A Computational Model of Avian Vocalizations

Yining Xie

A final project dissertation submitted in partial fulfilment
of the requirements for the degree of

Master of Science (MSc)

Acoustics and Music Technology

Acoustics and Audio Group

Edinburgh College of Art
University *of* Edinburgh

September 11, 2023

Supervisor 1: Dr Micheal Newton

Abstract

This research presents the development of a computational birdsong model in Matlab inspired by Mindlin's physical model. A key innovation is the introduction of a finite difference scheme, surpassing the efficiency of traditional Runge Kutta methods by over eight times. The study delves deep into the parameter space, revealing correlations between perceptual parameters, notably the fundamental frequency (f_0) and the spectral content index (SCI). These insights, while aligning with prior research, are uniquely highlighted using original remapping and visualization techniques. Efforts to invert the model introduced techniques like look-up tables, interpolation, and machine learning, with the latter two marking pioneering efforts in this domain. The perceptual parameter to control parameter approach promises enhanced model controllability and efficiency, setting the stage for future advancements in physics-based synthesis and potential applications in musical instrument design.

Declaration

I do hereby declare that this dissertation was composed by myself and that the work described within is my own, except where explicitly stated otherwise.

Yining Xie
September 11, 2023

Acknowledgements

I would like to extend my heartfelt gratitude to Dr. Michael Newton for his support and guidance throughout this final project. Special thanks to Dr. Vincent Lostanlen for introducing me to the topic of birdsong. Additionally, I am deeply appreciative of the entire staff and my peers in the AMT programme for making this year both productive and enriching.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Motivation

Birdsong, with its captivating and diverse nature, has long held the attention of humans. This fascination is not just limited to its aesthetic appeal but extends to a scientific curiosity about the mechanisms behind song production in birds. The avian syrinx, often likened to a musical instrument, serves as the primary source of these melodious sounds. Researchers such as Mindlin [4, 5], Fletcher [6] etc., have delved deep into understanding this mechanism, providing a foundation for our work. Building on this, Smyth [7, 8] has ingeniously extended such insights to the domain of musical instrument design. Furthermore, birdsong has piqued the interest of sound design artists, who often incorporate these melodious sounds into their artistic creations, underscoring the multifaceted appeal and significance of avian melodies.

1.2 Research Project

In this study, I developed a computational birdsong model in Matlab, building upon the foundational work of Mindlin's physical model. A notable advancement in this research is the is the introduction of a finite difference scheme, a first for this model type, which boasts an efficiency more than eight times that of the existing Runge Kutta method implementations.

Furthermore, an in-depth exploration of the parameter space unveiled correlations between perceptual parameters, specifically the fundamental frequency (f_0) and the spectral content index (SCI). These findings are consistent with previously published research but are illuminated in this work using innovative remapping and visualization techniques, providing a fresh perspective and complementing established methodologies.

Additionally, efforts were made to invert the model using a range of techniques, including look-up tables, interpolation, and machine learning. The incorporation of

interpolation is a pioneering effort for this model and the adoption of machine learning presents a novel direction in the realms of birdsong research and physics-based synthesis. This approach, leveraging perceptual parameter input to derive control parameter output, promises enhanced controllability and potentially inversion efficiency. Such advancements are poised to benefit more sophisticated models and have potential applications in musical instrument design.

1.3 Dissertation Structure

Following the introduction of the current chapter, Chapter 2 introduces the fundamental background knowledge of this work, including the dynamics, bifurcation, and mechanisms of birdsong production. It provides insights into the avian vocal system, its intricacies, and the inherent diversity. The chapter continues with a review of the current state of the art, highlighting existing physical models, nuances of control parameter mapping, and the emerging role of machine learning in physics-based modeling.

Chapter 3 provides a review of the timeline and development of the physical model utilized in this study. It delves into the physics and equations of the model, drawing its foundation from the model proposed by Mindlin's group. The chapter sequentially presents the equations for the syrinx, vocal tract (including the trachea and oropharyngeal-esophageal cavity (OEC), with a dimensional reduction achieved through dynamics. It concludes with a discussion on the multifaceted complexities inherent in the model.

Chapter 4 delves into the numerical implementation. It introduces the reader to the Runge-Kutta 4th Order (RK4) Method and its practical nuances. The chapter then transitions to the development and intricacies of the Finite Difference Method (FDM). Building on what was mentioned earlier, as a novel approach to this model, the efficiency of the FDM method stands out. The chapter concludes with a comparison of these methods, emphasizing their respective merits and limitations.

Chapter 5, "Evaluation and Analysis," embarks on a detailed study of bifurcation and its overarching influence on the parameter space. The chapter further explores added complexities, including diverse trajectories and the role of noise. The chapter further delves into perceptual parameter mapping, and discovers the correlations between fundamental frequency (f_0) and the spectral content index (SCI). Innovative remapping and visualization techniques are employed to shed light on these correlations, offering a nouvel insight. The chapter also delves into the efforts made to invert the model, emphasizing the pioneering incorporation of interpolation and the novel direction presented by machine learning in this domain. The chapter concludes with a thorough exploration of the inverse model and a comparison of the various techniques' efficacy.

The dissertation concludes with a reflective summary of the research's achievements, its contributions to the field of birdsong research and sound design, and potential avenues for future exploration in this domain.

Chapter 2

Theoretical Background

2.1 Dynamics and Bifurcation

Dynamics, a foundational pillar of physics, delves into the study of systems that evolve over time. It aims to understand how various elements within a system interact, change, and ultimately shape the system's behavior. These interactions and changes are often mathematically represented through differential equations, encapsulating the essence of a system's motion and behavior.

2.1.1 Key Terms

This section introduces key terms that would be useful in this work. They are given are the followings with reference from [9].

- **Equilibrium:** A state where all forces within a system are balanced, resulting in no change. Mathematically, for a system described by $\dot{x} = f(x)$, an equilibrium point x^* satisfies $f(x^*) = 0$.
- **Stability:** Refers to a system's capability to revert to its equilibrium after a disturbance. A system is deemed stable if minor disturbances lead to diminishing responses over time.
- **Linear Stability Analysis:** A method to ascertain the stability of an equilibrium by examining the behavior of minor perturbations around it. This involves linearizing the system around the equilibrium and analyzing the eigenvalues of the resulting Jacobian matrix.
- **Phase Space:** A multidimensional space where all possible states of a system are represented.
- **Phase Portrait:** A graphical representation in phase space, showcasing the system's trajectories.

- **Limit Cycle:** A closed trajectory in phase space. Mathematically, it's a periodic solution to the differential equations describing the system.
- **Parameter Space:** A space where different system parameters are plotted.
- **Bifurcation Diagram:** A graphical representation illustrating bifurcation points and system behavior as parameters vary.
- **Hopf Bifurcation:** A point where a system transitions between a stable equilibrium and a limit cycle. The onset of oscillatory behavior is typically indicated by a pair of complex conjugate eigenvalues crossing the imaginary axis.
- **Saddle Node Bifurcation:** A bifurcation where two equilibrium points merge and annihilate each other. It occurs when a real eigenvalue crosses zero.
- **Saddle-node on limit cycle(SNIC or referred as SNILC in some works):** A bifurcation where a limit cycle emerges from a saddle node on an invariant circle.

2.1.2 Bogdanov-Takens bifurcation

The **Bogdanov-Takens (BT) bifurcation** or **Bogdanov-Takens (BT) singularity** is a central concept in this work. It signifies a point where multiple system behaviors intersect. This leads to both a saddle-node and a Hopf bifurcation simultaneously, making it a particularly rich and complex bifurcation point in dynamical systems.

The BT bifurcation can be represented by the following ordinary differential equations (ODEs):

$$\dot{\xi}_0 = \xi_1, \tag{2.1}$$

$$\dot{\xi}_1 = \beta_1 + \beta_2\xi_0 + a_2\xi_0^2 + b_2\xi_0\xi_1. \tag{2.2}$$

This representation corresponds to a generic two-parameter unfolding of a codimension-2 BT when $a_2 b_2 \neq 0$.

In another context, the BT bifurcation can take the form as a generic three-parameter unfolding[10]:

$$\dot{\xi}_0 = \xi_1, \tag{2.3}$$

$$\dot{\xi}_1 = \beta_1 + \beta_2\xi_0 + \beta_3\xi_1 + a_3\xi_0^3 + b_2\xi_0\xi_1 + b_3'\xi_0^2\xi_1. \tag{2.4}$$

2.1.3 Normal Form

A normal form is essentially a simplified standard form of a system that captures its essential behavior near a particular point. One way to derive the normal form is as follows:

- **Identifying the Point of Interest:** Pinpoint the location in the system of interest, such as an equilibrium or bifurcation point.
- **Applying Taylor Expansion:** Around the identified point, use **Taylor expansion** to approximate the system's behavior. For a function $f(x)$, the expansion around $x = a$ is:

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2!} + \dots$$

- **Dimensional Reduction:** Truncate or neglect higher-order terms from the expansion that have minimal impact, simplifying the system's representation near the point of interest.
- **Achieving the Normal Form:** The truncated series becomes the system's **normal form**, a reduced-dimensional representation that captures the core dynamics near the point of interest.

2.2 Birdsong Production Mechanism

2.2.1 Avian Vocal System

The structure of bird vocal system is presented in figure 2.1. The primary components involved in birds' sound production include the lungs, bronchi, syrinx, trachea, larynx, mouth, and beak[1]. Not presented in the figure but no less essential is the air sac and a structure called oropharyngeal-esophageal cavity(OEC)[11].

Birds possess a respiratory system that is the most intricate and effective in terms

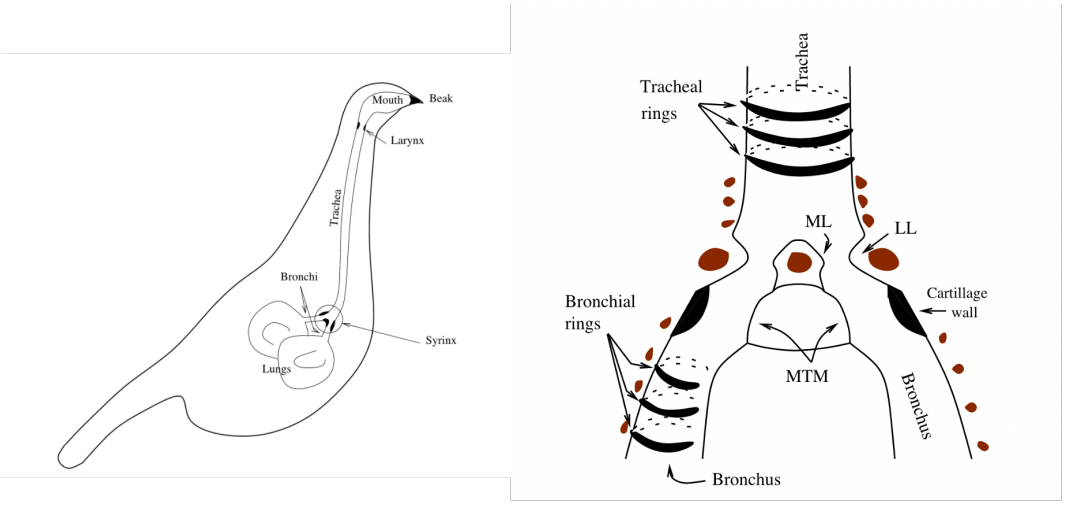


Figure 2.1: Left: avian sound producing organizations; Right:songbird syrinx[1]

of structure and function: a **lung-air sac system**[12]. In vocalization, this system provides a controlled and consistent airflow, which is critical for the modulation and sustenance of diverse avian sounds.

Syrinx

It is, however the **syrinx** that is the central organ in bird sound production, playing a significant role in bird taxonomy due to its varied anatomy across species. There are three main types of syrinx:

1. **Tracheobronchial**: Located at the trachea-bronchi junction.
2. **Tracheal**: Comprising complete cartilage rings.
3. **Bronchial**: Featuring incomplete C-shaped cartilage rings.

The distinction between these types can be nuanced due to overlapping characteristics.

When birds sing, airflow induces vibration in the medial syringeal tympaniform membrane (MTM) of each bronchus. The membranes have the ability to vibrate on their own, each with distinct base frequencies and patterns. These membranes operate based on pressure, similar to the reeds in woodwind instruments. However, while the membranes vibrate by opening up, reeds in woodwind instruments vibrate by closing[6]. However, more recent studies suggest that two soft tissues, medial and lateral labia (ML and LL), similar to human vocal cords, might be more crucial for sound production than MTM. Sound is generated when air flows through two oscillating tissues. [13]. However, the exact role and importance of MTM, ML, and LL can vary between bird species.

Trachea

The bird's **trachea**, positioned between the syrinx and larynx, acts as a resonator for sound. It consists of cartilage rings, typically complete (McLelland, 1989). The number of these rings varies with neck length, from about 30 in small passerines to around 350 in long-necked species like flamingos. In some birds, the trachea forms loops, extending its length beyond the neck, which may enhance sound frequency range (Gaunt et al., 1987). Unique tracheal adaptations in certain species, such as connections to air sacs or bifurcation, contribute to their distinct vocalizations.

Terminal Vocal Tract Components

Following sound initiation in the syrinx and its modulation in the trachea, the vocalizations pass through several anatomical structures before reaching the external

environment. The avian larynx, unlike in humans, lacks vocal folds and its role in sound production is debated. Birds' mouths act as cavity resonators with limited flexibility, and while they use their tongues to adjust the mouth's shape, the rigidity of most avian tongues restricts their sound production role. The beak, with its complex design, significantly modulates sound, especially through its opening and closing actions. While one traditional description of the vocal pathway includes the larynx, mouth, and beak, this study considers the oropharyngeal-esophageal cavity (OEC) as the final segment for avian vocal output, emphasizing its role in sound modulation, amplification, and directionality.

2.2.2 Diversity

The allure of birdsong lies not just in its melodic beauty but also in its remarkable diversity. This diversity, while making birdsong captivating, also introduces complexities in its modeling and understanding. A few facets of this diversity include:

Single vs. Dual Syrinx Usage: Birds are typically equipped with a syrinx, their unique vocal apparatus. However, the manner in which they use this organ can vary. While some species rely on a single syrinx to craft their songs, others harness both in tandem, resulting in a more complex and layered vocal output.

Tonal vs. Voiced Sounds: Birdsongs can be broadly categorized based on their acoustic properties. Some avian species produce pure, tonal sounds, reminiscent of musical notes. In contrast, others emit voiced sounds, which have a distinct texture and resonance, shaped by the bird's physiological attributes.

Mechanistic Variations: As highlighted in earlier sections, the mechanisms underpinning sound production can differ substantially among avian species. These variations can stem from differences in syrinx structure, the presence or absence of air sacs, and other anatomical distinctions.

Species-Specific Nuances: Delving deeper, individual bird species exhibit unique idiosyncrasies in their song production. These nuances can be influenced by a myriad of factors, from evolutionary lineage to habitat adaptations.

Grasping this diversity is paramount for both appreciating the richness of birdsong and for advancing computational modeling endeavors. By acknowledging and understanding these variations, we can ensure that the models we develop are both robust and versatile, capable of simulating a broad spectrum of avian vocalizations.

2.3 State of Art

2.3.1 Existing Physical Models

The avian vocal system has been the subject of numerous physical models over the years. Broadly, these models can be categorized into two primary branches. The first

branch draws parallels with reed instruments, focusing on membrane vibration. The second branch takes inspiration from human vocal mechanics, employing single, double, or multi-mass models. This dichotomy in modeling approaches mirrors the diverse theories surrounding song production. Elemans, in 2004, provided a comprehensive overview of the models available up to that point, as depicted in Figure 2.2.

