Algorithms and circuits for motor control and learning in the songbird

by

Michael E. Stetner

Submitted to the Department of Brain and Cognitive Sciences in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2019

© Massachusetts Institute of Technology 2019. All rights reserved.

Signature redacted

Brain and Cognitive Sciences Department

January 11, 2019

Signature redacted

Glen V. and Phyllis F. Dorflinger Professor and Professor of Neuroscience

Thesis Supervisor

Signature redacted

Matthew A. Wilson

Sherman-Fairchild Professor of Neuroscience and Picower Scholar

Director of Graduate Education for Brain and Cognitive Sciences
Algorithms and circuits for motor control and learning in the songbird

by

Michael E. Stetner

Submitted to the Department of Brain and Cognitive Sciences
on January 11, 2019
in Partial Fulfillment of the requirements for the Degree of
Doctor of Philosophy in Neuroscience

ABSTRACT

From riding a bike to brushing our teeth, we learn many of our motor skills through trial and error. Many biologically based trial and error learning models depend on a teaching signal from dopamine neurons. Dopamine neurons increase their firing rates to signal outcomes that are better than expected and decrease their firing rates to signal outcomes that are worse than expected. This dopamine signal is thought to control learning by triggering synaptic changes in the basal ganglia. What are the origins of this dopaminergic teaching signal? How do synaptic changes in the basal ganglia lead to changes in behavior? In this thesis, I study these questions in a model of skill learning — the songbird.

In the first part of my thesis, I develop a computational model of song learning. This model incorporates a dopaminergic reinforcement signal in VTA and dopamine-dependent synaptic plasticity in the singing-related part of the basal ganglia. I demonstrate that this model can provide explanations for a variety of experimental results from the literature.

In the second part of my thesis, I investigate a potential source of the dopaminergic error signal in VTA. I performed the first recordings from one cortical input to VTA: the dorsal intermediate arcopallium (Ald). Previous studies disagree on the role of Ald in behavior. Some studies argue that Ald contributes vocal error information to VTA. Other studies suggest that Ald is not involved in the computation of error signals, but is instead responsible for controlling head and body movements. I directly tested these hypotheses by recording single neurons in Ald during singing and during natural movements. My results support a motor role for Ald — Ald neurons had highly significant changes in activity during head and body movements. Meanwhile, following vocal errors Ald neurons had small but marginally significant decrease in firing rate. In a more detailed analysis, I developed an automated behavior classification algorithm to categorize zebra finch behavior and related these behavior classes to the activity of single units in Ald. My results support the hypothesis that Ald is part of a general-purpose motor control network in the songbird brain.

Thesis Supervisor: Michale S. Fee
Title: Glen V. (1946) and Phyllis F. Dorflinger Professor; Professor of Neuroscience
Acknowledgments

I would not have been able to complete this thesis without help from my coworkers, friends, and family. My committee members, Matthew Wilson, Ann Graybiel, and Bence Olveczky for their thoughtful questions, practical advice, and positive attitude. Most of all, I am grateful for the guidance from my advisor, Michale Fee. His energy, clarity of thought, and precise explanations were all things that tried so hard to imitate.

Michale also assembled a great group of graduate students and post docs that overlapped with me during my time in the lab: Tatsuo Okubo, Emily Mackevicius, Galen Lynch, Dmitriy Aronov, Aaron Andalman, Michael Long, Liora Las, Jesse Goldberg, Anusha Narayan, Maya Bronfeld, Yael Mandelblat-Cerf, Natasha Denisenko, Lena Veit, Husain Danish, Andrew Bahle, Michael Happ, Nader Nikbakht, and Joe Scherrer. I am thankful for their enthusiasm and at our marathon lab meetings and their camaraderie at our daily lunches together. I am also thankful for the behind-the-scenes work of the lab staff. Dan Rubin, Tim Currier, Shelbi Ferber, Anton Konovchenko, Collyn Messier, and Kailey Miller cared for our research subjects and performed excellent histology, and Margo Oleary helped keep us all organized.

I am especially appreciative of the lab members who contributed directly to the work in this thesis. Galen Lynch was always the first person I turned to with questions about anything technical, from electronics to statistics. His advice shaped pieces of every chapter. Shelbi Ferber helped with the video analysis in Chapter 3. Abe Akkas and Adrian Cho contributed code and ideas to the analyses in Chapter 4.

Outside of the lab, I had the kind support of my friends and family. Thank you to Megan Krench, Phil Isola, and Kean Jaime-Bustamante for the years of living together and sharing food, stories, and a cat. Thanks to Jon Malmiad for all the games we played together, and to Jessi and Mike Sanders for all the time we spent outdoors together. Finally, I would like to thank my family for their patience and encouragement, especially my wife Katie for being my greatest cheerleader, editor, audience, and partner.
# Table of Contents

Acknowledgments........................................................................................................................................................................... 5

Table of Contents..................................................................................................................................................................................... 7

Attributions and publications................................................................................................................................................................. 8

1 Introduction ......................................................................................................................................................................................... 10

2 Computational model of reinforcement learning........................................................................................................................... 33

3 Activity of neurons in AI$d$ during vocal errors and natural movements......................................................................................... 78

4 Neural coding of head and body movements in motor cortex of freely behaving zebra finches ......................................................... 112

5 Discussion............................................................................................................................................................................................ 146

Appendix A  Recording system for neural activity and movements ............................................................................................... 156

Bibliography ......................................................................................................................................................................................... 179
Attributions and publications

The concept for the model in Chapter 2, including the structure of the network and the learning rule, is taken from (Fee and Goldberg, 2011). The code is all my own. Chapters 3 and 4 are entirely my work.

None of this work has been published in a peer reviewed journal yet. I plan to submit Chapter 2 as a paper to PLOS ONE and combine Chapters 3 and 4 into a second paper to be submitted to PLOS ONE.
1 Introduction

We perform hundreds of complex motor sequences every day – from coordinating our fingers to tie our shoes to moving our lips and tongue and vocal cords to speak. These behaviors require extreme precision; our brains must bring the right body part to the right place at precisely the right time. These behaviors are not innate but learned through practice. With continued practice, these behaviors are maintained over a lifetime despite changes to our body. The mechanisms underlying the control and learning of these precise motor behaviors remain a topic of active research in neuroscience.

Neuroscientists study songbirds as a model of motor performance and learning, most commonly the zebra finch. Songbirds are natural vocal learners. Juvenile songbirds are born not knowing how to sing, but over development, they learn to imitate an adult tutor. While they are young, they hear an adult tutor sing and memorize this song. If juveniles are raised in isolation without an adult tutor, their song never converges to a stable adult song typical of juveniles raised naturally (Eales, 1985, 1987; Marler, 1981; Williams et al., 1993). Through several months of independent practice, they learn to produce an imitation of the song from their tutor (Immelmann, 1969; Tchernichovski et al., 2001). Their practice can occur alone, without an instructor to tell them when they make a mistake. Rather, they must be able to hear themselves sing; deafening juvenile birds prevents them from learning an accurate imitation of their tutor (Konishi, 1965; Marler and Waser, 1977). This implies they are judging the sound of their own song and producing a teaching signal internally.

In this thesis, I explore several ideas related to this internal teaching signal. In Chapter 2, I construct a computational model of the learning process and demonstrate how a simple teaching
signal can account for some surprising features of song learning. Then, I search for the source of the teaching signal by recording from a cortical brain area thought to carry auditory information during singing. Surprisingly, I discover that this brain area does not contain error related signals, but instead contains signals related to head and body movements. In Chapter 4, I further investigate these movement related signals. The remainder of this introduction reviews the literature on vocal production and learning in songbirds and how it relates to theories of learning in general.

1.1 Adult song production

Adult zebra finches sing a complex song, which is composed of repetitions of a single motif. The adult song is highly stereotyped because all renditions of the motif are nearly identical. The motif itself is about one second long and is composed of a fixed sequence of two to ten syllable types, separated by periods of silence. Each syllable type has distinct acoustic features, which are sometimes further subdivided into notes. Multiple renditions of the motif are sung in sequence to form a bout. The songbird is an attractive model for skill learning and performance because the entire process of song production and learning is confined to a small number of discrete brain nuclei that are wholly dedicated to singing. A simplified diagram of some of these brain areas is shown in Figure 1.1.
Production of the adult song requires a discrete set of brain nuclei called the motor pathway. At the top of the motor pathway sits the cortical nucleus HVC (used as proper name), which projects to downstream cortical nucleus RA (robust nucleus of the arcopallium) (Nottebohm et al., 1976). RA is similar to primary motor cortex in mammals (Dugas-Ford et al., 2012; Jarvis et al., 2005; Pfenning et al., 2014; Reiner et al., 2004), and it projects directly to the brainstem where it synapses with various motor control structures. These structures include the twelfth cranial nerve (nXII), which contains the motor neurons that innervate the syrinx. RA also projects to several midbrain nuclei not shown in Figure 1.1. It projects to midbrain respiratory centers nucleus retroambigualis and nucleus paraambigualis, as well as to the midbrain vocal center, the dorsomedial nucleus of the intercollicular complex (Vicario, 1991; Wild, 1993). In the following paragraphs, I review a series of studies of the motor pathway, and I describe a simple model for adult song production that focuses on the role of the descending pathway from HVC to RA to the brainstem.

In reality, HVC has additional connections that complicate this simple model. For the sake of completeness, I include a brief description of these extra connections here, but the function of these connections is outside the scope of this thesis. HVC receives input from multiple brain areas in the thalamus and cortex. In the thalamus, nucleus Uvaeformis projects to HVC (Nottebohm et al., 1982) and is thought to play a vital role in the production of a stereotyped, adult song (Danish et al., 2017; Hamaguchi et al., 2016). HVC is also reciprocally connected with auditory regions in the caudal mesopallium (Akutagawa and Konishi, 2010; Bauer et al., 2008), which may play a role in song learning (Roberts et al., 2017). Finally, HVC also receives input from the cortical nucleus interface (Foster and Bottjer, 1998; Nottebohm et al., 1982). The discussion of the motor pathway in rest of the thesis will only focus on HVC, RA, and the brainstem.
Multiple lines of evidence have implicated HVC in generating the stereotyped structure that is characteristic of adult song. In birds of any age, lesions of HVC cause the song to revert to a juvenile-like song consisting only of highly variable babbling (Aronov et al., 2008). As expected from a brain area driving stereotyped vocalizations, the neural activity in HVC is stereotyped and locked to the motif (Yu and Margoliash, 1996). Furthermore, the projection neurons in HVC have a pattern of activity that is not just highly stereotyped but is also highly sparse. During singing, the projection neurons are silent, except when emitting a high frequency burst of action potentials, which each last about six milliseconds and occur at the same time on each rendition of the motif with less than one millisecond jitter (Kozhevnikov and Fee, 2007). The bursting pattern is slightly different depending on the target of the projection neuron. All projection neurons project to one of two targets, RA in the motor pathway, or the basal ganglia nucleus Area X. HVC neurons that project to RA emit at most one burst per motif, while HVC neurons that project to Area X emit up to four bursts during the motif (Hahnloser et al., 2002; Kozhevnikov and Fee, 2007). As a population, the projection neurons produce bursts that are distributed roughly uniformly over the duration of the motif (Lynch et al., 2016; Picardo et al., 2016).

The sparse, stereotyped firing patterns in HVC have led to the hypothesis that HVC projection neurons provide a clock-like signal that controls the timing of the song. A variety of manipulations to HVC during singing specifically affect the temporal structure of the song. Perturbing the ongoing activity in HVC with electrical stimulation causes the song to terminate prematurely (Vu et al., 1994; Wang et al., 2008). Local cooling, which slows down the dynamics in the target brain region, slows the temporal structure of the song uniformly when applied to HVC, but not to RA (Long and Fee, 2008), but see (Hamaguchi et al., 2016). Since the song stretching is uniform, the dynamics in HVC are presumed to be controlling the song at all timescales. The
exact form of these dynamics is still an open question, but there are several hypotheses. In one hypothesis, HVC projection neurons form a synfire chain (Abeles, 1991; Jin et al., 2007; Li and Greenside, 2006), where the population of projection neurons active at one time directly activate the population active at the next time. Consistent with this model, intracellular recordings from HVC projection neurons found that their membrane potential is relatively constant until just before the beginning of a burst (Long et al., 2010). Alternative hypotheses incorporate a role for inhibitory interneurons in controlling the timing of HVC bursts (Gibb et al., 2009; Kosche et al., 2015; Yildiz and Kiebel, 2011). These results are all consistent with a hypothesis in which HVC generates the timing signals that control the song.

How do timing signals in HVC evoke stereotyped patterns of activity in the muscles of the vocal organ? In one model of song production, HVC drives stereotyped patterns of muscle activity through RA. Like HVC, the firing patterns in RA during singing are bursty, but the RA neurons are much less sparse (Leonardo and Fee, 2005; Yu and Margoliash, 1996). However, RA does not contribute to the temporal dynamics of the song. Manipulations such as electrical stimulation or mild cooling of RA do not alter the temporal structure of the song (Aronov et al., 2011; Long and Fee, 2008; Vu et al., 1994; Wang et al., 2008). Instead, RA is thought to contribute directly to muscle activation. The projection from RA to the motor neurons of the vocal organ is arranged myotopically (Vicario, 1991), and there is massive reduction in the number of cells from about 8000 RA neurons (Gurney, 1981) to only seven muscles (Düring et al., 2013). Within the population of RA neurons that project to each muscle, the sum of many converging RA neurons could create a smooth motor command for the downstream muscle (Leonardo and Fee, 2005). One hypothesis for vocal learning is that the process requires setting the proper synaptic weights from
the timing signal in HVC to the correct motor output in RA (Doya and Sejnowski, 1995; Fiete et al., 2007; Leonardo and Fee, 2005; Yu and Margoliash, 1996).

1.2 Song learning

Song learning proceeds through several stages. The earliest vocalizations are highly variable babbling, similar to the babbling of human infants (Doupe and Kuhl, 1999). This stage, also known as subsong, is characterized by syllables with a broad distribution of durations that cannot be separated into distinct syllable types (Aronov et al., 2008; Marler, 1981). Next, the first stereotyped elements of song appear in the form of syllables with a stereotyped duration, or “protosyllables” (Aronov et al., 2011). As the protosyllable gains stereotyped acoustic structure, it also splits into multiple distinct syllable types (Liu et al., 2004; Okubo et al., 2015). These syllables make up a motif that is further refined as the acoustic structure becomes more and more similar to the tutor song (Tchernichovski et al., 2001).

Throughout learning, the song gradually shifts from variable to stereotyped. The simultaneous reduction in variability and increase in similarity to the tutor song suggest that vocal exploration early in development is shaped by auditory feedback to match the desired goal. This gradual learning from exploration is shared by many other models that fall under the umbrella of reinforcement learning.

1.3 Reinforcement learning in mammals

Reinforcement learning (RL) is a conceptual framework for learning by trial and error. The origins of reinforcement learning date back to early animal behavior experiments and Thorndike’s law of effect: “Actions that produce a satisfying effect in a particular situation become more likely to occur again in that situation” (Thorndike, 1898). Taking inspiration from the way animals learn,
computer scientists developed a mathematical framework for learning using the three elements from Thorndike’s law: actions, situations or “states”, and satisfying effects or “rewards” (Sutton and Barto, 1998). A learner discovers associations between actions, states, and rewards using the final essential ingredient of a reinforcement learning system: exploration. By trying different actions in each state, the learner accumulates knowledge of which paths lead to the most reward. RL-based artificial intelligence systems have been very successful in learning to play a wide variety of games including backgammon (Tesauro, 1995), go (Silver et al., 2016, 2017), and a variety of video games (Mnih et al., 2013).

There are several practical problems in implementing reinforcement learning systems. First, the number of possible state-action combinations may be so large that it cannot be fully explored in a reasonable amount of time. This problem is called the curse of dimensionality. This problem is typically addressed by using a function to approximate the value of each state or state-action combination. When the system encounters a new situation, it estimates the value of the new situation based on similar situations from its past experience.

Another practical problem for RL systems is linking states and actions with subsequent rewards that may arrive after a long delay. For example, in chess, only the final move will give the immediate reward of a checkmate, but every previous move may have contributed to the victory. More generally, when a reward is received, the credit must be distributed among all past states and actions that caused this reward. A naïve solution is to remember the complete history of states and actions, and when a reward arrives, distribute it accordingly. A more common solution is to learn without waiting for the rewards to arrive. As a substitute for future rewards, use the estimated future rewards, perhaps from an approximated value function. Then, the teaching signal is the difference between the reward expected in the previous state and the actual and expected rewards.
from the new state. This reward prediction error or “temporal difference error” is a central concept in RL.

Biologically based models of reinforcement learning following discovery of reward prediction error signals in dopamine neurons. Dopamine neurons encode a wide variety information about rewards and other behavioral events. Dopamine neurons increase their firing rates after an animal receives an unexpected reward such as a morsel of food or a drop of juice (Ljungberg et al., 1992). They are also correlated with other behavioral events, such as salient or aversive events (Matsumoto and Hikosaka, 2009) and movement onsets (Howe and Dombeck, 2016; Jin and Costa, 2010). Within the diversity of dopamine signals, the signal that is relevant to reinforcement learning is reward prediction error, the difference between the reward and the expected reward. When an animal receives an unexpected reward, the reward prediction error is positive. If the reward is fully expected, for example due to the presentation of a conditioned stimulus, then the reward prediction error is zero. Finally, if an expected reward is omitted, the reward prediction error is negative at the time when the animal expected the reward. The reward prediction error was originally conceived by Rescorla and Wagner as a teaching signal for classical conditioning (Rescorla and Wagner, 1972). Reinforcement learning algorithms rely on a closely related quantity, the “temporal difference error” (Sutton and Barto, 1998). The signals in dopamine neurons correspond to this teaching signal (Bayer and Glimcher, 2005; Schultz, 2002; Schultz et al., 1997; Waelti et al., 2001), and the dopaminergic midbrain projects widely throughout the brain (Beckstead et al., 1979; Oades and Halliday, 1987; Swanson, 1982). Thus, the dopamine signal is positioned to be a global teaching signal for reward-related behaviors. Accordingly, manipulation of the activity of dopamine neurons can drive powerful changes in behavior. Electrical stimulation of the dopaminergic midbrain and surrounding areas is so reinforcing that animals that are given a
lever that triggers stimulation in VTA and surrounding brain areas will self-stimulate for hours, often until exhaustion (Corbett and Wise, 1980; Olds and Milner, 1954). Subsequent experiments demonstrated behavioral reinforcement could also be achieved by stimulating only the dopaminergic neurons in VTA (Tsai et al., 2009). To cause such a dramatic change in behavior, dopamine must cause profound changes in the brain.

Where does dopamine change the brain during learning? In many hypotheses, the site of learning is in the basal ganglia and more specifically in the striatum. The basal ganglia receive a massive input from the dopaminergic midbrain, and damage to the basal ganglia disrupts skill learning. For these reasons and many others, the basal ganglia are implicated in action selection and habit learning (Graybiel, 2008). The architecture of the basal ganglia supports the possibility of action selection based on a wide range of sensory and cognitive information about state. Thousands of synapses from a wide range of brain areas converge onto striatal medium spiny neurons (MSNs), the most prevalent cell type in the striatum (Kemp and Powell, 1971). These inputs from cortex have the potential to relay state information that could be used in selecting the best action for a given situation (Graybiel et al., 1994; Wickens and Arbuthnott, 2010). Changing the strengths of these corticostriatal synapses may be an important part of learning. MSNs change their activity during trial and error learning (Barnes et al., 2005; Jog et al., 1999), and such changes may be triggered by dopamine activity. In brain slices, glutamatergic inputs to MSNs undergo plasticity based on spike timing and dopamine rules (Shen et al., 2008), and this sort of learning rule is sufficient to solve the credit assignment problem (Izhikevich, 2007). Indeed, corticostriatal synapses are strengthened after learning to press a lever to trigger electrical stimulation of the dopaminergic midbrain (Reynolds et al., 2001), as well as other tasks (Xiong et al., 2015). But which corticostriatal synapses are strengthened? Answering this question is impossible without
knowing the representation of the information in the corticostriatal connections. Unfortunately, the cortical inputs have not been extensively characterized electrophysiologically (Turner and DeLong, 2000). In songbirds, however, there are only two cortical inputs to the basal ganglia in the song system, and both of them have been recorded during behavior.

1.4 The Anterior Forebrain Pathway: Brain areas for vocal learning in songbirds

Like skill learning in mammals, vocal learning in songbirds requires a basal ganglia circuit. In songbirds, this circuit is called the Anterior Forebrain Pathway (AFP, Figure 1.1B, and it includes brain areas in the cortex, basal ganglia, and thalamus (Figure 1.1A). Area X forms a cortico-basal ganglia-thalamo-cortical loop, similar to basal ganglia circuits in mammals. Area X receives input from two cortical areas, HVC and LMAN, and sends output to the thalamic nucleus DLM (medial portion of the dorsolateral thalamus). To complete the loop, DLM projects back to LMAN. Lesions to any of the nuclei in the AFP in juvenile birds prevent learning an accurate imitation, but lesions to the AFP in adult birds have minimal effects on their song. Since the AFP is not necessary for song production in adulthood, hypotheses about the function of the AFP focus on its role in programming the motor system to perform the adult song.

Many early studies of the role of the AFP in learning observed the changes in behavior after lesions to the output nucleus of the AFP, LMAN. Lesions of LMAN in juvenile birds prevent further learning and cause vocalizations to immediately become highly stereotyped, sometimes just repeating a single syllable. These prematurely crystalized songs are not good imitations of the tutor song (Bottjer et al., 1984; Scharff and Nottebohm, 1991). Even outside the context of natural song learning, LMAN lesions prevent the song from changing in other ways. Typically, deafening an adult bird with a mature, stereotyped song will cause the song to gradually degrade and become more variable (Nordeen and Nordeen, 1992). However, if LMAN is lesioned, the song remains
stereotyped even after deafening (Brainard and Doupe, 2000; Horita et al., 2008). The song also typically changes after damage to the tracheosyringeal nerve, which innervates the syrinx. With LMAN intact, birds add and remove syllables from their songs in the weeks following the nerve injury (Williams and McKibben, 1992), but if LMAN is lesioned before the nerve injury, the number of added and dropped syllables is reduced (Williams and Mehta, 1999). The common theme across all of these situations is that the song can only be variable and malleable when LMAN is intact.

Area X is the singing-related part of the basal ganglia, and early studies suggested that it played a complimentary role to LMAN in the song learning process. Lesions of Area X in young birds prevent them from learning an accurate imitation of their tutor song (Sohrabji et al., 1990). However, unlike LMAN lesions which immediately reduce variability, Area X lesions cause the song to remain in a permanent state of variable syllable structure and ordering (Scharff and Nottebohm, 1991). One interpretation of these results is that Area X reduces or shapes the variability generated by LMAN. I will return to this hypothesis in more detail in Section 1.6.

