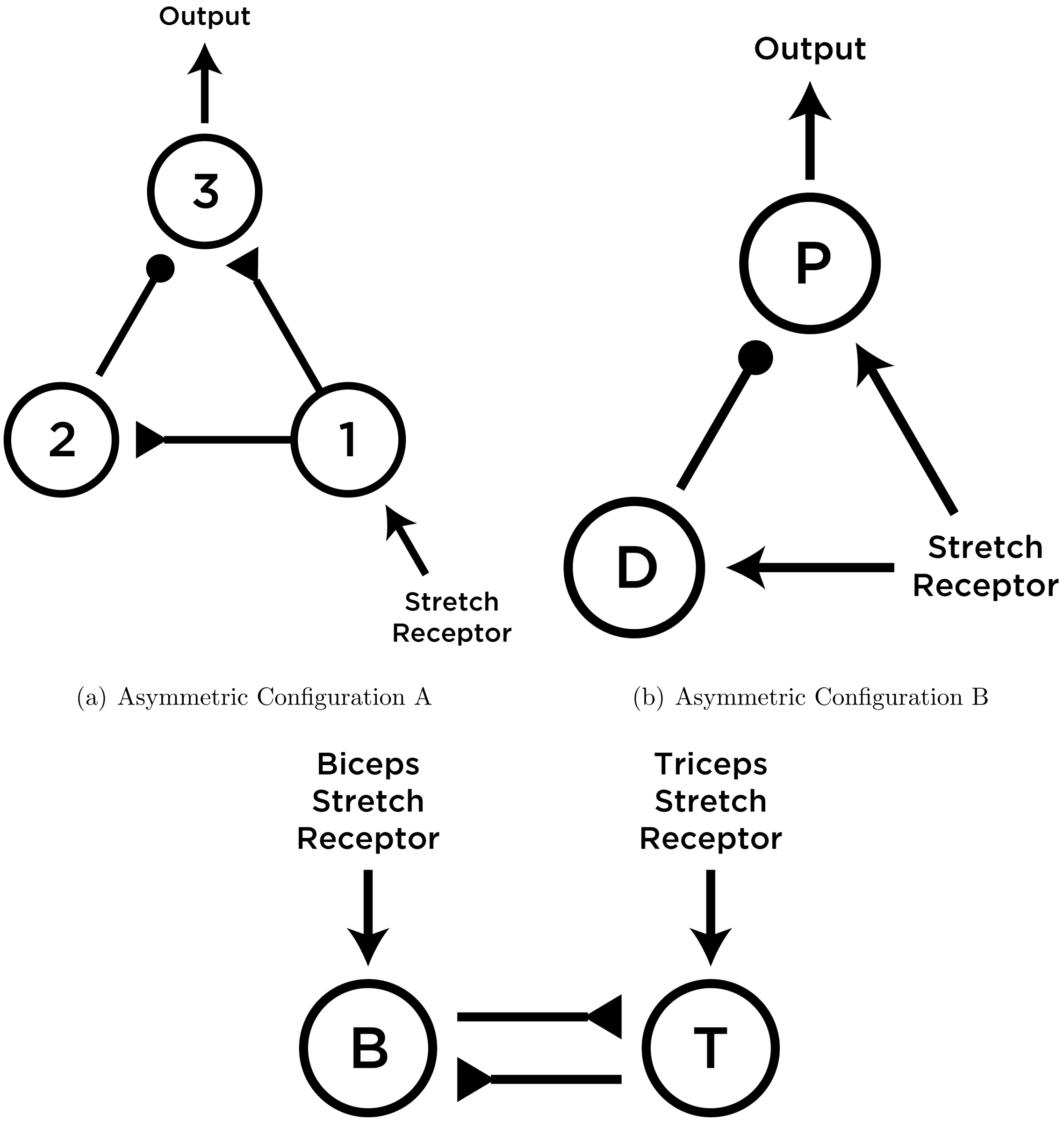
5.3 Simple PD-like Controller

Our goal for the remainder of this chapter will be to construct a neuron controller that behaves substantially like a classical PD (proportional-derivative) controller. Proportional control is simple with antagonistic motoneurons. The challenge is implementing derivative control that is tuneable, yet minimal. The three fundamental aspects of derivative control are delay, attenuation, and sign change. We'll discuss three possible networks, which are shown in Figure 5.1

Configuration A utilizes non-antagonistic motoneurons. Instead, it uses three neurons - one for inputting the error signal into the network, a direct path to the motorneuron via an excitatory synapse, and an indirect path to the motoneuron (here, Neuron 3). The indirect path has one interneuron (Neuron 2), which has an excitatory input synapse from the error neuron (Neuron 1) and which inhibits the motoneuron. The interneuron provides delay and attenuation, while the inhibitory synapse provides a sign change.

It's easy to see that this method has little need for an input neuron. Configuration B feeds the error signal directly into the motoneuron and the interneuron. This is likely the simplest model for controlling a single error signal in a fashion that is PD-like. A downside to these two schemes is that they are not symmetric. Notice that errors in one direction will elicit strong spiking behavior and activate derivative control through the inhibitory synapse, but errors in the other direction will not. We could consider two copies of this network arranged in an antagonistic fashion; this would require four neurons and two synapses.

Instead, consider a single excitatory synapse between two neurons. Figure 5.2 shows an output from such a simple system. The blue neuron is exited tonically by a step input. It triggers an excitatory synaptic response in the green neuron. Delay



(a) Asymmetric Configuration A

(b) Asymmetric Configuration B

(c) Symmetric Configuration C

Figure 5.1: Schematics for three PD-like neuron networks. Triangular connections represent excitatory synapses while solid circles represent inhibitory synapses.

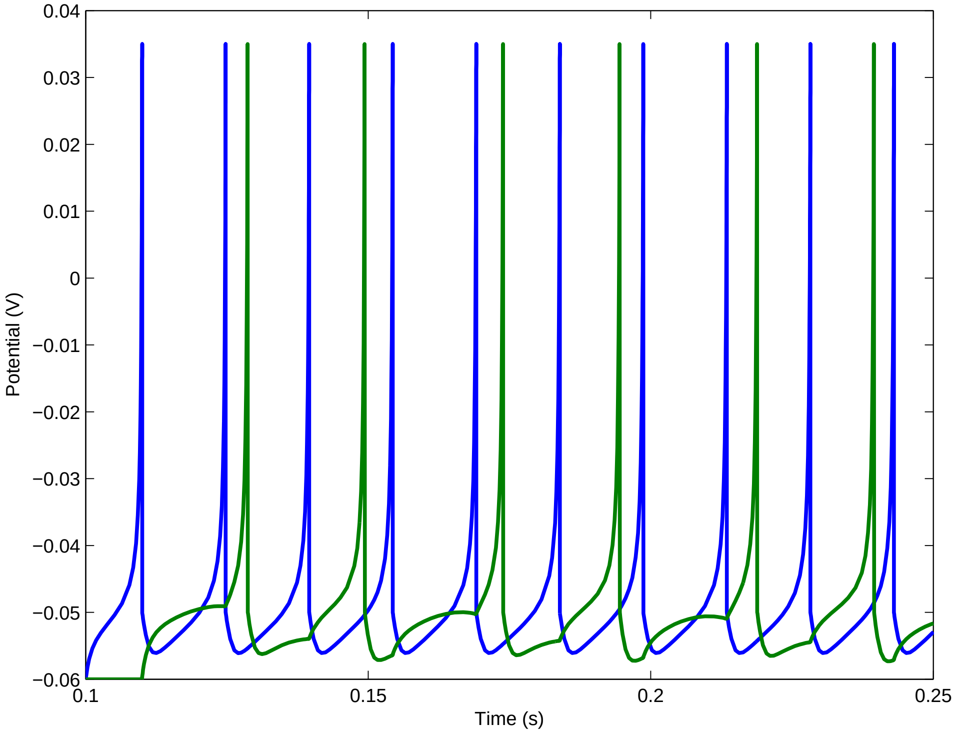


Figure 5.2: One excitatory synapse in isolation.

can be seen in the timing between the first spike for each signal. Attenuation can be understood as the increased distance between spikes in the green neuron.