Subsequent to Elemans' review, there has been a surge in models that attempt to bridge avian and human vocal production. For instance, Zaccarelli et al.[14]

Table I Comparison of models in literature. A. Brackenbury (1979b); B. Fletcher (1985); C. Fee et al. (1998); D. Fry (1998); E. Gardner et al. (2001); F. Laje et al. (2002); G. Fee (2002). [1] SD; Spring Damper model, [2] TM; Tympaniform Membrane

Model type	aeroacoustical			modified oscillators			
Model	A	B	C	D	E	F	G
Aim of model development							
Neuromuscular control			•	•	•	•	
Morphology							
General mechanisms	•	•					•
Bird group							
Passeriformes (song birds)		•	•	•	•	•	•
Non-songbirds	•						
MTM modelled by							
Moving piston	•						
Edge clamped drum		•					
Labia modelled by							
One/two mass SD[1]			•	•	•	•	
Multiple mass SD[1]							•
Input parameters							
Bronchial pressure	•	•	•	•	•	•	
Complex pressure wave		•					
Labial / membrane stiffness		•		•	•	•	•
Gating of flow						•	
Model output tested							
Generation sounds	•	•	•	•	•	•	
Acoustical power output		•		•			
Amplitude modulation				•			
Pressure gradient over TM[2]		•					

Figure 2.2: Models up to 2004, as summarized by Elemans[1]

experimented with two models: one that directly rescaled the human 2-mass model to emulate songbird vocalizations and another trapezoidal model tailored for the ring dove syrinx. While these models successfully demonstrated self-oscillations and yielded plausible parameter values, their outputs were not consistently corroborated with real-world observations.

Smyth and Smith ventured into another intriguing direction, leveraging the physics of birdsong as delineated by Fletcher[7][15]. Their model, rooted in the membrane model detailed in Section 2.2.1, incorporated Bernoulli flow as the nonlinear term responsible for self-oscillation. Notably, their work was less about biological accuracy and more about musical applications. They introduced innovative numerical methods in birdsong modeling, encompassing finite difference methods, waveguide synthesis, and discrete-time lumped models. However, a significant limitation of their approach was its inability to generate tonal sounds, underscoring the need for a deeper understanding of the underlying physics.

A particularly influential set of models, and the one this dissertation primarily relies on, has been developed by Mindlin and colleagues[4][5][16][17][18]. The standout features of this model include its ongoing updates, reflecting the latest biological discoveries, and its successful validation—sounds synthesized using this model have elicited responses from real birds. Rooted in the 1-mass model framework, a more detailed exploration of this model will be presented in the subsequent chapter.

2.3.2 Inverse Problem in Control Parameter Mapping

Contrary to the 'forward problem' where we predict the result from the cause, the **inverse problem** seeks the cause from the result. This has been an essential concept in many scientific domains. In some displines, the treatment of the inverse problem has been in pace with recent advancements in computational methods, optimization techniques, and machine learning, such as the use of regualization methods or neural networks.

In the topic of modeling birdsong discussed here, this mainly refers to **control parameter mapping**. So far relevant work in birdsong has been directed to **control parameter fitting** which is the main source of reference here. The main method uses the **lookup table**. In the work by Boari et al. [17]), a pre-computed table of control parameters was used, where each node corresponded to specific song features like the Spectral Content Index (SCI) and Fundamental Frequency (F0). Observed songs were cut into 20 ms segments and matched with the closest node in the table. Similarly in the work by Smyth et al. [19] control parameters were directly paired with the model's corresponding output power spectra and the maximum likelihood and minimum action method was used for inverse estimation.

In another recent work Arneodo et al. [20] explored the potential of **neural**

networks, using recorded neural activity to synthesize birdsong. Estimation of control parameters in the described biomechanical and physical model served as a middle step in the work. Even though this did not directly address the inverse problem, it pioneered the use of machine learning in parameter estimation in the field of birdsong.

One key challenge with inverse problems to put forward though is that they are often **ill-posed**, meaning that they do not have a unique solution, are highly sensitive to input data, or both. Ways to tackle this challenges are being explored with the advanced methods mentioned in the first paragraph. [21] In the field of birdsong, however, this has not been much studied.

2.3.3 Machine Learning in Physics-based Modelling

The combination of physics and machine learning has been a trend in many scientific fields, with various ways of leveraging the advantages of both disciplines [22]. This fusion has also seen significant advancements in the field of audio applications [23]. However, when it comes to applications in physics-based synthesis, there is still ample room for development. Current machine learning applications in synthesizer modeling typically focus on audio as either input or output [24], or primarily within the realm of creative synthesis [25], as opposed to physics-based synthesis.

The novel applications I envision in this context can encompass bridging synthesized sound with real-world sound, expanding data through faster computation, inversing physical parameters, and enhancing controllability. The latter is the primary focus of exploration here.

Chapter 3

Current Physical Model

As mentioned in Chapter 2, the model developed by Mindlin's group is chosen for this study due to its ongoing updates and validation through real bird responses. The following sections present the details of this model.

3.1 Syrinx

3.1.1 A Basic Model

Here describes a basic model that was studied in early 2000s. This model clearly explains the underlying physics, and holds for most nearly tonal birds, but cannot produce songs that are spectrally complex enough for other species such as zebra finch.[18]

The modeling here follows [18, 4, 26, 16], starting with a simple mass spring equation:

$$M\ddot{x} = -Kx - B\dot{x} + P_s - F_0 \tag{3.1}$$

with M being the mass of the labium, K being stiffness, B being dispersion term and P_s being the interlabial pressure.

F_0 is a force term that controls the labial's stationary position, which is constant and sometimes neglected in the modeling such as in [4, 16].

By treating F_0 as 0 as in [16] and in later updated models, dividing the equation by mass and rearrange it, it gives

$$\dot{x} = y, \tag{3.2}$$

$$\dot{y} = -kx - \beta y + p_s \tag{3.3}$$

Interlabial pressure

The interlabial pressure, described in detail in [4] takes reference from the 2 mass model of human vocal folds[27, 28] and calculates half the seperation of upper and lower edges as

$$a_1 = a_{10} + x + \tau y, \tag{3.4}$$

$$a_2 = a_{20} + x - \tau y, \tag{3.5}$$

where τ is the time taken for propagating wave to vertically travel half the distance of labia.

The average pressure therefore is calculated by

$$p_s = p_b \left(1 - \frac{a_2}{a_1}\right) \tag{3.6}$$

where p_b is the sublabial pressure.

One approximation is made here to give $1 - a_2/a_1 \sim y$, so that it arrives:

$$\dot{x} = y, \tag{3.7}$$

$$\dot{y} = -kx - (\beta - p_b)y \tag{3.8}$$

Nonlinear dispersion

At this point consider β as linear dispersion, we have the system reduced to simple harmonic oscillator when $\beta = p_b$, and it has damped oscillations when $\beta > p_b$. A nonlinear dispersion is added to avoid divergence of solutions when $\beta < p_b$ [18]. Taking the collision of labia into account, it is assumed to depend on both velocity and position [4] and is given by cx^2y. In some papers[26, 16], the final equations is given by

$$\dot{x} = y, \tag{3.9}$$

$$\dot{y} = -kx - (\beta - p_b)y - cx^2y \tag{3.10}$$

3.1.2 Updated Model

As said the previous model holds for most nearly tonal birds, it cannot produce songs that are spectrally complex enough for other species. Therefore further complexity was added in to produce sounds richer in harmonics.

First consider the complete form of (3.2)(3.3)(3.4)(3.5)(3.6), as well as the nonlinear

term we have

$$\dot{x} = y, \tag{3.11}$$

$$\dot{y} = -kx - \beta y - cx^2 y + p_b(\frac{\Delta a + 2\tau y}{a_{01} + x + \tau y}) \tag{3.12}$$

Different nonlinear refinements have been added in later work, which include having $k = k_1 + k_2 x^2$[29, 2], having $\beta = \beta_1 + \beta_2 y^2$[30]. The form more commonly used in later course of the work is given by

$$\frac{dx}{dt} = y, \tag{3.13}$$

$$\frac{dy}{dt} = (1/m)\left[-k(x)x - \beta(y)y - cx^2 y + a_{lab} p_{sub}(\frac{\Delta a + 2\tau y}{a_{01} + x + \tau y})\right] \tag{3.14}$$

[29, 30, 5] where a_{lab} is lateral labial area, p_{sub} is subglottal pressure, and $\Delta a = a_{01} - a_{02}$. Note here the equation is written without the redimention in (3.2)(3.3), so that the parameters k, b, f_0 have different dimention than previously described in the last subsection.

Bifurcation and reduction to normal form

As in figure A.1, the above system presents both Hopf bifurcation and a saddle node in a limit cycle (SNILC) bifurcation. The former generates tonal sound while the latter generates oscillations that are spectrally rich. Diverse ways of controlling frequency or doing frequency modulations are discussed in [2] that explains the need for reduction of the model that minimizes the number of parameters. The process is given in [30] starting from the Taylor expansion from the point where a Hopf curve meets

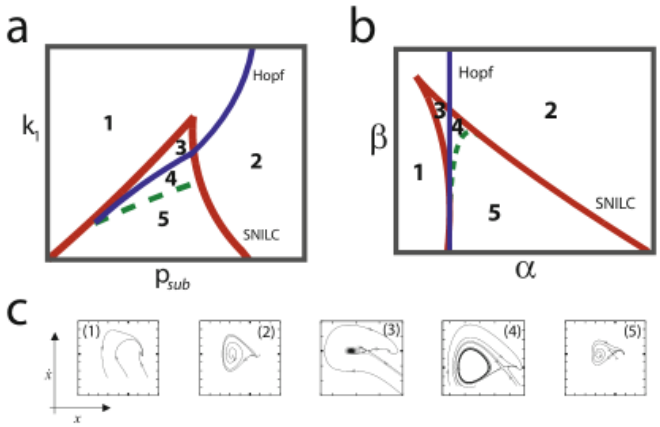


Figure 3.1: Parameter Space. a. original equations b. reduced equations c. phase portraits.[2]

tangentially a saddle-node curve, identified as Takens-Bogdanov singularity. The result, also after several updates in later years, gives:

$$\frac{dx}{dt} = y, \tag{3.15}$$

$$\frac{dy}{dt} = -\gamma^2\alpha - \gamma^2\beta x - \gamma^2 x^3 + \gamma^2 x^2 - \gamma xy - \gamma x^2 y \tag{3.16}$$

where γ is a time scaling factor. α and β are associated with air sac pressure(p) and labial tension (k) respectively. Previously p and k are identified as control parameters[28], and now they are represented by time-varying α and β.

3.2 Vocal Tract

3.2.1 Trachea

The trachea is approximated as a tube that has

$$p_i(t) = \alpha_{env}(t)x(t) - rp_i(t - \frac{2L}{c}) \tag{3.17}$$

$$p_o(t) = (1 - r)p_i(t - \frac{L}{c}) \tag{3.18}$$

where r is reflection coefficient, L is the length of trachea and c is velocity of sound in air.

$\alpha_{env}(t)$ is a coefficient proportional to the mean velocity of flow, while in the reconstructed syrinx function, flow is hard to calculate. In practice it is approximated

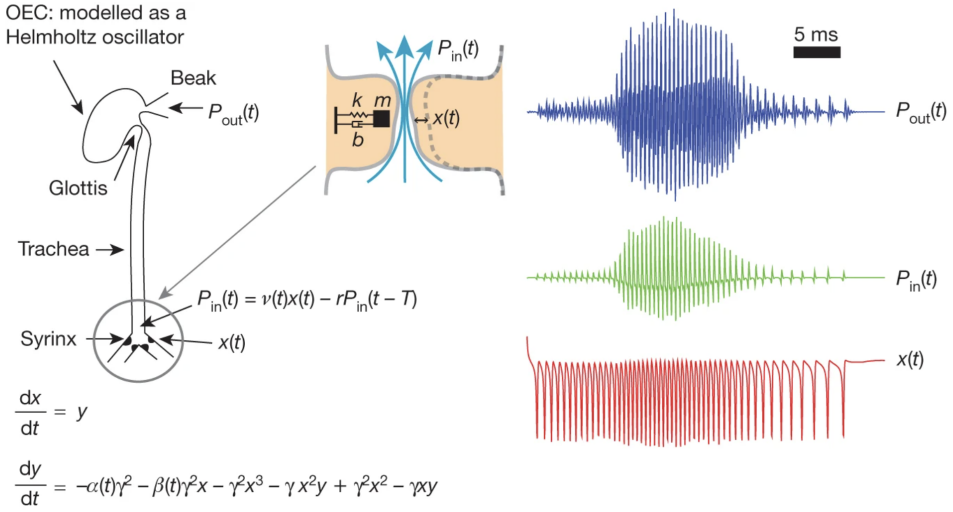


Figure 3.2: Avian vocalization mechanism[2]

by sound envelope since it has a monotonical relationship with pressure which is also monotonically related to the flow velocity.[17] In purely synthetic case, we approximate it to a constant.

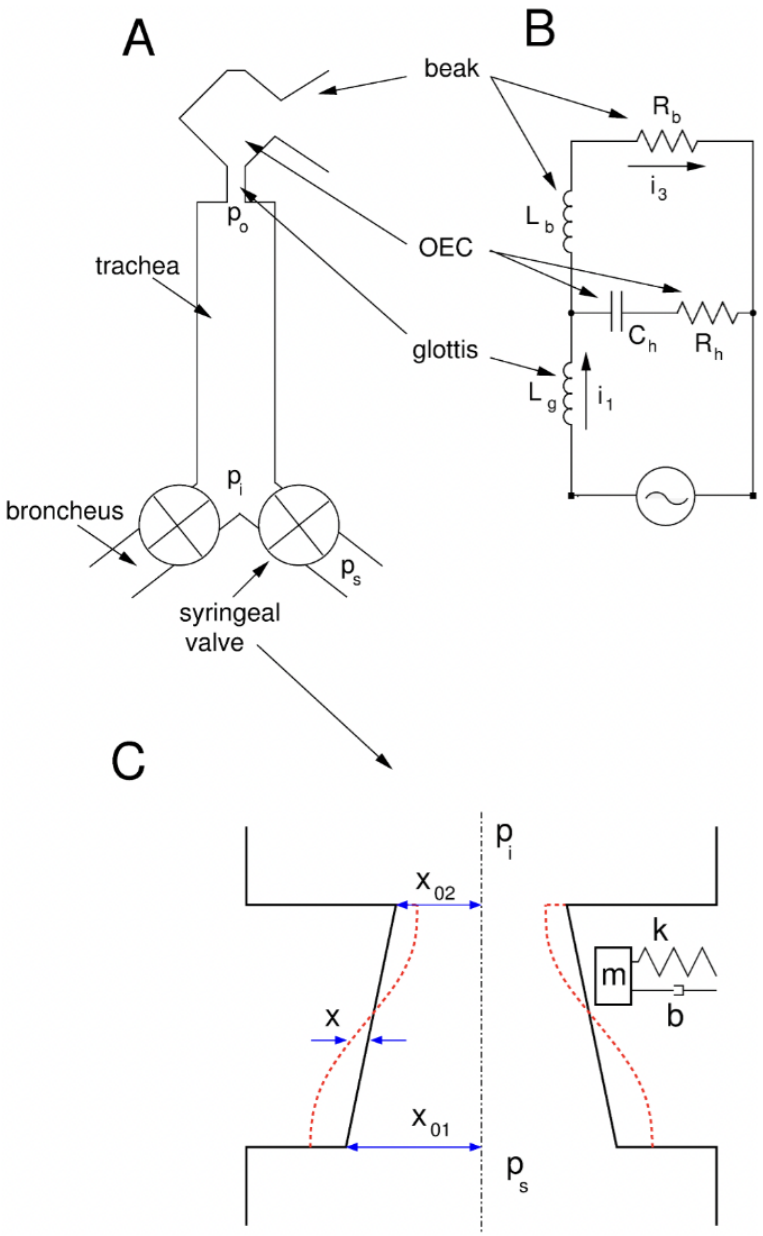


Figure 3.3: Physical models of bird vocal organs[3]

3.2.2 Oropharyngeal-esophageal Cavity

The oropharyngeal-esophageal cavity (OEC) is modelled as Helmhoz resonator[31, 32], and written into equivalent circuit as

$$\frac{di}{dt} = \Omega_1 \tag{3.19}$$

$$\frac{d\Omega_1}{dt} = -\frac{1}{L_g C_h} i_1 - R_h \left(\frac{1}{L_b} + \frac{1}{L_g}\right) \Omega_1 + i_3 \left(\frac{1}{L_g C_h} - \frac{R_b R_h}{L_b L_g}\right) \tag{3.20}$$

$$+ \frac{1}{L_g} \frac{dV_{ext}}{dt} + \frac{R_h}{L_g L_b} V_{ext} \tag{3.21}$$

$$\frac{di_3}{dt} = -\frac{L_g}{L_b} \Omega_1 - \frac{R_b}{L_b} i_3 + \frac{1}{L_b} V_{ext} \tag{3.22}$$

The drive V_{est} is proportional to the pressure at end of trachea $p_o(t)$ and the output of beak is proportional to $V_3 = R_b i_3$ hence proportional to i_3. With these considerations in mind, an alternative formulation of the equations is presented as follows[17]:

$$\frac{di_1}{dt} = a i_1 + b \Omega_1 + c i_3 + d \frac{dP}{dt} + eP \tag{3.23}$$

$$\frac{d\Omega_1}{dt} = f \Omega_1 + g i_3 + hP \tag{3.24}$$

3.3 Other Complexity

The complexity of this model can be expanded by considering several additional factors. One such factor is the inclusion of noise, as discussed in [11]. Another is the source-tract coupling, highlighted in [33]. Additionally, the 2-mass model for the syrinx adds another layer of intricacy. It's worth noting that both the coupling and the 2-mass model can lead to the emergence of subharmonics. However, these phenomena have been primarily studied in models that are simplified in other respects and are often overlooked in synthesis processes. The role of noise in this context will be further elaborated upon in Chapter 5.