Area X shares many features with the mammalian basal ganglia. Area X contains elements from both striatum and pallidum. It contains cell types that match cells mammalian striatum and pallidum in both anatomical and physiological features (Farries and Perkel, 2000, 2002; Goldberg and Fee, 2010; Goldberg et al., 2010). One of these cell types, the medium spiny neuron (MSN) bursts once per motif (Goldberg and Fee, 2010). Also like mammals, these MSNs have dopamine-dependent synaptic plasticity rules (Ding and Perkel, 2004).

1.5 Contributions of brain nuclei in the AFP to generating vocal variability

Perhaps the most salient feature of the AFP is its contribution to vocal variability. The lesions to the AFP discussed in the previous section all have some effect on variability, either
Reducing it or prolonging it. Which nuclei are involved in generating this variability and how do they do it? LMAN plays a central role in generating vocal variability. Lesion or pharmacological inhibition of LMAN reduces variability in vocalizations of birds of all ages, but the stereotyped structure of the song is preserved (Bottjer et al., 1984; Olveczky et al., 2005; Scharff and Nottebohm, 1991). In support of its role in driving variability, LMAN neurons have firing patterns that are highly variable, with some motif-locked activity (Aronov et al., 2008; Olveczky et al., 2005). While these data do not rule out the participation of other brain areas like Area X or RA in the generation of variability, at least some of the dynamics underlying this variability occur within LMAN. Mild cooling of LMAN in juvenile birds lengthens the duration of non-stereotyped syllables (Aronov et al., 2011).

Multiple lines of evidence support the idea that LMAN affects vocal output by directly activating neurons downstream in RA. LMAN projection neurons are glutamatergic and send collaterals to both Area X and RA (Nixdorf-Bergweiler et al., 1995; Vates et al., 1997). In RA, the variable activity from LMAN may add to stereotyped activity driven by HVC to cause random fluctuations in vocal output. Electrical stimulation of LMAN can cause instantaneous changes in acoustic features of the song (Kao et al., 2005), and pharmacologically inactivating LMAN reduces the variability of firing patterns in RA (Olveczky et al., 2011).

Does Area X contribute to the variability generated by the AFP? Multiple forms of variability persist after Area X lesions, suggesting that variability is generated elsewhere. Both babbling in juveniles (Goldberg and Fee, 2011) and motif-to-motif variability in pitch in adults (Ali et al., 2013) are unaffected by Area X lesions. However, an alternative view suggests that some forms of vocal variability depend on Area X (Kojima et al., 2018). This variability may originate with the variable spiking patterns of pallidal neurons inside Area X (Woolley and Kao,
2015; Woolley et al., 2014) or its glutamatergic interneurons (Budzillo et al., 2017). The preponderance of evidence supports the idea that Area X is not necessary for generating most forms of variability. For this thesis we will adopt the hypothesis that variability is generated in LMAN.

1.6 Circuit level model of reinforcement learning in the AFP

Further study of the contribution of the AFP to song learning has been challenging because natural song learning takes weeks to months and proceeds unpredictably (Deregnaucourt et al., 2004; Tchernichovski et al., 2001). Using a new operant conditioning paradigm, changes at a predictable target in the song can be learned in just hours. The song is monitored in real time by a computer, and the pitch of the song is computed at one target time in the motif. This pitch varies from rendition to rendition due to the influence of LMAN. When the pitch on the current rendition is below a threshold, a loud burst of white noise is played 50 milliseconds or less after the target time through a loudspeaker next to the cage. Over several hours, the bird learns to sing at a higher pitch to avoid triggering the noise (Andalman and Fee, 2009; Charlesworth et al., 2011; Tumer and Brainard, 2007; Warren et al., 2011). This learning by conditional auditory feedback requires the same brain areas as natural song learning. These pitch changes cannot be learned after lesions of Area X (Ali et al., 2013). Just as in natural song learning, the actively generated variability from LMAN is required too (Warren et al., 2011), but subsequent experiments demonstrated that LMAN contributes even more to the learning process.

Since the discovery that learning ceases after LMAN lesions, LMAN was hypothesized to send the motor pathway an instructional signal. It was unclear how LMAN could drive vocal variability while also giving an instructive signal. This instructional signal was thought to be auditory (Bottjer et al., 1984), the result of a comparison of the what the bird hears himself sing
against a stored memory of his tutor song. Initially, this theory found support from recordings LMAN neurons in anesthetized birds, which revealed auditory responses to playback of the bird’s own song or the song of his tutor (Doupe, 1997; Doupe and Konishi, 1991; Lewicki, 1996; Margoliash, 1983; Volman, 1996). However, subsequent studies ruled out an auditory role for LMAN by recording LMAN neurons in awake birds and finding no response to noise bursts played during singing (Leonardo, 2004).

Surprisingly, LMAN does relay an instructional signal to the motor pathway, but not an auditory one. Instead, the instructional signal comes from a bias embedded in the LMAN-driven variability. The variability is biased towards variants that reduce vocal errors. In conditional auditory feedback experiments, where a bird learns to change the pitch of his song at a targeted time to escape bursts of white noise, pharmacological inactivation of LMAN removes the variability as expected, but also removes the recently learned shift in pitch (Andalman and Fee, 2009; Warren et al., 2011). The learned shift in pitch returned after the pharmacological agent has washed out. Of course, this bias must be temporary because LMAN is not necessary for song production in adults. Indeed, if the pitch shift is maintained for multiple days, LMAN inactivation has a progressively smaller effect (Warren et al., 2011).

The transient role of LMAN in learning new song changes has led to a hypothesis in which song learning occurs in two stages proceeding in tandem. First, learning within the AFP allows LMAN to generate biased variability that biases the song towards variants that sound more similar to the tutor song. Second, the bias is consolidated into the HVC → RA pathway. For the remainder of the thesis, I will focus on the first learning process that occurs within the AFP.

Learning in the AFP is driven by reward prediction error signals from VTA. During conditional auditory feedback, neurons in VTA that project to Area X decrease their activity
immediately following a noise burst. Inversely, when a noise burst is omitted, these neurons increase their activity immediately after the target time (Gadagkar et al., 2016). This pattern of activity is a reward prediction error, very similar to the reward prediction error in mammalian VTA neurons. In this case, a noise burst is played at the target time on about 50% of renditions, so the expected reward is midway between the low reward of a noise burst and a high reward of an escape. Therefore, a noise burst is lower than the expected reward and an escape is higher.

Manipulating Area X-projecting VTA neurons directly is sufficient to drive learning, substituting for the noise burst. Just like noise bursts, inhibition of VTA neurons that project to Area X is aversive. For example, inhibiting VTA neurons when the bird sings at a lower pitch causes the bird to shift his song towards a higher pitch. Inversely, stimulating VTA neurons is reinforcing and causes birds to sing in the stimulated pitch range more often (Xiao et al., 2018). Several computational models of song learning have been proposed using this type of error signal (Doya and Sejnowski, 1995; Fee and Goldberg, 2011; Fiete et al., 2007).

This thesis focuses on the framework laid out by Fee and Goldberg (2011), in which a reinforcement signal from VTA strengthens corticostriatal synapses in Area X to drive bias in LMAN (Fee and Goldberg, 2011). Area X receives an image of the vocal variability via the axon collaterals from LMAN. When a reinforcement signal arrives from VTA, Area X can associate the reinforcement with the pattern of variability that caused it, in order to reproduce that pattern of LMAN activity again on future renditions. This computation must take place independently at every moment in the song, since the variant that makes the song “better” at one time may not make the song “better” at a different time in the motif. Therefore, the reinforcement signal must strengthen HVC synapses such that the timing signals from HVC drive the appropriate pattern of bias at each moment. This model hypothesizes a three-part learning rule that depends on the inputs
from HVC, LMAN, and VTA: dopamine input from VTA strengthens HVC synapses onto MSNs that recently received a signal from LMAN. In Chapter 2, I implement this model and show that it reproduces a wide variety of experimental results.

1.7 Origins of the reward prediction error signal

Like many biological models of reinforcement learning, my computational model depends on a reward prediction error (RPE), which experimental evidence suggests is carried by VTA neurons. A reward prediction error is just one of many signals that have been described in VTA neurons in birds and mammals; VTA neurons also carry a wide range of information relating to rewards, salient events, and motivation. To understand the learning process, it is crucial to understand how the brain computes the RPE signal in VTA neurons. In Chapter 3 of my thesis, I investigate the contribution of one area of the songbird brain to the RPE signal in VTA neurons.

The source of the reward prediction error signal in VTA is an area of active research. The output of VTA dopamine neurons is consistent with a simple subtraction of the actual reward minus the expected reward (Eshel et al., 2015), so it is tempting to imagine that just two inputs converge on VTA, one carrying reward information and the other carrying prediction information. Unfortunately, the truth is much more complicated.

1.7.1 Reward-related activity in the inputs to VTA in mammals

In mammals, VTA receives input from a wide range of brain areas, both cortical and subcortical (Geisler et al., 2007; Sesack and Grace, 2010; Watabe-Uchida et al., 2012), including from the lateral hypothalamus, ventral striatum, ventral pallidum, frontal cortex, and lateral habenula. Untangling the roles of brain areas is especially difficult because the areas that project to VTA are also interconnected with each other (Geisler and Zahm, 2005). No consensus has
emerged on how the signals are combined to form a reward prediction error (Watabe-Uchida et al., 2017). Nevertheless, I review a few elements of this circuit from studies in rodents and primates in the next few paragraphs, and then I will return to the possible sources of reward prediction error in birds.

One pathway for reward related signals comes from the lateral habenula (LHb). LHb projects directly to VTA, as well as indirectly though the rostromedial tegmental nucleus (RMTg). Neurons in both LHb and RMTg have responses that are the inverse of reward prediction error responses in VTA (Hong et al., 2011; Jhou et al., 2009; Matsumoto and Hikosaka, 2007). Their neurons decrease their firing rates upon presentation of a reward or a reward-predicting cue and increase their firing rates in response to an aversive stimulus or an omitted reward. LHb and RMTg send a GABAergic projection to VTA, and if this signal inhibits VTA dopamine neurons, it can reproduce the reward prediction error observed in those neurons. Even though it seems like all the information necessary to compute a reward prediction error is present in these neurons, they are not necessary for many of the response properties of VTA neurons. Even after lesions of LHb, VTA neurons still increase their firing rates in response to rewards and reward predicting cues. LHb lesions only eliminate the phasic decrease in VTA neuron activity when a reward is omitted (Tian and Uchida 2015). For its response to reward to persist after LHb lesions, VTA must receive reward-related signals from other parts of the brain.

Another source of reward signals comes from the orbitofrontal cortex (OFC). Single neurons in the OFC increase their firing rates when a mammal is presented with a rewarding stimulus or when a reward is expected (Feierstein et al., 2006; Padoa-Schioppa and Assad, 2006; Rolls, 2000; Schoenbaum and Eichenbaum, 1995; Tremblay and Schultz, 1999; Wallis and Miller, 2003). Lesions of OFC disrupt value-guided choice behavior, especially when the options change
in value (Butter et al., 1969; Izquierdo et al., 2004; Pickens et al., 2003). These lesions also change the activity of VTA dopamine neurons, but some reward information remains in their activity patterns (Takahashi et al. 2011). Further research will be necessary to finish untangling the roles of the LHb, OFC, and the many other brain areas that are connected to VTA.

1.7.2 Inputs to VTA in songbirds

A similar search is ongoing for the sources of auditory error signals in the VTA of songbirds. In zebra finches, Area X-projecting VTA neurons decrease their activity following a burst of white noise played during singing. When an expected noise burst is omitted, these neurons increase their firing rates (Gadagkar et al., 2016). This error signal is widely believed to be the result of a comparison between what a bird hears during singing and a stored memory of the tutor song. The components of this computation may be carried by the inputs to VTA, and several studies have explored the roles of these inputs in vocal learning.

One input to VTA, which is not the focus of my thesis, is from the ventral pallidum (VP), a part of the basal ganglia that is outside of Area X (Chen et al., 2018; Gale et al., 2008). Chen et al. discovered vocal error related information in VP in recent, unpublished work. In their recordings, VTA-projecting VP neurons carry reward-related information, including predicted reward and even reward prediction error (Chen et al., 2018). These signals may play an important role in computing the reward prediction error signals in VTA.

In my thesis, I have focused on inputs to VTA from a second set of inputs to VTA from cortical areas. In zebra finches, the only cortical input to VTA comes from the intermediate arcopallium (Gale et al., 2008). The intermediate arcopallium has been subdivided into two parts, ventral (Alv) and dorsal (Ald), and these two divisions may have different contributions to VTA.
AIV is required for vocal learning and sends a vocal error signal to VTA. Lesions of AIV in juvenile birds prevents them from learning an accurate imitation of the song of their tutor (Mandelblat-Cerf et al., 2014). In addition to lesion studies, neurophysiological evidence also supports a role for AIV in song learning. AIV receives input from auditory pathways (Kelley and Nottebohm, 1979; Mello et al., 1998; Vates et al., 1996), some of which have a phasic increase in their activity immediately following a noise burst during singing (Keller and Hahnloser, 2009). Similar to these auditory brain areas, neurons in AIV that project to VTA also have a phasic increase in activity after a noise burst is delivered during singing (Mandelblat-Cerf et al., 2014). If these AIV neurons inhibited the neurons in VTA that project to Area X, it could be responsible for the decrease in activity in those neurons following a noise burst.

In contrast, the role of AId in behavior is controversial, and a major focus of my thesis is to help resolve this controversy. Separate lines of research from different labs have focused on two different hypotheses for the role of AId. One hypothesis is that AId contributes to vocal learning by processing vocal errors, perhaps similar to AIV. The second hypothesis is that AId is not related to singing or vocalization at all, but is instead related to controlling head and body movements. Before explaining these two hypotheses in more detail, I will first review the anatomy of AId.

AId is part of a separate network of brain areas that parallel the song system in their anatomical positions and connectivity. This anatomy is depicted schematically in Figure 1.2. In the arcopallium, AId is immediately lateral to RA (Bottjer et al., 1989). These areas both send descending projections out of the telencephalon. They also project to adjacent regions in the dorsal thalamic zone (DTZ) (Bottjer et al., 2000), and these portions of the thalamus project to neighboring regions of nidopallium. In the song system, DLM projects to LMAN, while in the parallel network DTZ projects to LMANshell, a region of parvocellular cell bodies surrounding
LMAN (Johnson and Bottjer, 1992). LMANshell completes the loop by projecting back to AId directly and indirectly via dNCL (dorsal part of the caudolateral nidopallium) (Bottjer et al., 1989, 2000). The similarities between the anatomy and the song system and the anatomy of this parallel pathway implies that there may be some shared function between these two pathways.

One hypothesis is that the parallel pathway, including AId, is involved in vocal learning. Even though there are no known connections between this parallel pathway and the ascending auditory system, two of the nuclei in this pathway, dNCL and LMANshell, show physiological evidence of processing auditory information. In dNCL, immediate early gene expression is high after singing as well as after hearing playback of a tutor or conspecific song, compared to birds who did not sing or hear song playback (Bottjer et al., 2010). In LMANshell, recordings under anesthesia found neurons that respond to playback of the bird's own song or the tutor song (Achiro and Bottjer, 2013). During singing, about 10% of units in LMANshell either increase or decrease their activity during the production of syllables that are more similar to syllables in the tutor song (Achiro et al., 2017). It is not known if these signals are relayed to VTA because no recordings have been performed in AId.

Figure 1.2: Brain areas of the parallel pathway

29
The vocal learning hypothesis has also been tested with lesions of AId, but the results of these studies are difficult to interpret due to the anatomical relationship between AId and AIv. In one study, lesions of AId in juvenile birds was reported to impair imitation (Bottjer and Altenau, 2010). However, a subsequent study contradicted these results and found that AId lesions had no impact on learning. Instead, only lesions of AIv impacted learning (Mandelblat-Cerf et al., 2014). One possible explanation is that the earlier study lesioned both AId and AIv, since they are adjacent. Anatomically, not only are AIv and AId adjacent, they may also be partially overlapping. This overlap is possible because AIv is defined by its outputs: AIv is the portion of intermediate arcopallium that projects to VTA (Mandelblat-Cerf et al., 2014; Mello et al., 1998). On the other hand, AId is defined by its inputs: AId is the portion of intermediate arcopallium that receives a projection from LMANshell, part of the nidopallium surrounding the song nucleus LMAN (Bottjer et al., 2000; Johnson et al., 1995). By these definitions, there is a small portion on the medial edge of AId that overlaps with AIv. However, AId is largely distinct from the VTA projecting region AIv (Mandelblat-Cerf et al., 2014). Given that only a small portion of AId projects to VTA, it may seem unlikely that AId as a whole is contributing to the vocal error signals in VTA. Rather than focus on this small overlapping region, in this thesis, I investigate the role of AId as a whole by recording from single neurons throughout AId.

An alternative hypothesis for AId is that it not involved in singing or song learning at all, but instead controls head and body movements. The most striking evidence in favor of this hypothesis comes from a study of immediate early gene expression by Feenders et al. (2008). They examined immediate early gene expression throughout the zebra finch brain during two different activities, singing and hopping. During singing, immediate early gene expression is high in the nuclei of the song system, including HVC, RA, LMAN, and Area X, but low in the surrounding...
areas of the brain. During hopping, the expression pattern is reversed, with high expression in the areas surrounding the song system, including AId, LMANshell, and other areas in its parallel pathway. The juxtaposition of the song system with a network of hopping-related brain areas led Feenders et al. to hypothesize that the song system evolved from a more general motor control circuit (Chakraborty and Jarvis, 2015; Feenders et al., 2008). By this hypothesis, AId should be involved in motor control, not vocal learning.

Additional evidence supports the hypothesis that AId is involved in motor control. Anatomically, AId sends output to multiple subcortical motor centers including the midbrain reticular formation and the deep layers of the tectum (Bottjer et al., 2000). Consistent with a role in controlling movements, large lesions of AId cause akinesia (Mandelblat-Cerf et al., 2014). However, no recordings of AId have ever been performed in an awake bird.

Chapters 3 and 4 of my thesis test two hypotheses of AId function that have emerged from the literature. In Chapter 3, I test the hypotheses that AId is involved in processing vocal errors or controlling movements. I perform the first recordings of AId in an awake bird and record the same neurons during both singing and natural movements. I find evidence that AId predominantly encodes movement-related information with large modulations in activity around the times of head rotations and whole-body movements such as pecking. As a population, these neurons have a very small reduction in firing rate following noise bursts, so I cannot rule out that they are involved in processing vocal errors.

In Chapter 4, I further investigate the role of AId in movements. I design a microdrive for recording simultaneous neural activity and movement signals. To analyze the movement signals, I develop an algorithm to automatically classify natural zebra finch behavior and relate these behavior classes to the neural activity in AId.
1.1 Summary

Trial and error learning is ubiquitous in animal behavior. Among animal models of trial and error learning, the songbird has several advantages. Song learning involves a relatively small number of brain areas with known connectivity, and neural activity has been recorded from many of these brain areas during the same learning behavior. However, there are several notable gaps in understanding. First, there is no computational model that capture the recently discovered role of a basal ganglia circuit in driving biased variability during song learning. Second, there is some uncertainty about which brain areas are involved in computing an error signal. This thesis seeks to fill those gaps.

Chapter 2 constructs a computational model of song learning based on RL that provides a unified explanation for many behavioral results that could not be explained by previous models.

Chapter 3 contains the first recordings from Ald in an awake bird. Each neuron was recorded both during singing and during natural movements. These recordings lend strong support to the idea that Ald neurons are related to movements and not to processing vocal errors.

Chapter 4 further quantifies the head and body movements of freely moving birds and explores the relationship of neural activity in Ald to these movements.

Chapter 5 discusses the relationship between the song circuit and the surrounding brain regions and relates my research to other studies of motor control more generally.
2 Computational model of reinforcement learning

2.1 Introduction

The song learning process has inspired a series of computational models based on reinforcement learning. In the earliest reinforcement learning model of song learning, Doya and Sejnowski proposed that LMAN drives stochastic changes in HVC-RA weights on each rendition of the song. In the model, the vocal output is compared to a stored memory of the tutor song by Area X. Area X sends a reinforcement signal to LMAN that indicates if the perturbation in this trial made the song sound better than average. If so, the HVC-RA weights are modified to keep the perturbation (Doya and Sejnowski, 1995, 1998). This paper established an important framework that is adopted by future models of song learning, including in this paper. Variability from LMAN is correlated with a reinforcement signal to bias the song towards variants that give high reinforcement. Following this procedure, the HVC-RA weights undergo stochastic gradient ascent to maximize the reinforcement signal. Unfortunately, there is no known biophysical mechanism that can perturb synaptic strengths to a new value on every motif, as the Doya and Sejnowski model requires.

Subsequent experiments demonstrated that LMAN has a rapid, glutamatergic influence on the song, which inspired the next generation of song learning models. Electrical stimulation of LMAN causes rapid and transient fluctuations in the song (Kao et al., 2005). In light of this, Fiete et al. proposed a new model in which LMAN caused random fluctuations in RA activity. A reinforcement signal to RA strengthened HVC-RA synapses if an LMAN fluctuation caused the song to sound better than average (Fiete et al., 2007). Fiete et al. made an explicit distinction
between the two classes of synapses onto RA neurons. Each RA neuron receives a single “empiric” synapse from LMAN that drives random fluctuations in RA activity but does not change in strength during learning. Each RA neuron receives many plastic synapses from HVC that do change with learning. A plastic synapse is eligible for strengthening when it is coactive with the empiric synapse, and eligible synapses are strengthened when a reinforcement signal is received. Eligible synapses maintain an eligibility trace to allow for a delayed reward signal. One prediction of this model is that removing LMAN after learning will leave the learned song intact and only remove vocal variability.

Detailed studies of learning revealed that LMAN contributes more than just variability – the variability can be biased in a direction to reduce errors. In these studies, rapid learning was induced with a short, loud burst of noise which is played while the bird is singing (Andalman and Fee, 2009; Turner and Brainard, 2007). The noise is contingent on the pitch of the song at a single moment in the motif – for example, pitches above a threshold are hit with noise while pitches below the threshold escape punishment (Figure 2.1E). Over just a few hours, birds rapidly learn to sing at a lower pitch at the targeted time to avoid the noise. Surprisingly, this learned shift in pitch is driven by activity in LMAN. When LMAN is pharmacologically inactivated or its output to RA blocked after learning induced by conditional auditory feedback, the mean pitch returns to its pre-learning levels (Figure 2.1F) (Andalman and Fee, 2009; Warren et al., 2011). LMAN is carrying a signal that biases vocal output towards variants that escape the noise. There must be a separate learning process in the LMAN circuit that rapidly learns this bias.

The discovery of biased variability led to a conceptual model in which Area X learns to bias ongoing variability in LMAN (Fee and Goldberg, 2011). This is the model we implement in this chapter. In this model, Area X associates variations in LMAN activity with a dopaminergic
reinforcement signal from VTA. Then on subsequent renditions, Area X feeds back to reactivate the same LMAN neurons that were associated with reward. Of course, this computation must be performed locally at each time in the song – the increase in pitch that makes the song sound better at one time may not make the song sound better at another time. Therefore, the model proposes that plastic changes in the timing signals from HVC drive a bias that is specific to one time in the song. Fee and Goldberg proposed the following three-part learning rule for MSNs in Area X: an HVC-MSN synapse is strengthened when the HVC and LMAN inputs are co-active, followed by reward.