This leads us to propose an antagonistic two-neuron network with reciprocal excitation, Configuration C. The synapse dynamics provide delay and attenuation (see Figure 5.2), while the antagonistic nature provides the sign change. The collected equations are

$$
\begin{aligned}
C_1\dot{v_1} &= k_1(v_1 - v_{r,1})(v_1 - v_{t,1}) - u_1 + I_1 \\
C_2\dot{v_2} &= k_2(v_2 - v_{r,2})(v_2 - v_{t,2}) - u_2 + I_2 \\
\dot{u_1} &= a_1(b_1(v_1 - v_{r,1}) - u_1) \\
\dot{u_2} &= a_2(b_2(v_2 - v_{r,2}) - u_2) \\
\text{if } v_1 \geq +30mV, &\text{ then } \begin{cases} v_1 \leftarrow c_1 \\ u_1 \leftarrow u_1 + d_1 \end{cases} \\
\text{if } v_2 \geq +30mV, &\text{ then } \begin{cases} v_2 \leftarrow c_2 \\ u_2 \leftarrow u_2 + d_2 \end{cases} \\
S_{\infty,12} &= \begin{cases} \tanh((v_1 - v_{th,1})/v_{slope,1}) & \text{for } v_1 > v_{th,1} \\ 0 & \text{otherwise,} \end{cases} \\
S_{\infty,21} &= \begin{cases} \tanh((v_2 - v_{th,2})/v_{slope,2}) & \text{for } v_2 > v_{th,2} \\ 0 & \text{otherwise,} \end{cases} \\
\frac{dS_{12}}{dt} &= \frac{S_{\infty,12} - S_{12}}{\tau_{12}(1 - S_{\infty,12})} \\
\frac{dS_{21}}{dt} &= \frac{S_{\infty,21} - S_{21}}{\tau_{21}(1 - S_{\infty,21})} \\
I_1 &= I_{21} + I_{stretch} = g_{21}S_{21}(t)(v_1(t) - v_{rev,1}) + (\theta - \theta_d) \\
I_2 &= I_{12} - I_{stretch} = g_{12}S_{12}(t)(v_2(t) - v_{rev,2}) + (\theta_d - \theta),
\end{aligned}
\tag{5.6}
$$

where $\theta - \theta_d$ represents the input error (denoted $I_{stretch}$ to make analogy to stretch receptors). This only requires two neurons and two synapses, and it is symmetric in the case that the appropriate constants are symmetric. For the simulations, Table 5.1

Table 5.1: Neuron parameters

$a_i = 300$	$b_i = -2$	$c_i = -0.05$	$d_i = 0.1$	$C_i = 0.01$	$v_{slope,i} = 0.01$
$k_i = 700$	$v_{r,i} = -0.06$	$v_{t,i} = -0.04$	$v_{peak,i} = 0.035$	$v_{th,ij} = -0.02$	$\tau_{ij} = 0.04$
$v_{rev,i} = 0.02$					

gives the constants which were used for each neuron. g_{max} will be the parameter of inquiry.

5.4 Variable-size Window Moving Integrator Algorithm for Smoothing

The remainder of this chapter will detail a method for performing system identification (SysID) on the neuron circuit of Equation 5.6. Our goal will be to determine whether it behaves substantially like a PD controller. Our main tool for SysID parameter estimation will be MATLAB's nonlinear grey box routine. It uses a spatial measure of fit (which will be detailed in Section 5.6), but from the early trials, it became clear that this may be problematic when applied to a spiking neuron system. A spiking signal is stretched in the spatial domain and compressed in the temporal domain. More importantly, any such spiking signal will eventually be filtered by a larger mechanical system. In biology, this is the neuromuscular junction, which is not well understood [92]. In robotics, this is likely a motor.

We would like to retain the knowledge that filtering is occurring but abstract away from any particular plant/actuator model. This choice has some inherent risk if the filtering performed by different actuators perform in substantially different ways. We

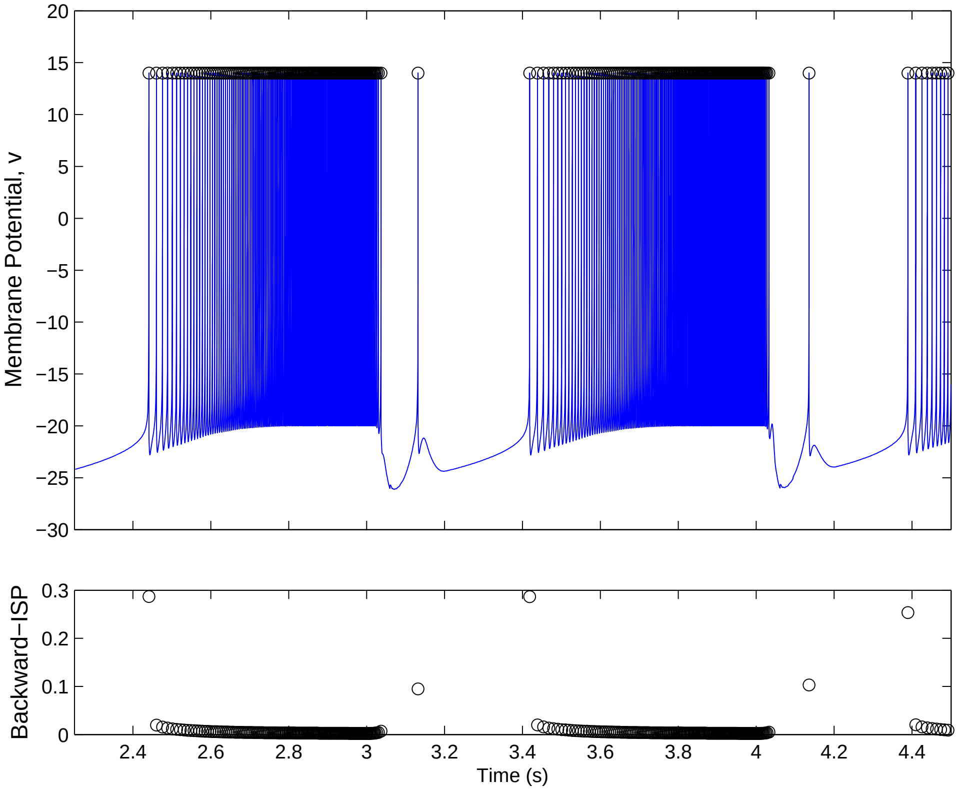


Figure 5.3: Example of spike identification (black circles) and ISP.

have two real choices here. 1) We can perform smoothing as part of the SysID and then check whether we need to make an alteration for various actuators' responses to the signal. 2) We can incorporate an actuator model in the SysID and then try to pull it back out afterward. We chose the former. Rather than trying to incorporate an entire plant in the SysID and then later trying to extract just the controller, we find it plausible that using a simple smoothing algorithm can help us mostly characterize the controller dynamics and that any actuator refinements can be added later. The smoothing algorithm proceeds as follows:

1. We identify each spike and calculate the backward-ISP (inter-spike period) for each spike. This is shown in Figure 5.3.