Chapter 4

Numerical Implementation

In the preceding chapter, a series of ordinary differential equations (ODEs) that model avian vocalizations were presented. While these equations provide a theoretical framework for understanding the dynamics of the system, practical applications often necessitate actual solutions. Analytical solutions are less feasible here, therefore we look into reliable numerical approximations.

Two primary numerical methods are explored: the Runge-Kutta method, a conventional approach for such avian models, and a finite difference method that I designed. Each method has its unique advantages, and a comparison, along with the rationale for selecting one over the other, will be discussed by the end of this chapter.

4.1 Runge Kutta Method

4.1.1 Introduction

The Runge-Kutta method is one of the widely-used numerical techniques for ODEs. It provides an iterative approach to approximate the solution of an ODE without requiring the explicit solution. The method is based on the principle of evaluating the slope of the solution at several points within each time step and then combining these slopes to produce an estimate of the solution.

4.1.2 4th Order Runge-Kutta Method

As one of the most commonly used variants of the Runge-Kutta method, the 4th order Runge-Kutta (RK4) is considered here to achieve higher accuracy while preserving reasonable computational cost. The method evaluates the function four times per time step. The form of equations used here follow [34][4] for dynamical systems and birdsongs respectively. For a system of equations

$$\dot{x} = f(x), \tag{4.1}$$

calculate:

$$k_1 = f(x_n)k \tag{4.2}$$

$$k_2 = f\left(x_n + \frac{1}{2}k_1\right)k \tag{4.3}$$

$$k_3 = f\left(x_n + \frac{1}{2}k_2\right)k \tag{4.4}$$

$$k_4 = f(x_n + k_3)k. \tag{4.5}$$

where k is time step size.

The solution is then updated as:

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{4.6}$$

4.1.3 Implementation

The implementation of the 4th order Runge-Kutta method is straightforward. We only need to feed the set of ODEs to solve into the numerical solver function, as it works the same for all ODEs. Before that, special attention needs to be paid to the equation for trachea, as it involves time delay and is calculated by delay line:

$$p_i^n = x^n - rp_i^{n-2L_d}, \quad p_o^n = (1-r)p_i^{n-L_d} \tag{4.7}$$

where $L_d = \frac{L}{ck}$ is the number of samples corresponding to the delay $\frac{L}{c}$, and k is the time step size, same with sampling period in this work.

Moreover, the equations for OEC also inherently a derivative that needs differential approximation. Backward difference is used here with details give in section 2.2.1. The approximation gives:

$$\dot{p_o}^n = \frac{p_o^n - p_o^{n-k}}{k} \tag{4.8}$$

Finally we design the ODE system, several ways are tested:

A. ODEs 1 for syrinx, delayline for trachea, ODEs 2 for OEC;

B. Delayline for trachea, ODEs for syrinx and trachea, which was the implememtation in [20];

C. All equations in one ODEs;

Note again since the model itself is an approximation, we pay more attention to the output waveform or spectrogram instead of accurate values, thus even though the above methods comes slightly different, all results are still valid and comparable.

4.2 Finite Difference Method

4.2.1 Difference Operators

In discrete time simulation, consider $u = u^n$ as an approximation to $u(t)$ at $t = nk$, where n is an integer and k is the given time step, which same as the previous section, equals sampling period in this work.

Unit shifts e_{t+} and e_{t-} are defined as

$$e_{t+}u^n = u^{n+1}, \quad e_{t-}u^n = u^{n-1}. \tag{4.9}$$

Forward, backward, and centered difference approximations to a first time derivative as (δ_{t-} is the operator used in 2.1.3 to approximate the derivative of trachea's output pressure):

$$\delta_{t+} = \frac{e_{t+} - 1}{k}, \quad \delta_{t-} = \frac{1 - e_{t-}}{k}, \quad \delta_{t\cdot} = \frac{e_{t+} - e_{t-}}{2k} \tag{4.10}$$

an approximation to a second time derivative as

$$\delta_{tt} = \frac{e_{t+} - 2 + e_{t-}}{(k)^2} \tag{4.11}$$

and various averaging operators as

$$\mu_{t+} = \frac{e_{t+} + 1}{2}, \quad \mu_{t-} = \frac{1 + e_{t-}}{2}, \quad \mu_{t\cdot} = \frac{e_{t+} + e_{t-}}{2}. \tag{4.12}$$

4.2.2 State-Space Form

The subsequent section presents a linear state-space representation. While this does not strictly involve finite difference itself, it will be employed and discretized in the modeling of the OEC. The representation can be described as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.13}$$

$$y = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{4.14}$$

where $\mathbf{x}$ is state vector, $\mathbf{u}$ is input vector and y is the output.

4.2.3 Trapezoid Rule

The trapezoidal rule works by calculating the area beneath the curve of the given function via representing it as a trapezoid. It has:

$$\int_{t-k}^{t} f(x(\tau))\, d\tau \approx \frac{k}{2}\left[f(x(t)) + f(x(t-k))\right] \tag{4.15}$$

In discrete time this gives

$$x^n = x^{n-1} + \frac{k}{2}\left[f(x^n) + f(x^{n-1})\right] \tag{4.16}$$

In state-space formalism it writes with $\mathbf{I}$ being identity matrix:

$$(\mathbf{I} - \frac{k\mathbf{A}}{2})\mathbf{x}^{n+1} = (\mathbf{I} + \frac{k\mathbf{A}}{2})\mathbf{x}^n + \frac{k}{2}\mathbf{B}(\mathbf{u}^n + \mathbf{u}^{n+1}) \tag{4.17}$$

$$y^n = \mathbf{C}\mathbf{x}^n \tag{4.18}$$

The stability condition has the system is unconditionally stable when All the eigenvalues of matrix $\mathbf{A}$ have non-positive real parts.

4.2.4 Implementation

The implementation was developed incrementally, capitalizing on the model's source-filter separation assumption. It is referred to model D in the next section.

Syrinx

Rewrite the equation (3.15)(3.16) as

$$\frac{d^2x}{dt^2} = -\gamma(x + x^2)\frac{dx}{dt} - \gamma^2\alpha - \gamma^2\beta x - \gamma^2 x^3 + \gamma^2 x^2 \tag{4.19}$$

Use the difference operators in section 2.2.1 and represent the cubic term x^2 as $x^2\mu_t.x$, we arrive at the following operator form:

$$\delta_{tt}x = -\gamma(x + x^2)\delta_t.x - \gamma^2\alpha - \gamma^2\beta x - \gamma^2 x^2\mu_t.x + \gamma^2 x^2 \tag{4.20}$$

The update form is therefore given by

$$\begin{aligned} x^{n+1} = {} & \frac{1}{k\gamma(x^n + (x^n)^2) + k^2\gamma^2(x^n)^2 + 2} \\ & \times \left(-2k^2\gamma^2\alpha(i) - 2k^2\gamma^2\beta(i)x^n + 2k^2\gamma^2(x^n)^2 + 4x^n\right. \\ & \left.+(-k^2\gamma^2(x^n)^2 + k\gamma x^n + k\gamma(x^n)^2 - 2)x^{n-1}\right) \end{aligned} \tag{4.21}$$

Trachea

The trachea is still modelled as delay lines given by (4.7):

$$p_i^n = x^n - rp_i^{n-2L_d}, \quad p_o^n = (1-r)p_i^{n-L_d}$$

OEC

Rewrite the equations as follows:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{p} \tag{4.22}$$

$$\mathbf{y} = \mathbf{C}^T\mathbf{x} \tag{4.23}$$

where

$$\mathbf{x} = \begin{bmatrix} i_1 \\ \Omega_1 \\ i_3 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} \dot{p_o} \\ p_o. \end{bmatrix} \tag{4.24}$$

The derivative $\dot{p_o}$ is computed in a manner consistent with (4.8). The system matrices read

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ a & b & c \\ 0 & f & g \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ d & e \\ 0 & h \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{I} = \text{eye}(3). \tag{4.25}$$

Following (4.17) the final update form gives

$$\mathbf{x}^{n+1} = \left(\mathbf{I} - \frac{k\mathbf{A}}{2}\right)\left(\left(\mathbf{I} + \frac{k\mathbf{A}}{2}\right)\mathbf{x}^n + \frac{k}{2}\mathbf{B}\left(\mathbf{p}^{n+1} + \mathbf{p}\right)\right) \tag{4.26}$$

4.3 Comparison and Discussion

4.3.1 Time Efficiency

As previously mentioned and supported by literature, the Runge-Kutta 4th order method boasts a higher accuracy compared to the finite difference method we've developed. However, when the output waveform and spectrogram are sufficiently similar, computational efficiency becomes our primary concern. This is illustrated in table 4.1.

The table reveals that the finite difference method operates approximately 8.5 times more swiftly than the most efficient RK4 implementation. This advantage can be further enhanced during the data generation phase, which will be discussed later. Consequently, this model has been chosen for in-depth evaluation and further discussions.

	A	B	C	D
TI, 0.2s sound	0.619281s	0.764654s	0.543868s	0.063981s
TV, 0.5s sound	1.571319s	1.745776s	1.235279s	0.146235s

Table 4.1: Run Time Comparison. TI: time-invariant alpha, beta TV: time-varying alpha, beta. Step Size: 1/(4*48000) s.

4.3.2 Stability

In this section, we briefly touch upon the topic of stability. Given that the RK4 method can be viewed as a replication of established literature, and considering that none of the referenced papers addressed stability, it is not a focal point in our discussion either. As for the finite difference method, the inherent nonlinearity of the functions suggests stability calculations based on energy conservation. However, given that our model is anchored in equations that have been mathematically or dynamically transformed, the conventional energy conservation principle may not hold true in this context. Despite these complexities, we undertook a cursory stability assessment by varying the step size, and reassuringly, no instability was observed within our target variable range.

Chapter 5

Evaluation and Analysis

This chapter evaluates the model simulated in the previous chapter using the finite difference method by comparing the bifurcation curves, output waveforms, and exploring the effect of control parameters. Additionally, it investigates the possibility of inverse mapping of perceptual parameters to control parameters.

5.1 Bifurcation and Parameter Space

5.1.1 Calculation and Plot

First we explore the parameter space and calculate bifurcation. Revisit the equation of syrinx after reduction to normal form (3.15) (3.16):

$$\begin{aligned}\frac{dx}{dt} &= y, \\ \frac{dy}{dt} &= -\gamma^2\alpha - \gamma^2\beta x - \gamma^2 x^3 + \gamma^2 x^2 - \gamma x y - \gamma x^2 y\end{aligned}$$

We first linearize the equations around fixed points:

$$\frac{dx}{dt} = y = 0, \tag{5.1}$$

$$\frac{dy}{dt} = -\gamma^2\alpha - \gamma^2\beta x - \gamma^2 x^3 + \gamma^2 x^2 - \gamma x y - \gamma x^2 y = 0 \tag{5.2}$$

Therefore we have:

$$\alpha = -\beta x + x^2 - x^3 \tag{5.3}$$

that gives the relationship between fixed points, alpha and β. To calculate bifurcation, a geometrical approach is taken here:

Instead of explicitly finding roots, we interpret equation (5.3) as two curves: $f(x) = \alpha$ and $g(x) = -\beta x + x^2 - x^3$. In this context, equation (5.3) can be equated to

$f(x) = g(x)$, indicating that the fixed points correspond to the intersections of $f(x)$ and $g(x)$ geometrically. These curves also delineate stable and unstable regions. Therefore, to identify the critical points where stability transitions occur, we look for points where the two curves are tangent to each other. This gives

$$\beta = 2x - 3x^2 \tag{5.4}$$

In coding, we find x roots first, given by

$$x_{1,2} = \frac{1 \pm \sqrt{1 - 3\beta}}{3} \tag{5.5}$$

with x required to be real, there is the constraint $\beta \leq \frac{1}{3}$. We can therefore have α from equation (5.3) that gives saddle-node bifurcation. Hopf bifurcation could be given directly by $\alpha = 0$. The plot of the curves in the parameter space is presented as figure 5.1, which is identical to the graph presented in the literature.

It is also helpful to have an explicit expression of α in terms of β. Bring equation (5.5) into equation (5.3) we can arrive at

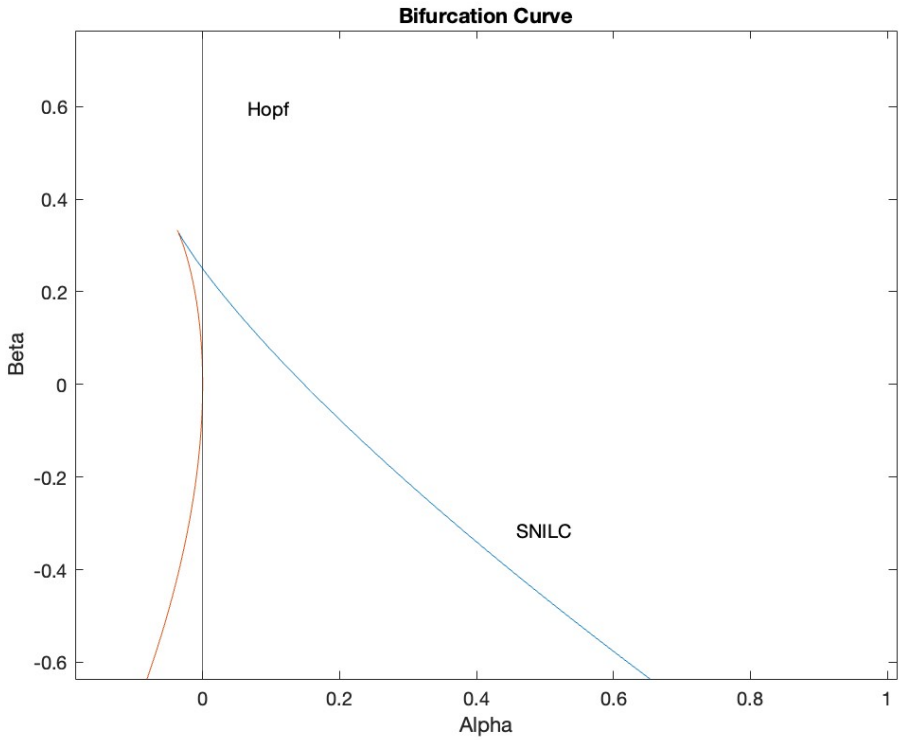


Figure 5.1: Bifurcation curves in (α, β) parameter space

$$a = -\frac{1}{27}\left(-2 + 9\beta \pm (-2 + 6\beta)\sqrt{1 - 3\beta}\right) \tag{5.6}$$

5.1.2 Influence on Output

We're now turning our attention to how bifurcations affect the output. Based on findings from Amador et al.[11], the phonetic region is the top right area of the parameter space, the part above the SNILC bifurcation and to the right of the Hopf bifurcation.

Here presents first 2 situations, one born in Hopf bifurcation in figure 5.2, which gives a tonal sound with a defined frequency and zero amplitude. The other born in SNILC bifurcation in figure 5.3 has the spike-like wave with infinite period and spectrally rich. These features along with the characteristic waveforms all align well with those described in literature[3].