The Fee and Goldberg model retains the plastic/empiric synapse distinction from the Fiete et al. model. Each model MSN receives a single empiric synapse from LMAN and a single plastic synapse from HVC representing the neurons that are active at a single moment in the song.

Borrowing from the mammalian reinforcement learning literature, Fee and Goldberg hypothesized that the reinforcement signal to Area X comes from dopaminergic neurons in VTA and SNc. This prediction was confirmed a few years later with the discovery of a vocal error signal in dopamine neurons that project to Area X (Gadagkar et al., 2016). The activity in these dopamine neurons decreased following a noise burst during singing, but increased when an expected noise burst was omitted.

Alternative models for song learning combine vocal exploration and auditory feedback in different ways. For example, auditory feedback can be used to build a forward model of the vocal production system that predicts the auditory feedback that would be created by a given motor command. This prediction could be used in place of a real-time reinforcement signal to train the song production circuitry (Troyer and Doupe, 2000). Another alternative is to learn an inverse model, rather than a forward model. An inverse model can translate a desired auditory stimulus
into the motor commands necessary to recreate that stimulus. Once learned through exploration and feedback, such an inverse model could produce an imitation by replaying the auditory memory of the tutor song through the inverse model (Hahnloser and Ganguli, 2013).

In this chapter, we implement the Goldberg and Fee model as a firing rate-based neural network and discover that it quantitatively reproduces several features of song learning. I begin by demonstrating that the model can learn a simple song. Next, I explore some of the potential weakness of this reinforcement learning model. In general, reinforcement learning models scale poorly with the number of degrees of freedom and with the delay and precision of the reinforcement signal. I demonstrate that the model is robust to the number of degrees of freedom in the song and the shape of the reward signal.

I also demonstrate that the model can explain several surprising behavioral results from conditional auditory feedback experiments. The model reproduces the precise temporal structure of learning during conditional auditory feedback. In a careful analysis of learning during conditional auditory feedback (CAF), Charlesworth et al (2011) found that the pitch changes in a small window about 20ms around the target time. This observation was especially interesting because the exact temporal structure of this learning can be predicted from the average of the trials that escape punishment (Charlesworth et al., 2011). The precision of this learning is especially surprising considering that the reinforcement signal may be delayed and imprecise in time. Our model reproduces this feature of learning, but only when the reward signal is indeed broad.

In another set of experiments based on CAF, Charlesworth et al. demonstrated that LMAN is capable of latent learning (Charlesworth et al., 2012). In Bengalese finches, blocking the projection of LMAN to RA leaves about 70% of the vocal variability intact (Charlesworth et al., 2012; Warren et al., 2011). Performing CAF does not result in a pitch change as long as the block
on LMAN->RA is in place. However, as soon as the blockade is removed, the song immediately has a bias that allows it to escape the noise, as if this bias was learned covertly but was only able to be expressed through the connection from LMAN to RA. Crucially, no bias is learned at all if LMAN activity is inhibited using muscimol. We demonstrate that our model is capable of covert learning in this fashion by receiving a copy of the non-LMAN variability through the RA->DLM->LMAN projections, as suggested by Charlesworth et al.

In the final sections of this chapter, we extend the model in more speculative directions. First, we explore how MSNs in Area X might develop their characteristic sparse firing patterns. In our original model, each MSN was active once per motif because it received input from a single HVC neuron. In reality, there is massive convergence from cortical neurons onto MSNs (Kincaid et al., 1998), yet individual MSNs are active at only a single moment during singing (Goldberg and Fee, 2010). We demonstrate that adding long-term depression (LTD) and lateral inhibition allows each MSN to develop this selectivity from initial all-to-all connectivity during the learning process.

Finally, we extend the model to include more of the internal circuitry of Area X. Like mammalian basal ganglia, Area X contains two pathways: the direct and indirect pathways (Farries et al., 2005). In models of mammalian basal ganglia function, the direct pathway activates movements while the indirect pathway inhibits movements (Albin et al., 1989; Alexander and Crutcher, 1990a; DeLong, 1990; Mink, 1996). Our original model only contains the direct pathway, so in the final section we extend these ideas to our learning model in which the indirect pathway biases song output away from variants that sound worse. This compliments the direct pathway, which biases the song towards variants that sound better.
2.2 Methods

Our implementation of the Fee and Goldberg model uses units with smoothly varying activities representing the aggregated firing rates of groups of hundreds to thousands of neurons in Area X and connected brain regions. The model includes HVC, LMAN, Area X, DLM, and VTA. The roles of each of these brain areas are described in the following sections.

2.2.1 HVC

HVC produces stereotyped timing signals that uniformly cover the entire song. Each HVC projection neuron bursts highly selectively and reliably at 1-3 moments in the song (Figure 2.1B) (Kozhevnikov and Fee, 2007). For simplicity, HVC units in our model burst once per motif. Each unit $H_i$ emits a single burst at a unique time $t_i$ with a burst that lasts $\tau_H$ milliseconds in the motif and is otherwise silent. Each burst is shaped like one period of a sine squared curve and sequential bursts are staggered by 1 ms and scaled so that the sum of all HVC activity is always one:

$$H_i(t) = \begin{cases} \sin^2\left(\frac{\pi}{\tau_H}(t - t_i)\right) & t_i \leq t < t_i + \tau_H \\ 0 & \text{otherwise} \end{cases}$$

2.2.2 LMAN

LMAN produces biased variability in the output of the model. The output of each LMAN unit is a combination of a constant baseline level of activity ($L_0$), intrinsic variability ($Z$), and excitatory input from DLM that drives bias ($D$):

$$L^n_k(t) = [L_0 + Z^n_k(t) + D^n_k(t)]_0$$

Both $Z$ and $D$ can be positive or negative and cause fluctuations around baseline LMAN activity $L_0$. The random noise process $Z$ was constructed so that the pitch fluctuations from the
model matched the dynamics of natural pitch variations from young adult zebra finches (see Section on Pitch Fluctuations below).

There are two units in LMAN: activity in the first LMAN unit ($L_{UP}$) increases the pitch of the song and activity in the second unit ($L_{DOWN}$) decreases the pitch ($p$) around a baseline (Figure 2.4E):

$$p^n(t) = L^n_{UP}(t) - L^n_{DOWN}(t)$$

where $n$ is the motif number. This pitch is the output of the model.

2.2.3 Area X and DLM

Area X drives temporally localized bias in LMAN fluctuations in our model. Each LMAN unit is connected with a separate circuit of bias-driving neurons in Area X and DLM. One channel of this circuit is shown in Figure 2.2B. Each MSN unit receives input from a single HVC unit and a single LMAN unit. Within each channel, there is a complete representation of time - there is an MSN unit for each HVC unit. There are 50 MSN units per channel. The activity of each MSN unit $M_j$ is determined by its HVC input $H_i$ and learned synaptic weight, $w_{ji}$. MSN unit activity is constrained to be nonnegative:

$$M^n_j(t) = [w^n_{ji}H_i(t)]_0$$

The weights are initialized to 0.001 and change with learning (see Section on learning below). Note that LMAN activity does not directly affect the output of MSN units even though each MSN receives an afferent from LMAN. In our model, the only role for LMAN in Area X is to control synaptic plasticity (see Section on plasticity below).

All of the MSNs from a channel converge on a single pallidal neuron:

$$P^n_k(t) = - \sum_j M^n_j(t)$$
Each pallidal unit inhibits a single DLM unit in each channel:

\[ D_k^n = -P_k^n(t) = \sum_j M_j^n(t) \]

Both DLM and Pallidal units can have both positive and negative activities, representing fluctuations around a high baseline firing rate. We define bias as the influence of DLM on the song:

\[ \text{bias}_n(t) = D_{UP}^n(t) - D_{DOWN}^n(t) \]

Strong weights can drive a temporally precise bias in LMAN units. A strong \( w_{ij} \) allows a burst in HVC unit \( i \) to drive a burst in MSN unit \( j \) (Figure 2.2C). A burst in the MSN unit will inhibit firing in the pallidal unit which will disinhibit the DLM unit. Finally, the DLM unit will drive increased activity in the LMAN unit (Figure 2.2C).

2.2.4 VTA

In our model, a single VTA unit carries a reward prediction error signal, the result of a comparison between vocal output and the template. The comparison begins by computing the instantaneous vocal error \( e(t) \), which is absolute value of the difference between the vocal output \( p \) and the template pitch \( \hat{p} \) (Figure 2.3A):

\[ e^n(t) = |p^n(t) - \hat{p}(t)\hat{p}(t)|^2 \]

Reward is the opposite of this error. In addition to a sign inversion, the reward signal \( r \) is delayed and blurred with a kernel \( B \), which we chose to be a Gaussian with standard deviation \( \sigma_B = 12.5 \) ms unless otherwise specified. The peak of the reward occurs at a \( 4\sigma_B = 50 \) ms delay.

\[ r^n(t) = -\sum_{\tau=0}^{8\sigma_B} B(\tau)e(t - \tau) \]

\[ B(\tau) = \frac{1}{\sqrt{2\pi\sigma_B^2}} \exp\left(-\frac{(\tau - 4\sigma_B)^2}{2\sigma_B^2}\right) \]
Finally, the reward prediction error $\delta$ is the difference between the actual reward and the expected reward $\tilde{r}$:

$$\delta^n(t) = r^n(t) - \tilde{r}^n(t)$$

The expected reward is calculated for each timestep in the motif as the average of the recent rewards received. It is initialized to zero and updated using the reward prediction error (Sutton and Barto, 1998):

$$\tilde{r}^{n+1}(t) = \tilde{r}^n(t) + \gamma \delta^n(t)$$

where $\gamma = 0.2$ is the learning rate. In this framework, small vocal errors cause small reward prediction error signals that can guide learning. The reward prediction error is positive if the song is more similar to the template than it is on average. The output of the VTA unit is the reward prediction error $\delta$. This signal can be positive and negative, representing fluctuations above and below a baseline level of dopamine neuron activity.

2.2.5 Learning

Phasic dopamine signals trigger synaptic plasticity in HVC-MSN synapses to create a bias that reduces vocal errors. When an LMAN fluctuation causes an increase in the reinforcement signal $\delta$, HVC-MSN synapses are strengthened such that HVC will drive the same pattern of LMAN activity on subsequent renditions of the song. This synaptic plasticity follows a three-part learning rule including the activity of HVC, LMAN, and VTA afferents to MSNs.

LMAN units gate plasticity of HVC-to-MSN synapses to restrict plasticity to a single LMAN channel. It is crucial that plasticity be restricted to the active LMAN channel so that pitch will change in the right direction. For example, if activity in the pitch-up LMAN channel increases the pitch to more closely match the template and causes a positive reward prediction error, the
synapse in the pitch-up channel should be strengthened, not the synapse onto the pitch-down channel, to produce higher pitches at this time on future renditions. A synapse is eligible for plasticity if it receives coincident LMAN and HVC activity. This tag is convolved with the same kernel as used for reward to make an eligibility trace:

\[ b_{ji}^n(t) = \sum_{\tau=0}^{8\sigma_B} L_k^n(t - \tau) H_i^n(t - \tau) B(\tau) \]

When a reward arrives, eligible synapses are changed proportional to the reward prediction error:

\[ w_{ji}^{n+1} = w_{ji}^n + \eta \sum_t b_{ji}^n(t) \delta^n(t) \]

2.2.6 Analysis of pitch fluctuations

In the model, LMAN-driven fluctuations in vocal output were matched to the power spectrum of pitch fluctuations observed in young adult zebra finches. To measure these pitch fluctuations, nine birds aged 79 to 135 days post hatch were isolated in sound attenuating chambers and their vocalizations were recorded at a 40 kHz sampling rate with custom software written in MATLAB. Songs were segmented into syllables based on sound amplitude, and syllables were classified based on their acoustic features. A target region was manually selected in syllables with harmonic stacks longer than 20 ms for further analysis (Figure 2.4A). For each bird, pitch measurements in the target interval were normalized to percent difference from the mean pitch.

One day of singing consisting of 283 to 3464 renditions of the target syllable was used for analysis in each bird. Renditions in which the pitch changed faster than 100 Hz/ms at any time during the target interval were discarded.
Model LMAN fluctuations were generated by applying a frequency domain filter to Gaussian white noise. The white noise was transformed into the Fourier domain and multiplied by the average amplitude spectrum of the measured pitch fluctuations from one bird and inverse Fourier transformed back into the time domain. The noise is normalized to have a variance of one.

In some simulations, we explored the role of the temporal structure of these LMAN fluctuations by speeding up or slowing down the variability. To change the timescale of these fluctuations, the frequency domain filter was stretched by a factor of up to 100 or compressed by a factor of up to 10. For frequencies less than 33 Hz, we resampled the spectrum, linearly interpolating where necessary. For frequencies greater than 33 Hz, we fit a line to the log-log spectrum between 33 and 96 Hz and stretched that linear fit to calculate the new spectrum. For making slower LMAN fluctuations, the spectrum was compressed linearly, and the fit to the tail of the spectrum was used to extrapolate the power spectrum. The total power of LMAN fluctuations is always normalized to one.

2.2.7 Analysis of learning rate with

We simulated additional degrees of freedom by adding noise sources to the reward signal. Each LMAN channel learns independently, so adding additional degrees of freedom only decreases the learning rate by diluting the reward signal. The error signal is now:

$$e^n(t) = \sum_{m=1}^{M} (p^n_m(t) - \hat{p}_m(t))^2$$

Where $M$ is the number of degrees of freedom and $p_m(t)$ is the vocal output in the $m$-th dimension at time $t$. The total error is normalized to have the same total variance for any number of degrees of freedom.
2.2.8 Simulating conditional auditory feedback

Conditional auditory feedback is a behavioral paradigm that induces rapid vocal learning. The pitch of the song is monitored in real-time, and if the pitch at a targeted time in the motif is below a threshold, a loud burst of white noise is played through a loudspeaker next to the cage. Within hours, the bird learns to raise the pitch of his song at the target time to avoid triggering the noise. Typically, the threshold is chosen to have about 70% hits based on baseline songs (Andalman and Fee, 2009; Charlesworth et al., 2011; Tumer and Brainard, 2007; Warren et al., 2011).

In the model, we implemented conditional auditory feedback by manipulating the error signal. The pitch was measured at time step 100, and if the pitch was below a threshold, the error is set to be equal to 400 for the duration of the noise burst (Figure 2.3A). We used a noise duration of 2 ms, and a threshold that caused about 70% of renditions to be hit with noise at the beginning of learning. We chose an extremely short burst of noise so that the duration of the noise burst would not affect the time course of learning.

Figure 2.3 shows the reward prediction error computation from several trials of simulated conditional auditory feedback. When the model produces a pitch that is below threshold, a 20 ms burst of noise is given (Figure 2.3B). The noise burst causes a large error (Figure 2.3C) or equivalently a low reward (Figure 2.3D). After subtracting the expected reward, motifs that receive noise have a negative reward prediction error while motifs that escape the noise have a positive reward prediction error (Figure 2.3E). This dopaminergic error signal gates plasticity on the MSN units to create an adaptive bias in pitch that allows the model to sing at a pitch above the threshold and escape punishment.
2.2.9 Simulating covert learning

In one CAF experiment, Bengalese finches were able to learn changes to their song even when the LMAN→RA connection was blocked pharmacologically (Charlesworth et al., 2012). We simulated this type of learning by adding a source of variability in RA, which could be relayed to LMAN via the RA→DLM→LMAN pathway. We implemented the RA-generated variability using the same Gaussian noise process that was used to create the LMAN noise. Each RA channel has its own independent variability source \( Y_k^n(t) \), so RA output is now:

\[
R_k^n(t) = \left[ L_0 + L_k^n(t) + \sqrt{\alpha} Y_k^n(t) \right]_0
\]

The relative contributions of LMAN- and RA-generated noise to the total vocal output is controlled by the parameter \( \alpha \). Total LMAN output is the same as the simpler model, except with the noise process scaled by \( \sqrt{1 - \alpha} \):

\[
L_k^n(t) = \left[ L_0 + \sqrt{1 - \alpha} Z_k^n(t) + D_k^n(t) \right]_0
\]

RA-generated variability can reach LMAN through DLM, which carries the sum of inputs from Area X and RA:

\[
D_k^n(t) = -P_k^n(t) + \nu R_k^n(t - 1)
\]

The parameter values were chose as \( \alpha = 0.65 \) and \( \nu = 0.25 \) were chosen to match the experimental observation that 65% of the total variance remains when LMAN output to RA is blocked or LMAN is inactivated (Figure 2.7B) (Charlesworth et al., 2012; Warren et al., 2011).

2.2.10 Simulating the indirect pathway

We added the indirect pathway to our model. In the indirect pathway, a separate set of indirect MSN units project to a GPe-like pallidal unit in Area X. The GPe-like pallidal unit inhibits
the SNr-like pallidal unit that also receives input from the direct pathway MSN units (Figure 2.8A). The GPe-like pallidal unit $G$ aggregates the output of all the MSN units $I$ in its channel:

$$G_k^n(t) = -\sum_j I_j^n(t)$$

The SNr-like pallidal cell now receives an inhibitory projection from the GPe-like pallidal unit in addition to input from all the direct pathway MSN units:

$$P_k^n(t) = -\sum_j M_j^n(t) - G_k^n(t)$$

The indirect pathway MSNs are identical to the direct pathway MSNs described earlier except their learning rule is inverted. HVC synapses onto indirect pathway MSN units are strengthened in response to negative reward prediction errors:

$$w_{ji}^{n+1} = w_{ji}^n - \eta \sum_t b_{ji}^n(t)\delta^n(t)$$

This is the opposite of the learning rule for the direct pathway, and indirect MSN units have the opposite effect on LMAN. For example, a random LMAN fluctuation makes the song different from the template and causes a negative reward prediction error. The weight from the HVC unit that was active at the time of the fluctuation onto an indirect pathway MSN unit is eligible, and this weight is strengthened.

### 2.2.11 Learning sparseness

For simplicity, this model only contains the direct pathway and model MSNs receive only the LMAN noise directly instead of the output from LMAN which contains noise and bias. The eligibility trace is computed as:
We implemented all-to-all connections between MSNs and with random weights uniformly distributed between 0 and 0.005. All weights followed the learning rule described earlier – eligible synapses are strengthened by a positive reinforcement signal and synapses become eligible when their HVC and LMAN inputs are active simultaneously. Each synapse on each MSN maintains its own independent eligibility trace that is computed with the HVC input at that synapse and the LMAN input that is shared across the entire MSN. All of the simulations in this section were performed with 100 HVC units and 300 MSN units in a single channel. Simulations were run for 10,000 iterations.

2.3 Results

2.3.1 LMAN fluctuations

Before simulating learning in our model, we wanted to construct the model so that it produces variability that matches the temporal patterns observed in the pitch variability of real birds. We measured the precise temporal structure of pitch variability from singing birds and generated random fluctuations with matching power spectra for our model LMAN units. We computed the fluctuations in pitch from the mean during harmonic stacks in the songs of nine birds. Each bird had slightly different temporal dynamics in pitch fluctuations. The diversity in the speed of pitch fluctuations can be seen in individual examples (Figure 2.4A), the average autocorrelation (Figure 2.4B) and the average power spectra (Figure 2.4C) from each bird.
The random component of LMAN activity in our model units matches the dynamics of pitch fluctuations of the middle example in Figure 2.4A. For each LMAN unit, \( N \) Gaussian white noise segments, each \( T \) samples long, are filtered to match the power spectrum of pitch fluctuations from the harmonic stack shown in Figure 2.4A (center). The spectrum of the LMAN noise matches the power spectrum of the bird (Figure 2.4D).

This procedure is repeated to generate a separate noise source for each LMAN unit. For each LMAN unit, the generated noise is added to a baseline activity level and thresholded at zero to calculate the output of that LMAN unit. The output of the pitch-up and pitch-down LMAN units are combined downstream to control a single variable – pitch (Figure 2.4E and 2.1G).

### 2.3.2 Model Performance

Here we demonstrate that our model can learn a simple song. We choose our template song to be one cycle of a sine wave with period 200 ms. A network of 50 HVC units, 2 LMAN units control a single variable representing the pitch of the vocal output. The learning of the model is measured as bias, the output of the AFP without the random component. The bias is initially zero at all times in the song, but after 10,000 trials of learning the bias closely matches the template with a different MSN unit driving the song at each time (Figure 2.5A-B).

Next, we characterized the performance of the model under more challenging conditions. First, we expanded the number of degrees of freedom that the model had to learn to control. In our learning simulation from the previous paragraph, the model only controlled a single degree of freedom representing pitch. The number of degrees of freedom in the song is not known. There are seven muscles on each side of the syrinx, so we evaluated how our learning algorithm scales with increasing degrees of freedom. With increasing degrees of freedom, the difference between the vocal output and the template takes longer to reach the same steady state value (Figure 2.5C).
We measured the learning rate as the time constant of an exponential fit to the first 500 trials of learning (Figure 2.5D). Even with 32 degrees of freedom, the model requires less than 3500 trials to learn the song. Real birds sing over 50,000 motifs over the course of learning their song (Johnson et al., 2002).

We also explored the effect of the temporal characteristics of the reward signal. During conditional auditory feedback, VTA neurons have a phasic response lasting about 100 ms following a 50 ms burst of noise (Gadagkar et al., 2016). The response properties of VTA neurons have not been measured during natural song learning, so we wanted to evaluate the performance of the model over a wide range of possibilities. We simulated learning with the standard deviation of the Gaussian reward kernel, $\sigma_B$, ranging from 1 to 64 ms to understand the effect of this parameter on performance of the model (Figure 2.5E). Learning is slower with wider reward kernels with learning rate scaling linearly with the standard deviation of the reward kernel (Figure 2.5F).

2.3.3 Temporal Resolution of Learning

In this section, we test if the model can reproduce recent behavioral results from conditional auditory feedback experiments. In one study, Charlesworth et al. analyzed the temporal structure of the learned change in pitch after CAF. Even though the noise is delivered contingent on pitch at a single instant in the song, the induced pitch change is spread over 13.9 ms centered on the target time (Charlesworth et al., 2011). This temporal extent in learning is can be predicted from the average of the trials that avoid triggering the noise burst. We wondered if our model could explain this unique feature of CAF learning.
We simulated the CAF protocol by triggering a 2 ms-long noise burst when the pitch at 100 ms in the song was below a threshold (Figure 2.6A). The threshold was chosen so that about 70% of renditions were punished before learning. Punished renditions had a negative reward prediction error while unpunished renditions had a positive RPE (Figure 2.6B). The model rapidly learns to avoid the noise, and the learning is spread around the tens of milliseconds near the target time (Figure 2.6C).

After learning, we measured the temporal structure of learning to see if it matched the predictions of Charlesworth et al. Like Charlesworth et al., we define learning as the bias at the end of learning, normalized to a maximum of one. We found that the learning is more precise than the reward signal, but less precise than a single HVC burst. As predicted, it closely matches the average of the rewarded trials (Figure 2.6D).