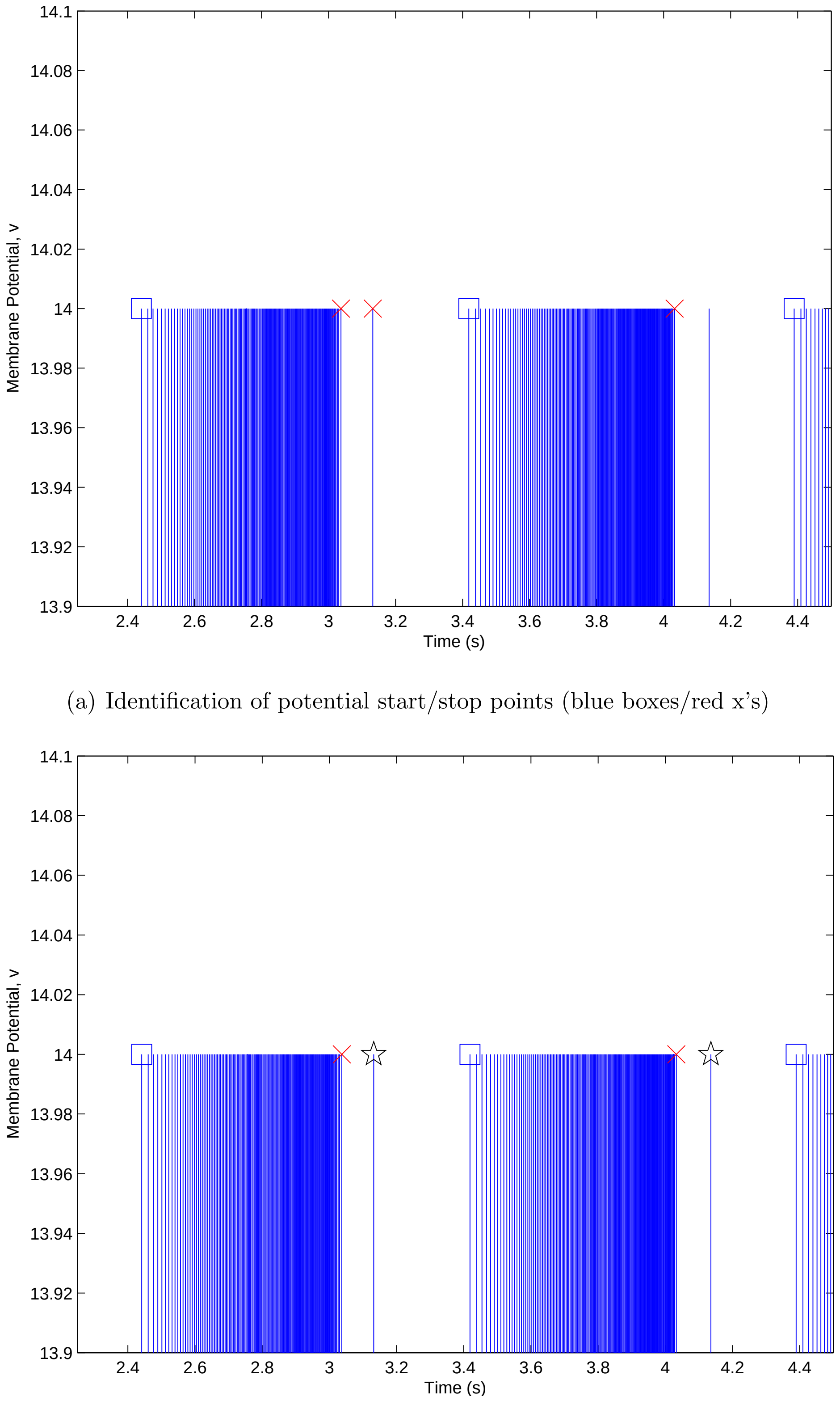
2. We identify the beginning/end of each burst using relative ISP (e.g., if $ISP_2 > 3ISP_1$, then spike 1 is potentially the end of a burst. This is shown in Figure 5.4. Notice that two subsequent spikes are identified as possibly being the end of a burst. This is corrected in the next step.

3. We identify potential singletons using absolute ISP (if the forward and backward ISP are both above 12 times the median ISP for a sufficiently excited signal, we consider it a singleton). We correct beginning/end points as necessary. This is shown in Figure 5.4.

4. At all points within the burst, we smooth signal y to signal y_1 with

$$y_1(t) = \frac{1}{ISP(t)} \int_{t-ISP(t)/2}^{t+ISP(t)/2} y(t')dt'. \tag{5.7}$$

   This is shown in Figure 5.5.



(a) Identification of potential start/stop points (blue boxes/red x's)

(b) Correction for singletons (black stars)

Figure 5.4: Set up smoothing regions where ISP is meaningful.

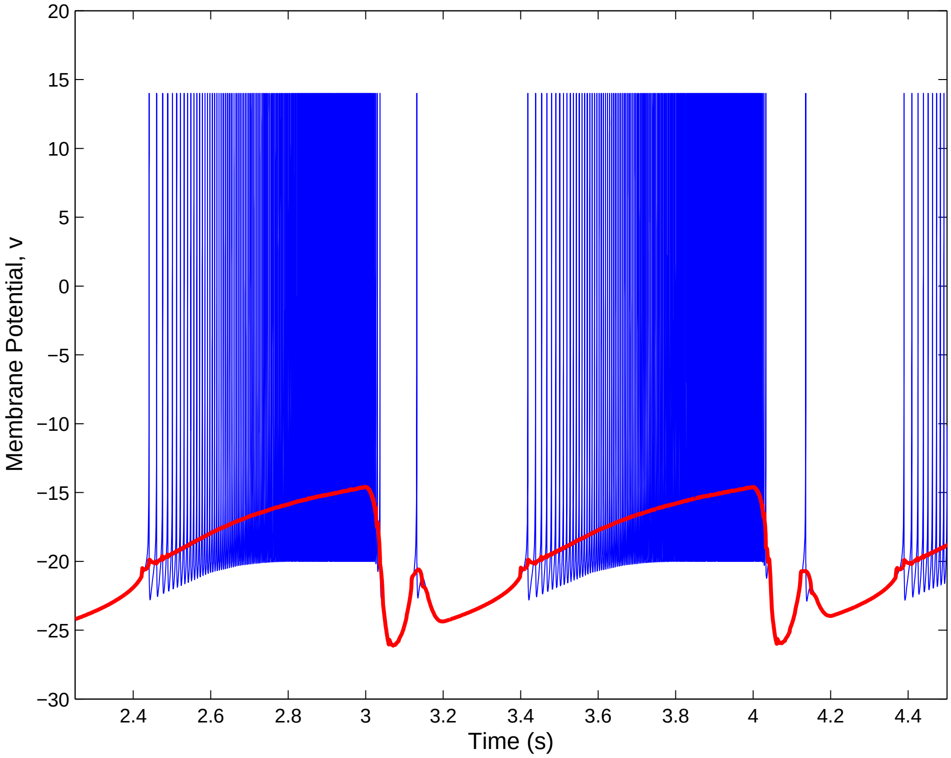


Figure 5.5: Smoothed signal.

Remark 1. *Our definition of singletons, beginnings of bursts, and ends of bursts, is entirely motivated by the desire to perform smoothing using ISP. It is my opinion that there is no theoretically valid method of distinguishing between spikes and bursts in general.*

An additional benefit to smoothing the signal is that we can rediscretize with a larger timestep, reducing the computation time for the SysID routine. Notice further that any choice of filter or smoother will provide tradeoffs. Our hypothesis is that if the neuron circuit is performing differentiation, it is doing so at a very fast timescale or instantaneously in analog. Therefore, we prefer to not have a preprocessing routine introduce a delay, so a windowed smoother is ideal.

5.5 Nonlinear Steady-State Response

In order to capture a nonlinear 'proportional' state response, we computed the steady-state output of the neuron circuit for constant input values, $I_{stretch}$. The results appear in Fig. 5.6. The blue curve was produced by steadily increasing the input, then the input was decreased back to zero for the red curve.

Note that in the two high synapse-strength cases, there are two instances which appear to be possible hysteresis. The first instance (input<3 in both cases) is genuine hysteresis. It is caused by the fact that the synapses are strong enough to incite persistent mutual excitation (i.e., once the neurons start firing, the synapse strength alone is enough to keep them firing indefinitely).

It is not clear yet whether the second instance (input$\approx$4 for g=2.1544 and input$\approx$7 for g=3.1623) actually represents significant hysteresis. In the tested cases, it is only 2-3 points wide. Recalling Sec. 5.2, we send the error signal to the first neuron and its additive inverse to the second. This means that for large inputs, the second neuron is