By setting α and β as constants, it becomes even more evident that as control parameters approach the SNILC bifurcation, the spectrogram becomes richer in content. Figures 5.4 and 5.5 illustrate sounds originating away from and near the SNILC bifurcation, respectively. This observation will be further substantiated in subsequent sections.

5.2 More Complexity

5.2.1 Various Trajectories

To further our understanding of the avian vocal system, we can design additional trajectories, each offering unique insights. Here presents some examples in figure 5.6, 5.7 and 5.8. Corresponding audio representations of these trajectories are provided in the supplementary materials.

5.2.2 β with Noise

In [11] [17], it was highlighted that the absence of noise in synthetic gestures could elicit only weak responses from birds. Following these findings, we experimented with introducing noise into our model as well. Specifically, Gaussian noise, three times smaller in magnitude than that of the range of β, was added. A comparative analysis of plots with and without noise is presented in figure 5.9. While the graphical differences might appear subtle, the auditory output with the introduced noise is perceptibly more authentic. Their corresponding audio samples are also available in the supplementary materials.

5.3 Perceptual Mapping

Manipulating control parameters directly in bird vocalization models presents a challenge. Determining how these parameters influence perceptual outcomes, such as frequency, is not always straightforward. To address this, a perceptual mapping method was introduced. This approach bridges the gap between the abstract parameters and the sounds that are perceived, facilitating the production of desired vocalizations and simplifying the inverse problem.

Throughout the development of this mapping, a correlation between perceptual parameters (f_0, SCI) was unveiled independently. This discovery was made possible by the introduction of a novel remapped parameter, which visually highlighted the correlation. This insight offers a fresh perspective, complementing and enriching the current literature on the subject.

5.3.1 Perceptual Parameters

In the literature, particularly in the works of Mindlin's group, two primary parameters have been utilized for the inverse mapping of bird songs: f_0 and SCI. These parameters are not only pivotal for the inverse process but also serve as effective metrics for perceptual mapping. We adopt these parameters for our study and provide their definitions below:

- f_0: The fundamental frequency;
- SCI (Spectral Richness Index): This index quantifies the richness of the song's spectrum. It is calculated using the formula:

$$\text{SCI} = \frac{SpectralCentroid}{f_0} \tag{5.7}$$

  where the Spectral Centroid is calculated by $\frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}$.

5.3.2 Data Acquisition

This section details the method employed for data collection pertaining to mapping. In this phase, control parameters were treated as constants. Vectorization was executed based on the prior model script to facilitate parallel computation and expedite data generation. The control parameter ranges were defined as β: [-0.6490, 2.5] and α: [0.0025, 0.6686], yielding an f_0 range between [375,6047], consistent with the range observed in zebra finch songs.

Initial attempts to approximate the SNILC bifurcation using either a straight line or a polynomial curve resulted in a minimum frequency exceeding 1000Hz, which did

not align with our desired frequency range. Consequently, a more refined strategy was devised.

To ensure a balanced distribution of data, an additional parameter, c, was introduced. It was computed based on the desired β range, determined by values exponentially spaced between [-9.9658,1.1699]. This was added to the linearly spaced [-0.65, 0.25] to achieve the targeted β range. α was derived along the bifurcation using equation 5.6, with a slight offset of 0.0025. The data points are illustrated in figures 5.10 and 5.11:

The challenge of accurately estimating f_0 was tackled. Several methods, including auto-correlation and the YIN technique, were evaluated. The most consistent results were obtained by identifying the first peak of either power spectrum or frequency plot post the application of a 2% highest peak threshold, in line with findings from other academic research. Notably, existing studies favored the power spectrum [17][19], whereas either the power or direct FFT produced the same results in this work.

For consistency and potential future real-world sound analyses, a Gaussian windowing technique was employed during data collection. With step size of (1/4*48000)s, a window size of 8192 samples (0.04s) and a 50% overlap, data could be able to be captured at 20ms intervals. This level of detail was deemed appropriate for analyzing bird songs, which are characterized by their swift frequency transitions. The chosen window size and overlap were found to strike an optimal balance between efficiency and precision.

It was also noted that the initial output samples displayed varied amplitudes. While these could have been excluded, the decision was made to retain them, given their minimal influence on the overall findings.

Performance-wise, the script showcased impressive efficiency, generating 10,000 datasets of 8192 samples (0.4s) with a step size of 1/(4*48000) in just 9.3 seconds for all model outputs and 13.6s for the complete dataset, including f_0 and SCI calculations.

5.3.3 Visualization

The relationship between the control parameters (including the remapped c) and the perceptual parameters is visualized in figure 5.12, 5.13.

5.3.4 Correlation Between f_0 and SCI

Notably, figure 5.13 clearly highlights the correlation between f_0 and SCI. Another way of displacing such correlation could be done directly by plotting the relationship of f_0 and SCI, given in figure 5.14

5.4 Inverse

Given the correlation identified in the preceding section, it's logical to simplify the inversion process by focusing solely on the data of (β) and f_0, a strategy also employed in [17]. In addition to the look up table method introduced in [17], two alternative methods are explored. One involves interpolation between the data points, inspired by [35]. The other employs a machine learning approach, an original idea in this work. The methods discussed can be succinctly summarized as:

A. Perceptual Parameter (f_0) → Lookup Table → Control Parameter (β or c)

B. Perceptual Parameter (f_0) → Interpolation → Control Parameter (β or c)

C. Perceptual Parameter (f_0) → Support Vector Machine → Control Parameter (β or c)

Furthermore, while this study doesn't delve into it, the inverse process holds potential not just for enhanced control and playability but also for deducing control parameters directly from raw audio and subsequent resynthesis. A general workflow for this would center around:

Audio → Our Inverse → Resynthesized Sound

In the implementions, the parameter α is fixed at a value of 0.256, which corresponds to $\beta = -0.15$ on the SNILC bifurcation curve. A new script is designed to generate a dataset comprising the parameters α (fixed), β, c, and f_0, which is subsequently saved in the data0.mat file. Additionally, the updated relationships among f_0 and β or c are visualized in figure 5.15.

To ensure the accuracy and robustness of the data, especially for its later use in the inverse table, the number of samples is extended to 6*8192, equivalent to a duration of 0.24 seconds. This extension is crucial to prevent any potential overlap or repetition of f_0 values in the dataset.

5.4.1 Lookup Table

As previously mentioned, the utilization of a lookup table is straightforward: simply identify the point in the table that has the smallest difference from the given frequency. Any pair of data in figure 5.15 will yield the same result using this method. A single transformation of f_0 to a logarithmic scale is performed to better align with human auditory perception. To assess the accuracy of our model, we determine the relative error between the synthesized fundamental frequency, $f_{0_{\text{act.}}}$, and the anticipated fundamental frequency, $f_{0_{\text{exp.}}}$. The relative error is computed as:

$f_{0_{\text{exp.}}}(Hz)$	440	880	1760	3520	5920
β	-0.1478	-0.1306	-0.0580	0.4478	2.1001
$f_{0_{\text{act.}}}(Hz)$	435.9	885.9	1743.8	3543.8	5939.1
$error(\%)$	0.923	0.675	0.923	0.675	1.158

Table 5.1: Example $f_{0_{\text{exp.}}}$, resulting β, resulting fundamental frequency $f_{0_{\text{act.}}}$ and error by lookup table.

$f_{0_{\text{exp.}}}(Hz)$	440	880	1760	3520	5920
β	-0.1478	-0.1308	-0.0557	0.4371	2.0847
$f_{0_{\text{act.}}}(Hz)$	440.6	881.3	1757.8	3520.3	5920.3
$error(\%)$	0.142	0.142	0.124	0.009	0.005

Table 5.2: Example $f_{0_{\text{exp.}}}$, resulting β, resulting fundamental frequency $f_{0_{\text{act.}}}$ and error by interpolation.

$$\text{error} = \frac{\left|f_{0_{\text{act.}}} - f_{0_{\text{exp.}}}\right|}{f_{0_{\text{exp.}}}} \tag{5.8}$$

Example $f_{0_{\text{exp.}}}$, resulting β, resulting fundamental frequency $f_{0_{\text{act.}}}$ and error are given in the table 5.1.

5.4.2 Interpolation

Several attempts were made to fit known functions, including linear, polynomial, logarithmic, exponential, sigmoid etc. curves, to the plots in figure 5.15. However, none provided a satisfactory fit. Consequently, linear interpolation between each data point was employed, aligning with the approach used in [35]. The data pair ($log(c)$, $log(f_0)$) is used here. Some results transformed back in β and f_0 from this method are presented in the table 5.2.

5.4.3 Support Vector Machine

Finally Support Vector Machine is used for the regression. Different data pairs are explored as well and the pairs of (log(f_0), β) and (f_0, β) give the best result. Kernels and hyperparameters are set as table 5.3. Fitted curves are given in figure 5.16, 5.17.

We also compute the Mean Squared Error (MSE) to assess the accuracy of our model. The MSE is mathematically defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{\text{pred},i})^2 \tag{5.9}$$

where n represents the total number of data points. For the two fittings we considered, the computed MSE values are 2.6559×10^{-4} and 2.7957×10^{-4}, respectively.

	Kernel Function	Box Constraint	Kernel Scale
($\log(f_0)$, β)	rbf	100	1.1
(f_0, β)	rbf	10	2200

Table 5.3: Kernels and hyparameters for SVM

$f_{0_{\text{exp.}}}(Hz)$	440	880	1760	3520	5920
β	-0.1437	-0.1547	-0.0330	0.4144	2.0628
$f_{0_{\text{act.}}}(Hz)$	576.6	3609.4	1926.6	3469	5896.9
$error(\%)$	31.04	310.2	9.464	1.456	0.391

Table 5.4: Example $f_{0_{\text{exp.}}}$, resulting β, resulting fundamental frequency $f_{0_{\text{act.}}}$ and error by SVM regression on ($log(f_0)$, β).

Example $f_{0_{\text{exp.}}}$, resulting β, resulting fundamental frequency $f_{0_{\text{act.}}}$ and error by SVM of pair ($\log(f_0)$, β) and pair (f_0, β) are given in table 5.4, 5.5 respectively.

5.4.4 Comparison and Conclusion

When comparing the results from the tables, it is evident that the interpolation approach yields the most accurate results, which is to be expected. By creating a continuous link between neighboring data points, or in coding terms, by interpolating a vast amount of data between these points, the interpolation method overcomes the inherent limitation of the direct lookup table approach. Specifically, the lookup table can only provide results that exist within its original dataset, making it incapable of offering values outside this set.

The accuracy of the support vector machine is contingent on both the dataset it is trained on and the specifics of the fitting process. In the current configuration, it demonstrates superior performance at higher frequencies compared to lower ones. This discrepancy is likely attributed to the denser distribution of data at lower frequencies. However, it is noteworthy that the support vector machine can discern the general trend in data that could not be mapped to any recognized function in our experiments. As the model evolves to accommodate more parameters, the support vector machine might prove more invaluable, especially when tackling ill-posed problems.

$f_{0_{\text{exp.}}}(Hz)$	440	880	1760	3520	5920
β	-0.1303	-0.1491	-0.0649	0.4410	2.0646
$f_{0_{\text{act.}}}(Hz)$	890.6	384.4	1687.5	3529.7	5896.9
$error(\%)$	102.4	56.32	4.119	0.275	0.391

Table 5.5: Example $f_{0_{\text{exp.}}}$, resulting β, resulting fundamental frequency $f_{0_{\text{act.}}}$ and error by SVM regression on (f_0, β).

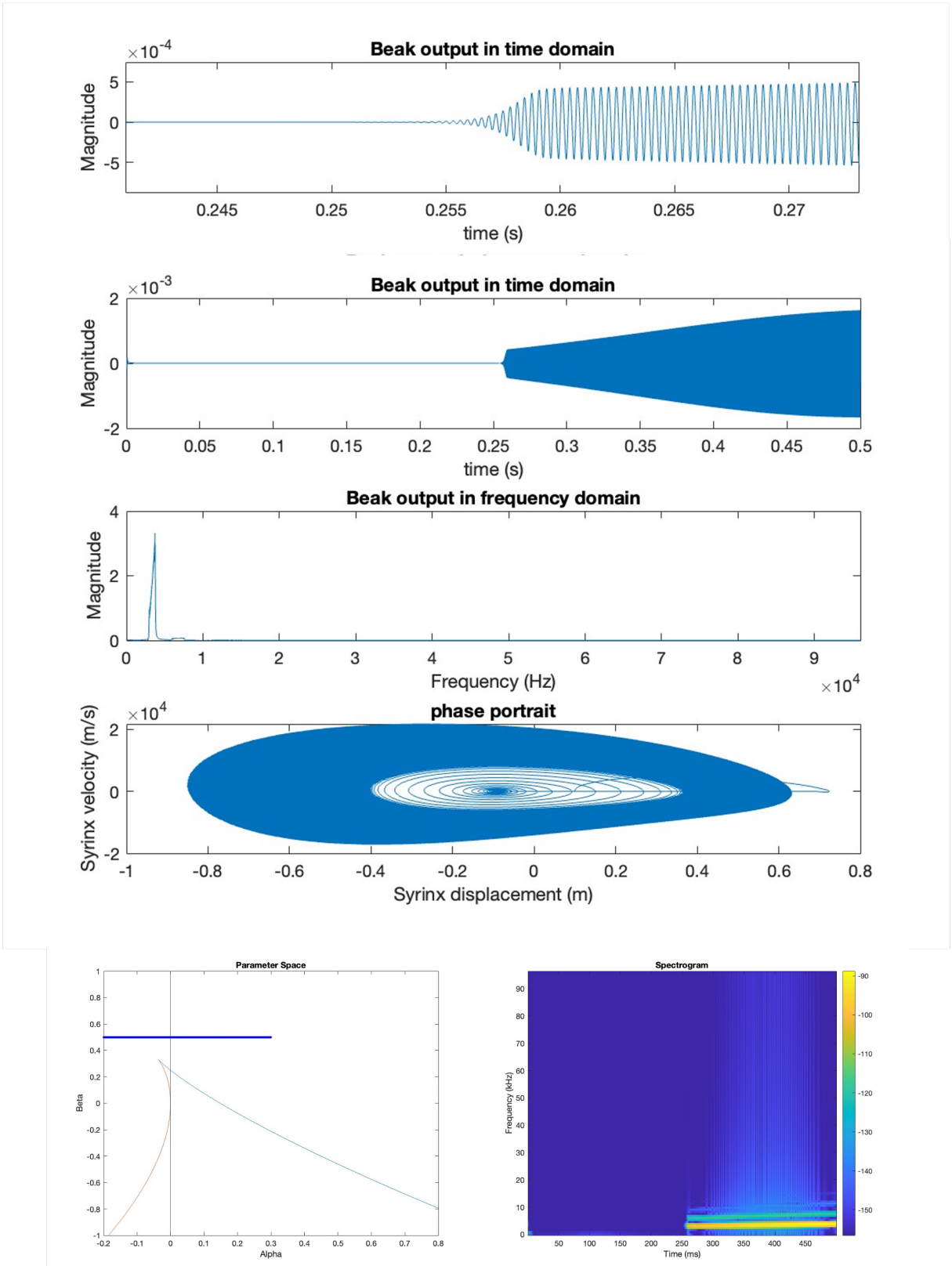


Figure 5.2: Sound originating from the Hopf bifurcation. Displayed from top to bottom are: beak output in the time domain (zoomed in), beak output in the time domain (full length), beak output in the frequency domain, phase portrait of the syrinx, trajectory in the (α,β) space (bottom left), and the spectrogram (bottom right).