We changed the speed of LMAN fluctuations to demonstrate that they are causally linked to the width of learning. Compressing the spectrum makes the fluctuations slower and changes the shape of the trials that escape punishment. Slower LMAN fluctuations induce learning that is more spread out around the trigger time (Figure 2.6E-F).

The temporal structure of the learning falls into two different regimes depending on the relationship between the timescale of pitch fluctuations and the timescale of the reward signal. When the width of the pitch fluctuations is narrower than the width of reward, the width of learning is highly correlated with the width of LMAN fluctuations. However, when LMAN fluctuations are wider than the width of the reward, the learning is limited by the width of reward (Figure 2.6G). All widths are measured as the full width at half of the maximum. The same pattern holds when the width of the reward kernel is adjusted while the timescale of LMAN fluctuations remains the same. We simulated CAF with normal LMAN fluctuations and reward kernels ranging from
widths 0.6 to 35 ms, corresponding to values of $\sigma_B$ from 0.25 to 15 ms. When the width of reward is smaller than the width of LMAN fluctuations, the reward signal constrains the temporal resolution of learning (Figure 2.6H). In our model, the timescale of learning is only determined by the exact reinforcement history if the reward signal is less precise than the LMAN fluctuations.

### 2.3.4 Covert learning

We also simulated the results of a second study in which learning can occur even when the connection from LMAN to RA is blocked pharmacologically. In Bengalese finches, lesions or pharmacological inactivations of LMAN spare about two-thirds of vocal variability which is thought to be generated elsewhere (Charlesworth et al., 2012). Surprisingly, Charlesworth et al. found that extra-LMAN variability can be used to learn, but the learning can only be expressed through LMAN. LMAN variability can be removed from the song while preserving activity in LMAN by blocking NMDA receptors in the downstream motor nucleus RA. Synaptic transmission from LMAN to RA is almost entirely through NMDA receptors but the glutamatergic input from HVC to RA is mostly non-NMDA (Mooney, 1992).

When LMAN input to RA is blocked with AP-5, no learning occurs during conditional auditory feedback. However, as soon as the AP-5 is washed out, an adaptive pitch change appears immediately, as if it was learned while the block was in place but could not be expressed. If activity in LMAN is blocked completely with GABA receptor agonist muscimol no learning appears even after the drug is washed out. Charlesworth et al. hypothesized that LMAN was receiving a copy of variability generated elsewhere and transmitting that variability to Area X. The authors hypothesized that variability generated in the motor system could be relayed to Area X from RA through DLM and LMAN (red path in Figure 2.7A). Area X learns to bias LMAN activity, but this bias can only affect vocal output through LMAN.
To model this experiment, we added a noise source in RA to produce the vocal variability that remains after LMAN inactivation or AP-5 infusion into RA. We also added the connection between RA and DLM with strength v (Figure 2.7A). Vocal variability generated in RA can reach Area X through DLM and LMAN. Crucially, we hypothesize that the RA-DLM is tightly topographic and forms a closed topographic loop between LMAN, RA, and DLM. Then, variability in one channel in RA can be relayed to the matching channel in Area X.

This model learns to shift its vocal output to avoid loud noise bursts. A single run of the model learning under conditional auditory feedback is shown in Figure 2.7C. Beginning at the 200th motif, a large error signal is delivered when the pitch at 100ms in the song is below zero. The pitch at 100ms rapidly shifts upward to avoid this error (black line). After the 1000th motif, the synaptic weights are held constant during a washout period to measure the vocal output after learning.

To reproduce covert learning, we simulated CAF with the connection from LMAN to RA removed during learning. When the LMAN→RA connection is blocked, variability generated in RA travels to DLM to LMAN, so LMAN activity is correlated with the remaining variability in vocal output. Even though the pitch at the target time does not change while the LMAN-RA block is in effect, a bias gradually accumulates in Area X. This bias is computed as what the vocal output would be if the LMAN-RA connection was intact and there was no noise in LMAN or RA. When the LMAN-RA connection is restored during the washout period (motifs 1000-1200), the bias is immediately apparent in the vocal output (Figure 2.7D). When all LMAN output is blocked and RA variability cannot be relayed to Area X, no bias is learned and the vocal output remains at its pre-learning levels even during the washout period (Figure 2.7E). These results are summarized in Figure 2.7F.
2.3.5 Indirect Pathway Reduces Variability

Now we turn to more speculative extensions of the model. So far, our simulations of Area X have only included the direct pathway, where signals from MSN→pallidal neuron→DLM can bias the song towards variants that lead to increased reinforcement. In this section, we add a simulation of the indirect pathway through Area X. We demonstrate that the indirect pathway can play a complimentary role of biasing the song away from variants that lead to lower reinforcement. Increased drive through the indirect pathway decreases the excitatory drive from DLM to LMAN. We hypothesized that the indirect pathway would allow the AFP to reduce variability in the song. Learning to increase activity of the indirect pathway in both the pitch up and pitch down channels would lead to less LMAN activity and less variability.

To demonstrate that the indirect pathway can reduce variability, we simulated a modified version of conditional auditory feedback. Pitches above 1 or below -1 were punished while a middle range near the mean escaped punishment (Figure 2.8B). After learning, the variations in the song are reduced at the target time (Figure 2.8C and 2.8E), so the song no longer triggers noise bursts (Figure 2.8D). The indirect pathway is active at this time to cause the reduction in variability (Figure 2.8G) while activity in the direct pathway remains low (Figure 2.8F).

2.3.6 Learning Sparseness

In a second extension of the model, we simulate more realistic connections between HVC neurons and MSNs in Area X. Up to this point, we have assumed a one-to-one wiring between HVC and MSN units. This simplification allowed us to accurately reproduce the sparse firing patterns of MSNs but it is not an accurate representation of the known anatomy of corticostriatal afferents. MSNs receive thousands of excitatory synapses onto their dendritic spines, which likely
come from many different cortical neurons (Kincaid et al., 1998). Therefore, we desired to extend our model by beginning with all-to-all connectivity between HVC and MSNs (Figure 2.9A). Using a three-part learning rule similar to the previous sections (see Methods), the model learns an accurate imitation of the template (Figure 2.9B). However, the activity of the MSN units no longer matches the sparse activity observed experimentally. Instead, each MSN is active throughout the song. The weight matrix at the end of learning shows that each MSN has a similar weight profile, mirroring the amplitude of the song (Figure 2.9C). This is very different from the sparse firing patterns of real MSNs (Figure 2.2A).

We wondered if we could modify the learning rule such that MSNs would develop their characteristic sparse activity patterns during the learning process. To be active at just one time in the motif, MSNs should have a strong connection from only one HVC neuron. To restrict the number of strong HVC synapses onto each MSN, we added a reward-independent depression rule that weakens an HVC→MSN synapse if its HVC input is not active when the MSN is active:

\[ \Delta w_{ji} = \zeta \sum_t \Theta(H_i(t))\Theta(M^n_j(t)) \]

where \( \Theta \) is the step function that is one when its argument is greater than zero and zero otherwise. \( \zeta \) is a parameter to control the learning rate. Following this rule, once one HVC→MSN synapse is strong enough to drive the MSN unit, other synapses onto that neuron are suppressed.

This process is happening constantly and in parallel with the reward-dependent plasticity that guides the vocal output towards the template. Once one input becomes strong enough to drive the MSN unit above threshold, that activity weakens all the other inputs to that MSN. Different MSNs have different inputs reach threshold first due to differences in the random initial weights. With this depotentiation rule, the model learns to imitate the template (Figure 2.9D) and individual
MSNs have strong weights at only a single time in the song (Figure 2.9E). However, the distribution of MSNs depends strongly on the initial conditions, which can lead to gaps in coverage, like at t=70ms in Figure 2.9D-E. The number of MSNs active at each time step varies from 0 to 18 with no MSNs active at t=70ms (Figure 2.9H).

To correct these gaps in coverage, we added lateral inhibition between MSNs to build a uniform distribution of MSNs throughout the song. The eligibility trace now will depend on postsynaptic depolarization $V$ instead of LMAN input directly:

$$b_{ji}^n(t) = \sum_{\tau=0}^{8\sigma_B} V_{ji}^n(t - \tau) H_i^n(t - \tau) B(\tau)$$

$$V_j^n(t) = \left[ Z_i^n(t) + \sum_i w_{ji} H_i(t) - \sum_{j' \neq j} u_{jj'} \Theta \left( M_{j'}(t) \right) \right]_0$$

Note that the post synaptic depolarization is shared among all synapses in the MSN and is a combination of noise from LMAN, the sum of the inputs from HVC, and inhibition from other MSNs. Now, if many MSNs are active at time $t$, the postsynaptic depolarization $V_j$ in other MSNs will be forced to zero and these MSNs will not be able to learn from rewards at this time. Note that this inhibition is only included in the learning rule. The output of MSN units is still a function of HVC input alone.

Now, with lateral inhibition in addition to the depression rule, the sparse MSNs distribute themselves more evenly. The model learns to match the template, as before (Figure 2.9F), and MSNs have strong weights from HVC at only a single time in the song (Figure 2.9G). There are 0
2.4 Discussion

In this chapter, I developed a computational model of song learning based on a conceptual model by (Fee and Goldberg, 2011). I demonstrated that this model can account for a number of surprising behavioral results from the literature, including covert learning (Charlesworth et al., 2012). I also developed two speculative extensions to the model. First, by adding an indirect pathway to the modeled circuitry in Area X, I gave the model a way to reduce variability in the song. Second, I demonstrated how a modification to the learning rule allows MSNs to develop and maintain their sparse firing patterns.

Further extensions to the model could make it more biologically plausible. First, the model could be rebuilt using spiking neurons instead of firing rates. Another improvement for the model could come from implementing a more realistic mapping from brain signals to vocal output. Currently, there is just a linear transformation between them, but in reality the vocal organ is highly nonlinear (Fee et al., 1998). These dynamics are captured by existing biomechanical models of the vocal organ, which can faithfully reproduce the acoustic structure of zebra finch song using only a few parameters (Amador et al., 2013; Perl et al., 2011). Future work could train my model to control these biomechanical parameters.

2.4.1 Features of Area X that are not included in the model

The computational model here presents a highly simplified view of Area X. The model only includes three inputs to Area X: HVC, LMAN, and VTA. Within Area X, the circuitry in the model is very streamlined. For most of the simulations, Area X contains only two cell types which
form the classical “direct pathway” through the basal ganglia (Albin et al., 1989): MSNs that receive inputs from LMAN and HVC, and pallidal neurons that receive input from MSNs and project out of Area X to the thalamus. In Section 2.3.5, I also include one additional pallidal cell type that is part of the classical “indirect” pathway through the basal ganglia. This streamlined architecture omits several anatomical features of Area X. Although this model was sufficient to reproduce several features of vocal learning, future models may incorporate the additional components to provide a more complete account of the function of Area X.

Future extensions to the model could incorporate the full diversity of cell types within Area X. Area X contains at least seven different cell types, corresponding to cell types observed in the mammal striatum, pallidum, and potentially the subthalamic nucleus. The striatal cell types are the medium spiny neuron, which is included in the model, as well as tonically active, low threshold spiking, and fast-spiking interneurons, which are not included. The tonically active neurons are thought to be cholinergic (Farries and Perkel, 2002; Goldberg and Fee, 2010), and recordings of these neurons in mammals implicate these neurons in processing rewards or other salient stimuli. Both types of pallidal neurons are already included in the model. Finally, Area X also contains a small number of glutamatergic neurons that synapse onto pallidal neurons (Budzillo et al., 2017). These neurons may be analogous to the glutamatergic neurons found in the subthalamic nucleus of mammals, but their function in Area X is unknown.

The model also omits two additional inputs to Area X. One of these additional inputs comes from paraHVC, a region in the nidopallium that is adjacent to the medial edge of (Foster and Bottjer, 1998). The role of paraHVC in song production and song learning is unknown. Another input to Area X comes from a region in the thalamus that in turn receives input from the deep cerebellar nuclei (Pidoux et al., 2018). This projection is interesting because it provides an
opportunity for the cerebellum to contribute to vocal learning, perhaps in a similar way that it contributes to motor learning in mammals (Thach et al., 1992). The contributions of the cerebellum to vocal learning in songbirds is an emerging area of research, and the model may be expanded to include a role for the cerebellum as future experiments clarify its role.

2.4.2 Additional phases of song learning that are not included in the model

A second avenue for improvement is to capture the full vocal learning process into a single unified model. The model in this chapter focuses only on the learning of bias in the AFP, but does not include a mechanism for transferring this bias to the HVC→RA synapses in the motor pathway. A complimentary model from Tesileanu et al. (2017) focuses on this second stage of learning. Their model incorporates a bias learning stage that is similar to the algorithm implemented in this chapter.

Recent research has also modeled an earlier phase in song learning in which the timing signals develop in HVC. In adulthood, the sequence of HVC bursts is stable (Katlowitz et al., 2018; cf. Liberti et al., 2016), but in juveniles it quite dynamic. A recent study by Okubo et al. (2015) found that HVC neuron activity is initially shared among newly developing syllables, but each syllable gradually develops a distinct representation in HVC. In current models of HVC development, HVC chains can form on their own, without the involvement of the AFP (Fiete et al., 2010). The chains may split when driven by alternating signals from one of the other afferents to HVC, without the involvement of Area X (Mackevicius, 2018; Okubo et al., 2015).

The timing signals in HVC can also change in adult birds. Ali et al. (2013) developed a paradigm to train birds to either lengthen or shorten individual syllables by playing bursts of white noise contingent on the duration of these syllables. They found that the temporal changes in song correlate with a change in the temporal structure of neural activity in HVC. Surprisingly, these
changes can be learned even after Area X has been lesioned. The model of learning in this chapter does not include a mechanism for learning these changes to song timing.

What is the AFP doing when the HVC sequence does not cover the full motif? There are spectral changes to the song occurring at this time (Okubo et al., 2015; Tchernichovski et al., 2001), which may be driven by Area X. What error signal is driving these changes? It is not clear how vocalizations comprising one stereotyped protosyllable interspersed with other variable syllables would be compared against a template that included an entire stereotyped motif. One possibility is that the protosyllable is being compared against one particular syllable in the template. Recordings from VTA in juvenile birds may answer this question.

2.4.3 Other functions of the basal ganglia

The computational model presented in this chapter focuses on the role of the basal ganglia in learning. The model does not include a role for Area X in the performance of a fully learned adult song. Instead, the adult song is presumed to be stored entirely in the cortical areas of the motor pathway, HVC and RA. This is inspired by studies which found that lesions of Area X in adulthood does not affect the performance of the song (Sohrabji et al., 1990). However, these results are at odds with some studies of learning in mammals where the basal ganglia is required for the performance of a learned skill (Atallah et al., 2007). Some learned motor tasks can be performed even after the motor cortex is lesioned (Kawai et al., 2015). These results suggest that some motor skills can be executed by subcortical regions of the brain, possibly including the basal ganglia. Generalizing my computational model to learning tasks in mammals will require a greater understanding of the contributions of the basal ganglia to motor performance.
In addition to learning, the basal ganglia have been implicated controlling the details of movements, including the response vigor, speed, and kinematics (Dudman and Krakauer, 2016; Turner and Desmurget, 2010). Even in songbirds, some evidence suggests that Area X contributes to making the song more stereotyped when a male is performing for a female instead of practicing alone (Leblois et al., 2010). These aspects of basal ganglia function could be included in a future extension of the model.

2.4.4 Biological tests of the model

The model makes a number of specific predictions about the roles for Area X and its afferents in the learning process. Future experiments could test these predictions through experiments in learning birds. In this section, I propose two specific experiments for testing the assumptions and predictions of the model.

2.4.4.1 Test 1: Area X is required for the expression of bias

The most fundamental prediction of the model is that Area X is necessary for the expression of bias in the anterior forebrain pathway. The temporal precision of the bias signal strongly suggests that it depends on timing signals from HVC. After learning, the model predicts that bias is driven by HVC driving a time-varying bias in LMAN through Area X. No experiments have differentiated the role of Area X in learning versus expression of bias. Lesions of Area X prevent song learning, both during natural development (Scharff and Nottebohm, 1991; Sohrabji et al., 1990) and during conditional auditory feedback (Ali et al., 2013). However, these studies cannot exclude the possibility that Area X is only required for the initial learning of bias and not for its expression.

In an alternative model, HVC may drive bias through an alternate route by sending timing information to RA to DLM to LMAN. DLM neurons receive time-varying excitatory drive from
RA during singing (Goldberg and Fee, 2012). Setting the appropriate synaptic weights between RA and DLM could drive bias in LMAN without input from Area X. During learning, Area X may program these RA→DLM synaptic weights so that they drive the appropriate bias. Under this alternative hypothesis, once the bias is fully programmed into the RA→DLM synapses, Area X is no longer required for the expression of bias.

To test specifically for the role of Area X in the bias expression, Area X must be inhibited or removed from the circuit when there is bias in the AFP. First, develop bias in the AFP by shifting the pitch of a target syllable using conditional auditory feedback. Immediately after learning the pitch shift, remove Area X using pharmacology, lesion, or other method. Finally, measure the bias at the target without Area X.

A wide variety of manipulations may be effective in removing the influence of Area X, but care must be taken to not disrupt normal functioning in LMAN. Pharmacological agents like tetrodotoxin and muscimol can inhibit activity in Area X, including pallidal neurons that project to DLM. These inhibitory pallidal neurons have a very high firing rate during singing (Goldberg et al., 2010) and have a powerful influence on the spike times of DLM neurons (Goldberg et al., 2012). Like any acute manipulation, transiently removing this inhibition may have acute effects that go beyond just removing bias (Otchy et al., 2015). The pallidal activity may remain intact by using glutamate receptor blockers like CNQX to block only the excitatory drive from HVC and LMAN. However, I tried performing this experiment by infusing CNQX into Area X and this caused the birds to stop singing. If pharmacological methods fail, a lesion of Area X may be sufficient, but would not be reversible.
2.4.4.2 Test 2: HVC-X neurons drive bias

If Area X is required for the expression of bias, a further test is necessary to demonstrate the bias is driven by its input from HVC. To test this, we will stimulate HVC$_X$ neurons to demonstrate that they are sufficient to drive bias. First, we will induce learning using conditional auditory feedback at a specific time in the song, T1. At this point, the bias is hypothesized to be driven by the HVC$_X$ neurons active at time T1. Therefore, reactivating these neurons at a different time in the song, T2, should cause the bias learned at T1 to be expressed at T2.

The biggest challenge in this experiment is reactivating the HVC neurons active at one time in the song. Rather than try to somehow capture the HVC neurons that are naturally active at T1, a more feasible strategy is to activate a small random subset of HVC neurons. This subset of HVC neurons could be selected by injecting a small amount of virus carrying the gene for channelrhodopsin (ChR2). The virus will cause ChR2 expression in a random set of HVC neurons – many of these neurons will not be naturally active at the learning time T1. We will stimulate them with light on every rendition of the song, triggered on the acoustic features of the song at that time. We will give the light stimulus 50 ms before the CAF-targeted time in the song because electrical stimulation in HVC evokes changes in the song approximately 50 ms after the time of the stimulus (Vu et al., 1994). We will restrict stimulation to only HVC$_X$ neurons by shining light in Area X to stimulate the axon terminals of HVC$_X$ neurons. This technique has been used previously to selectively stimulate individual projections between brain areas (Cruikshank et al., 2010; Gradinaru et al., 2007, 2009; Lee et al., 2010; Zhang and Oertner, 2007).

Initially, stimulating the HVC$_X$ neurons should not have any effect on song. Next, we will induce learning at time T1 while also stimulating at time T1. After CAF, the pitch of the song at
T1 will be measurably higher. At this point, we hypothesize that the HVC\textsubscript{x} neurons that are active at T1 (including the neurons we activated with light) are driving this change in LMAN.