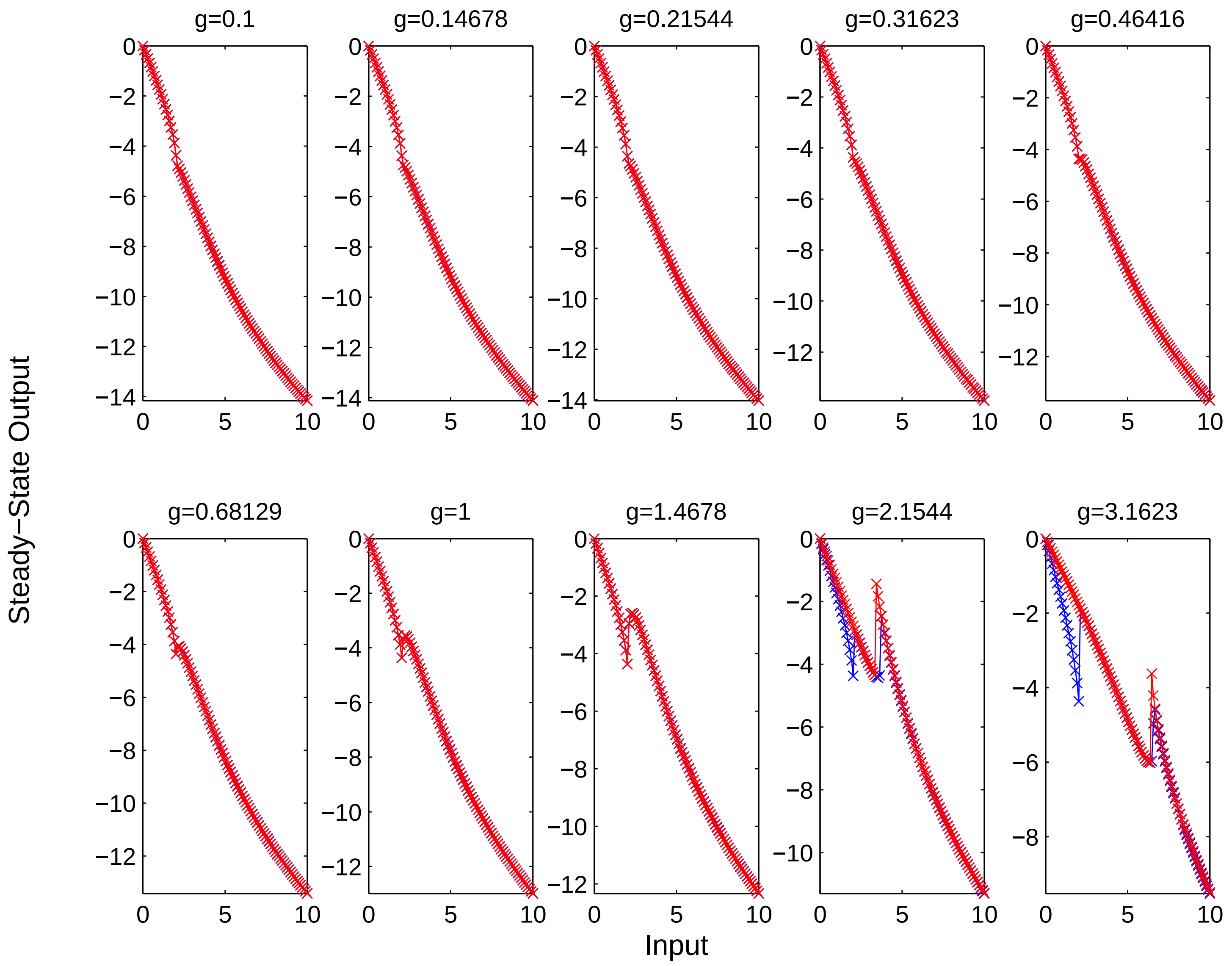


Figure 5.6: Steady-State 'proportional' response.

receiving a large inhibitory input. Even though the synapse may be wide open, this inhibitory input overcomes the excitatory synapse, leaving the neuron in a quiescent state.

5.6 System Identification Procedure

In order to capture several standard modes as well as a realistic signal from a system, we constructed a single large input signal to use for further SysID estimations. It can be seen in Fig. 5.7, and is composed of the following elements. First, we took a series of trajectories produced by a realistic motor/pendulum system simulation. Then, a chirp comes into existence, sweeps through a frequency range, and then fades back out. This is followed by a few step functions and, finally, a ramp.

Model controllers which use a dynamic method to perform differentiation will likewise produce an artificial delay. While the neuron circuit may be introducing some delay, our hypothesis is that if differentiation is happening, it's happening very fast or instantaneously in analog. Therefore, for the input signal shown in Fig. 5.7, we use a delay-free central difference method to compute the derivative for the first ten seconds and then use a delay-free analytic derivative elsewhere. This signal is also fed into the model controller for the SysID routine.

The flow of our procedure is shown in Fig. 5.8. MATLAB's nonlinear grey box parameter estimator was the main tool for performing system identification. For each value of synapse strength, we fed the input signal through the neuron circuit. Using the output, we estimated the parameters of the model controller. Internally, the optimization routine minimizes the cost function

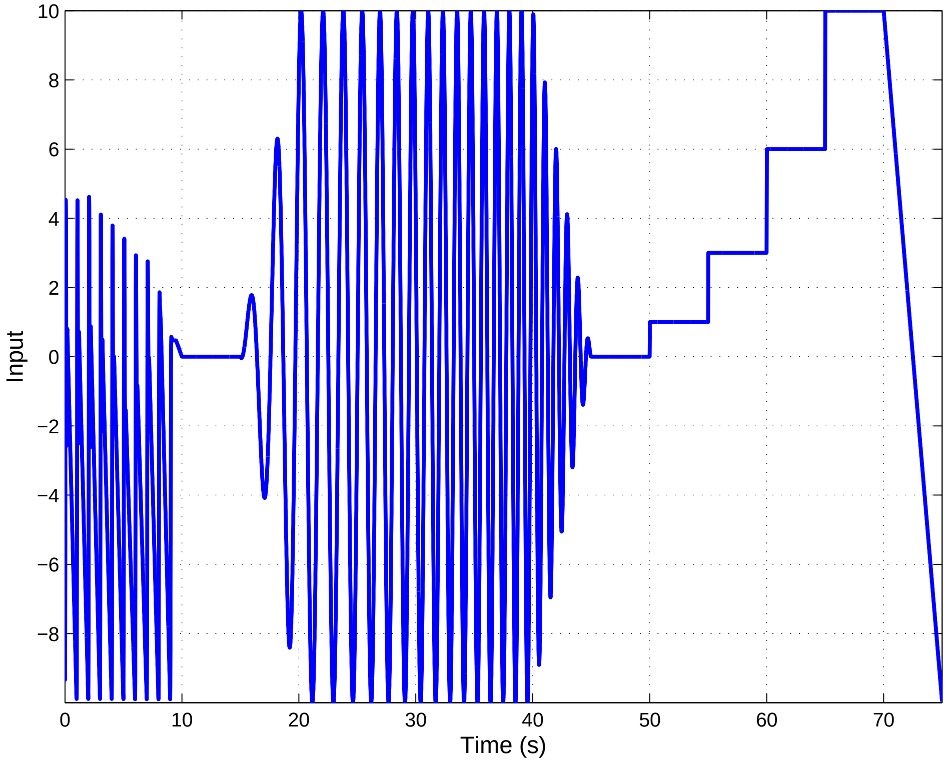


Figure 5.7: Larger input signal for SysID routines.

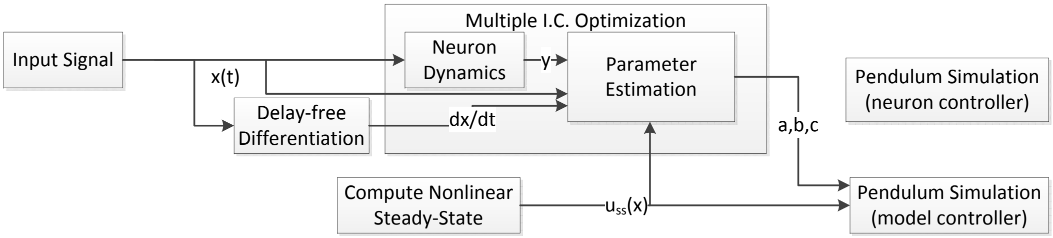

Multiple I.C. Optimization
Input Signal
x(t)
Neuron Dynamics
y
Parameter Estimation
Delay-free Differentiation
dx/dt
a,b,c
Pendulum Simulation (neuron controller)
Compute Nonlinear Steady-State
$u_{ss}(x)$
Pendulum Simulation (model controller)

Figure 5.8: Depiction of estimation scheme, where the estimated parameters a, b, and c are given in Equation 5.10.

$$J = \sum_{i=1}^{N} (y(t_i) - \hat{y}(t_i))^2, \tag{5.8}$$

where y is the output of the neuron circuit and $\hat{y}$ is the output of the model controller. The routine outputs a fit value (represented as a percentage) defined by

$$\text{fit} = 100 \left(1 - \frac{||y - \hat{y}||}{||y - \text{mean}(y)||}\right). \tag{5.9}$$

From early results, we determined that a sudden shift in parameters occurred for high synapse strength cases. While this could be the result of a bifurcation, it's also possible that the routine was getting carried away by a local minimum. Since each parameter estimation begins with the output of the previous estimation (from the neighboring synapse strength), picking between multiple local minima may be a phenomenon of the order in which we perform the estimations. Implementing a true global search algorithm was computationally burdensome, so we compromised with an algorithm which is likely to identify a separate local minimum and pick the better option. The algorithm can be described as follows:

1. Sweep through the synapse strength cases, performing parameter estimation for each case.

2. Identify the maximum distance (in parameter space) between any two solutions. Call it R_{max}.

3. Set an $R_{min} = R_{max}/10$.

4. For each synapse strength $g_{max,i}$, keep a record of initial conditions which have been used to perform parameter estimation. If any other synapse strength has parameters that are further than R_{min} from all previous initial conditions used

for $g_{max,i}$, then perform parameter estimation again using this point as the initial condition.