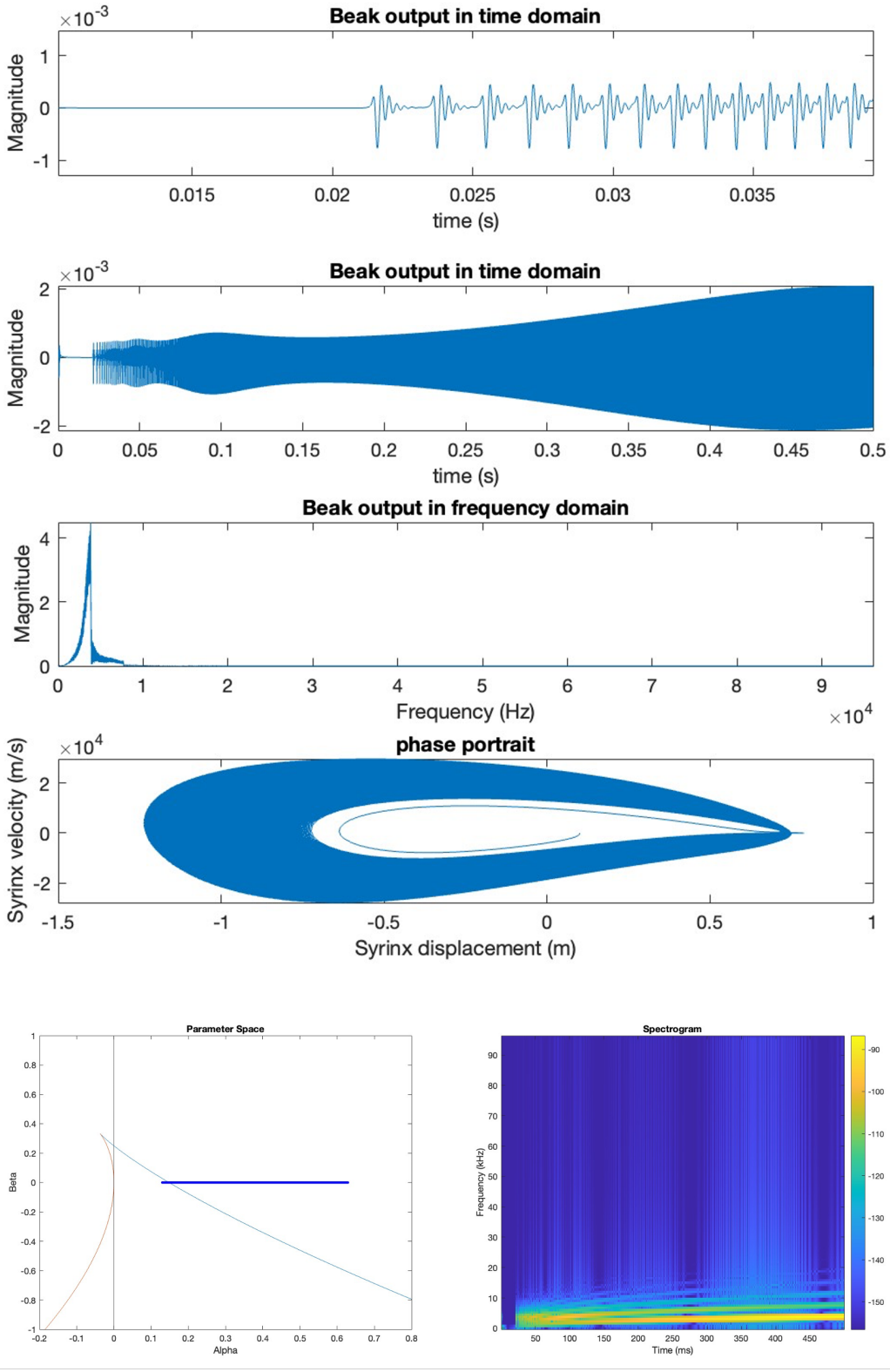


Figure 5.3: Sound originating from the SNILC bifurcation. Displayed from top to bottom are: beak output in the time domain (zoomed in), beak output in the time domain (full length), beak output in the frequency domain, phase portrait of the syrinx, trajectory in the (α,β) space (bottom left), and the spectrogram (bottom right).

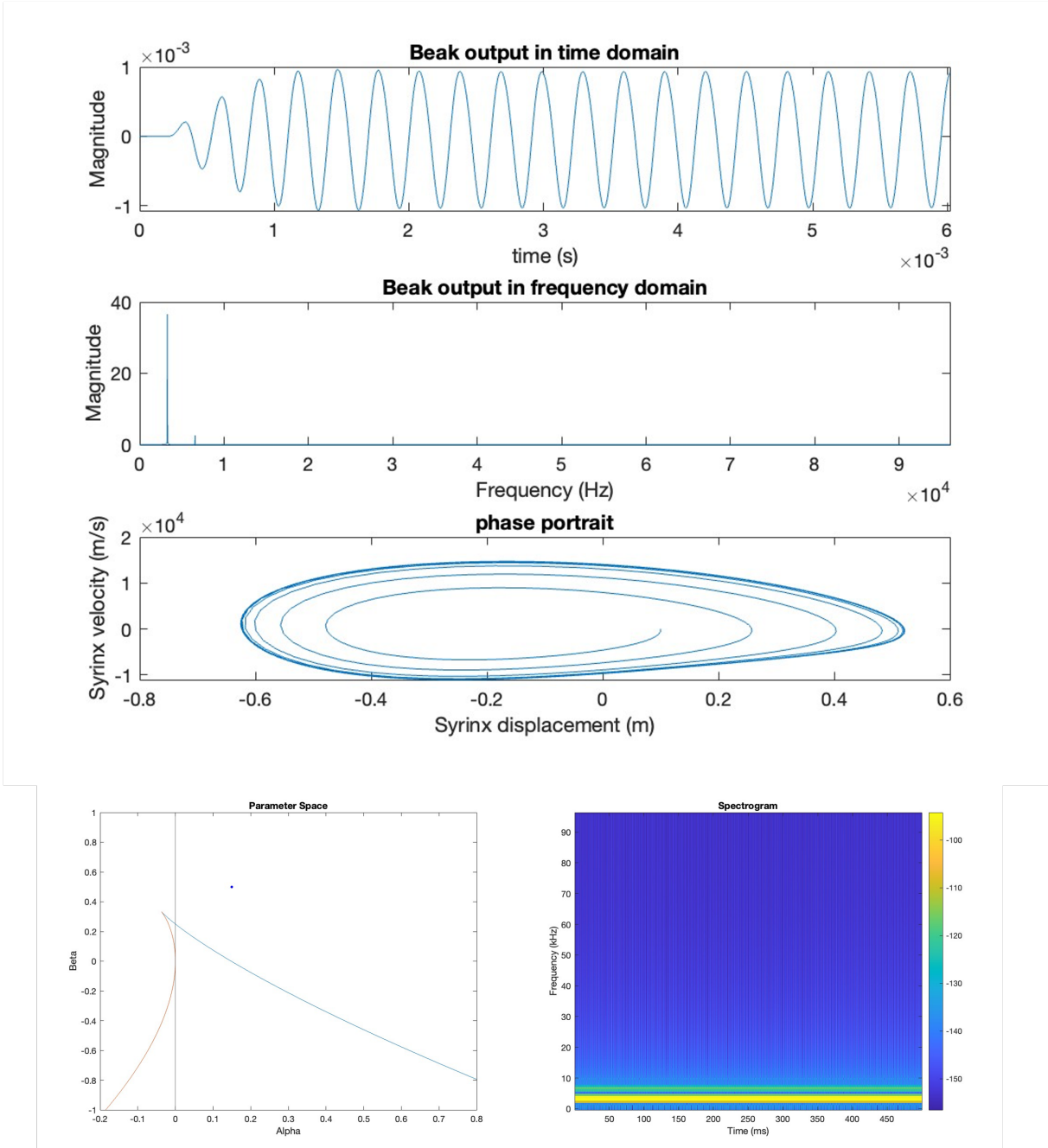


Figure 5.4: Sound generated away from the SNILC bifurcation. From top to bottom, the figure displays: beak output in the time domain (zoomed in), beak output in the frequency domain, phase portrait of the syrinx, position in the (α, β) space (bottom left), and the spectrogram (bottom right).

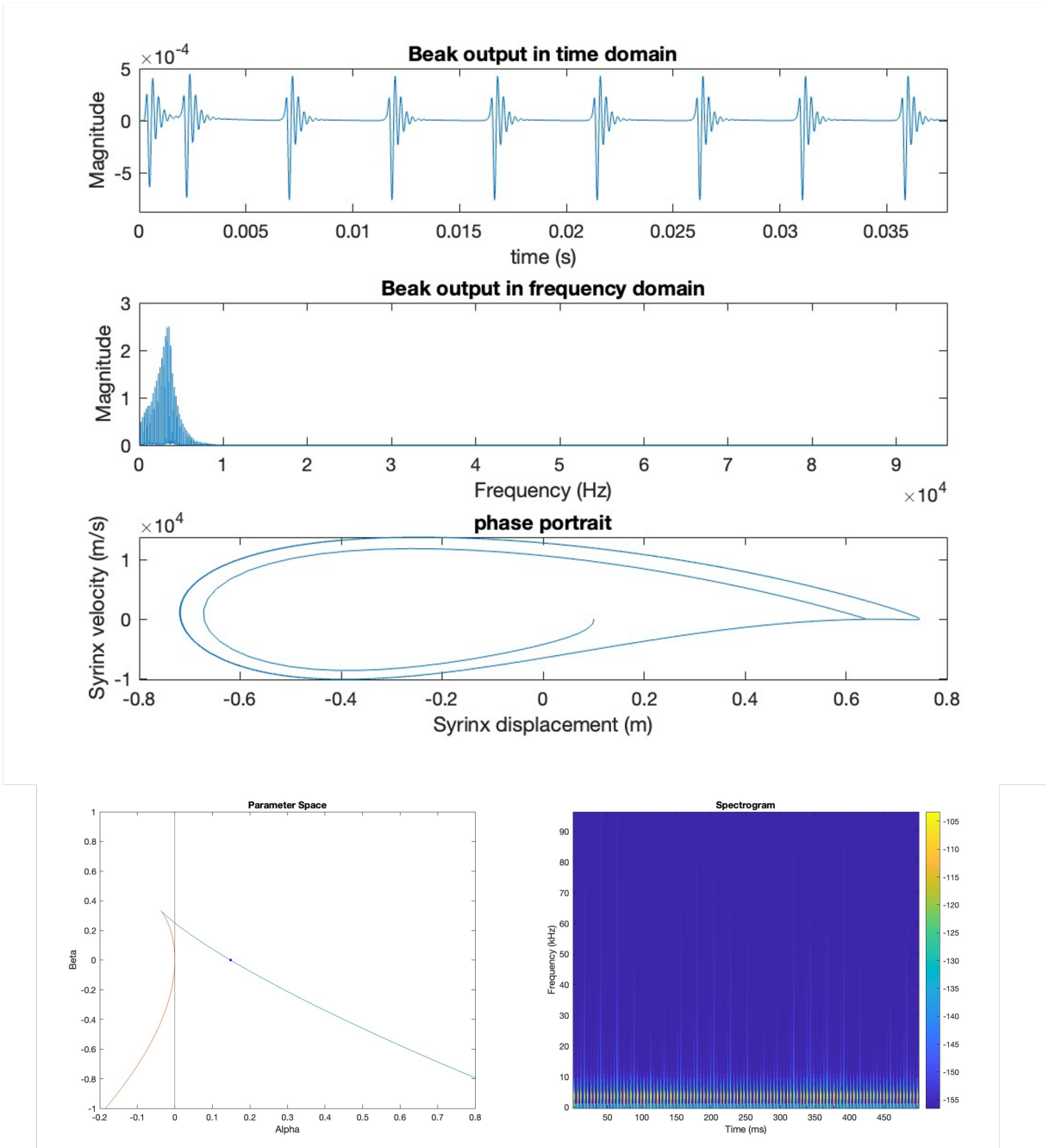


Figure 5.5: Sound generated in proximity to the SNILC bifurcation. From top to bottom, the figure displays: beak output in the time domain (zoomed in), beak output in the frequency domain, phase portrait of the syrinx, position in the (α, β) space (bottom left), and the spectrogram (bottom right).

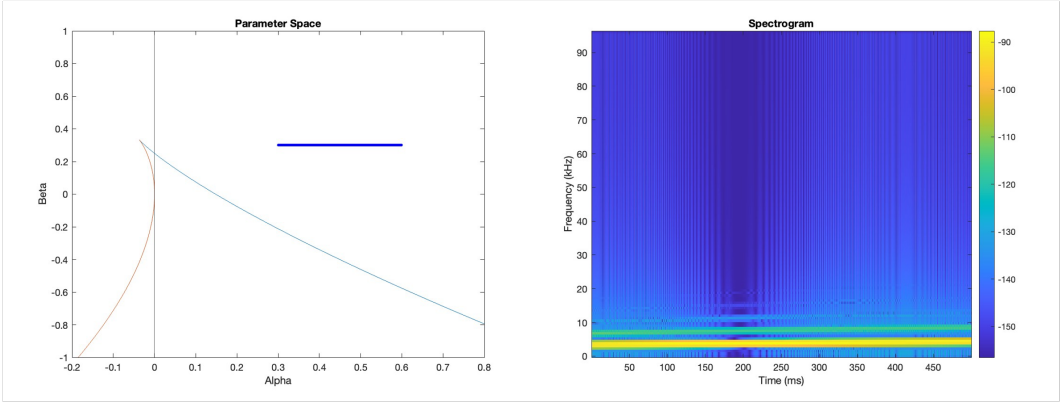


Figure 5.6: Example A: parameter trajectory in the (α, β) space (left) and the spectrogram(right).

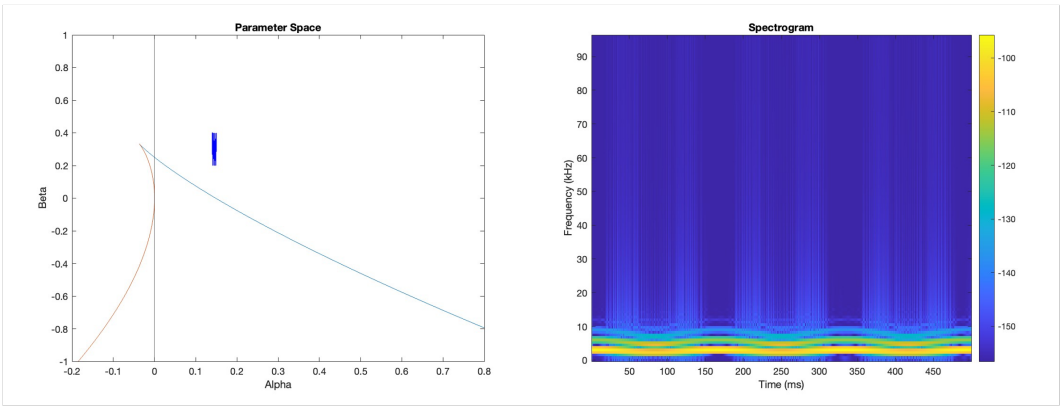


Figure 5.7: Example B: parameter trajectory in the (α, β) space (left) and the spectrogram(right).

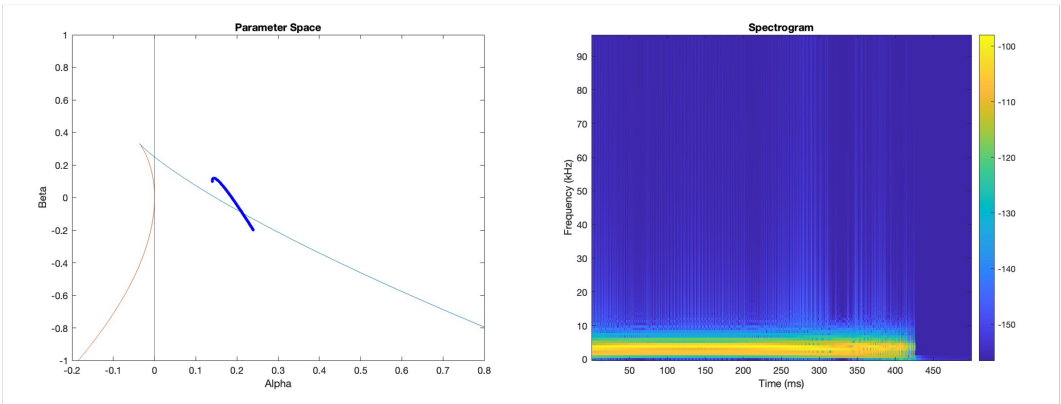


Figure 5.8: Example C: parameter trajectory in the (α, β) space (left) and the spectrogram(right).