After inducing a pitch change at T1 with CAF, we will stop CAF and stimulate at test time T2 instead of the learning time T1. We hypothesize that the light-activated HVC\textsubscript{x} neurons are driving the bias, so stimulating at T2 should cause a change in the mean pitch at this time, even though conditional auditory feedback never induced learning at this time.
Figure 2.1: Anatomy and physiology of the song system. (A) Diagram of the brain areas controlling singing. (B) Activity of single units in HVC during singing. The activity of seven HVC projection neurons (bottom) is aligned to a spectrogram of the song (top). Each row represents one rendition of the song and each tick mark is one spike. Each projector bursts highly selectively at 1-3 times per rendition. (C) Inactivation of LMAN eliminates vocal variability. Left: Pitch (mean subtracted) from multiple renditions of a short segment of song from a bird with LMAN intact. Right: Mean-subtracted pitch from multiple renditions of the same syllable after LMAN activity is blocked with local injection of tetrodotoxin. (D) Activity of one single unit in LMAN during multiple renditions of the song. Each row represents one rendition and each tick mark represents one spike. Example spectrogram of the song is above. (E) Spectrogram of two renditions of a song during conditional auditory feedback. The first rendition of the target syllable escapes punishment, but the second rendition is hit with a burst of white noise. (F) Pitch of targeted syllable during conditional auditory feedback. Each point is the average pitch of the targeted syllable on one rendition of the song. The black line is the average pitch over the renditions. Over several hours, the pitch decreases to avoid the noise, but when activity in LMAN is blocked with tetrodotoxin (red), the average pitch returns to its initial state. (G) Model of bias generation. Two parallel channels through LMAN, Area X, DLM drive upward and downward pitch fluctuations. When the pitch is biased downwards, the pitch-down channel becomes more active than the pitch-up channel.
Figure 2.2: Model MSNs drive bias in vocal variability. (A) Spikes in six putative MSNs recorded from a juvenile songbird (61-65 dph) aligned to an example motif of the song. In addition to being selective for a single time in the song, putative MSNs are distributed sparsely through the song. (B) Model of temporally-precise bias generation. A strong synapse between HVC unit 2 and MSN unit 2 (red) drives bias at time 2. (C) Model output to generate bias. HVC unit 2 drives MSN unit 2 (red) which in turn drives activity in the pitch-up channel of LMAN. (D) Model for learning bias. HVC-MSN synapse on MSN 2 is eligible (red) because its HVC and LMAN inputs were active simultaneously. A delayed reinforcement signal from VTA strengthens the eligible synapse.
Figure 2.3: Reward prediction error. We hypothesize that VTA sends an error signal that depends on the similarity between the song and a memorized template. (A) Error is proportional to the square of the difference between template pitch and the pitch that the model sings. Here the template pitch is 800 Hz. In conditional auditory feedback, a short burst of white noise is played if the pitch is below a threshold. When pitch is below 800 Hz in this example, the noise burst causes a large error. Dotted line shows the error in the absence of CAF. (B) Example of two pitch traces produced by the model. The red trace is below 800 Hz at 50ms (arrow), so it is punished with a noise burst from 50-60 ms (red bar). The black trace is above 800 Hz at 50ms, so it escapes punishment. (C) Error over time for the song in (B) with conditional auditory feedback. The noise burst causes a large error at 50-60 ms that is much larger than the errors caused by natural fluctuations in pitch. (D) Reward over time for the two song renditions in (B). Reward is the opposite of error, convolved with a Gaussian kernel. The standard deviation of the kernel is 12.5 ms and the peak is four standard deviations (50 ms) delayed from the error. The noise burst causes a dip in reward (red). Black dotted line shows the average reward over 100 motifs. Since 50% of motifs receive noise, the expected reward is halfway between the hit and escape variants. Note the low reward in response to the noise burst (red). (E) Reward prediction error over time for the two motifs in (B). Reward prediction error is difference between actual reward and the average, or predicted, reward. On average, the RPE is zero at all times. Positive RPE means that the motif was more rewarding than the average (as in the motif in green that escaped the white noise feedback). Negative RPE means that the motif was less rewarding than average (as in the red motif that received a white noise burst as punishment).
Figure 2.4: Model LMAN fluctuations. (A) Example pitch fluctuations from three birds with fast (left), medium (center), and slow (right) pitch fluctuations. Top: Spectrograms of syllables with harmonic stacks. White boxes mark the segments of the syllables with a well-defined pitch that are analyzed further. Bottom: Three example pitch traces from the white-boxed segments above. (B) Average autocorrelation of pitch fluctuations from nine birds. Colors correspond to the examples in A. (C) Average power spectra of pitch fluctuations. Colors correspond to colors in B. (D) Power spectrum of pitch fluctuations from example bird in center of A (teal) and model pitch fluctuations (black). (E) Diagram of generating variability in the model. LMAN units in the pitch-up and pitch-down channel each generate variable activity with a spectrum in D. These activities are combined to make a variable pitch.
Figure 2.5: Model performance. (A) The model learns to sing a simple song. The song is 200 ms long with 50 units in HVC. The template is one period of a sine wave. Top: The template (black) and the bias (i.e. the song without LMAN variability, purple) after 5000 learning trials. Bottom: The activity in five example MSN units after 5000 learning trials. Only MSNs in the pitch-up channel are shown. (B) Mean squared error between bias and template over 5000 learning motifs. The error rapidly decreases as the model learns the template in (A). (C) Additional degrees of freedom reduce learning rate. Model learns the template in (A), but with increasing amounts of uninformative noise in the reward signal. In A-B, variability in a single dimension, pitch. In these simulations we added additional sources of uncorrelated variability to the reward signal to emulate simultaneous learning across many degrees of freedom. There were 2, 4, 8, 16, or 32 uncorrelated noise sources. Each curve is the average of ten simulations. Each noise source contributes an equal amount of variance to the reward signal, and total variance of the reward signal is normalized to be the same in all cases. (D) Time to learn as a function of the number of independent noise sources. The time to learn is the trial on which the mean squared error first drops below 3, shown as a dashed line in C. The time to learn increases as the square root of the number of independent noise sources. Error bars show standard error of the mean from ten simulations. (E) Mean squared error through learning with different reward signals. The standard deviation of the reward kernel was set to 1, 2, 4, 8, 16, 32, or 64 ms (compare to 12.5 ms for A-C). Increasing the standard deviation of the reward kernel makes the reward signal less temporally precise and more delayed from the original error. The eligibility trace was also convolved with the reward kernel, so the eligibility trace and reward remained temporally matched in all simulations. (F) Time to learn as a function of the standard deviation of the reward kernel. Time to learn is the first motif on which the mean squared error is less than 3 (dashed line in panel E). As the reward becomes wider and less precise, the time to learn increases. Error bars show standard error of the mean from ten simulations.
Figure 2.6: Temporal resolution of learning. (A) Example model output during conditional auditory feedback. If the output at the target time (time step 100, yellow arrow) is below a threshold, it is punished with a high error signal (red traces). If the output is above the threshold, it escapes punishment (black traces). (B) Reward prediction error signals during CAF. Each trace corresponds to one rendition of the song in panel A. Red traces: negative reward prediction error signals during punished trials. Black traces: positive reward prediction error during trials that escape punishment. (C) The learned bias after CAF. The bias is temporally precise, with changes spanning 14 ms at half maximum and centered on the target time. (D) Learned bias compared to other relevant signals in learning. Learning (blue) is similar to the average of the trials that escaped punishment (black). The timescale of learning is broader than a single HVC burst (cyan), but narrower than the reward prediction error resulting from the noise burst (green). (E-F) Same as B-C, but in a model with slower LMAN-driven variability. In this regime, the timescale of learning is most similar to the timescale of the reward prediction error signal, but narrower than the average of the trials that escaped punishment. (G) Width of learning versus width of the autocorrelation of LMAN-driven vocal variability. Each point represents one simulation with a different timescale for LMAN fluctuations. For smaller LMAN widths, the learning width is near the identity line. For wider LMAN widths, the learning width is near the width of the reward signal (horizontal line). (H) Width of learning versus the width of the reward signal. Each point represents one simulation with a different time course for the reward signal. The width of the LMAN fluctuations was held constant (horizontal line). For smaller reward widths, the width of learning is near the identity line.
Figure 2.7: Covert learning (A) Diagram of model for covert learning. This model contains a separate source of vocal variability in RA. The relative magnitudes of the vocal variability generated within LMAN and within RA is controlled by the parameter $\alpha$. The model also contains a connection from RA to DLM with a strength controlled by the parameter $\nu$. The LMAN→RA connection (dotted line) is removed during the covert learning phase of these simulations. (B) Standard deviation of the vocal output, normalized to 1 for the normal case. Each bar is the mean of 10 simulations, with error bars representing the standard error of the mean. (C-E) Vocal output and AFP bias at the target time for example simulations. Green trace is the AFP bias at the target time on each motif. Black trace is the vocal output at the target time on each motif. Gray block marks the motifs that are subject to conditional auditory feedback. Red bar above the traces represents the motifs where the circuit is manipulated by removing the LMAN→RA connection or setting LMAN output to zero. For the final 200 motifs, the model runs fully intact. (F) Summary of vocal output at the end of CAF and during washout for each condition. Each bar is the average of 10 simulations, with error bars representing the standard error of the mean.
Figure 2.8: Model of variability reduction using the indirect pathway. (A) Diagram of model including direct and indirect pathways. (B) Vocal output of 25 example motifs before CAF for variability reduction. Output that is between the two thresholds (yellow arrows) escapes punishment (black traces). Vocal output that is above the upper threshold or below the lower threshold is punished with a simulated noise burst (red traces). (C) Vocal output of 25 example motifs after learning. (D) Fraction of motifs hit with punishment over the course of learning. (E) Histogram of vocal output at the target time before learning (gray) and after learning (black). Red lines mark the thresholds for conditional auditory feedback. (F) Sum of all of the activity in the direct pathway MSNs during the last motif after learning. (G) Sum of all activity in the indirect pathway MSNs during the last motif after learning.
Figure 2.9: Learning with all-to-all connectivity between HVC and MSNs. (A) Diagram of model with all-to-all connectivity between HVC and MSNs (red lines). (B) Song learning in model with all-to-all connectivity using. Dotted line is the bias at the end of learning and the gray line is the template. (C) Weight matrix for HVC→MSN weights at the end of learning for the simulation show in panel B. All MSNs have similar weights that are spread over a wide range of HVC inputs. (D) Same as panel B, but for a model which heterosynaptic competition between HVC synapses on each MSN. (E) Weight matrix for model including heterosynaptic competition. (F) Learned bias for a model which includes heterosynaptic competition and lateral inhibition between MSNs. (G) Weight matrix for a model which includes heterosynaptic competition and lateral inhibition between MSNs. (H) Average number of MSNs active at each time in the song at the end of learning. Blue trace corresponds to a model with heterosynaptic competition and lateral inhibition (same as panels F-G). Green trace corresponds to a model with heterosynaptic competition only (same as panels D-E). Each trace is the average of ten simulations. (I) Standard deviation of the density of MSN activity from panel G. Error bar is the standard error of the mean over ten simulations.
3 Activity of neurons in AId during vocal errors and natural movements

The previous chapter demonstrated the capabilities of a neural network model that learns from a dopaminergic reward prediction error signal. This error signal has been observed in VTA neurons that project to Area X (Gadagkar et al., 2016) in the form of a decrease in activity following a burst of white noise played during singing and an increase in activity when this noise burst is omitted. The origins of this signal are still a topic of active research.

Research into the source of vocal error signals in VTA on its inputs from the telencephalon. The only telencephalic input to VTA comes from the intermediate arcopallium, adjacent to RA (Gale et al., 2008). The intermediate arcopallium is composed of two subdivisions: ventral (AIV) and dorsal (AId). AIV is defined as the part of intermediate arcopallium that projects to VTA and it has a vital role in normal song learning. Lesions of AIV in juvenile birds prevent them from learning an accurate imitation of their tutor songs (Mandelblat-Cerf et al., 2014). Anatomically, AIV is well positioned to process vocal error signals because it receives input from auditory pathways (Kelley and Nottebohm, 1979; Mello et al., 1998; Vates et al., 1996). Some of these auditory areas have a phasic increase in their activity immediately following a noise burst during singing (Keller and Hahnloser, 2009). Similarly, neurons in AIV that project to VTA also have a burst of activity after a noise burst is delivered during singing (Mandelblat-Cerf et al., 2014). AIV could be responsible for the decrease in activity of VTA neurons following a noise burst if the AIV neurons inhibited the neurons in VTA. However, AIV activity alone probably cannot account for the increase in VTA activity observed when a noise burst is omitted, since the AIV neurons have a
low baseline rate and cannot decrease their rate below baseline when a noise burst is omitted. Nonetheless, the signal in AIv is likely important for computing the teaching signal in VTA.

The role of AId in behavior is controversial, and a major focus of my thesis is to help resolve this controversy. Separate lines of research from different labs have focused on two different hypotheses for the role of AId. One hypothesis is that AId contributes to vocal learning by processing vocal errors, perhaps similar to AIv. The second hypothesis is that AId is not related to singing or vocalization at all, but is instead related to controlling head and body movements. This chapter will directly test these two hypotheses by recording single neurons in AId. Before explaining these two hypotheses in more detail, I will first review the anatomy of AId.

3.1 AId is part of a circuit that parallels the song system

AId is part of a network of brain areas that parallel the song system in their anatomical positions and connectivity. This anatomy is depicted schematically in Figures 3.1B–3.1C. In the arcopallium, AId is immediately lateral to RA (Bottjer et al., 1989). These areas both send descending projections out of the telencephalon. They also project to adjacent regions in the thalamus (Bottjer et al., 2000), and these portions of the thalamus project to neighboring regions of nidopallium. In the song system, DLM projects to LMAN, while in the parallel network the dorsal thalamic zone (DTZ) projects to LMANshell, a region of parvocellular cell bodies surrounding LMAN (Johnson and Bottjer, 1992). LMANshell completes the loop by projecting back to AId directly and indirectly via dNCL (dorsal part of the caudolateral nidopallium) (Bottjer et al., 1989, 2000).

All of the connections through this cortico-thalamic AId→DTZ→LMANshell→dNCL→AId loop are topographic (Bottjer et al., 2000; Iyengar et al., 1999; Johnson et al., 1995). In younger birds this topography is less refined, with axons of thalamic neurons projecting to both
LMAN and LMANshell (Iyengar and Bottjer, 2002; Iyengar et al., 1999), and axons of LMAN neurons projecting to both RA and AId (Miller-Sims and Bottjer, 2012).

These anatomical similarities imply that the parallel pathway has a similar function to the song system. There are two competing hypotheses for this shared function. One is that both the song system and the parallel pathway are involved in vocal learning. The song system handles the motor aspects of song learning while the parallel pathway may handle the auditory processing (Bottjer and Altenau, 2010). An alternative hypothesis is that both networks are involved in motor control. The song system is responsible for voluntary control of the syringeal and respiratory muscles, while the parallel pathway may be responsible for control of other muscles including the head, wings, and legs (Feenders et al., 2008). Each of these hypotheses has some experimental support, as detailed in the next section.

The parallel circuit does have several unique features that differentiate it from the song system. Unlike the song system, where the basal ganglia nucleus Area X only receives cortical input from the nidopallium (LMAN and HVC), the medial striatum in the parallel circuit receives input from the arcopallium, AId (Iyengar et al., 1999). This is just one example of the differences in the basal ganglia circuitry of the parallel pathway and the song system (Reiner et al., 2004).

3.1.1 Hypothesis 1: AId is part of a vocal learning circuit

Physiological evidence supports the hypothesis that the parallel pathway is processing auditory information. In dNCL, immediate early gene expression is high after singing as well as after hearing playback of a tutor or conspecific song, compared to birds who did not sing or hear song playback (Bottjer et al., 2010). In LMANshell, recordings under anesthesia found neurons that respond to playback of the bird’s own song or the tutor song (Achiro and Bottjer, 2013). Furthermore, during singing, about 10% of units in LMANshell either increase or decrease their
activity during the production of syllables that are more similar to syllables in the tutor song (Achiro et al., 2017). However, unlike AIv, AId has no known connections to the ascending auditory system. Although some have speculated that there may be a connection between dNCL and high level auditory areas such as caudomedial nidopallium and caudal mesopallium (Bottjer et al., 2010), anatomical evidence of such a connection is still missing.

Some lesion studies support a role for AId in vocal learning. In one study, lesions of AId in juveniles were found to impair imitation (Bottjer and Altenau, 2010). However, a subsequent study failed to find any song learning deficit after lesions of AId (Mandelblat-Cerf et al., 2014). Instead, Mandelblat-Cerf et al. attribute the learning deficit to a lesion of AIv. The proximity of AIv and AId makes it difficult to know if the deficits caused by lesions to intermediate arcopallium is due to the loss of a signal carried by inputs from dNCL and LMANshell, which project to AId, or due to the loss of a signal from auditory regions, which project to AIv. Recordings from AId during singing could determine what signals these neurons are carrying.

3.1.2 Hypothesis 2: AId is part of a motor control circuit

Anatomically, the descending outputs of AId support a potential role as a premotor region. Aside from a possible output to VTA, its other midbrain targets are motor-related. One of these targets is the reticular formation, which controls head and beak movements in other bird species (Dubbeldam, 1998; Dubbeldam and Den Boer-Visser, 1994). AId also targets the deep layers of the tectum (Bottjer et al., 2000), which is involved in orienting behaviors in birds and mammals (Gandhi and Katnani, 2011). Consistent with its downstream connectivity, large lesions of AId including the most lateral parts cause akinesia (Mandelblat-Cerf et al., 2014).

Gene expression patterns also suggest that AId is involved in motor control. The gene expression patterns in AId are most similar to layer V motor cortex in mammals (Pfenning et al., 81
Immediate early gene expression patterns also support a motor role for the parallel circuit in zebra finches. After hopping on a treadmill, there is high immediate early gene expression in AId, MST, and LMANshell, among other regions. By contrast, after singing there is low immediate early gene expression in the parallel network and high expression in the nuclei of the song system (Feenders et al., 2008).

Both of these hypotheses are missing crucial neurophysiological evidence. What is the activity of single neurons in AId during behavior? In this chapter, I perform the first recordings of AId in an awake bird in order to determine if they are encoding information about vocal errors or body movements.

3.2 Methods

I recorded from single neurons in using a chronically implanted, motorized microdrive that I implanted into AId. During surgery, I located AId by stereotaxic coordinates as well as by its response to electrical stimulation of the upstream brain area LMANshell. After the end of the experiment, I confirmed the location AId histologically using anterograde tracer injected into LMANshell during the implant surgery.

After recovering from surgery, I recorded from single units in AId which I identified by their response to electrical stimulation in LMANshell. This setup is depicted schematically in Figure 3.1D. Neurons were localized in AId by their orthodromic response to electrical stimulation in upstream brain area LMANshell (Figure 3.1E). At the end of the experiment, recording locations were verified histologically by making a small electrolytic lesion through one of the recording electrodes at its final position and reconstructing the recording sites in serial sections (Figure 3.1F). For each neuron, I assessed its responses to vocal errors as well as natural head and body movements. To evaluate the response to vocal errors, I recorded the neurons while the bird sang
with noise bursts played at a targeted time in the song. To evaluate the activity of AIid neurons during movements, I also recorded video of the bird while he moved around his cage.

The subjects in this experiment were three male zebra finches aged 36 to 134 days post-hatch (dph). They were born and raised in the zebra finch breeding facility at the Massachusetts Institute of Technology (Cambridge, Massachusetts), and all care and experimental procedures were approved by the Massachusetts Institute of Technology Committee on Animal Care. At age 36 to 80 dph, subjects were isolated in sound attenuating chambers. After they began singing in isolation, a microdrive was implanted into AIid in the right hemisphere.

3.2.1 Surgical procedures

In the first step of surgery, an anterograde tracer (Alexa Fluor 488, dextran 10000 MW, anionic, fixable, Invitrogen) was injected into LMANshell to localize AIid in histology. Injections were made both medial and lateral to LMAN. The exact sites and volumes for the injections varied from bird to bird. A total volume of 64.4 nL to 110.4 nL was injected over multiple smaller injections medial to LMAN (5.30 AP, 0.95–1.15 ML, 1.85–2.00 DV) and another 64.4 nL to 110.4 nL was injected in a similar fashion lateral to LMAN (5.30 AP, 1.95–2.15 ML 1.85–2.00 DV).

After injecting tracer, a bipolar stimulating electrode (Okubo et al., 2014) was placed in LMANshell to identify AIid by its orthodromic response during surgery and during chronic recordings. The medial pole of the stimulating electrode was placed at 5.30 AP, 0.95–1.15 ML, 2.00 DV and the lateral pole was placed at approximately 1.0 mm lateral of the medial pole.

Next, AIid was localized by the following procedure. The head was angled at 40 degrees from the horizontal, where the horizontal is determined by placing a glass rod in the groove at the front of the skull and rotating the head until the rod is parallel to the surgical bench. A craniotomy was made over AIid at the following coordinates relative to the bifurcation of the mid-sagittal sinus:
0.0 mm to 1.4 mm anterior and 2.6 mm to 3.6 mm lateral. Using a carbon fiber electrode (Carbostar-1, Kation Scientific), the lateral edge of RA was found by searching for its characteristic tonic activity 2.6 mm lateral to the midline and 0.0 to 1.4 mm anterior to lambda, the bifurcation of the midsagittal sinus. Then, from the lateral edge of RA, the spatial extent of AI\textsubscript{d} was mapped by finding the spatial extent of the orthodromic response to bipolar electrical stimulation in LMAN\textsubscript{shell}. Typically, this extended to 3.2 mm lateral from the midline and spanned 0.4 mm in the anterior/posterior direction. Finally, the microdrive was implanted so that the medial most electrode was 0.2 mm lateral to the edge of RA and centered with respect to AI\textsubscript{d} in the anterior/posterior direction. The microdrive was equipped with five electrodes that extended in the medial/lateral direction with the most lateral electrode about 0.8 mm lateral to the edge of RA.

3.2.2 Recording procedures

Neural signals were recorded a miniature motorized microdrive. These neural signals were high-pass filtered with a cutoff frequency of 300 Hz and low-pass filtered with a cutoff frequency of 15 kHz. The neural signals and audio from the cage microphone were digitized and recorded on a computer running custom software written in MATLAB.

The audio was also processed in real-time to detect targets for noise bursts song in a manner similar to previous studies (Andalman and Fee, 2009; Charlesworth et al., 2011; Tumer and Brainard, 2007; Warren et al., 2011). These noise bursts were used to test the response of AI\textsubscript{d} neurons to vocal errors. The noise bursts were 50 ms in duration, 90 dB SPL in amplitude, and were played through a loudspeaker next to the cage. The noise bursts were played at a targeted time in the song. This target time was detected in real-time by a combination of filters applied to the audio signal from the cage in real time by a digital signal processor (RX8 Multi-I/O Processor,
Tucker Davis Technologies). The noise burst was only played on a fraction of detected target times, chosen at random.

The target time was detected in different ways depending on the structure of the song. For mature birds with a stereotyped motif, the filters were designed to detect a single moment in the motif using a combination of acoustic features. First, to target a single moment in the target syllable, I used the sound amplitude to detect syllable onset. Sound amplitude was measured as a moving average of the squared sound waveform over 33 samples (1.3 ms). Then, to detect the identity of the syllable, I measured spectral features such as pitch. Pitch was measured with a bank of six finite impulse response (FIR) filters (Andalman and Fee, 2009). Three of the filters, the on-pitch filters, were centered on the fundamental frequency and the first two harmonics of the pitch to be detected. The other three filters, the off-pitch filters, were centered in the gaps between the harmonics. The pitch was detected when the ratio of power in the on-pitch filters to the power in all six filters is above a threshold. If a sound was perfectly on the pitch we are detecting, all of the power will be in the on-pitch filters. On the other hand, broadband sound will have power in both the on-pitch and off-pitch filters. For syllables that do not have a well-defined pitch, I used a similar setup, but only with one “on” filter and one “off” filter, each of which has a wide passband about 4 kHz wide. In all cases, the parameters of the filters, including the cutoff frequencies, are customized for each bird using a custom program written in MATLAB. For juvenile birds without a stereotyped motif, the procedure is similar except the noise bursts were targeted at random during singing. For these birds, only a sound amplitude filter was used. In these juvenile birds, 19–30% of syllables were hit with noise.

To analyze the response of A1d neurons to movements, video of the bird was recorded using a webcam (Logitech Webcam Pro 9000) with a variable frame rate. To synchronize the video
recording with the neural and audio recordings, two clicks were played through a loudspeaker next to the cage. Video was aligned by finding the latency that maximized the cross-correlation between the webcam audio and the data acquisition system audio.

After recording, spike events were detected in the neural data by threshold. First, the electrode signals were high-pass filtered (750 Hz cutoff frequency) with an 80th order finite impulse response (FIR) filter with a Hann window. A threshold was chosen that detected spikes in the filtered signal. Any waveforms that had a substantially different spike waveform were removed by hand. All single units were verified to have an inter-spike interval distribution free from refractory period violations.

3.2.3 Analysis of neural activity during singing

After recovery from surgery, I recorded single neurons in A1d during singing. The song was recorded by a microphone next to the cage. This audio signal segmented into syllables based on the smoothed squared amplitude. Syllables and introductory notes were labeled by hand based on their acoustic structure using a custom program in MATLAB. Calls were left unlabeled and were not analyzed further.

To measure the effect of singing on the activity of A1d neurons, I compared the firing rate during singing to the firing rate outside of singing. Recordings were divided into periods of singing and non-singing based on the occurrence of labelled song syllables. Singing periods or “bouts” were recording segments at least 0.2 seconds long where each gap between syllables is 2.0 seconds or less. Any part of the recording that is not during a bout was designated “non-singing”. The firing rate during singing is the number of spike events during singing periods divided by the sum of the durations of all the singing periods, and the same for non-singing periods. I used a permutation test to determine if the observed differences in firing rate between singing and non-singing periods
were significant. For each neuron, I randomly shuffled the “singing” or “non-singing” labels for the periods and recomputed the difference in mean firing rates using the permuted labels. I repeated this procedure 50,000 times to build a null distribution of firing rate differences. The observed firing rate difference would be significant if it was outside the middle 95% of values in the null distribution.