5. Save the best estimates.

5.7 Model Controller Results

The input system was run through the neuron controller described by Equation 5.6 and smoothed according to Section 5.4. Then, the system identification procedure optimized the parameters of a modified PD controller, described by

$$u = k(\theta) + \dot{\theta}(a + b\theta^2 + c\theta^4), \tag{5.10}$$

where the variables a, b, and c are being estimated by the SysID routine and $k(\theta)$ is the curve defined by the steady-state 'proportional' response. Multiple controller models were tried, but we report only the best fit. The fit quality and estimated parameters for each synapse strength are shown in Figure 5.9. For the low synapse strength cases, the fits are good, and the linear damping parameter is increasing in strength, as expected (while the other effects are much smaller, showing that the derivative term is mostly linear). A large shift occurs near the point at which the hysteresis develops - the point at which initiation of spiking causes self-sustained spiking. The fit also goes down at this point, but rebounds above. Future work should focus on this region, perhaps pinpointing the bifurcation with numerical continuation.

Nevertheless, we compare the neuron controller to the model controller on a simple nonlinear damped pendulum, chosen for its familiar characteristics with regard to controllers. Figure 5.10 shows a mock-up of the neuron circuit driving the pendulum. Figure 5.11 shows the pendulum response to the neuron circuit and the model

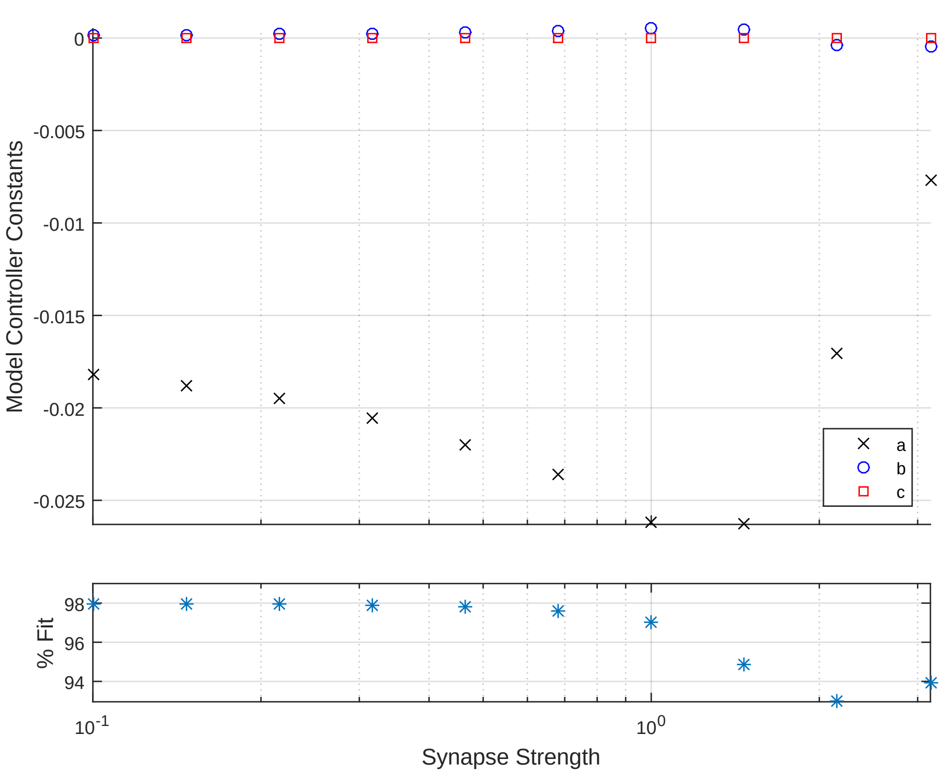


Figure 5.9: Results from system identification.

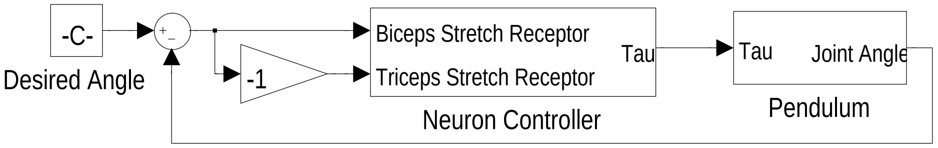
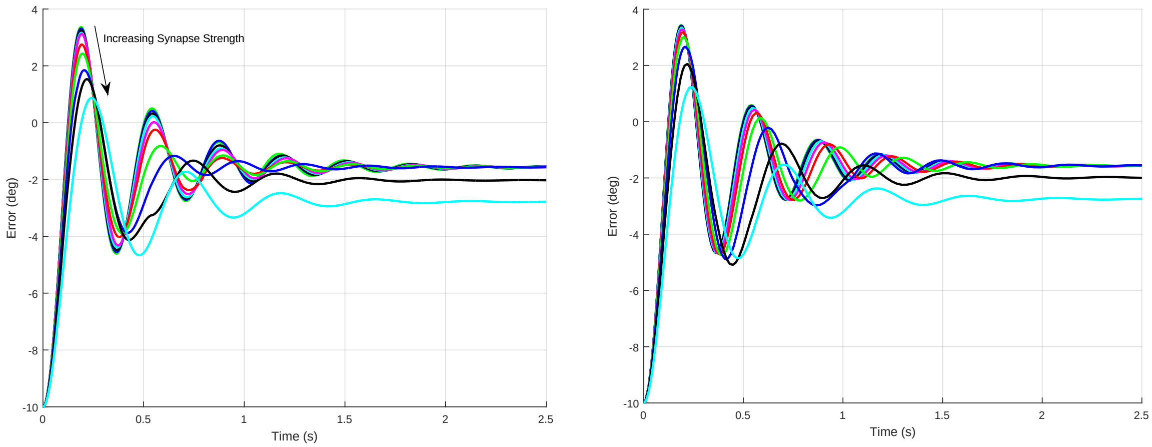


Figure 5.10: Diagram of neuron controller, where the stretch receptor input is given as in Equation 5.6.



(a) Neuron Controller (Equation 5.6)

(b) Model Controller

Figure 5.11: Comparison of controllers driving a simple pendulum. The synapse strengths used are the same as those of Figure 5.6, increasing in the direction of the arrow.

controller.

The general characteristics are similar. Since the steady-state part of the controller, $k(\theta)$ varies with synapse strength, the steady-state response does as well. Notice that for high synapse-strength cases, $k(\theta)$ is nonmonotonic, which enables the possibility of producing multiple steady-states for a single synapse-strength. While we do not include a figure showing this, we note that it has been seen in simulations.

5.8 Chapter Summary

Biological neurons come in many shapes, sizes, and flavors. Biological synapses do, as well. It will take many studies to begin producing genuinely general results which elucidate large-scale design principles for varieties of neurons and varieties of networks. Instead of starting with many neurons (i.e. ’neuron soup’), we focused on a simple two-neuron, two-synapse network composed of identical components. Izhikevich's neuron model and Rabinovich's synapse model are a compromise between total biological realism, dynamical systems similarity, and manufacturing capability. Nevertheless, the key insights of delay, attenuation, and sign change allowed us to construct a PD-like network.

We detailed a smoothing scheme alongside a system identification procedure to analyze the circuit. The controller exhibits derivative control, but also has a nonlinear steady-state response with hysteresis behavior and a discontinuous jump for high synapse-strength cases. These results had high fit values away from what appears to be a bifurcation, and this was confirmed by simulations of a pendulum being controlled by the neuron circuit and the model controller.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The objective of this dissertation was to explore strategies inspired by nature, particularly with respect to flapping flight. We focused on three main control schema: coupled limit-cycle generation, mode switching, and low-level neuromorphic control. High-level flight controllers were tested in simulation and experimentation. In addition, a PD-like neuron circuit was analyzed using system identification techniques.

Flapping fliers often produce complicated, coupled wing motions which drive complicated, oscillatory locomotion through the air. The first step in this dissertation was to analyze a method of producing those wing motions with coupled limit cycle oscillators. The goal here was to provide a reduced set of control parameters with which we could design a top-level controller. We chose the Hopf oscillator for several of its properties: smooth, robust, and symmetric. Coupled together with proportional coupling, the oscillators can synchronize. We gave rigorous proofs describing the conditions under which these networks will exhibit phase synchronization. Phase synchronization is one of the key tools to utilizing limit cycle CPG control for flapping flight. In addition, coupled Hopf oscillators provide capabilities for frequency modulation, amplitude modulation, and fast inhibition via bifurcation.