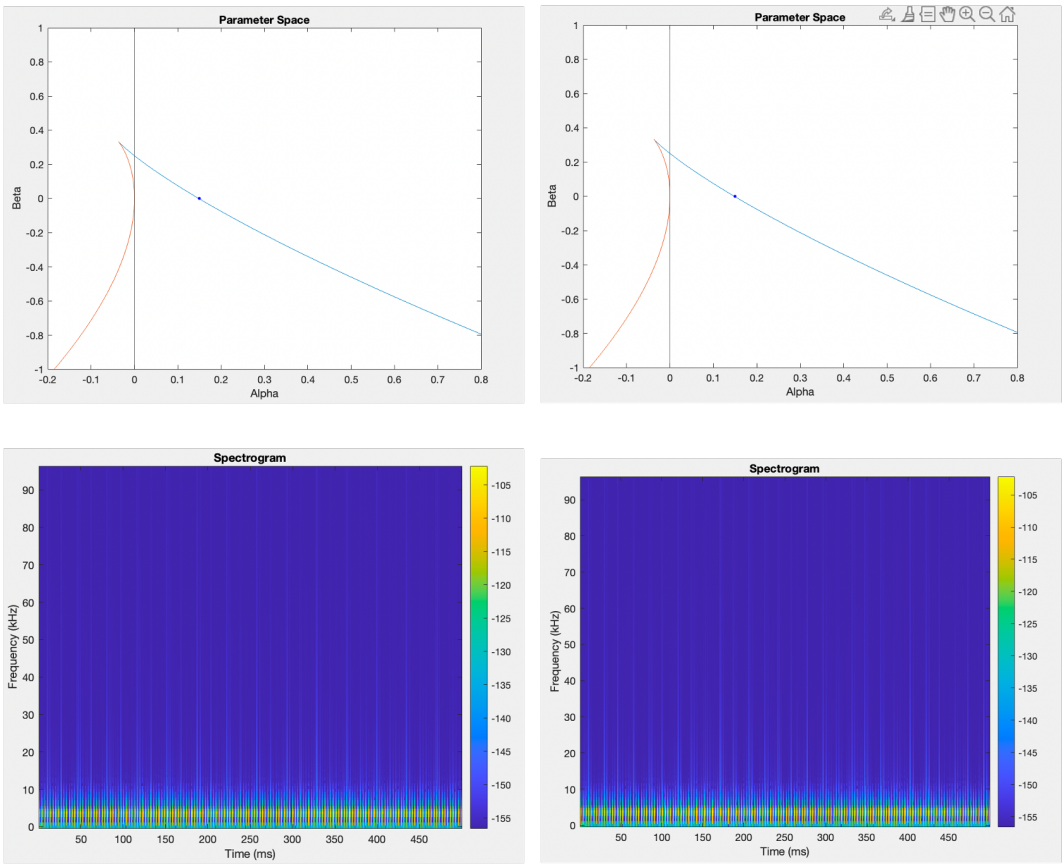


Figure 5.9: Comparison of parameter trajectories and spectrograms with and without noise. Left: Without noise - parameter trajectory in the (α,β) space (top) and its corresponding spectrogram (bottom). Right: With noise - parameter trajectory in the (α,β) space (top) and its corresponding spectrogram (bottom). The only difference in the parameters is the inclusion of noise.

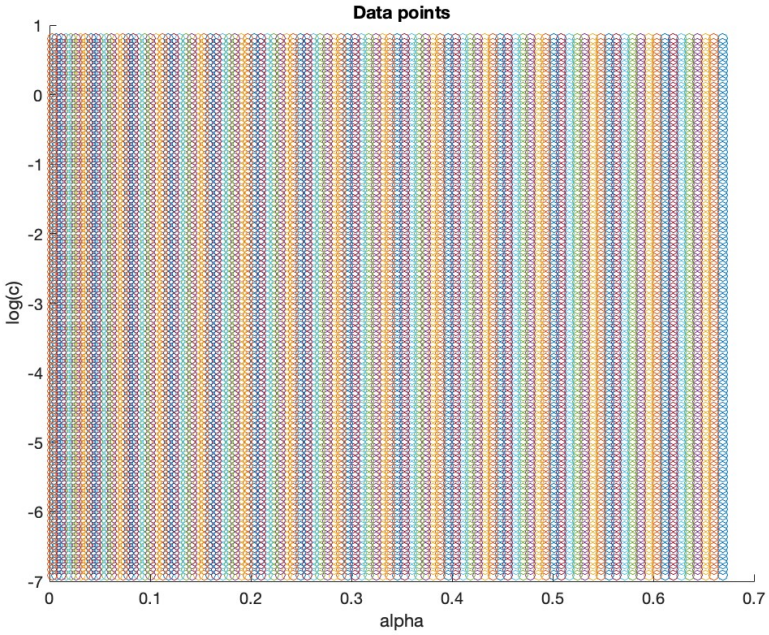


Figure 5.10: (α, $log(c)$) data points

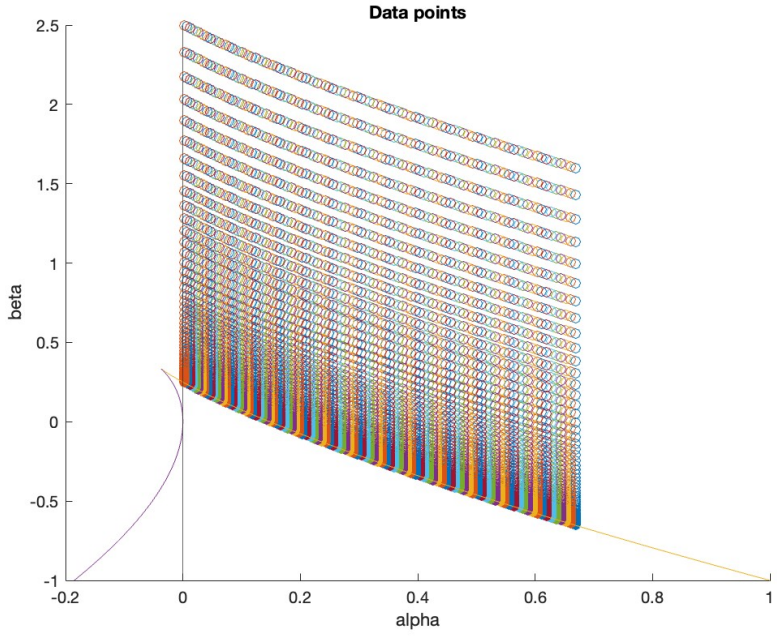


Figure 5.11: (α, β) data points

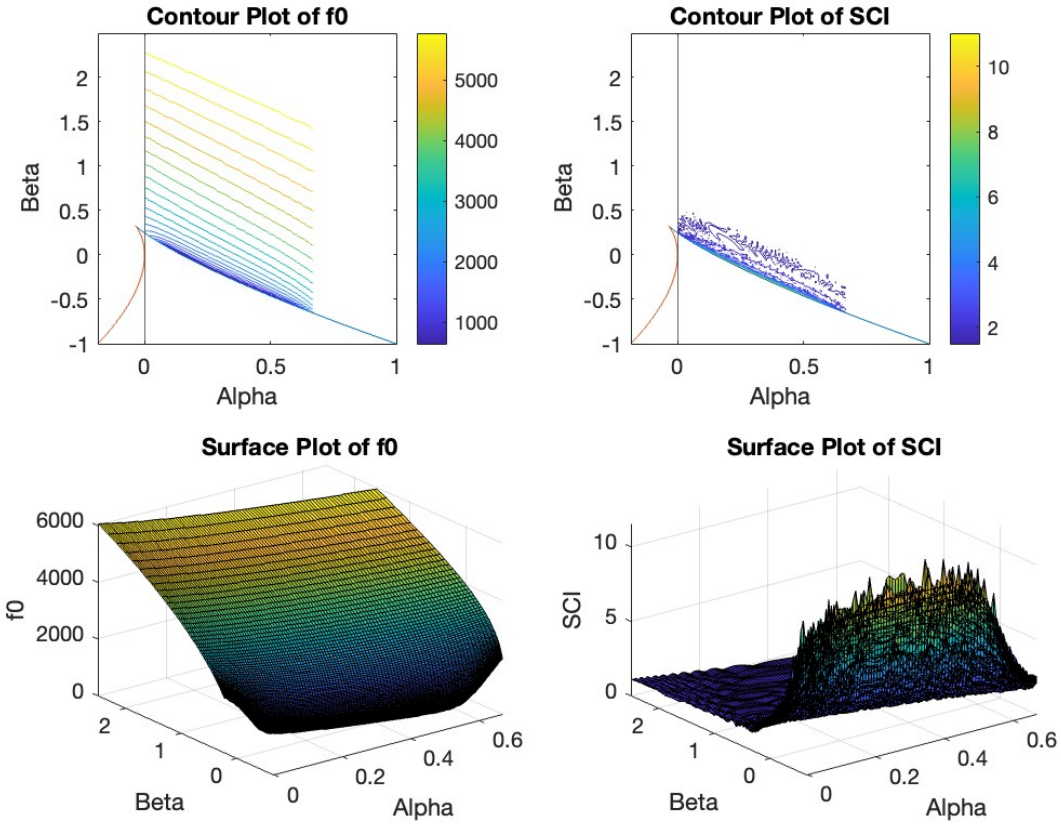


Figure 5.12: Relationship between (α, β) and (f_0, SCI)

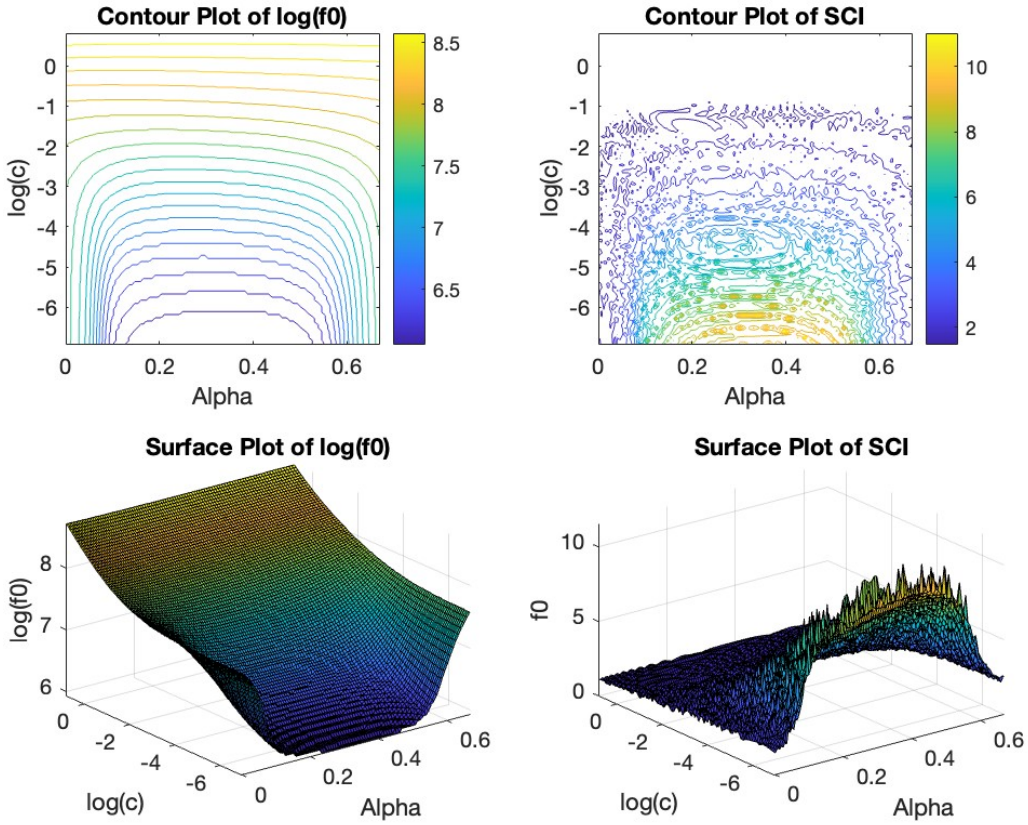


Figure 5.13: Relationship between (α, c) and (f_0, SCI)

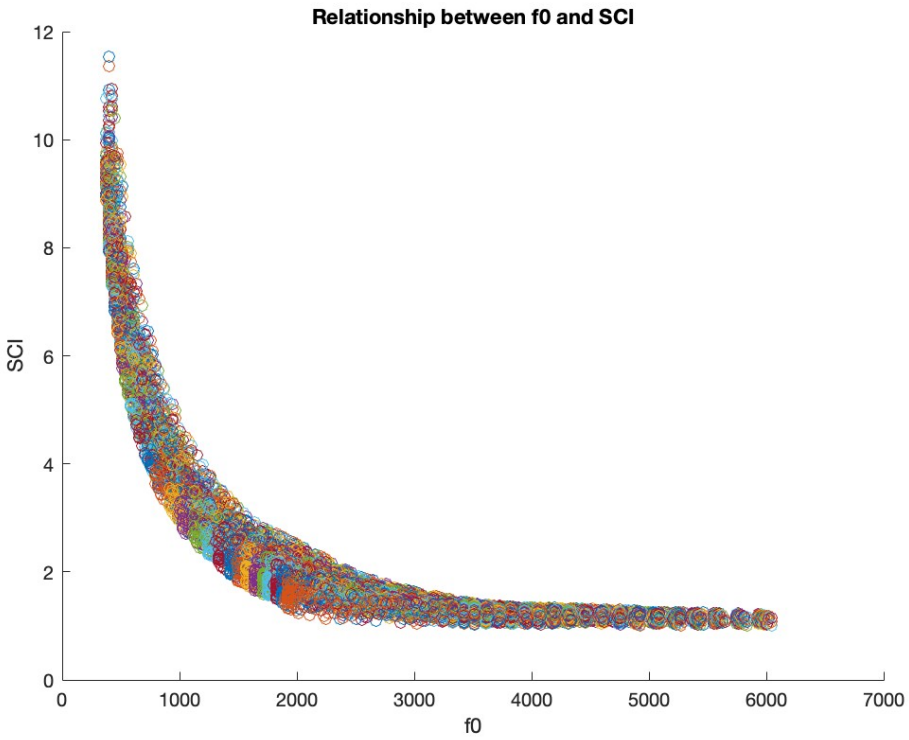


Figure 5.14: Correlation between (f_0, SCI)