I tested the significance of the difference in firing regularity between singing and non-singing periods using a similar process. As a measure of regularity, I computed the coefficient of variation (CV) of the inter-spike intervals during singing and during non-singing. The difference between the singing CV and the non-singing CV was compared against a null distribution constructed as before, by permuting the “singing” and “non-singing” labels. The observed CV difference would be significant if it was outside the middle 95% of values in the null distribution.

3.2.4 Monte Carlo analysis of response to vocal errors

How much does the firing rate change following a noise burst? The simplest test would be to measure the firing rate after distorted targets and compare it to the firing rate after undistorted targets. However, I found that this approach did not account for slow but relatively large variations in the firing rate that occurred over seconds. Given a large amount of data, these fluctuations should average out between distorted and undistorted targets, but my dataset was too small. Therefore, I chose to explicitly correct for the variability in firing rate at each target. For the i-th, I measured the change in firing rate between the 150 ms immediately before the target ($r_{pre}$) and the 150 ms immediate after the target ($r_{post}$):

$$\Delta(i) = r_{post}(i) - r_{pre}(i)$$
If a neuron responds to a noise burst, the change in firing rate ($\Delta$) should be significantly different between targets that were distorted versus targets that were undistorted. For juveniles, where the targets could be anywhere in the motif, I selected a sample of undistorted targets that matched the distorted targets in the latency from syllable onset. For each distorted target, I picked an undistorted target that occurred at the same time after syllable onset. I computed the mean of $\Delta$ over distorted targets ($h(i) = 1$) and undistorted targets ($h(i) = 0$):

$$\overline{\Delta_{\text{distorted}}} = \frac{\sum_i \Delta(i) h(i)}{\sum_i h(i)}$$

$$\overline{\Delta_{\text{undistorted}}} = \frac{\sum_i \Delta(i) (1 - h(i))}{\sum_i (1 - h(i))}$$

The difference between these means is the effect of the noise ($b$), which I measured as a fractional change relative to the global mean firing rate of the neuron, $\bar{r}$:

$$b = \frac{\overline{\Delta_{\text{distorted}}} - \overline{\Delta_{\text{undistorted}}}}{\bar{r}}$$

To test the significance of this noise effect, I compared it to a distribution of the effect sizes under the null hypothesis as calculated using a Monte Carlo method. I computed a surrogate by randomly shuffling the distorted/undistorted labels ($h(i)$) and recomputing the effect, $b$. I repeated this process 10,000 times to build a distribution of possible effect sizes under the null hypothesis. The p-value of the measured effect is the fraction of surrogate values that are more extreme than the measured value of $b$. This analysis gives one p-value for the noise effect for each neuron, for a total of 15 p-values. To correct for multiple comparisons, I used the Benjamini-Hochberg procedure to estimate the false discovery rate (FDR) (Benjamini and Hochberg, 1995) for the 15 p-values and only called a result significant if FDR < 0.05.
To compute the standard error of my estimates of the noise effect \( b \), I used the following uncertainty propagation technique. For the numerator, I estimated the standard error of the mean from the data. Note that the three quantities in the equation for \( b \) are all means, so their sampling distributions are normal:

\[
\sigma_{\Delta_{\text{distorted}}} = \frac{\text{std}(\Delta_{\text{distorted}})}{\sqrt{N_{\text{distorted}}}}
\]

\[
\sigma_{\Delta_{\text{undistorted}}} = \frac{\text{std}(\Delta_{\text{undistorted}})}{\sqrt{N_{\text{undistorted}}}}
\]

Since these are two normal distributions, the standard deviation of their sum is:

\[
\sigma_1 = \sqrt{\frac{\sigma^2_{\Delta_{\text{distorted}}} + \sigma^2_{\Delta_{\text{undistorted}}}}}{}}
\]

For the denominator, I assumed that the mean rate was the mean of a Poisson distribution, so the standard error of the mean firing rate is:

\[
\sigma_2 = \frac{\sigma_{\bar{\mathcal{P}}}}{\sqrt{N_{\text{spikes}}}} = \frac{\sqrt{\bar{\mathcal{P}}}}{\sqrt{N_{\text{spikes}}}}
\]

Finally, to get the standard error of \( b \), I use the following formula for the standard deviation of the ratio of two normal distributions (Dunlap and Silver, 1986):

\[
\sigma_b = \sqrt{\frac{\sigma_1^2 + \chi_1^2 \sigma_2^2}{\bar{\mathcal{P}}}}
\]

where:

\[
\chi_1 = \Delta_{\text{distorted}} - \Delta_{\text{undistorted}}
\]

In the last step of this analysis, I combined the effects \( b \) from each neuron to calculate one combined effect for the population. I estimated the combined effect using a random effects model (Borenstein et al., 2011), which is typically used in the meta-analysis of pharmaceutical
studies to combine the results of multiple independent clinical trials. The mean effect is a weighted sum of the effects from each neuron:

\[ b_{meta} = \frac{\sum_{j=1}^{J} W_j^* b_j}{\sum_{j=1}^{J} W_j^*} \]

where \( J \) is the number of neurons. The weights are given by:

\[ W_j^* = \frac{1}{\sigma_{b_j}^2 + T^2} \]

where:

\[ T = \sqrt{\frac{Q - (J - 1)}{C}} \]

\[ Q = \sum W_j b_j^2 - \left( \frac{\sum W_j b_j}{\sum W_j} \right)^2 \]

\[ C = \sum W_j - \frac{\sum W_j^2}{\sum W_j} \]

Finally, the standard error of the combined effect is given by:

\[ \sigma_{b_{meta}} = \sqrt{\frac{1}{\sum W_j^*}} \]

3.2.5 Statistical power for Monte Carlo analysis

One interpretation of a failure to find a significant noise effect is that the neuron does not carry information about the noise burst. To support this interpretation, I need to demonstrate that the failure to find a significant result is not due to other methodological problems, such as
insufficient data. To this end, I performed simulations to measure the statistical power of the Monte Carlo analysis for each of my neurons.

For each neuron, I generated simulated datasets where I added a change in firing rate around the time of the noise bursts while preserving other properties of the experiment. I preserved the mean firing rate of the neuron as well as the number of distorted and undistorted. The simulated spike counts in the 150 ms window before the noise burst were drawn from a Poisson distribution with a mean rate equal to the global mean firing rate of the neuron. The simulated spike counts in the 150 ms window after the noise burst were drawn from a Poisson distribution with a mean rate calculated in one of two ways, both of which are based on the measured change in firing rate following noise burst in AIv neurons reported by Mandelblat-Cerf et al. (2014) (their Figure 8E). The first method was additive: the mean rate following a noise burst was the sum of the global mean firing rate of the neuron from my experiment and the mean increase in firing rate in the window from the AIv data. The second method was multiplicative: the mean firing rate following a noise burst was the product of the global mean firing rate and the factor by which firing rates increased their rates in the AIv data (1.7). I simulated 1000 datasets per neuron and performed the Monte Carlo analysis on each dataset. To reduce the computation time, I only generated 1000 surrogates for each Monte Carlo test. Finally, the statistical power for the neuron is the fraction of datasets where I detected a significant noise effect.

3.2.6 Generalized linear model analysis of response to vocal errors

As an alternative test for the significance of the noise burst effect, I fit a generalized linear model (GLM) to the spike count data. For the i-th neuron, the expected number of spikes in the window \( y_i \) can be modeled as:
\( y_i \sim Poisson[\lambda] \)

\[
\log(\lambda) = \alpha_i + \beta_i h
\]

Where \( h=1 \) when there is noise and \( h=0 \) otherwise. The parameters of this model are related to the firing rate following the target time in noise burst trials \( (\lambda_{h=1}) \) and the firing rate following the target time in trials without a noise burst \( (\lambda_{h=0}) \):

\[
\begin{align*}
\lambda_{h=0} &= \exp(\alpha_i) \\
\lambda_{h=1} &= \exp(\alpha_i + \beta_i) = \exp(\alpha_i) \exp(\beta_i) = \lambda_{h=0} \exp(\beta_i) = G_i \lambda_{h=0}
\end{align*}
\]

\( G_i \) is the change in firing rate in the presence of noise, which is modeled as a multiplicative gain on the firing rate without noise. The i-th neuron had a significant response to noise if the p-value of the coefficient \( \beta_i \) was less than 0.05. Like the Monte Carlo analysis, this analysis gives one p-value for the noise effect of each neuron for a total of 15 p-values. To correct for multiple comparisons, I used the Benjamini-Hochberg procedure to estimate the false discovery rate (FDR) \((Benjamini and Hochberg, 1995)\) of these 15 hypothesis tests and only called a result significant if \( \text{FDR} < 0.05 \)

Then, to combine across birds, I fit a second model where each neuron has its own baseline firing rate related to \( \alpha_i \), but all neurons share the same noise burst effect in a single parameter \( \beta \):

\( y_i \sim Poisson[\alpha_i + \beta h] \)

3.2.7 Analysis of head and body movements

For analysis, the video was transcoded into raw video with a fixed frame rate of 15 frames per second. The resulting video was labeled by hand, frame-by-frame using the video annotation tool ANVIL \((Kipp, 2014)\). The video was labeled for grooming, pecking, chewing, and head movements along three orthogonal axes: yaw (left or right), pitch (up or down), and roll (left or
right), as shown in Figure 3.8A. Head movements along different dimensions are not mutually exclusive. For example, a single movement where the head rotates up and to the right would be labeled as both pitch up and yaw right. On the other hand, grooming, pecking, and chewing are each exclusive. All of the annotations were performed by an observer who was blind to the neural data.

To test for significance of each movement type, we computed the mean firing rate between movement onset and movement offset across all occurrences of that movement. We compared this measured firing rate to a null distribution computed for 100,000 surrogate datasets. In each surrogate dataset, each movement is randomly moved to a new position in the data that does not overlap with a movement from the type being tested. The p-value for each movement is the fraction of surrogates that are more extreme than the observed mean firing rate. This analysis gave one p-value for each of the nine movement types for each of the fifteen neurons, for a total of 135 p-values. To correct for multiple comparisons, I used the Benjamini-Hochberg procedure to estimate the false discovery rate (FDR) (Benjamini and Hochberg, 1995) across these 135 tests and only called a result significant if FDR < 0.05.

3.3 Results
To distinguish between the vocal learning and motor hypotheses for the function of AId, I recorded from single units in AId during both singing and natural head and body movements. The same units were recorded in both behaviors. The units were recorded with a miniature, motorized microdrive chronically implanted into AId (see Section 3.2), depicted schematically in Figure 3.1D.

First, I characterized the firing patterns of AId neurons to compare them with the firing patterns other regions of arcopallium, RA and Alv, which have been recorded in previous studies. These two regions of arcopallium have been well-characterized and have very different firing
patterns. RA contains two cell types, projection neurons and putative interneurons. These cell types can be distinguished by their spike waveform as well as their firing patterns. RA projection neurons fire regularly at 20 to 30 Hz outside of singing and have broad spike waveforms (Leonardo and Fee, 2005). In AIV, projection neurons fire at only 1 to 10 Hz (Mandelblat-Cerf et al., 2014). Ald neurons have mean firing rates of $15.9 \pm 7.8$ Hz (mean ± standard deviation). The distribution of mean firing rates of Ald neurons is broadly distributed between 3 Hz and 30 Hz (Figure 3.2A). Similar to neurons in RA, Ald neurons have spike widths that follow a bimodal distribution with one group at 150 μs and another at about 300 μs (Figure 3.2C). However, unlike RA neurons, Ald neurons were not highly periodic. Their inter-spike interval distributions were distributed broadly (Figure 3.2B).

How do the firing patterns of Ald neurons change with singing? In RA, projection neurons change their firing patterns dramatically during singing by transitioning from regular spiking to a highly irregular pattern of bursts and pauses (Leonardo and Fee, 2005; Vu et al., 1994). In contrast, AIV neurons maintain the same low firing rate regardless of singing state. Once again, Ald is not exactly like either of these other brain areas. In 10 of 15 Ald neurons I recorded, the mean firing rate was not significantly different during periods of singing and non-singing (Figure 3.2D, see Section 3.2.3). On the other hand, spiking variability decreased during singing. The coefficient of variation of the inter-spike interval distribution is significantly lower during singing (Figure 3.2E, $p < 0.05$, Wilcoxon sign-rank test). In the following sections, I investigate if these modulations in firing carry information about vocal errors or head and body movements.

3.3.1 Response of Ald neurons to vocal errors
Distorted auditory feedback has been used by previous studies to look for evidence of auditory processing across the songbird brain. These stimuli evoke short latency responses in the
auditory forebrain (Keller and Hahnloser, 2009), AIV (Mandelblat-Cerf et al., 2014), and in dopaminergic neurons in VTA that project to the song system (Gadagkar et al., 2016). These same stimuli fail to evoke responses during singing in the nuclei of the song system, such as LMAN (Leonardo, 2004) and HVC (Hamaguchi et al., 2014; Kozhevnikov and Fee, 2007).

To test if AId is involved in auditory processing of what the bird hears while singing, I recorded the response of the neurons to distorted auditory feedback (Figure 3.3A). During singing, a digital signal processor monitors the song in real time and detects a predetermined target time. On a portion of these targets, selected at random, a loud burst of white noise is played through a loudspeaker next to the cage (see Section 3.2.2).

On first examination, AId neurons did not seem to modulate their firing rates following distorted auditory feedback. In peri-stimulus time histograms (PSTHs) aligned to the target time, the firing rate seemed highly similar following distorted auditory feedback and undistorted targets (Figures 3.3–3.4). If the neurons were sensitive to vocal errors, the firing rates after the target time should be different in distorted versus undistorted trials. However, the average of this difference across all recorded AId neurons appeared flat (Figure 3.5).

I performed a more detailed analysis of the firing rates following distorted auditory feedback by measuring the difference in firing rate before and after the target time. The distribution of these differences for distorted and undistorted targets from five example neurons is shown in Figure 3.6B–3.6F. To test if the means of these distributions were significantly different, I performed a Monte Carlo analysis (see Section 3.2.4). After correcting for multiple comparisons, none of the individual neurons had a significant response (Figure 3.6A, top).

One interpretation of a failure to find a significant noise effect is that the neuron does not carry information about the noise burst. To support this interpretation, I need to demonstrate that
the failure to find a significant result is not due to other methodological problems, such as insufficient data. To this end, I performed simulations to measure the statistical power of the Monte Carlo analysis for each of my neurons. The statistical power is the probability of discovering a significant change in firing rate following the onset of the noise burst when such a change is truly present. In these simulations, each neuron had a change in firing rate similar to the mean change in firing rate observed in Alv neurons by Mandelblat-Cerf et al (2014). I calculated this change in firing rate two different ways: additive and multiplicative (see Section 3.2.5). The statistical power for each neuron estimated using both the additive and the multiplicative methods are listed in Table 1. Using the multiplicative method, 10 of 15 neurons have a power above 0.8, while using the additive method, 6 of 15 neurons have a power above 0.8.

<table>
<thead>
<tr>
<th>Neuron Identifier</th>
<th>Mean firing rate (Hz)</th>
<th>Trial Counts</th>
<th>Statistical Power, multiplicative</th>
<th>Statistical Power, additive</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>31.0</td>
<td>53 47</td>
<td>1.000</td>
<td>0.458</td>
</tr>
<tr>
<td>C</td>
<td>4.8</td>
<td>79 53</td>
<td>0.687</td>
<td>0.968</td>
</tr>
<tr>
<td>D</td>
<td>22.0</td>
<td>78 32</td>
<td>0.993</td>
<td>0.553</td>
</tr>
<tr>
<td>E</td>
<td>16.9</td>
<td>70 19</td>
<td>0.873</td>
<td>0.495</td>
</tr>
<tr>
<td>F</td>
<td>4.2</td>
<td>38 38</td>
<td>0.460</td>
<td>0.983</td>
</tr>
<tr>
<td>G</td>
<td>11.7</td>
<td>37 13</td>
<td>0.531</td>
<td>0.453</td>
</tr>
<tr>
<td>H</td>
<td>14.9</td>
<td>84 45</td>
<td>0.983</td>
<td>0.824</td>
</tr>
<tr>
<td>I</td>
<td>23.7</td>
<td>78 33</td>
<td>0.998</td>
<td>0.562</td>
</tr>
<tr>
<td>J</td>
<td>25.0</td>
<td>49 34</td>
<td>0.995</td>
<td>0.460</td>
</tr>
<tr>
<td>K</td>
<td>9.6</td>
<td>24 16</td>
<td>0.451</td>
<td>0.509</td>
</tr>
<tr>
<td>L</td>
<td>56.7</td>
<td>50 21</td>
<td>1.000</td>
<td>0.187</td>
</tr>
<tr>
<td>M</td>
<td>43.1</td>
<td>10 5</td>
<td>0.611</td>
<td>0.093</td>
</tr>
<tr>
<td>N</td>
<td>3.1</td>
<td>391 391</td>
<td>0.999</td>
<td>1.000</td>
</tr>
<tr>
<td>O</td>
<td>24.2</td>
<td>281 281</td>
<td>1.000</td>
<td>0.998</td>
</tr>
<tr>
<td>P</td>
<td>10.3</td>
<td>539 539</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Even though there is no significant result detected in the individual neurons, there may still be a very small modulation that I did not have sufficient statistical power to detect on the single neuron level. To gain more statistical power, I combined the results from all of the neurons into an
estimated population mean. I combined the individual neurons using a random effects model, which assumes that each individual effect is randomly drawn from a normal distribution around an overall mean (Borenstein et al., 2011). The overall mean is estimated by a weighted sum of the individual means, where the weight is related to the uncertainty in each individual effect (see Section 3.2.4). By this analysis, the population of AId neurons as a whole have a small (8.8%) but statistically significant decrease in their firing rates following a noise burst (Figure 3.6B, bottom).

I performed a second analysis to compute an overall mean effect using a generalized linear model (GLM). First, I modeled each neuron separately, where the spike count in the 150ms window following the target time is a Poisson-distributed random variable. Distributions of these counts for five example neurons are shown in Figure 3.7B–3.7F. The rate of the Poisson process is a function of a baseline firing rate as well as the presence of distorted auditory feedback (see Section 3.2.6). From these individual fits, none of neurons had a significant relationship between the presence of a noise burst and the spike rate after the target time after correcting for multiple comparisons (Figure 3.7A, top).

To get an overall effect from this GLM analysis, I fit a model that included data from all of the neurons and had just one parameter to represent the shared distorted auditory feedback response across all of the neurons. In this model, the overall effect was a 5.1% reduction in spiking rate following distorted auditory feedback (p=0.033, Figure 3.7A, bottom). This agrees with the results from the Monte Carlo analysis; both tests failed to find an effect on the individual neuron level but found a very small effect when combining across all neurons.

3.3.2 Response of AId neurons to head and body movements

Immediate early gene expression results from previous studies suggest that neurons in AId have increased activity during head and body movements. Therefore, we hypothesized that single
units in AId would modulate their firing rates during head and body movements. To measure the responses of single units in AId to head and body movements, we annotated videos by hand, frame by frame. The video annotators were blind to the neural data. The videos were annotated for grooming, pecking, and head rotations in three orthogonal directions: yaw (left or right), pitch (up or down), and roll (left or right). A raster and peri-stimulus time histogram for the best head movement response are shown in Figure 3.8C. The firing rate increases beginning after the onset of pecking movements. It does not modulate its firing rate around any other movements (Figure 3.8D).

Overall, 10 of the 15 single units had a significant modulation during at least one movement after correcting for multiple comparisons (see Section 3.2.7). At least one neuron responded to each of the movement labels (Figure 3.9A) and most neurons responded to more than one movement type (Figure 3.9B).

In most of the neurons, the magnitude of the movement-related firing rate change was larger than the noise-related firing rate change in that same neuron. For each neuron, I compared the firing rate change from the most significant movement from each neuron to the firing rate change from the Monte Carlo analysis of distorted auditory feedback responses (Figure 3.10), and found that the movement-related responses were larger.

3.4 Discussion

The results give clear support to the motor hypothesis AId function, but cannot rule out a contribution of AId to processing vocal errors. Consistent with the motor hypothesis, 10 out of 15 single units modulate their firing rates during at least one category of head or body movements. None of the neurons had a significant response to noise bursts played during undirected singing when analyzed individually. Furthermore, this noise burst modulation is much smaller in
magnitude than what has been observed in other brain areas. As a population, AId neurons decrease their firing rates by about 5% after a noise burst, but in Alv, firing rates increase by 100-200% (Mandelblat-Cerf et al., 2014). However, there is a small but significant decrease in firing rate following noise bursts, so we cannot rule out that AId is carrying some auditory information alongside the motor-related signals. Another possible explanation is that these distorted auditory feedback responses are actually due to changes in movements in response to the noise. For example, the birds may flinch or freeze momentarily. Future experiments can test this hypothesis by recording movements during singing and fitting a single unified model that includes movement and auditory feedback responses.

Why is AId lacking the auditory responses that have been observed upstream in LMANshell? First, many of the recordings in LMANshell have been under anesthesia (Achiro and Bottjer, 2013), which may be different than during wakefulness. Similarly, previous studies of LMAN and HVC found neurons that responded to playback of the bird’s own song while under anesthesia (Doupe, 1997; Doupe and Konishi, 1991; Lewicki, 1996; Margoliash, 1983; Volman, 1996), but these neurons are not responsive to distorted auditory feedback during singing (Leonardo, 2004). One possibility is that auditory signals are gated off in the parallel pathway when the bird is awake and moving. Another possibility is that in awake birds, singing or hearing song may be correlated with movement. I have observed that zebra finches who are resting quietly will often begin hopping around their cage upon playback of any zebra finch song.

The tonic activity of some AId neurons is reminiscent of the activity of RA neurons outside of singing. However, Ald neurons never seem to transition to a burst-silence pattern of firing like RA neurons exhibit during singing. Perhaps the precise bursting of RA neurons is unique to the particular requirements of song production. Another possibility is that Ald neurons are capable of
precise bursting, but they were never observed in the behavior that would put them into this state. For example, they may enter a bursty state during flight, but the birds were unable to fly in the recording chamber.

For the neurons that do respond to movements, the causal role in initiating or controlling movements is not clear from these recordings. Given the output of AId to brainstem motor centers, it is reasonable to speculate that they may be involved in motor control, but the timing of their firing suggests that they may only begin firing after movement initiation. I cannot rule out that their activity is actually the result of proprioceptive or other sensory activity related to the movement. Future experiments could clarify the causal role of AId in generating movements, for example stimulating AId and observing an evoked movement or muscle twitch.

Future work could clarify the relationship between movements and the neural activity in AId with a more quantitative analysis. The current study was limited by the method for labeling movements from video. For example, we might want to know if the neurons in AId are premotor, as one might expect given the projection from AId to midbrain motor centers. However, the interval between frames is 33 milliseconds, so this method of movement detection does not have sufficient temporal resolution to answer this question. Taking video at a higher frame rate or measuring movements by another method is necessary.