Fast inhibition implicates a switched systems framework. We cast the limit-cycle problem as one of multiple, possibly non-equilibrium steady-states. Using Lyapunov functions to link the spatial and temporal domains, we built entry-sets and a no-escape set with an associated dwell time. While this was a true generalization of previous work, our use of Lyapunov derivatives introduced some complications. In the end, however, our method produces tighter bounds and is applicable to systems with more useful dynamics.

These methods were applied in both simulation and experimentation. A full 6DOF dynamic simulation captured realistic aerodynamics using blade-element theory and quasi-steady assumptions. Through a combination of frequency control, symmetric and anti-symmetric phase differences, and symmetric and antisymmetric amplitude modulation, we controlled all body modes through ascending and turning flight. In addition, we switching between flapping and gliding rapidly for altitude regulation by changing the bifurcation parameter.

We developed an 8DOF RoboBat and mounted it on a compound pendulum, isolating the longitudinal modes. Open-loop experiments supported the hypothesis that flapping/lead-lag phase differences were a potential control parameter for pitch dynamics and would have less affect on the other states. Closed-loop experiments confirmed this hypothesis by allowing us to stabilize an unstable system and exert a measurable amount of control authority.

Desire for low-level reflex behavior and a control schema that could satisfy size, weight, and power constraints for a variety of micro vehicles, we also investigated spike-bursting neuron control. We selected Izhikevich's neuron model and Rabinovich's synapse model for their combination of biological realism, ability to produce relevant dynamic phenomena, and propensity to be implementable in hardware. A two-neuron, two-synapse, mutually-excitatory circuit was developed to produce PD-like control behavior in an antagonistic set-up, exploiting the properties of delay and attenuation. Interpolating on inter-spike distance allowed us to exploit a variable-size window integrator smoothing algorithm allowed us to provide high-quality smoothed signals from the spiking signal.

Since global optimization is too time-consuming, we developed a not-entirely-local optimization algorithm to help avoid some possible local minima. Using this in combination with system identification techniques, high fit values supported the hypothesis that the neuron circuit was acting much like a PD controller. Finally, driving a pendulum with both the model controller and the neuron circuit produced substantially similar results, confirming the hypothesis.

6.2 Future Work

As this dissertation touched broadly on multiple areas of interest, there are many problems which can be explored in continuing work. In particular, the neuron circuits in Chapter 5 represent a very new intersection between neuroscience and engineering, leaving much to be done. The following is a list of potential applications and challenges which follow from this work:

- Explicit characterization of death of oscillation and the relation to Splay states: This dissertation provided some new insight into Splay states and the inner workings of contraction theory. It remains to characterize these states explicitly or to determine whether coupling structures for networks of size greater than two can mitigate the phenomenon. The onset of global contraction may also correspond to a bifurcation which extends the observed behavior to be provable for all coupling gains.

- Theoretical bridge between network control and switching: This dissertation required strong assumptions on switched systems which may be difficult to satisfy for a complicated plant, particularly when being driven by complicated networks. This gap must be addressed from both sides. Oscillator network-driven plants need methods to guarantee passivity-like properties or other boundedness properties. In turn, one could attempt to weaken the switched Lyapunov assumptions in order to utilize these properties.

- Onboard, open-air flapping flight: This dissertation provided experimental validation only on a constrained compound pendulum with external measurements provided by encoders. A platform which produces sufficient thrust for forward flight and which remains stable and controllable by means of only the two main wings is an admirable goal. To confirm the capabilities beyond question, sensing and processing should be all onboard, allowing continuous outdoor flight apart from encoders or external camera systems.

- Gain ranges for neuron circuits: This dissertation showed a small range of PD-like gains for a simple neuron circuit. Parameter selection, mechanical design, or populations of neurons could be used to extend these ranges. Populations of neurons could be used to improve robustness to chip failures. There appears to be a qualitative change in the high synapse-strength cases. Bifurcation analysis could shed light on methods to directly extend the range of useful synapse strengths along with the gains, directly.

- Self-tuning neuron circuits: If we think of a dynamic controller as a plant, itself, then this dissertation did a kind of open-loop analysis on it. We may want to add circuit complexity which is specifically designed to provide self-tuning capabilities. This open loop analysis may inform the particular goals which are set for such a circuit.

- Modular, hierarchical neuron structures: This dissertation only analyzed a very simple neuron circuit with rather simple control properties. However, the overall theme clearly points to developing circuits which oscillate to produce locomotion, select behaviors, and/or incorporate strong reflex responses. The smoothing and system identification tools used here, along with insight gained about the intuitive properties of neurons and synapses, may be useful for proposing or analyzing other structures. Structures with complicated functionality can be imported and analyzed for suitability in engineered projects. Those structures can be combined hierarchically to produce a whole-vehicle neuromorphic controller, which could even be decentralized in the same way that the peripheral nervous system operates in mammals or sucker ganglia operates in octopuses.

Bibliography

[1] Mueller, T. J., *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Application*, Progress in Astronautics and Aeronautics, AIAA, 2001.

[2] Shyy, W., Lian, Y., Tang, J., Viieru, D., and Liu, H., *Aerodynamics of Low Reynolds Number Flyers*, Cambridge University Press, New York, NY, 2008.

[3] Azuma, A., *The Biokinetics of Flying and Swimming*, AIAA, 2nd ed., 2006.

[4] Kato, N. and Kamimura, S., *Bio-Mechanisms of Swimming and Flying: Fluid Dynamics, Biomimetic Robots, and Sports Science*, Springer Verlag, 2008.

[5] Norberg, U. M., *Vertebrate Flght: Mechanics, Physiology, Morphology, Ecology and Evolution*, Springer-Verlag, 1989.

[6] Deng, X., Schenato, L., Wu, W. C., and Sastry, S. S., "Flapping Flight for Biomimetic Robotic Insects: Part I-system Modeling," *IEEE Trans. on Robotics*, Vol. 22, No. 4, 2006, pp. 776–788.

[7] Deng, X., Schenato, L., Wu, W. C., and Sastry, S. S., "Flapping Flight for Biomimetic Robotic Insects: Part II-flight Control Design," *IEEE Transactions on Robotics*, Vol. 33, No. 4, 2006, pp. 789–803.

[8] Frampton, K. D., Goldfarb, M., Monopoli, D., and Cveticanin, D., "Passive Aeroelastic Tailoring for Optimal Flapping Wings," *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Application*, AIAA, 2001, pp. 473–482, T. J. Mueller, Ed.

[9] Ho, S., Nassef, H., Pornsinsirirak, N., Tai, Y.-C., and Ho, C.-M., "Unsteady Aerodynamics and Flow Control for Flapping Wing Flyers," *Progress in Aerospace Sciences*, Vol. 39, 2003.

[10] Issac, K. K. and Agrawal, S. K., "An Investigation Into the Use of Springs and Wing Motions to Minimum the Power Expended by a Pigeon-sized Mechanical Bird for Steady Flight," *Journal of Mechanical Design, Transactions of the ASME*, Vol. 129, No. 4, Apr. 2007.

[11] Roget, B., Sitaraman, J., Harmon, R., Grauer, J., Conroy, J., Hubbard, J., and Humbert, S., "A Computational Study of Flexible Wing Ornithopter Flight," *Proc. of the 26th Applied Aerodynamics Conference*, 2008, Honolulu, HI, AIAA 2008–6397.

[12] Dickinson, M. H., Lehmann, F. O., and Sane, S. P., "Wing Rotation and the Aerodynamic Basis of Insect Flight," *Science*, Vol. 284, 1999, pp. 1954–1960, June.

[13] Jones, K. D., Lund, T. C., and Platzer, M. F., "Experimental and Computational Investigation of Flapping Wing Propulsion," *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Application*, 2001, pp. 307–336, T. J. Mueller, Ed., AIAA,.

[14] Kawamura, Y., Souda, S., Nishimoto, S., and Ellington, C. P., "Clapping-wing Micro Air Vehicle of Insect Size," *Bio-Mechanisms of Swimming and Flying: Fluid Dynamics, Biomimetic Robots, and Sports Science*, Springer Verlag, 2008, pp. 319–330, N. Kato and S. Kamimura, Ed.

[15] Larijani, R. F. and DeLaurier, J. D., "A Nonlinear Aeroelastic Model for the Study of Flapping Wing Flight," *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Application*, AIAA, 2001, pp. 399–428, T. J. Mueller, Ed.