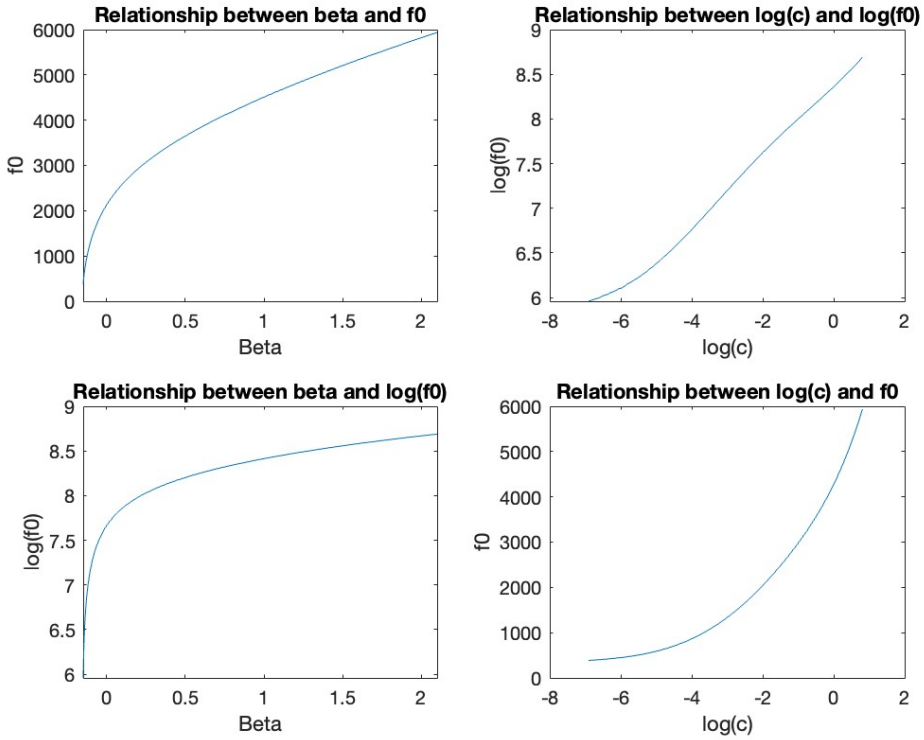


Figure 5.15: Relationship between $f0$ and β or c with α fixed to 0.256

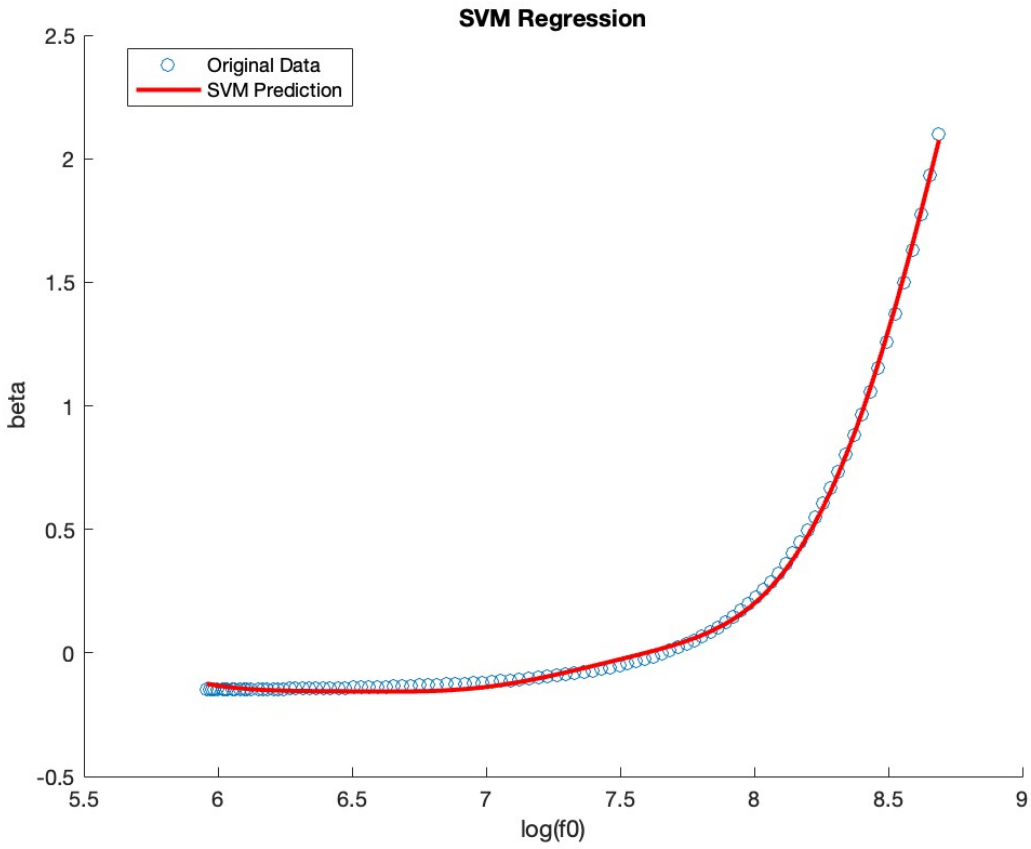


Figure 5.16: SVM regression on $(log(f_0), \beta)$

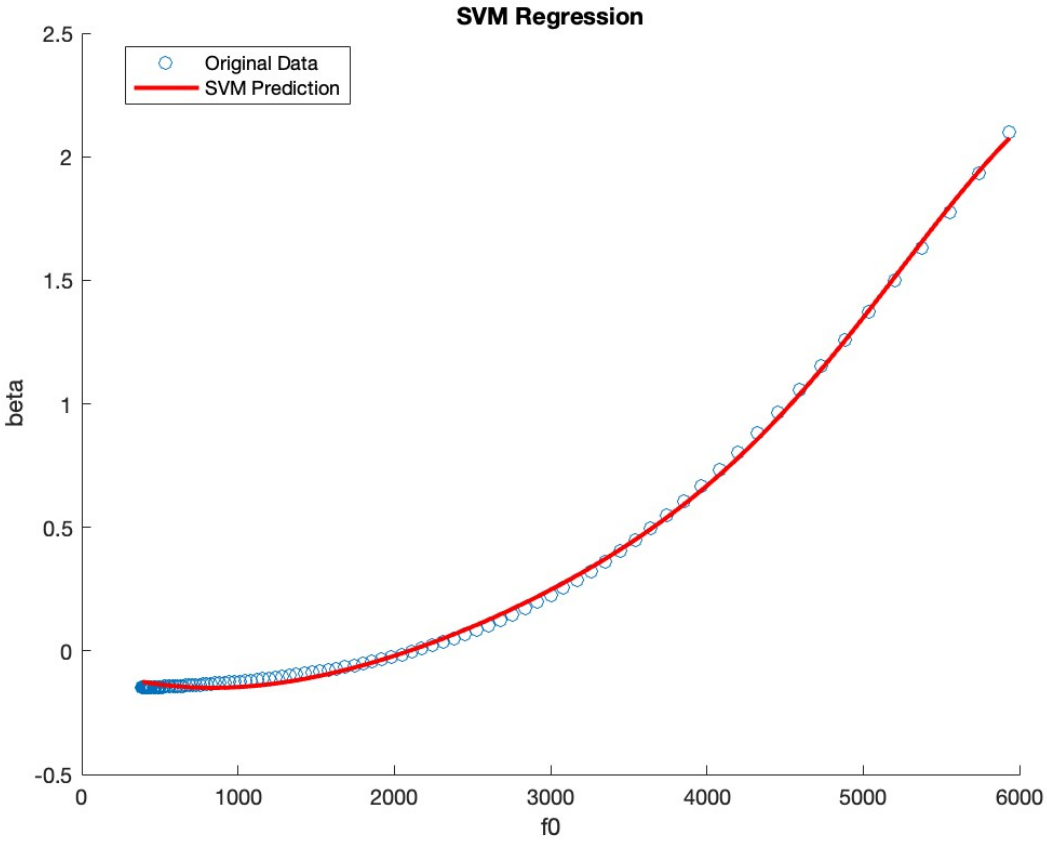


Figure 5.17: SVM regression on (f_0, β)

Chapter 6

Conclusions

This journey into the world of birdsong has been both enlightening and challenging. A computational birdsong model was successfully developed in Matlab, introducing an efficient finite difference scheme that surpasses existing methodologies. This unique approach to unveiling correlations in the parameter space has shed new light on the intricate world of birdsong. The incorporation of interpolation and machine learning techniques for model inversion signifies a promising direction for future research in this domain.

While significant results have been achieved, the realm of birdsong research is vast, and there remains much to explore. The initial venture into machine learning, especially the perceptual parameter input-output model, showcases the potential of integrating advanced techniques into traditional research. However, it's important to acknowledge that this approach, while promising, still requires further refinement and exploration.

Future Work

Moving forward, there's potential to refine the physical model to enhance its generalization capabilities. Exploring dual voices, introducing additional parameters, and studying various species are areas ripe for investigation. Analytical frequency solutions also present another intriguing avenue for exploration. Moreover, achieving real-time implementation on platforms like MaxMSP is a desired objective.

The integration of machine learning into this research marked an exhilarating advancement. Although the concept wasn't fully actualized due to the discovery of correlations, it remains a exciting attempt. Current machine learning applications in synthesizer modeling primarily involve audio as either input or output, often within the creative synthesis sphere rather than physics-based synthesis. This approach, which centers on scalar interpretable features, is ideally aimed at facilitating easier control for creation and parameter inversion from real-world sounds using the same training. It introduces a novel perspective. Nonetheless, it's crucial to ensure that the parameters employed aren't correlated. The potential of this approach, especially in terms of

enhancing playability and possibly resynthesis, is vast and certainly merits further investigation.

Appendix A

Example Codes

A.1 A Computational Model of Avian Vocalizations - model D

```
% ------------------------------------------------------------------------
% A Computational Model of Avian Vocalizations - model D
% - MSc Acoustics and Music Technology Final Project
%
% Author: Yining Xie
% Date: 9/9/23
%
% This scrip simulates birdsong based on a published model and parameters
% in [1].
%
% Finite difference method is used here, and the model mplementation was
% developed incrementally, capitalizing on the modelDs source-filter
% separation assumption. It is the implementation in section 4.2.4 in the
% dissertation.

% ------------------------------------------------------------------------
% Different sets of alpha, beta are given below with different spectral
% features.
%
% With velocity of syrinx as approximation of pressure input it has
% richer waveforms and similar to those in paper (both Mindlin's
% and Fletcher's) - zoom in to see.
%
% There is clip/impulse at the beginning which should be reasonable
% mathmatically though not physically. To avoid this, you may simply cut
% the first few samples.

% Reference:
% [1] Boari et al., [Automatic Reconstruction of Physiological Gestures Used in a Model of Birdsong Production].
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%function out = bird2015D()

clear all; close all; clc;

%% basic parameters ---------------------------------------------------

Fs = 4*48000;                   % Sampling rate [Hz] ------ there is aliasing under 44800
Ts = 0.5;                       % total duration [s]
```

```
%% derived parameters ------------------------------------------------------

k = 1/Fs;                           % step size
Nf = floor(Ts/k);                   % number of samples

%% avian structual parameters ----------------------------------------------

% trachea
Tr = 0.0002;                        % time it takes to a sound wave to traverse the trachea back and forth once
r = 0.1;                            % reflection coefficient
dlength = round(Tr/k);              % delayline length

% OEC
a = -540e6; b=-7.8e3; c=1.8e8; d=1.2e-2; e=7.2e-1;
f = -0.83e-2; g=-5e2; h=1e-4;

% matrices for OEC
A = [0,1,0;a,b,c;0,f,g];            % Re(eig(A))<0
B = [0,0;d,e;0,h];
% C = [0;0;1];
I = eye(3);

% syrinx
gamma = 24000;

%% initialize --------------------------------------------------------------

% dlinebuf = zeros(2*dlength,1); % delayline buffer in the case commented out below

pin = zeros(Nf, 1);                 % input pressure
pb = zeros(Nf, 1);                  % pressure reflected back
pout = zeros(Nf, 1);                % output pressure

tvec = linspace(0,Ts,Nf);           % time vector
fvec = Fs*[0:Nf-1]/Nf;              % frequency vector

tab = linspace(0,1,Nf);             % time vector for alpha, beta's trajectory design - so that their paths are irrelev

x00 = 0.1;                          % initial oscillator displacement
y0 = 0;                             % initial oscillator velocity
x2 = x00;                           % initial syrinx displacement
```

```
x1 = x00+k*y0;                  % second syrinx displacement
x = zeros(Nf,1);                % output syrinx displacement
o = zeros(Nf,1);                % output syrinx velocity
out = zeros(Nf,1);              % beak output

iv = zeros(3,1);                % OEC state vector
u = zeros(2,1);                 % initial pressure input to OEC
u1 = zeros(2,1);                % second pressure input to OEC
```

```
%% input - design trajectories ---------------------------------------------
% control parameters are identified as alpha, beta which can both be
% initialized as zeros(Nf, 1);
%
% change alpha to constant: multiply 0 before linspace
% alpha = 0.15+0*tab;
% change to impulse input:
% alpha(1) = 0;
% same for beta
% -------------------------------------------------------------------------

% % this is a spectrally rich one around SNIC bifurcation
alpha = 0.149+0*0.01*tab;
beta = 0+0*0.5*tab;
%
% % this is a tonal one
% alpha = 0.15+0*0.01*tab;
% beta = 0.5+0*0.5*tab;
%
% % time-varying
% alpha = 0.3+0.3*tab;
% beta = 0.3+0*0.5*tab;
%
% % one born in Hopf - oscillation starts from 0
% alpha = -0.2+0.5*tab;
% beta = 0.5+0*0.5*tab;
%
% % one born in SNIC
% alpha = 0.13+0.5*tab;
% beta = 0+0*0.5*tab;
```

```
%
% % it is of course possible to make other ones
% alpha = 0.14+0.01*tab;
% beta = 0.3+0.1*cos(6*pi*tab);
%
% % one more
% alpha = 0.14+0.1*tab.^2;
% beta = -0.5*tab.^2+0.2*tab+0.1;
```

```
%% main loop
tic
for i = 1:Nf

    % syrinx
    % update states
    x(i) = x2;
    y = (x1-x2)/k;

    % compute updates
    xx = 1/(k*gamma*(x1+x1^2)+k^2*gamma^2*x1^2+2)*(-2*k^2*gamma^2*alpha(i) ...
        -2*k^2*gamma^2*beta(i)*x1+2*k^2*gamma^2*x1^2+4*x1+(-k^2*gamma^2*x1^2+k*gamma*x1+k*gamma*x1^2-2)*x2);

    % shift states
    x2 = x1;
    x1 = xx;

    % write output velocity
    o(i)=y;


    % trachea
    % previous pressure output
    p_pre = pout(i);

    % delay
    if i>dlength
        pin(i) = y+pb(i-dlength);
        pb(i) = -r*pin(i-dlength);
        pout(i) = (1-r)*pin(i-dlength);
    else
```

```
        pin(i) = y;
        pb(i) = 0;
        pout(i) = 0;
    end

% above turns out to be faster
%         pin(i) = x(2)-r*dlinebuf(2*dlength); %consider changing to x1/dx(1)
%         pb(i) = -r*dlinebuf(dlength);
%         pout(i) = (1-r)*dlinebuf(dlength);
%
%         dlinebuf(end) = pin(i);
%         dlinebuf = circshift(dlinebuf,1);
%

    % derivative of trachea output pressure of trachea
    dp = (pout(i)-p_pre)/k;

    % OEC
    % pressure input to OEC
    u1 = [dp;pout(i)];

    % state update
    iv = (I-k*A/2)\((I+k*A/2)*iv+k/2*B*(u+u1));

    % input shift
    u = u1;

    % output
    out(i) = iv(3);

end
toc
```

```
%% make sound
soundsc(out,Fs)
```

```
%% plots

% output in time domain
subplot(3,1,1)
```

```
plot(tvec,out)
xlabel('time (s)')
ylabel('Magnitude')
title('Beak output in time domain');

% output in frequency domain
subplot(3,1,2)
Xf = fft(out);        % transform output

plot(fvec,abs(Xf))
xlim([0 Fs/2])
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('Beak output in frequency domain');

% phase portrait
subplot(3,1,3)
plot(x,o)
xlabel('Syrinx displacement (m)')
ylabel('Syrinx velocity (m/s)')
title('phase portrait');

% spectrogram
figure;
windowSize = 256;
overlap = round(windowSize/2);
nfft = 2^nextpow2(windowSize);
spectrogram(out, windowSize, overlap, nfft, Fs, 'yaxis');
colorbar;
title('Spectrogram');

% parameter space
figure;
[alpha0,beta0]=bir2();
plot(alpha0,beta0)
hold on
xline(0);
hold on
scatter(alpha, beta, 10, 'b', 'filled');
hold off
xlabel('Alpha')
```

```
ylabel('Beta')
xlim([-0.2 0.8]);
ylim([-1, 1]);
title('Parameter Space')
```

```
%% function for bifurction diagram
function [alpha0,beta0]=bir2()
% this function gives bifurction diagram, see dissertation for detailed
% explanation of the equations

    N = 1000;
    x0 = zeros(N,2);
    beta0 = linspace(-1, 1/3, N)';
    x0(:,1) = (1+sqrt(1-3*beta0))/3;
    x0(:,2) = (1-sqrt(1-3*beta0))/3;
    alpha0 = -beta0.*x0+x0.^2-x0.^3;

end


%end
```

Appendix B

The Original Final Project Proposal

MSc Acoustics and Music Technology
Final Project Proposal

Differentiable White-Box Modelling of Virtual Analog with a comparison of Trapezoid, Mid-Point and Port-Hamiltonian methods

Student name: Yining Xie
UUN: s2172442
Proposed supervisor: Prof. Stefan Bilbao

1 Background

Traditional white box virtual analog modelling has been long studied with different approaches, such as state-space methods, wave digital filters and port-Hamiltonian systems[1]. Recently, following the development of differentiable IIR filter[2], Esqueda et al. introduced a differentiable form that incorporates the white box numerical method with machine learning technology to address the limited accuracy of the traditional component-wise circuit modelling.[3] Despite its successful implementation, only trapezoid rule and computationally expensive iterative methods are used in this study, and a comparison between different numerical methods, though proposed, is not studied there. Besides, it also reports a problem in low-frequency response, multiple local minima and notes that learned parameter values may not reflect real circuit values.