Finally, video labelling is inherently subjective. A small portion of the videos were labelled by a second viewer, and the agreement between labelers was only about 80%. In the next chapter, I develop an objective, quantitative way to measure movements.
Figure 3.1: Brain circuitry and localization of Ald. (A) Schematic of song control brain nuclei. Includes the song motor pathway that controls adult song in HVC to RA to motor neurons in nXII. Also includes a learning circuit of HVC, Area X, DLM, and LMAN. (B) Schematic of brain areas in the parallel pathway, including Ald, which is the focus of this chapter. (C) Brain areas from the song system (green) and the parallel pathway (purple). Each nucleus in the song system is adjacent to a nucleus in the parallel pathway. (D) Schematic of chronic neural recordings. A bipolar stimulating electrode is placed in LMANshell and a recording electrode is in Ald. (E) Evoked response of an example single unit in Ald to stimulation in LMANshell. The stimulus is at time zero, followed by orthodromic activity at 2 to 5 milliseconds. (F) Histological confirmation of recording site in Ald. This is a coronal section with lateral to the right and dorsal to the top. Fluorescent label is visible in RA, the sphere towards the medial edge, and in Ald, the oblong region lateral to RA. An asterisk marks the location of the small electrolytic lesion created through the electrode after the final recording.
Figure 3.2: Properties of Ald neurons. (A) Mean firing rate distribution of all Ald single units. Rates are broadly distributed between 3 Hz and 30 Hz. (B) Example inter-spike interval histogram from one example neuron. (C) Distribution of spike width for all Ald neurons. Width is measured as the time between the peak and trough of the spike waveform. Example waveforms are shown above the histogram for one narrow unit and one broad unit. (D) Mean firing rate during singing vs. non-singing periods. Red dots represent neurons where the rate during singing is significantly different from the rate during non-singing. Black dots represent neurons where there is no significant difference. (E) Difference in spiking variability during singing and non-singing periods. Each linked pair represents the coefficient of variation of the inter-spike interval distribution for a single Ald neuron. As a population, Ald neurons have a significantly lower coefficient of variation during singing periods.
Figure 3.3: Response of Ald neurons to vocal errors during singing. (A) Example spectrogram (top) and neural data. The red arrow marks the onset of a noise burst played during singing. (B-F) Example spectrogram (top), raster (middle) and peri-stimulus time histogram (PSTH, bottom) of spikes relative to the target time for noise burst during singing for five example neurons. In the upper half of the PSTH, the red shaded region marks the duration of the noise burst. In the bottom half of the PSTH, there were no noise bursts, but the red vertical line marks the target time. Yellow shaded regions represent song syllables.
Figure 3.4: PSTH of noise response. (G-H) PSTH of neural responses around the target time (t=0). Red trace is the average of targets hit by noise and black trace is the average of targets that were undistorted. Combined with Figure 3.3D–3.3F, this is all of the neurons in the dataset.
Figure 3.5: Averaged PSTH for all A1d neurons in response to noise bursts delivered during singing. The PSTH shows the difference between distorted and undistorted trials. Shaded region marks the 95% confidence intervals for the mean, as determined by bootstrapping.
Figure 3.6: Monte Carlo analysis of response to vocal errors. (A) Each point represents the mean difference in firing rate between distorted and undistorted targets for one neuron. Error bars represent the standard error of the mean. Single units are labeled B-P, corresponding to their labels in Figures 3.3-3.4. At the bottom, the summary effect combines across all neurons. (B-F) Histograms of spike count differences for example neurons. Difference in the number of spikes 150 ms before target and 150 ms after target for undistorted trials (blue) and distorted trials (red). These are the same example neurons as in Figure 3.3B–3.3F.
Figure 3.7: GLM analysis of response to vocal errors. (A) Each point represents the multiplicative gain on the firing rate of the neuron in the presence of a noise burst. Error bars represent the 95% confidence intervals. Single units are labeled B-P, corresponding to their labels in Figures 3.3-3.4. At the bottom, the summary effect combines across all neurons. (B-F) Histograms of spike counts in the 150ms window following the target time on escape trials (blue) and hit trials (red). These are the same example neurons as in Figure 3.3B–3.3F.
Figure 3.8: Response of example Ald neuron to movements. (A) Diagram of the six rotations that were annotated in the video. (B) Example of raw data with dark blocks marking movements (top) and accompanying neural recording (bottom). (C) Raster aligned to onset of pecking movements for this example neuron. Red blocks mark the duration of the head movement. (D) Peri-stimulus time histograms of all head rotations for this example neuron. The top PSTH is constructed from the raster in C.
Figure 3.9: Number of neurons with significant movement responses. (A) Number of neurons with a significant response for each movement class. A single neuron is counted more than once if it has significant responses to more than one movement. (B) Movement selectivity of each neuron. Here every neuron is counted only once, and neurons without any significant movement responses are also included. Overall 10/15 neurons have significant responses to at least one movement, with 7 of 15 having significant responses to more than one movement type.
Figure 3.10: Comparison of responses to noise burst and movement. Each point represents one neuron. Each y value is the mean difference in firing rate between distorted trials and undistorted trials in a 150 ms window after the target time. Each x value is the mean difference in firing rate between the rate during the most significant movement class and the mean firing rate. Error bars represent the standard error of the mean.
4 Neural coding of head and body movements in motor cortex of freely behaving zebra finches

4.1 Introduction

In this chapter, I perform a more sophisticated experimental and quantitative analysis of the relationship between neural activity in AId and head and body movements. My recordings from Chapter 3 demonstrated that single neurons in AId carry minimal information about vocal errors but modulate their firing rates during various head and body movements. To gain a better understanding of this movement-related activity, I will address some of the weaknesses of the manual annotation process from the previous chapter. The reliance on frame-by-frame labeling of videos made movement classification qualitative, subjective, and time-consuming. The labels were limited to nine different categories, including head rotations, pecking, chewing, and grooming, which are only a subset of the zebra finch behavioral repertoire. In this chapter, I describe hardware and software for automatically classifying a wide variety of zebra finch behaviors. I designed a new microdrive that includes inertial sensors in order to make quantitative measurements of head movements while also recording the activity of single neurons. Next, I developed an automated movement classification algorithm to sort the movement data into clusters of similar movements. Finally, I analyzed the relationship between these movement clusters and the activity of single units in AId.

A wide variety algorithms exist to quantitatively identify behavioral elements of diverse animal species, including rodents, hydra, nematodes, and drosophila. These algorithms have their roots in ethology, the study of natural behavior, which hypothesizes that natural behavior is composed of a sequence of simpler components (Tinbergen, 1951), such as feeding, grooming,
and walking. Behavior classification algorithms are designed to identify the behavioral modules that form the building blocks of natural behaviors. Although behaviors such as locomotion are common to nearly all animal species, the mechanical details vary widely by species; rodents walk by moving their four legs, hydra somersault by reaching and grasping with their many tentacles, and nematodes crawl without any appendages at all. Each behavior classification algorithm must be tailored to accommodate the animal of interest, but all of the algorithms follow a common pipeline.

Most behavior analysis algorithms have a common goal — to analyze raw behavior measurements and produce category labels. For example, raw video of a mouse in his enclosure is automatically divided into periods of resting, walking, grooming, and eating. Techniques for behavior classification vary widely, in both the format of the raw data and the computational techniques employed in categorization. Raw data may come from a video camera outside the cage or from inertial sensors such as accelerometers affixed to the animal. Behavioral categories can range from high-level descriptors like resting and walking to more specific labels like left and right turns.

No behavior analysis maps directly from raw data to category because learning such a mapping is impractical with high-dimensional raw data. This is an example of a well-known problem in machine learning, the curse of dimensionality: as the dimensionality of the input data increases, exponentially more data is required to maintain the same level of classification. Therefore, before classification, all algorithms take an extra step of feature extraction, which reduces the dimensionality of the data while preserving relevant information. A complete pipeline for behavior analysis begins with raw data, goes through several stages of feature extraction, and only then is classified into discrete movement groups.
The computational machinery underlying these algorithms include both supervised and unsupervised machine learning techniques. Supervised approaches rely on a “ground truth” classification provided by a human expert. The supervised algorithm learns to match the ground truth labels; when it is presented with a new segment of behavior, it will classify it the same way as a human expert. On the other hand, unsupervised algorithms do not use any “ground truth” labels; they only rely on the underlying structure of the data. The unsupervised version of classification is clustering. Clustering algorithms find groups of similar movements without any input from a human expert, so they can discover new categories of movement that an expert may not have noticed. Unsupervised approaches also go beyond classification to include several approaches to dimensionality reduction.

In the next few sections, I review the methods from several recent studies in automated behavior analysis. These studies include a wide variety of methods, including both supervised and unsupervised approaches.

4.1.1 Instrumentation for automated behavioral analysis

Raw movement data is typically acquired by video camera or inertial sensors. The most common setup is a single overhead camera that looks down on the environment, which has been used with nematodes (Albrecht and Bargmann, 2011; Stephens et al., 2008, 2011), rodents (Wiltschko et al., 2015), drosophila (Berman et al., 2014; Branson et al., 2009), and hydra (Han et al., 2018). In most of these studies, the animal was constrained to only move in two dimensions, corresponding to the focal plane of the camera. However, one study was able to capture movement in three dimensions, including movements towards the camera, using a single overhead depth camera (Wiltschko et al., 2015). For some species, an overhead camera cannot capture some
movements, such as paw movements in rodents that are obscured by the body, so sometimes the camera is placed on the side of the cage (Jhuang et al., 2010).

In addition to video cameras, inertial sensors such as accelerometers and gyroscopes are also commonly used for recording movement data. Inertial sensors are widely used to identify behaviors in humans, as they are a component of nearly all smartphones (Anguita et al., 2013; Lv et al., 2018; Reyes-Ortiz et al., 2016; Wu et al., 2012). Inertial sensors have several advantages and disadvantages compared to video recording. Inertial sensors are small enough to be fully wearable allowing the animal free range of motion (Venkatraman et al., 2010). On the other hand, inertial sensors provide less complete information about body position and movements. They are usually affixed to one part of the body, and therefore cannot directly measure the movements of other body parts. Measuring absolute orientation is particularly difficult due to sensor drift. Nevertheless, inertial sensors are still attractive because they also are much lower dimensionality, with a typical setup having only three accelerometers and three gyroscopes to cover the three spatial dimensions.

4.1.2 Feature extraction

Raw sensor values can be difficult to translate into specific movements. This is especially true for video data, where a slight change in posture or the movement of one appendage may change thousands of pixel values. Rather than examining the raw pixel values directly, a more parsimonious explanation can come from reducing the raw data to a smaller number of meaningful features. For example, a worm can be represented as a single curve down the center of its body, with the shape of the curve specified by its local angle of curvature at each point (Stephens et al., 2008). Naturally, the method for computing a low-dimensional representation of posture varies by species, but posture estimation is an important first step in algorithms for rodents (Wiltschko et al.,
and hydra (Han et al., 2018) as well. Other algorithms take a different approach to feature extraction by relying on generic measurements of video features, such as moving edges (Berman et al., 2014; Han et al., 2018).

In addition to spatial components, features also include temporal information because behaviors are composed of movements that unfold over time. In many studies, the features come from the frequency domain following a Fourier or wavelet decomposition of the data (Berman et al., 2014; Venkatraman et al., 2010; Wiltschko et al., 2015). Others use low level visual features that include motion (Han et al., 2018; Jhuang et al., 2010).

Often, a feature extraction pipeline contains multiple stages, including dimensionality reduction using unsupervised learning techniques. The most common algorithm for dimensionality reduction is principal component analysis (PCA), which decomposes points into an orthogonal basis set such that the first component holds the maximum possible variance, the second component holds the second most, and so on. To reduce the dimensionality, the components that capture a small amount of variance are discarded. The resulting dimensions are each a linear combination of the original dimensions. PCA has been used to extract a low-dimensional representation of posture in nematodes (Stephens et al., 2008, 2011), drosophila (Berman et al., 2014), and mice (Wiltschko et al., 2015).

Another approach is to use a non-linear dimensionality reduction technique, like t-distributed stochastic neighbor embedding (t-SNE). t-SNE is a technique for a low-dimensional representation such that points that are close together in the high-dimensional space will also be close together in the low-dimensional space (Maaten and Hinton, 2008). Unlike PCA, t-SNE favors preserving the local structure of the data. For points that are far apart in the high-dimensional space, PCA will also keep these points far apart in the low-dimensional space, but t-
SNE may allow these points to be close together in the low-dimensional space if such an arrangement allows better preservation of the relationships of nearby points. t-SNE was first applied to behavioral data Berman et al. (2014) to classify the behaviors of drosophila. Since then, it has also been used to cluster hydra behavior (Han et al., 2018), and in this chapter I will use t-SNE to cluster songbird behavior.

4.1.3 Classification of natural behaviors

In the final step, classification algorithms divide all of the behaviors into discrete categories based on their features. Supervised learning methods used for behavior classification include support vector machines and neural networks (Han et al., 2018; Venkatraman et al., 2010; Wiltschko et al., 2015). The main weakness of these algorithms is their reliance on a hand-labeled training set, which is time-consuming to create. Supervised algorithms can only detect movement classes that have already been labeled by a human in the training set; they cannot discover new movement classes.

Unsupervised techniques discover groups do not use any “ground truth” labels and instead find clusters of behavior based on the structure of the data alone. One particular unsupervised technique is to cluster points using the watershed transform (Meyer, 1994) after embedding movements in a 2-D space using t-SNE. After classification, a human expert watches exemplar videos from each behavior class and assigns an intuitive label (Berman et al., 2014; Han et al., 2018; Stephens et al., 2008, 2011; Wiltschko et al., 2015). The watershed approach was introduced to behavior classification of drosophila by Berman et al. (2014) and subsequently used in hydras (Han et al., 2018), and I will use this approach on songbird behavior in this chapter.
4.1.4 Case study: behavioral classification in drosophila

Now I will take a closer look at the full pipeline for one particular study: the classification of drosophila behavior by Berman et al. (2014). This study is the source of the dimensionality reduction and classification elements that I will use for my own data later in this chapter. Berman et al. clustered the full repertoire of ground-based behavior of drosophila. They acquired video of the fly with an overhead camera that acquired 40,000 pixels per frame at 100 frames per second.

After some preprocessing to register each frame to a reference image, Berman et al. performed a multistep feature extraction procedure. First, they extracted what they termed "postural modes" of the fly through a multistep process. They took the radon transform of the image and reduced the dimensionality by keeping only the transform components that captured the most variance of the images. Next, they reduced the dimensionality further by performing PCA on the remaining transform components and keeping only the 50 components that capture the most variance. These 50 components are the postural modes.

For the next step in feature extraction, Berman et al. computed the spectrograms of their postural modes. Each of the 50 spectrograms has 25 frequency channels, for a total dimensionality of 1250. At this stage it might seem counterproductive to increase the dimensionality, but they needed a way to capture the temporal aspect of the rhythmic behaviors of the fly. In one final dimensionality reduction step, Berman et al. use t-SNE to embed the data into just two dimensions.

After embedding the data with t-SNE, Berman et al. classified the points in the embedded space using unsupervised techniques. First, they transformed the embedding into a smooth heat map by convolving each point with a Gaussian kernel. Then they clustered the embedded points by performing a watershed transform on the heat map. Each local maximum in the heat map
became a cluster center. Points were assigned to a cluster by walking up the gradient of the heat map to the nearest cluster center. After clustering, human observers assigned labels to each cluster.

4.1.5 Analysis of avian motor behavior

A number of earlier studies have characterized the singing behavior of songbirds (Mandelblat-Cerf and Fee, 2014; Mets and Brainard, 2018; Tchernichovski et al., 2000, 2004), but much less work has been done to characterize their head and body movements. Birds have a rich behavioral repertoire including social behaviors, nest building, flying, grooming, and eating (Zann, 1996). Most of these behaviors have not been quantitatively analyzed at all. Analysis of songbird head and body movements has focused on their courtship dance. In some bird species, the dance can be highly elaborate, with jumps and leg and wing movements synchronized to vocalizations (Dinsmore, 1970). In zebra finches, the courtship dance is more humble, with some beak movement and hopping that is loosely correlated with the onset of singing (Ullrich et al., 2016; Williams, 2001). All of these behaviors have been catalogued by description in prose, or by manual video analysis. One recent study has automated the analysis of head movements during one specific behavior — prey killing in shrikes. Shrikes kill their prey by a rapid head movement while holding the prey in their beaks. Using automated analysis of high speed video, this head movement was found to be highly stereotyped (Sustaita et al., 2018). The study focused on one particular behavior, but it did not analyze any movements outside of this specialized prey-killing maneuver. My work in this chapter is the first application of modern automated behavior classification to a wide range of natural, non-vocal songbird behavior.

In this chapter, I describe an automatic analysis of the head and body movements in birds, combined with analysis of the firing patterns of single units in Ald during these movements. I design a new microdrive that can record movement data as well as single unit activity from freely
behaving zebra finches. Next, I cluster zebra finch movements using unsupervised techniques, which capture a wide variety of movement classes including some behaviors that were not included in the previous chapter, such as hopping. Finally, I analyze the relationship between these behavior clusters and single unit activity in AIld. I find that 16 of 29 neurons modulate their firing rates during specific movement classes.

4.2 Methods

4.2.1 New recording system for head motion and single unit activity

To capture movement signals alongside neural activity, I designed a new miniature motorized microdrive with a nine-axis inertial measurement unit (IMU). A schematic of the design of the microdrive and apparatus is show in Figure 4.1A. The microdrive on the head has sensors for movement data and single unit activity. The microdrive is connected via a thin cable to a data logger that records the movement data. The electrode signals pass through the data logger and travel through a motorized commutator to separate signal processing and data acquisition hardware. The following sections describe each component of the recording system in detail. Further details and assembly instructions can be found in Appendix A.

4.2.1.1 Microdrive

The microdrive is a totally new design that incorporates electrodes for recording single unit activity as well as inertial sensors to record movement data. The body of the microdrive is based on a single custom printed circuit board (PCB) which includes two electrically isolated subsystems: an analog system for neural recording and a digital system for movement recording. When fully assembled, the microdrive weighs 1.0 g and measures 20.9 mm x 12.1 mm x 5.5 mm. The outside
of the PCB is covered with a clear plastic cover made from overhead transparency to protect the motor and the electronics from dirt and water.

One side of the PCB holds the components for recording neural data (Figure 4.1D). Similar to previous microdrives for zebra finches (Fee and Leonardo, 2001; Okubo et al., 2014), this design uses movable metal microelectrodes to record single unit activity. For the recordings in this chapter, I used three platinum-iridium electrodes with 3 MΩ impedance (P120035.0A3, MicroProbes for Life Science), but other electrode impedances and materials could be used instead. The metal microelectrodes are affixed to a shuttle, which can be moved up and down in unison by a miniature brushless DC motor (Faulhaber, 0206 A 001 B 021:1 Y2825). Each electrode is connected by a thin platinum-iridium wire (90% Pt, 10% Ir; PFA coated; 0.002” bare, 0.0040” coated; Cat No. 776000, A-M Systems) to a solder pad on the PCB. The electrode signal goes through a one-pole RC high pass filter with a cutoff of 14.7 Hz to eliminate any DC offset from the electrode. Next, the filtered electrode signals are amplified by a factor of 100 using an instrumentation amplifier (AD8224, Analog Devices). The reference for this amplifier comes from a platinum-iridium virtual ground wire (Cat No. 776000, A-M Systems) inserted into the brain near the electrodes. The amplified electrode signals go to a connector (A75243-001, Omnetics Connector Corporation) on the opposite side of the PCB, where they travel up a thin flexible cable to the data logger and the rest of the system.

The opposite side of the PCB contains the digital electronics for capturing movement information. The movement measurements are performed by a nine-axis IMU, which has a gyroscope, accelerometer and magnetometer along each of the three spatial dimensions (MPU-9250, Invensense). I only record from the accelerometers and gyroscopes because the magnetometers caused large artifacts in the electrode signals. The accelerometers and gyroscopes
are sampled at 200 Hz and have dynamic ranges of ±2000 degrees per second for the gyroscopes and ±4 g for the accelerometers, where g is the acceleration due to gravity of Earth. The IMU communicates with the data logger devices through a two-wire I2C bus.

4.2.1.2 Data logger for movement signals

The main function of the data logger is to communicate with the IMU on the microdrive. The data logger receives the neural and movement signals from the microdrive and sends them each to separate recording systems. The bottom of the data logger PCB connects to the microdrive cable, and the top of the data logger PCB connects to the commutator. The pre-amplified neural signals pass through the data logger unaltered, while the movement signals are saved on the data logger.

The data logger PCB is designed to mate with a microcontroller, such as an Arduino or Teensy (PJRC), which runs custom software to initialize the IMU and record its data. The microcontroller polls the IMU for new data once every five milliseconds (200 Hz sampling rate), and writes the data to an SD card. This digital communication causes very large transients on the power and ground wires that serve the microcontroller and IMU. In order to minimize the effect of these transients on the extremely sensitive neural recordings, I found that the power and ground for the digital circuitry had to be isolated from the analog power and ground. Therefore, the microcontroller and IMU are powered by a battery pack consisting of three AA batteries attached to the data logger.

The IMU signals do not travel any farther up the signal processing chain, but the IMU recordings made by the microcontroller need to stay synchronized with the neural recordings made by the computer farther up the chain. To solve this synchronization problem, the microcontroller emits a unique sequence of pulses that travel through the commutator to be captured by the neural
recording system. Then, after the fact, the two recordings can be synchronized by aligning the pulse train logged by the microcontroller with the pulse train recorded alongside the neural data. I discuss this procedure in more detail in Section 4.2.1.5.

The final function of the data logger is to measure the torque on the cable and relay this signal to the motorized commutator. At the bottom of the data logger PCB, the cable from the microdrive mates with a connector attached to a torque sensor assembly. The connector is attached to one end of a hollow tube, which is attached to the inside of a ball bearing. When the cable rotates, the connector and tube rotate too, inside the ball bearing. To sense the rotation of the tube, a small Hall sensor is fixed to the PCB behind the tube. The Hall sensor measures the magnetic field of a small magnet attached to the tube. Rotations of the tube change the orientation of the magnet relative to the fixed Hall sensor and thus change the magnetic field detected by the sensor. This torque signal is amplified on the data logger PCB using an instrumentation amplifier (INA2141U, Texas Instruments), and sent through the commutator to control the motorized commutator to counteract the rotation.

4.2.1.3 Signal processing and acquisition

Upstream from the motorized commutator, the electrode signals pass through further filtering and amplification before they are recorded. First, the neural signals go through a high-pass filter to attenuate signals below 300 Hz. In this high-pass filter stage, the neural signals are also amplified by 10, to give the system a total gain of 1000. Next, the neural signals go through a low-pass filter with a cutoff frequency of 15 kHz for anti-aliasing before finally going to the data acquisition system (PCle-6351 and BNC-2090A, National Instruments), which samples the signals at 40 kHz. The synchronization signal goes directly to a data acquisition system without any filtering or amplification.
4.2.1.4 Video recording

For a subset of recordings, simultaneous video was recorded to validate our subsequent classification attempts. Video was captured at 1280x720 pixels resolution, 120 frames per second by a GoPro Hero 4 camera.