[16] Michelson, R., Helmick, D., Reece, S., and Amarena, C., "A Reciprocating Chemical Muscle (RCM) for Micro Air Vehicle Entomopter Flight," *Proc. of the Association for Unmanned Vehicle Systems, International*, 1997, pp. 429–435.

[17] Steltz, E., Avadhanula, S., and Fearing, R. S., "High Lift Force with 275 Hz Wing Beat in MFI," *IEEE Int. Conf. on Intelligent Robots and Systems*, 2007, San Diego, CA.

[18] Wood, R. J., "The First Takeoff of a Biologically-inspired At-scale Robotic Insect," *IEEE Transactions on Robotics*, Vol. 24, No. 2, 2008, pp. 341–347.

[19] Albertani, R. J., Stanford, B., DeLoach, R., Hubner, J. P., and Ifju, P., "Wind-tunnel Testing and Modeling of a Micro Air Vehicle with Flexible Wings," *Journal of Aircraft*, Vol. 35, No. 3, 2008, pp. 1025–1032.

[20] Hall, K. C., Pigott, S. A., and Hall, S. R., "Power Requirements for Large-amplitude Flapping Flight," *Journal of Aircraft*, Vol. 35, No. 3, 1998, pp. 352–361.

[21] Tian, X., Iriarte-Diaz, J., Middleton, K., Galvao, R., Israeli, E., Roemer, A., Sullivan, A., Song, A., Swartz, S., and Breuer, K., "Direct Measurements of the Kinematics and Dynamics of Bat Flight," *Bioinspiration and Biomimetics*, Vol. 1, 2006, S10-S19.

[22] Swartz, S. M., "Allometric Patterning in the Limb Skeleton of Bats: Implications for the Mechanics and Energetics of Powered Flight," *Journal of Morphology*, Vol. 234, 1997, pp. 277–294.

[23] Swartz, S. M., Bishop, K. L., and Ismael-Aguirrre, M.-F., "Dynamic Complexity of Wing Form in Bats: Implications for Flight Performance," *Functional and Evolutionary Ecology of Bats*, Oxford University Press, Oxford, UK, 2005.

[24] Taylor, G. K. and Zbikowski, R., "Nonlinear Time-periodic Models of the Longitudinal Flight Dynamics of Desert Locusts Schistocerca Gregaria," *J. R. Soc. Interface*, Vol. 2, 2005, pp. 197–221.

[25] Wang, Z. J., "Aerodynamic Efficiency of Flapping Flight: Analysis of a Two-stroke Model," *The Journal of Experimental Biology*, Vol. 211, 2008, pp. 234–238.

[26] Willis, D. J., Peraire, J., Drela, M., and White, J. K., "A Numerical Exploration of Parameter Dependence in Power Optimal Flapping Flight," 2006, AIAA Paper 2006–2994.

[27] Zbikowski, R., Ansari, S. A., and Knowles, K., "On Mathematical Modelling of Insect Flight Dynamics in the Context of Micro Air Vehicles," *Bioinspiration and Biomimetics*, Vol. 1, 2006, pp. 26–37.

[28] Swartz, S. M., Iriarte-Diaz, J., Riskin, D. K., Song, A., Tian, X., Willis, D. J., and Breuer, K. S., "Wing Structure and the Aerodynamic Basis of Flight in Bats," *Proc. of the 45th AIAA Aerospace Science Meeting*, Reno, NV, 2007.

[29] Liberzon, D., *Switching in Systems and Control*, Birkhäuser Boston, 2003.

[30] Alpcan, T. and Başar, T., "A stability result for switched systems with multiple equilibria," *Dynamics of Continuous, Discrete and Impulsive Systems Series A: Mathematical Analysis*, Vol. 17, 2010, pp. 949–958.

[31] Webb, A., Davies, S., and Lester, D., "Spiking Neural PID Controllers," *Neural Information Processing*, Springer, 2011, pp. 259–267.

[32] Proctor, J. and Holmes, P., "Reflexes and preflexes: on the role of sensory feedback on rhythmic patterns in insect locomotion," *Biological cybernetics*, Vol. 102, No. 6, 2010, pp. 513–531.

[33] Izhikevich, E., *Dynamical Systems in Neuroscience: The Geometery of Excitability and Bursting*, Prentice Hall, 2007.

[34] Nowotny, T., Zhigulin, V. P., Selverston, A. I., Abarbanel, H. D. I., and Rabinovich, M. I., "Enhancement of Synchronization in a Hybrid Neural Circuit by Spike-Timing Dependent Plasticity," *The Journal of Neuroscience*, Vol. 23, 2003, pp. 9776–9785.

[35] Hooper, S. L., "Central Pattern Generators," *In Nature Encyclopedia of Life Sciences*, 2001.

[36] Bay, J. S. and Hemami, H., "Modeling of a Neural Pattern Generator with Coupled Nonlinear Oscillators," *IEEE Transactions on Biomedical Engineering*, 1987.

[37] Bernstein, N., *The Coordination and Regulation of Movements*, Pergamon Press, Oxford, 1967.

[38] Collins, J. J. and Stewart, I. N., "Coupled Nonlinear Oscillators and the Symmetries of Animal Gaits," *Nonlinear Science*, Vol. 3, 1993, pp. 349–392.

[39] Golubitsky, M., Stewart, I., Buono, P., and Collins, J., "Symmetry in Locomotor Central Pattern Generators and Animal Gaits," *Nature*, Vol. 401, No. 19, 1999, pp. 693–695.

[40] Hooper, S. L., Guschlbauer, C., von Uckermann, G., and Buschges, A., "Natural Neural Output That Produces Highly Variable Locomotory Movements," *Journal Neurophysiology*, Vol. 96, 2006, pp. 2072–2088.

[41] Lewis, M. A., Tenore, F., and Etienne-Cummings, R., "CPG Design Using Inhibitory Networks," *Proc. of the 2005 IEEE Int'l Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 3682–3687.

[42] Matsuoka, K., "Mechanisms of Frequency and Pattern Control in Neural Rhythm Generators," *Biological Cybernetics*, Vol. 56, 1987, pp. 345–353.

[43] Sekerli, M. and Butera, R. J., "Oscillations in a Simple Neuromechanical System: Underlying Mechanisms," *Journal of Computational Neuroscience*, Vol. 19, No. 2, 2005, pp. 181–197.

[44] Grillner, S., "Neurological Bases of Rhythmic Motor Acts in Vetebrates," *Science*, Vol. 228, 1985, pp. 143–149.

[45] Iwasaki, T. and Zheng, M., "Sensory Feedback Mechanism Underlying Entrainment of Central Pattern Generator to Mechanical Resonance," *Biological Cybernetics*, Vol. 94, No. 4, 2006, pp. 245–261.

[46] Seo, K., Chung, S.-J., and Slotine, J.-J. E., "CPG-based Control of a Turtle-like Underwater Vehicle," *Proc. of the Robotics: Science and Systems (RSS)*, Switzerland, June 2008.

[47] Ijspeert, A. J., "Central Pattern Generators for Locomotion Control in Animals and Robots: A Review," *Neural Networks*, Vol. 24, No. 4, 2008, pp. 642–653.

[48] Ayers, J. and Rulkov, N., "Controlling Biomimetic Underwater Robots With Electronic Nervous Systems," *Bio-Mechanisms of Swimming and Flying: Fluid Dynamics, Biomimetic Robots, and Sports Science*, Springer Verlag, 2008, pp. 295–306, N. Kato and S. Kamimura, Ed.

[49] Ijspeert, A. J., Crespi, A., and Cabelguen, J.-M., "Simulation and Robotics Studies of Salamander Locomotion: Applying Neurobiological Principles to the Control of Locomotion in Robots," *Neuroinformatics*, Vol. 3, No. 3, 2005, pp. 171–195.

[50] Morimoto, J., Endo, G., Nakanishi, J., and Cheng, G., "A biologically inspired biped locomotion strategy for humanoid robots: modulation of sinusoidal patterns by a coupled oscillator model," 2006.

[51] Dickinson, W. B. and Dickinson, M. H., "The Effect of Advance Ratio on the Aerodynamics of Revolving Wings," *Journal of Experimental Biology*, Vol. 207, 2004, pp. 4269–4281.

[52] Ellington, C., "The Aerodynamics of Hovering Insect Flight. I-VI," *Philosophical Transactions of the Royal Society of London B Biological Sciences*, Vol. 305, 1984, pp. 1–181.

[53] Strogatz, S., *Nonlinear Dynamics and Chaos With Applications to Physics , Biology, Chemistry, and Engineering*, Perseus Books Group, Cambridge, MA, 1994.

[54] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Prentice Hall, 1991.