2 Aim

This project primarily aims to implement trapezoid method, midpoint method[4][5] and the port-Hamiltonian method[6] respectively for differentiable white box modelling of virtual analog. The non-iterative scheme introduced in [7] could also be examined. All these methods besides the trapezoid one have not been previously implemented in the differentiable white box modelling context, and their performance would be studied and compared here. Besides, I consider parameterizing to allow user control. Additionally, methods to avoid iterations such as the non-iterative scheme[7], and explicit port Hamiltonian scheme[8] may be studied, which could be the most important and most difficult if not impossible part. Moreover, and if possible, I would also like to address the low-frequency response error as mentioned before. Furthermore, I may look into evaluation with different loss functions, as described in [9], aliasing still is a problem in white box modelling, and the possibility to generalize the code for arbitrary circuits.

3 Methodology

All parts of this project are fairly challenging, especially when there is no existing data or code to learn from and for many numerical modelling schemes, I also need to derive by myself. Work could be broken down into the following parts:

3.1 Data

Input and output data: I design it to be a 3 minutes mixture of white noise, pure sine tone, sine sweep and a short piece of music, similar to [3].
Circuit: For simple circuits such as diode clipper, it could be simulated from LTspice [10], which though, takes time to learn. If possible, I also consider building it on a breadboard and

recording physical responses. If it ended up with other circuits such as tube screamer, recording from actual Ibanez Tube Screamer TS808 could be considered.

3.2 Numerical schemes

As mentioned above, trapezoid method, mid-point method and port-Hamiltonian method are of interest. Even though there is reference from [11] for the former two, I prefer to use the state update methods in [12], hence need to derive the mid-point scheme by myself. Port-Hamiltonian method is the most challenging one, as it is an entirely new system we have never studied before. There is some reference from [6]. The pyphs[13] github repository shall also be looked into.

3.3 Iteration

This is a very important aspect to study, as iteration significantly increases computational cost especially in differentiable white box modelling, and could make this project impossible to finish in time if all methods studied need it. In the very limited existing relevant paper, the model is either linear hence does not need iteration, or nonlinear and iteration is involved, and there is not yet a good solution to this.

As practiced in previous coursework, look-up table together with Lagrange interpolation could be considered to avoid iteration. However, as there is also no previous experience of including this method to replace Newton Raphson iteration in a Python neural network code, its feasibility is to be studied. And as look-up table should change with parameters too, this may also not increase efficiency at all.

Besides, the non-iterative scheme could also be discussed, though its stability cannot be guaranteed.

For the port-Hamiltonian scheme, I am interested in if the explicit scheme that succeeded in [14] for KORG35 and MoogVCF could be applied to other nonlinear circuits such as distortion ones as well. Otherwise I may also directly work on non-iterative nonlinear MoogVCF which I have previously implemented successfully in Matlab as a beyond the basics PBMMI assignment.

3.4 Incorporating into Neural Network

The code in [3] has been deleted, and I shall take time to build it by myself. Adequate time needs to be reserved for tuning.

3.5 Parameterizing

[3] described a possible method for parameterizing, while the actual implementation is still to experiment with.

3.6 Low-frequency response error

This is also reported in [3], and from my analysis, this could be because the new parameter learned by neural network successfully compensates for parasitic capacitance, which improves especially the performance in relatively high frequencies where parasitic capacitance exists, and which takes a larger part of the entire frequency domain of interest, the low-frequency domain where parasitic capacitance is less obvious is hence compromised for better overall performance. This also explains why the learned parameters are different from actual circuit values.

Possible approaches I consider are: 1) similar to the parameterizing technique, let the neural network learn a function instead of a set value, which shall be closer with the real case. 2) modify the numerical design to take this into consideration. This could be harder to realize, but possibly of great benefit to the white box numerical modelling itself.

For this to be taken into consideration, data from real circuit instead of simulated one may be necessary.

3.7 Generalization

If time allows: consideration is to give a general (likely state space) structure and let the machine learn/optimize parameters alongside outputs. I have done something similar previously in Matlab as beyond the basics in PBMMI, while its feasibility with machine learning is to be experimented with.

3.8 Evaluation

With MSE as loss function, it is possible to directly plot the model output in comparison with the ideal output. Other loss functions may be applied too, and for perceptual related evaluation, mel spectrogram and MUSHRA listening test could be considered. As to aliasing, I probably would not have time to reduce it, a comparison of it using different schemes can also be of interest, as there have also not been such comparisons before.

3.9 Device

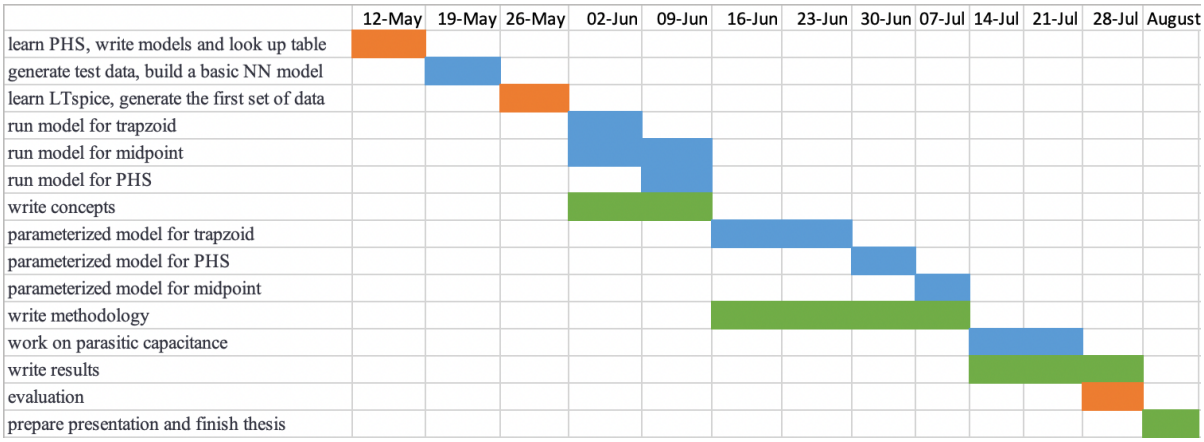
May need device to build and test circuits
May need Ibanez Tube Screamer TS808
May need more computational resource other than my own laptop

4 Timetable

4.1 Milestones

26/5 build 1st NN model (can be linear but parameterized)
7/6 finish 1st nonlinear trapezoid model
10/6 finish 1st nonlinear midpoint model
16/6 finish 1st nonlinear PHS model
30/6 finish parameterized trapezoid model
07/7 finish parameterized PHS model
14/7 finish parameterized midpoint model
28/7 finish any additional work
02/8 finish evaluation
12/8 finish first draft of dissertation

4.2 Gantt Chart



5 Bibliography

[1] S. D'Angelo, *Lightweight Virtual Analog Modeling*. 2018.

[2] B. Kuznetsov, J. D. Parker, and F. Esqueda, 'Differentiable IIR filters for machine learning applications'.

[3] F. Esqueda, B. Kuznetsov, and J. D. Parker, 'Differentiable White-Box Virtual Analog Modeling', in *2021 24th International Conference on Digital Audio Effects (DAFx)*, Vienna, Austria: IEEE, Sep. 2021, pp. 41–48. doi: 10.23919/DAFx51585.2021.9768272.

[4] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, vol. 14. 1996. doi: 10.1007/978-3-662-09947-6.

[5] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied Mathematics, 2007. doi: 10.1137/1.9780898717839.

[6] R. Müller, 'Time-continuous power-balanced simulation of nonlinear audio circuits: realtime processing framework and aliasing rejection', 2021. doi: 10.13140/RG.2.2.18643.71204.

[7] M. Ducceschi and S. Bilbao, 'Non-Iterative Simulation Methods for Virtual Analog Modelling', *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 3189–3198, 2022, doi: 10.1109/TASLP.2022.3209934.

[8] N. Lopes, T. Hélie, and A. Falaize, 'Explicit second-order accurate method for the passive guaranteed simulation of port-Hamiltonian systems', *IFAC-Pap.*, vol. 48, no. 13, pp. 223–228, Jan. 2015, doi: 10.1016/j.ifacol.2015.10.243.

[9] A. Wright and V. Välimäki, 'Perceptual Loss Function for Neural Modelling of Audio Systems'. arXiv, Nov. 20, 2019. Accessed: Apr. 10, 2023. [Online]. Available: http://arxiv.org/abs/1911.08922

[10] J. Najnudel, 'Power-Balanced Modeling of Nonlinear Electronic Components and Circuits for Audio Effects', 2022.

[11] M. Porter, 'Virtual Analog Modeling of Guitar Effects Circuits', 2019.

[12] A. Carson, 'Aliasing Reduction in Virtual Analogue Modelling', 2020. doi: 10.13140/RG.2.2.29405.44008.

[13] 'pyphs/pyphs'. PyPHS, Apr. 07, 2023. Accessed: Apr. 23, 2023. [Online]. Available: https://github.com/pyphs/pyphs

[14] M. Danish, S. Bilbao, and M. Ducceschi, 'Applications of Port Hamiltonian Methods to Non-Iterative Stable Simulations of the KORG35 and MOOG 4-Pole VCF', in *2021 24th International Conference on Digital Audio Effects (DAFx)*, Vienna, Austria: IEEE, Sep. 2021, pp. 33–40. doi: 10.23919/DAFx51585.2021.9768301.

Bibliography

[1] S. Fagerlund, "Acoustics and physical models of bird sounds," in *Seminar in acoustics, HUT, Laboratory of Acoustics and Audio Signal Processing*. Citeseer, 2004.

[2] A. Amador and G. B. Mindlin, "Low dimensional dynamics in birdsong production," *The European Physical Journal B*, vol. 87, pp. 1–8, 2014.

[3] E. M. Arneodo, Y. S. Perl, F. Goller, and G. B. Mindlin, "Prosthetic avian vocal organ controlled by a freely behaving bird based on a low dimensional model of the biomechanical periphery," *PLoS computational biology*, vol. 8, no. 6, p. e1002546, 2012.

[4] G. B. Mindlin and R. Laje, *The physics of birdsong*. Springer Science & Business Media, 2005.

[5] G. B. Mindlin, "Nonlinear dynamics in the study of birdsong," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 9, 2017.

[6] N. H. Fletcher, *Acoustic systems in biology*. Oxford University Press, 1992.

[7] T. Smyth and J. O. Smith, "The sounds of the avian syrinx—are they really flute-like?" in *DAFX 2002 Proceedings*, 2002.

[8] T. Smyth, "Applications of bioacoustics to musical instrument technology," Ph.D. dissertation, Ph. D. thesis, Stanford University, 2004.

[9] S. H. Strogatz, *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[10] Y. A. Kuznetsov, "Practical computation of normal forms on center manifolds at degenerate bogdanov–takens bifurcations," *International Journal of Bifurcation and Chaos*, vol. 15, no. 11, pp. 3535–3546, 2005.

[11] A. Amador, Y. S. Perl, G. B. Mindlin, and D. Margoliash, "Elemental gesture dynamics are encoded by song premotor cortical neurons," *Nature*, vol. 495, no. 7439, pp. 59–64, 2013.

[12] J. N. Maina, "Development, structure, and function of a novel respiratory organ, the lung-air sac system of birds: to go where no other vertebrate has gone," *Biological Reviews*, vol. 81, no. 4, pp. 545–579, 2006.

[13] F. Goller and O. Larsen, "New perspectives on mechanisms of sound generation in songbirds," *Journal of Comparative Physiology A*, vol. 188, pp. 841–850, 2002.

[14] R. Zaccarelli, C. P. Elemans, W. Fitch, and H. Herzel, "Modelling bird songs: voice onset, overtones and registers," *Acta acustica united with acustica*, vol. 92, no. 5, pp. 741–748, 2006.

[15] T. Smyth and J. O. Smith, "The syrinx: Nature's hybrid wind instrument," *Journal of the Acoustical Society of America*, vol. 112, no. 5, p. 2240, 2002.

[16] R. Laje and G. B. Mindlin, "Diversity within a birdsong," *Physical review letters*, vol. 89, no. 28, p. 288102, 2002.

[17] S. Boari, Y. S. Perl, A. Amador, D. Margoliash, and G. B. Mindlin, "Automatic reconstruction of physiological gestures used in a model of birdsong production," *Journal of neurophysiology*, vol. 114, no. 5, pp. 2912–2922, 2015.

[18] M. Trevisan and G. Mindlin, "New perspectives on the physics of birdsong," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1901, pp. 3239–3254, 2009.

[19] T. Smyth, J. S. Abel, and J. O. Smith, "The estimation of birdsong control parameters using maximum likelihood and minimum action," in *Proceedings of SMAC*, vol. 3, 2003, pp. 413–416.

[20] E. M. Arneodo, S. Chen, D. E. Brown, V. Gilja, and T. Q. Gentner, "Neurally driven synthesis of learned, complex vocalizations," *Current Biology*, vol. 31, no. 15, pp. 3419–3425, 2021.

[21] S. I. Kabanikhin, "Definitions and examples of inverse and ill-posed problems," 2008.

[22] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating physics-based modeling with machine learning: A survey," *arXiv preprint arXiv:2003.04919*, vol. 1, no. 1, pp. 1–34, 2020.

[23] D. Herremans and C.-H. Chuan, "The emergence of deep learning: new opportunities for music and audio technologies," pp. 913–914, 2020.

[24] L. Gabrielli, S. Tomassetti, S. Squartini, and C. Zinato, "Introducing deep machine learning for parameter estimation in physical modelling," in *Proceedings of the 20th international conference on digital audio effects*, 2017.

[25] P. Modler, "Neural networks for mapping hand gestures to sound synthesis parameters," *Trends in Gestural Control of Music*, vol. 18, no. 2.2, p. 2, 2000.

[26] R. Laje, T. J. Gardner, and G. B. Mindlin, "Neuromuscular control of vocalizations in birdsong: a model," *Physical Review E*, vol. 65, no. 5, p. 051921, 2002.

[27] I. R. Titze, "The physics of small-amplitude oscillation of the vocal folds," *The Journal of the Acoustical Society of America*, vol. 83, no. 4, pp. 1536–1552, 1988.

[28] T. Gardner, G. Cecchi, M. Magnasco, R. Laje, and G. B. Mindlin, "Simple motor gestures for birdsongs," *Physical review letters*, vol. 87, no. 20, p. 208101, 2001.

[29] A. Amador and G. B. Mindlin, "Beyond harmonic sounds in a simple model for birdsong production," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 4, 2008.

[30] J. D. Sitt, E. M. Arneodo, F. Goller, and G. B. Mindlin, "Physiologically driven avian vocal synthesizer," *Physical Review E*, vol. 81, no. 3, p. 031927, 2010.

[31] T. Riede, R. A. Suthers, N. H. Fletcher, and W. E. Blevins, "Songbirds tune their vocal tract to the fundamental frequency of their song," *Proceedings of the National Academy of Sciences*, vol. 103, no. 14, pp. 5543–5548, 2006.

[32] T. Riede, N. Schilling, and F. Goller, "The acoustic effect of vocal tract adjustments in zebra finches," *Journal of Comparative Physiology A*, vol. 199, pp. 57–69, 2013.

[33] E. M. Arneodo and G. B. Mindlin, "Source-tract coupling in birdsong production," *Physical Review E*, vol. 79, no. 6, p. 061921, 2009.

[34] S. Pajevic, "Nonlinear dynamics and chaos: Steven h. strogatz, addison-wesley, reading, massachusetts, 1994," 1995.

[35] C.-i. Wang, T. Smyth, and Z. C. Lipton, "Estimation of saxophone control parameters by convex optimization," in *CIM14, Conference on Interdisciplinary Musicology: proceedings. Conference on Interdisciplinary Musicology (9th: 2014: Berlin, Germany)*, vol. 2014. NIH Public Access, 2014, p. 280.