[55] Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J.-M., "From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model," *Science*, Vol. 315, No. 5817, 2007, pp. 1416–1420.

[56] Pham, Q.-C. and Slotine, J.-J. E., "Stable Concurrent Synchronization in Dynamic System Networks," *Neural Networks*, Vol. 20, No. 1, 2007, pp. 62–77.

[57] Doya, K., Kimura, H., and Kawato, M., "Neural Mechanisms of Learning and Control," *IEEE Control Systems Magazine*, Vol. 21, No. 4, 2001, pp. 42–54.

[58] Norberg, U. M. and Winter, Y., "Wing Beat Kinematics of a Nectar-feeding Bat, Glossophaga Soricina, Flying At Different Flight Speeds and Strouhal Numbers," *J. of Experimental Biology*, Vol. 209, 2006, pp. 3887–3897.

[59] Taga, G., "A Model of the Neuro-musculo-skeletal System for Anticipatory Adjustment of Human Locomotion During Obstacle Avoidance," *Biological Cybernetics*, Vol. 78, No. 1, 1998, pp. 9–17.

[60] Riskin, D. K., Willis, D. J., Iriarte-Diaz, J., Hendrick, T. L., Kostandov, M., Chen, J., Laidlaw, D. H., Breuer, K. S., and Swartz, S. M., “Quantifying the Complexity of Bat Wing Kinematics,” *Journal of Theoretical Biology*, Vol. 254, 2008, pp. 604–615.

[61] Hedrick, T. L. and Biewener, A. A., “Experimental Study of Low Speed Turning Flight in Cockatoos and Cockatiels,” *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2007.

[62] Chung, S.-J. and Slotine, J.-J. E., “Cooperative Robot Control and Concurrent Synchronization of Lagrangian Systems,” *IEEE Transactions on Robotics*, Vol. 25, No. 3, 2009, pp. 686–700.

[63] Lohmiller, W. and Slotine, J.-J. E., “Contraction Analysis for Nonlinear Systems,” *Automatica*, Vol. 34, No. 6, 1998, pp. 683–696.

[64] Sepulchre, R., Paley, D. A., and Leonard, N. E., “Stabilization of Planar Collective Motion: all-to-all communication,” *IEEE Trans. Autom. Control*, Vol. 52, No. 5, May 2007, pp. 811–824.

[65] Wang, W. and Slotine, J.-J. E., “On Partial Contraction Analysis for Coupled Nonlinear Oscillators,” *Biological Cybernetics*, Vol. 92, No. 1, 2005, pp. 38–53.

[66] Elowitz, M. B., *Transport, Assembly, and Dynamics in Systems of Interacting Proteins*, PhD dissertation, Princeton University, Department of Physics, 1999.

[67] Morimoto, J., Endo, G., Nakanishi, J., Hyon, S.-H., Cheng, G., Bentivegna, D., and Atkeson, C. G., “Modulation of Simple Sinusoidal Patterns by a Coupled Oscillator Model for Biped Walking,” *Proc. of the 2006 IEEE Int’l Conference on Robotics and Automation*, 2006, pp. 1579–1584.

[68] Rossignol, S., Dubuc, R., and Gossard, J.-P., “Dynamic sensorimotor interactions in locomotion,” *Physiological reviews*, Vol. 86, No. 1, 2006, pp. 89–154.

[69] Park, H.-W., Ramezani, A., and Grizzle, J., “A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking,” *Robotics, IEEE Transactions on*, Vol. 29, No. 2, 2013, pp. 331–345.

[70] Chen, G., Moiola, J. L., and Wang, H. O., “Bifurcation control: theories, methods, and applications,” *I.J. Bifurcation and Chaos*, Vol. 10, No. 3, 2000, pp. 511–548.

[71] Hespanha, J. P., Liberzon, D., Angeli, D., and Sontag, E. D., “Nonlinear Norm-Observability Notions and Stability of Switched Systems,” *IEEE Trans. on Auto. Cont.*, Vol. 50, 2005, pp. 154–168.

[72] Bacciotti, A. and Mazzi, L., "An invariance principle for nonlinear switched systems," *Systems & Control Letters*, Vol. 54, 2005, pp. 1109–1119.

[73] Mancilla-Aguilar, J. L. and Garcia, R., "An extension of LaSalle's invariance principle for switched systems," *Systems & Control Letters*, Vol. 55, 2005, pp. 376–384.

[74] Mastellone, S., Stipanović, D. M., and Spong, M. W., "Stability and Convergence for Systems with Switching Equilibria," *Proc. of the 46th IEEE Conference on Decision and Control*, 2007.

[75] Xu, X., Zhai, G., and He, S., "On practical asymptotic stabilizability of switched affine systems," *Nonlinear Analysis: Hybrid Systems*, Vol. 2, No. 1, 2008, pp. 196–208.

[76] LaSalle, J., "Some Extensions of Liapunov's Second Method," *IRE Transactions on Circuit Theory*, Vol. 7, No. 4, 1960, pp. 520–527.

[77] Bhatia, N. P., "On Asymptotic Stability in Dynamical Systems," *Mathematical Systems Theory*, Vol. 1, No. 2, 1967, pp. 113–127.

[78] Seo, K., Chung, S.-J., and Slotine, J.-J. E., "CPG-based Control of a Turtle-like Underwater Vehicle," *Autonomous Robots*, Vol. 28, No. 3, 2010, pp. 247–269, Control of Locomotion: From Animals to Robots.

[79] Chung, S.-J. and Dorothy, M., "Neurobiologically Inspired Control of Engineered Flapping Flight," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 440–453.

[80] Chung, S.-J., Dorothy, M., and Stoner, J. R., "Neurobiologically Inspired Control of Engineered Flapping Flight," *AIAA Infotech at Aerospace and Unmanned Unlimited Conference and Exhibit*, Seattle, WA, Apr. 2009, AIAA Paper 2009-1929.

[81] Thomas, A. L. R. and Taylor, G. K., "Animal Flight Dynamics I. Stability in Gliding Flight," *Journal Theoretical Biology*, Vol. 212, 2001, pp. 399–424.

[82] Kuang, P. D., Dorothy, M., and Chung, S.-J., "RoboBat: Dynamics and Control of a Robotic Bat Flapping Flying Testbed," *AIAA Infotech at Aerospace Conference and Exhibit*, St. Louis, MO, Mar. 2011, AIAA Paper 2011-1435.

[83] Paranjape, A. A., Chung, S.-J., and Selig, M. S., "Flight Mechanics of a Tailless Articulated Wing Aircraft," *Bioinspiration and Biomimetics*, Vol. 6, 2011.

[84] *Fundamental Neuroscience (Fourth Edition)*, edited by L. R. Squire, D. Berg, F. E. Bloom, S. du Lac, A. Ghosh, and N. C. Spitzer, Academic Press, San Diego, 2013.

[85] Rabinovich, M. I., Varona, P., Selverston, A. I., and Abarbanel, H. D. I., "Dynamical Principles in Neuroscience," *Reviews of Modern Physics*, Vol. 78, 2006, pp. 1213–1265.

[86] Izhikevich, E. M., "Which Model to Use for Cortical Spiking Neurons?" *IEEE Transactions on Neural Networks*, Vol. 15, No. 5, 2004, pp. 1063–1070.

[87] Shilnikov, A., "Complete dynamical analysis of a neuron model," *Nonlinear Dynamics*, Vol. 68, 2012, pp. 305–328.

[88] Marder, E. and Bucher, D., "Central pattern generators and the control of rhythmic movements," *Current Biology*, Vol. 11, No. 23, 2001, pp. R986 – R996.

[89] Wijekoon, J. H. B. and Dudek, P., "Compact silicon neuron circuit with spiking and bursting behavior," *Neural Networks*, Vol. 21, 2008, pp. 524–534.

[90] Wang, X.-J., "Synaptic Basis of Cortical Persistent Activity: the Importance of NMDA Receptors to Working Memory," *The Journal of Neuroscience*, Vol. 19, No. 21, 1999, pp. 9587–9603.

[91] Izhikevich, E. M., Gally, J. A., and Edelman, G. M., "Spike-timing Dynamics of Neuronal Groups," *Cerebral Cortex*, Vol. 14, 2004, pp. 933–944.

[92] Brezina, V., Orekhova, I. V., and Weiss, K. R., "The neuromuscular transform: the dynamic, nonlinear link between motor neuron firing patterns and muscle contraction in rhythmic behaviors," *Journal of neurophysiology*, Vol. 83, No. 1, 2000, pp. 207–231.