4.2.1.5 Synchronization

Synchronization is a challenge because I have recordings from two or three sources: neural data on a computer, movement data on the microcontroller, and sometimes video on a battery-powered camera. Each of these recording devices has its own clock, and the clocks need to be synchronized to analyze all the signals together. My strategy for synchronization is to have one component, the microcontroller, generate a synchronization signal that is recorded by the others. Then, after recording, I shift and stretch each source such that the synchronization signals match. Most of the time, the synchronization signal is low (0 V), but goes high (3.3 V) for a 30 microsecond pulse at random, on average once per second with a refractory period of \(1/15\) seconds. This signal is sent through the commutator and is recorded alongside the neural data.

For video recordings, I used two different methods to record the synchronization signal. In some recordings, I put the synchronization signal through a speaker amplifier connected to a loudspeaker next to the cage. Each synchronization pulse made a click sound through the speaker that was recorded by the microphone in the video camera. In other recordings, I used the synchronization pulse to drive an LED that is in the field of view of the camera. For this strategy, the synchronization pulse alone was too short to be captured by the camera. To ensure that the camera could see the pulse, I elongated each pulse to 33 ms using a Master-8 programmable pulse stimulator.
After recording, I aligned the sync pulses to a reference clock, which I arbitrarily chose to be the clock on the neural recordings. Alignment was performed by a linear regression between reference synchronization times (from the neural recordings) and their corresponding times in the other data source. Due to errors in the synchronization pulse detection process, finding which sync pulses in one source correspond to the pulses in another source is not straightforward. The matching algorithm must be robust to missing and extraneous sync pulses.

I matched corresponding synchronization pulses from different data sources by taking advantage of the random intervals between pulses. Even though each interval between sequential pulses is random, there are so many pulses that the duration of one interval is not informative enough to get a definitive match. Instead, I considered a sequence of multiple (usually four or five) consecutive intervals together as a sort of barcode for one moment in the recording. For each of these barcodes in the reference, I found the best matching barcode in the other dataset. If the match was good, i.e. if the sum of the absolute differences between the intervals in the sequence was below a preset threshold, then the sync pulses from these two barcodes were matched.

4.2.2 Surgical methods

Surgical methods were the same as the previous chapter.

4.2.3 Neural recordings

All neural recordings were performed with three 3-MΩ platinum-iridium electrodes (PI20035.0A3, MicroProbes for Life Science). All single units were verified to be in A1d by a combination of stereotaxic coordinates, tonic activity patterns, and orthodromic response to electrical stimulation in LMANshell. Similar to the A1d units recorded in the previous chapter, these neurons have mean firing rates that are broadly distributed between 0 and 30 Hz (Figure 4.3A).
4.2.4 Computing the spike-triggered average

To analyze the relationship between neural activity and the raw gyroscope signals, I computed the spike-triggered average. The spike-triggered average was computed for each neuron as the mean gyroscope signals around spikes. First, I computed an up-sampled version of the gyroscope signals with a 1 kHz sampling rate as a vector, \( g(t) \), by linearly interpolating the original 200 Hz sampled signal. Next, I binned the spike times into 1 ms bins corresponding to the samples of \( g(t) \). The resulting binned spike count, \( n(t) \), is one when there is a spike in the bin at time \( t \) and zero otherwise. Due to the small bin width, there was never more than one spike per bin. Using the spike counts \( n(t) \) and the gyroscope signals \( g(t) \), I computed the spike triggered average, \( \phi \), as follows:

\[
\phi(\tau) = \frac{1}{N} \sum_{t} n(t) (g(t - \tau) - \bar{g})
\]

where \( N \) is the total number of spikes, \( \bar{g} \) is the average gyroscope signal, and \( \tau \) is the lag between spike and gyroscope signal.

The significance of the spike-triggered average was computed by comparing the observed extrema to a null distribution of extrema calculated using a Monte Carlo procedure. The spike trains were circularly shifted by a random interval to destroy any relationship between spikes and movement, and a new spike triggered average was computed on the shifted data. The observed extrema of the spike triggered average was statistically significant if it was more extreme than 95% of the extrema from the null distribution.
4.2.5 Movement clustering

Given the phasic nature of the movements, it is natural to segment the IMU signals into movements. The segmentation was performed based on the net angular speed, $|g|$. First, stationary periods were detected when $|g|$ remained below 120 degrees per second for 75 ms. Periods where the bird is not stationary for 150–450 ms are movements (Figure 4.5A).

I analyzed these movements using an algorithm inspired by Berman et al. (2014). First, I constructed a high-dimensional vector of features from each movement. Then, these high-dimensional features were embedded in a low-dimensional space using t-SNE. Finally, the embedding was segmented into different movement classes using a watershed transform. The primary difference between the algorithm in this chapter and the algorithm developed by Berman et al. is in the construction of the feature vectors. Berman et al. computed features in a moving window that passed over all of the movement data, but I only compute one feature vector for each movement. I discard the stationary periods for this analysis.

For each movement, I build a high-dimensional feature vector based on the IMU readings during the movement. The accelerometer and gyroscope signals for each movement are extracted (Figure 4.5B) and each is padded with zeros on the edges to make each 90 samples (450 ms) long (Figure 4.5C). These six padded segments are then concatenated to make a single 540-dimensional vector for each movement (Figure 4.5D).

Finally, I used the same strategy as Berman et al. for classification: embedding using t-SNE followed by clustering using a watershed algorithm. We embed each feature vector into a two-dimensional space using t-SNE (Figure 4.6A). This is a nonlinear dimensionality reduction technique that preserves local structure. That is, if points are nearby in the high-dimensional space, then they will also tend to be nearby in the embedding, but if they are far apart in the high-
dimensional space, then their relationship is not very constrained in the embedded space. Performing this embedding on N points requires calculating all N^2 pairwise Euclidean distances between them in the high-dimensional space. To perform t-SNE, we used the implementation in scikit-learn (Pedregosa et al., 2011) which uses Barnes-Hut approximation to reduce the computational complexity of finding an embedding (Van Der Maaten, 2014).

Next, we clustered the movements in the 2-D embedded space using a watershed algorithm. The embedded movements were transformed into a 2-D histogram by dividing the embedded space into 200 x 200 pixels. The counts of the histogram were then smoothed with a 2-D Gaussian distribution to make a heat map (Figure 4.6B). Each of the local maxima of this heat map was taken as the center of a cluster and nearby bins were added to cluster of the nearest uphill maxima using a watershed algorithm. This segmented the embedded space into about 100 different movement classes (Figure 4.6C). After clustering, I watched exemplar videos from each class and labeled each as a head turn, hop, peck, or other. Interestingly, even though the sensor was attached to the head, the algorithm was able to classify movements that involve the whole body, like hops.

4.2.6 Analysis of firing patterns during movements

To analyze the firing patterns during movement clusters, I used the spikes from a window around each movement. For each movement, I extracted the spikes from a window starting 200 ms before the start of the movement and ending 200 ms after the end of the movement. Since movements had different durations, the windows were different sizes for each movement. To calculate the mean firing rate of a cluster, I took the mean of the mean firing rates of its constituent movements. Only movement classes with at least 10 occurrences during the neural recording were included in this analysis.
Using the mean firing rate during each movement might underestimate the modulations if there are both increases and decreases in the firing rate over the duration of the movement. As an alternative measurement of the size of the modulations, I computed the standard deviation of the peri-stimulus time histogram (PSTH). I constructed a PSTH using the spikes from 200 ms before the start of the movement to 500 ms after the start of the movement. This window is the same for every movement, but it may extend up to but only 50 ms after the longest movements. The PSTH bins are 50 ms in duration. Only movement classes with at least 10 occurrences with neural data were included in this analysis.

4.3 Results

To quantify head and body movements and their relationship to brain activity in A1d, we developed a new microdrive to record single unit activity and movement of the head in freely behaving birds. Subjects were tethered by a thin flexible cable while the microdrive records neural data from three electrodes and movements with a nine-axis IMU. The IMU includes an accelerometer, which measures acceleration in three orthogonal directions (Figure 4.1B), and a gyroscope, which measures angular velocity around the same axes (Figure 4.1C). The gyroscope signals are all approximately zero when the bird is stationary, interspersed with brief excursions as the bird makes small head movements. Occasionally there is a longer excursion representing a hop or other complex movement. These movements are rhythmic: the power spectrum of the net angular speed is periodic with a peak frequency of 4.54 Hz (Figure 4.2). The accelerometers show a similar pattern of punctuated equilibrium, except the signal is nonzero when the bird is stationary, due to the acceleration due to gravity.
4.3.1 Relationship of brain activity to motion

Single units in A1d have activity that is correlated with the movement signals by several different metrics. First, their spikes are coherent with the rhythm of movements (Figure 4.3E). For 13 of 29 single units, the coherence between the spike train and the angular speed of the IMU has a significant peak. Second, the spike-triggered average of the gyroscope signals shows a significant peak in 18 of 29 single units. An example of a significant spike triggered average is shown in Figure 4.3B-D and Figure 4.4.

If the neurons in A1d are premotor, the spikes should precede movements. I estimated the lag between spikes and movement by finding the lag of the peak of the spike triggered average. For most of the neurons, however, the peak in the spike triggered average gyroscope signals occurs before the spike (Figure 4.4F), as if the movements are preceding the spikes.

4.3.2 Classification of movements

Next, I investigated the possibility that the neural activity is modulated during specific types of movements. Based on the results from video annotation in the previous chapter, we expect that single units will be selective for head movements in particular directions and to high level behaviors like pecking and grooming. To analyze these correlations, I developed an automatic classification algorithm to identify and cluster movements (see Section 4.2.5). As part of the classification process, each movement was represented by one point in a low-dimensional space (Figure 4.6A). Similar movements were clustered together by performing a watershed transform on the smoothed density of movements (Figure 4.6B–4.6C). At the end of this process, tens of thousands of movements were clustered into about 100 different categories. Each category was labeled as hopping, head movement, pecking, or other by a human observer (Figure 4.6D).
performed this classification separately for each bird, since the IMU was mounted in a slightly
different position on each.

4.3.3 Relationship of brain activity to movement classes

In addition to being coherent with the IMU signals, many single units had activity related
to our different movement classifications. We computed a mean firing rate for each movement
(see Section 4.2.6). To visualize the relationship between our movement classes and single unit
activity in A1d, we colored the movement maps by mean firing rate of the neurons (Figure 4.7),
where red represents a firing rate above the global mean firing rate and blue represents a below-
average firing rate. Movements classes with fewer than 10 movements with neural data were
excluded from this analysis and are represented with gray outlines in the map. Some neurons have
higher activity in one or a few specific movement classes, while others have higher or lower
activity across a wide swath of adjacent movement classes (Figure 4.7). Overall, 15 of 29 neurons
have a significantly different pattern of firing rates across movement classes (p < 0.05, Kruskal-
Wallis test, Bonferroni correction for multiple comparisons).

This measurement of mean firing rate is a crude and incomplete description of the neural
activity during a movement. For example, this analysis would not discover a significant
modulation in a neuron that responds strongly to a movement by first increasing then decreasing
its firing rate, if these modulations averaged out to be near the baseline firing rate. To address this,
we also examined the standard deviation of the peri-stimulus time histogram (SDPSTH) during a
movement. We colored the movement maps by the SDPSTH during each movement (Figure 4.8).
Overall, 16 of 29 neurons have an SDPSTH that was significantly different across the movement
classes (p < 0.05, Kruskal-Wallis test).
These movement maps demonstrate that many neurons are sensitive to movements, in agreement with our results of the raw gyroscope signals (Section 4.3.1). In measuring the activity during each movement class independently, I neglected the fact that nearby movement classes in t-SNE space may have similar gyroscope and accelerometer readings. Indeed, from our movement map results in Figure 4.8, it appears that some neurons have similar activity levels for a broad swath of adjacent movement clusters. In these cases, the spike-triggered average model may be a more parsimonious explanation of the movement-related activity we have observed. To compare the STA model with the movement map model, we predicted the mean firing rates during each movement based on the STA model and compared this prediction to the measured firing rate during each movement class (Figure 4.9). Overall, the STA model predicts a modest fraction of the variance in the results from the class-by-class movement analysis.

4.4 Discussion

In this chapter, I expanded on the discovery from the previous chapter that the activity of single units in A1d is modulated during head and body movements. I confirmed these results in a new set of recordings that also used a more quantitative and objective classification of zebra finch behavior. To measure movements, I developed a new microdrive with an inertial measurement unit to capture the rotations and acceleration of the head. I segmented the raw motion signals into discrete movements, and I developed an algorithm to classify the movements into about 100 different movement categories, including head movements, pecking, and hopping. I discovered that single units in A1d had different activity in particular classes of movements. An analysis of raw gyroscope signals without segmenting into movements confirmed this relationship. These results support the hypothesis that A1d is related to head and body movements, but the exact
contribution of A1d to movement planning and execution is still an open question. I address this topic in Chapter 5.

4.4.1 Future directions for automated behavior analysis

In this chapter, I developed an algorithm to classify a wide range of zebra finch behaviors, but the classification here is far from complete. Future studies can expand upon this work by adding the ability to record and analyze a wider range of zebra finch behavior. Most notably, I have not combined my movement analysis with song recording. Simultaneous recording of head movements and singing would address a number of open questions. First, such recordings would supplement existing studies of “dancing” during song production by providing a quantitative description of dancing movements. Measurements of movement during singing is also important to fully resolve the roles of A1d in vocal error processing versus head and body movements. Birds may move differently after hearing a loud burst of white noise, for example they may flinch or stop moving momentarily. These small changes in movement may explain the small effect of noise bursts on the activity of A1d neurons that I found in Chapter 3. Quantitative measurements of movements during singing in the presence of noise bursts could be used to build a single unified model of A1d activity as a function of both vocal errors and motion.

Another limitation in the current study is that the experimental apparatus prevents the zebra finches from engaging in their full behavioral repertoire. By design, the recording setup prevents the birds from flying so that they do not become tangled in the thin flexible cable that connects them to the rest of the recording apparatus. For similar reasons, the birds are kept in isolation during the experiment, and therefore they cannot express many of their social behaviors. To allow flight and social interaction, the movement recording system would have to be untethered. Such a device would be similar to existing recording devices with a microphone and accelerometer that
are carried like a backpack by adult zebra finches (Anisimov et al., 2014). However, the device would need to be modified to also record neural data.

Other features of zebra finch behavior are outside what is available to head-mounted inertial sensors. For example, my microdrive could not detect chewing. In Chapter 3, I found A1d neurons that were correlated with chewing, but I did not analyze chewing responses in this chapter because I could not detect any inertial sensor patterns associated with chewing. Manual examination of video and corresponding gyroscope signals during chewing confirmed that there were no changes in the gyroscope signal. To capture chewing, additional instrumentation would be necessary, such as a video camera or other sensor that can track the beak.

In general, motion can be measured and classified on a wide range of levels, each of which informs our understanding of motor coding. For example, chewing can be further broken down into a complex sequence individual muscle movements:

Crop samples show that every seed is dehusked before swallowing. Slow motion analysis of video recordings of domesticated birds shows that the mechanical aspects of dehusking are complex (Nelson 1993). On grasping the seed, the eyelids close for an instant. Next, the tongue positions the seed about a third of the way up from the tip of the bill on one side of the mandible. The tongue rotates the seed so that the margins of the lemma and palea, the two scales that form the husk, contact the edges of the upper and lower mandibles. While held in place with the tongue, the two mandibles then close on the seed. Both mandibles have rapid movements with vertical and lateral components and act in synchrony so that resulting shearing forces first crack the seal between the husk and the kernel, and subsequently, further dexterous mandibulations force the edge of the upper mandible down between the lemma and palea and strip them away from the kernel. The husk is allowed to fall from the bill. Positioning of the seed and the mandibular movements vary according to the type of the seed and its size. (Zann, 1996)

Other movements that I analyzed, like hopping and pecking, are likely also composed of a precise sequence muscle activations in the leg, wing, and back. It is possible that analyzing motion on the muscle level might lead to a more parsimonious explanation for A1d activity. At the other extreme, extended sequences of movements can be classified based on their goals or endpoints, like pecking
at a target, or hopping to a perch. The classification algorithm I developed in this chapter operates at just one level of this spectrum. Some algorithms can capture multiple levels automatically through hierarchical classification (Vogelstein et al., 2014). One avenue for future research is to compare the neural activity to multiple levels of movement classifications to find which level provides the most accurate explanation for the neural activity. However, this approach might just lead to a drawn out debate over small differences in decoding performance. For example, in the primate literature, a decades long debate remains unresolved about whether single neurons in primate motor cortex are encoding muscle activity, reach direction, or something else (Churchland et al., 2012; Evarts, 1968; Georgopoulos et al., 1986; Kakei et al., 1999; Kalaska, 2009; Mussa-Ivaldi, 1988). Ultimately, an understanding of how movements are represented in one brain region may require comparative studies of the full network of brain areas controlling movement.
Figure 4.1: Microdrive with integrated IMU. (A) Schematic of microdrive electronics. Top: Electronics to amplify, filter, and digitize neural signals. Three electrodes are amplified on the microdrive by an instrumentation amplifier. The amplified signals are relayed through a thin, flexible cable to a signal processing apparatus that filters and digitizes the signals. Bottom: IMU is part of the microdrive and communicates via I2C over a long, thin cable. The IMU data is stored by a microcontroller on an SD card. The IMU circuit is electrically isolated from the neural recording circuitry. (B) Three orthogonal axes for measuring acceleration. X (green) is positive towards the beak, Y (blue) is positive towards the ventral side of the head, and Z (red) is positive to the left of the head. (C) Three orthogonal rotations measured by the gyroscopes on the IMU. A positive X rotation is a roll of the head to his right. A positive Y rotation is a yaw or sideways head turn to the right. A positive Z rotation is pitching the beak down. (D-E) Photographs of microdrive, viewed from the top (D) and bottom (E).
Figure 4.2: Power spectral density of the net angular speed for one example bird.
Figure 4.3: Characterization of firing patterns of Ald neurons and relationship to gyroscope signals. (A) Histogram of mean firing rates of all Ald neurons recorded for this chapter. (B-D) Spike triggered average of gyroscope signals for one neuron. Extrema outside of the gray bands are statistically significant. (E) Coherence between spike train and the net angular speed of the head. These are significantly coherent between 2-6 Hz. (F) Latency to peak of spike-triggered average for all neurons recorded in Ald. (G) Trajectory of example spike-triggered average in panels B-D. An arrow is shown every 250 ms to mark the roll angle of the head. (H) Power spectral density of the net angular speed.
Figure 4.4: Spike triggered average of gyroscope signals from six example units for which the most data was recorded. Extrema outside of the shaded region are statistically significant.
Figure 4.5: Construction of feature vectors for movement classification. (A) Example IMU signals segmented into movements (shaded regions). The subsequent panels show the construction of the feature vector for the last movement, highlighted in yellow. (B) The three accelerometer and three gyroscope signals from this movement are extracted. (C) These IMU signals are padded with zeros (gray) at the edges until they are each 90 samples long. (D) The padded IMU signals are concatenated into one long feature vector.
Figure 4.6: Embedding and clustering of movements for one example bird. (A) 2-D embedding of all movements from one bird using t-SNE. Each point represents one movement. (B) Heatmap of the embedding. (C) Heatmap with cluster boundaries superimposed. The cluster boundaries are determined by watershed algorithm from all of the local maxima of the heat map. (D) Labels of each movement cluster, as determined by a human expert.
Figure 4.7: Mean firing rate during each movement cluster for six example neurons. Each cluster is colored by its mean firing rate in a window from 200 ms before movement onset to 200 ms after movement offset. Red indicates a firing rate above the mean firing rate of the neuron, and blue indicates a firing rate below the mean. Clusters outlined in gray occurred fewer than ten times during the recording of the neuron.
Figure 4.8: Standard deviation of the peri-stimulus time histogram of each movement cluster for six example neurons. Clusters outlined in grey occurred fewer than ten times during the recording of the neuron.
Figure 4.9: Comparison of spike-triggered average model against the results from movement clustering. Each panel shows the results from one single unit in Ald. Each point represents one movement cluster. On the x-axis is the predicted standard deviation of the PSTH (SDPSTH) of each movement cluster based on the gyroscope signals of each movement in that cluster. On the y-axis is the observed SDPSTH of each movement cluster. Each plot is fit with a linear regression.
5 Discussion

At the beginning of this thesis, I constructed a computational model of song learning that explains several surprising behavioral results from the vocal learning literature. One key component of this model is a dopaminergic reward prediction error signal from VTA. While this signal has been observed in VTA neurons, its origins are not fully understood. I investigated one potential source of the error signals in VTA, the cortical brain area Ald. The role of Ald in song learning is controversial. The existing literature is split between two competing hypotheses for the function of Ald. The first hypothesis is that Ald is involved in processing vocal errors signals that are relayed to VTA. The second hypothesis is that Ald is involved in controlling the movements of the head and body. In Chapter 3, I directly tested these hypotheses by recording from Ald in awake birds. I found that single neurons in Ald increased their firing rates during specific head turns and other movements. In Chapter 4, I performed a more quantitative analysis of the relationship between Ald neurons and movements. I developed a hardware system to record movements with inertial sensors alongside the activity of single neurons. I also created an automatic movement classification algorithm to cluster movements into distinct classes. This second study confirmed that many single units in Ald are activated during specific types of movements.

My recordings from Ald support a hypothesis that Ald and its network of connected brain areas form a network for motor control of non-vocal behavior. This hypothesis was first suggested by Feenders et al. (2008) after observing that Ald and several other brain areas had high immediate early gene expression after hopping but not after singing. Subsequent studies of the expression patterns of a wider set of genes provided additional evidence that the song system and the
surrounding parallel pathway are closely related to the motor cortex in mammals (Pfenning et al., 2014). My work is the first electrophysiological study to corroborate the role of this brain network in motor control. Future studies will need to record from other brain areas in this network to determine their respective roles in controlling movements.

5.1.1 Movement-related signals in the brain

Movement-related signals are found in a wide variety of brain regions. As expected, some of the regions with movement related activity are directly upstream from motor neurons. These neurons are in layer V of primary motor cortex, where some neurons have activity during and immediately before movements (Economo et al., 2018).

Motor-related signals have also been observed in brain regions that are further removed from motor output. In premotor cortex, which is further upstream from the motor outputs, the motor related signals occur earlier before the onset of movement (Alexander and Crutcher, 1990b). Parietal cortex encodes the spatial locations of goals for reaching movements (Andersen et al., 1997).

Movement-related signals have also been observed in brain regions that are far removed from the motor outputs. The mere presence of a correlation between neural activity and movement is not taken as evidence that a neuron is directly involved in motor control. For example, neurons in the subiculum are correlated with running speed (Anderson and O’Mara, 2004). Even in primary visual cortex, some neurons are correlated with running speed (Dipoppa et al., 2018).

Where does AID fit into these movement-related brain areas? Anatomically, AID neurons are similar to the output neurons in layer 5 of primary motor cortex. Like pyramidal neurons in primary motor cortex, AID neurons project to the reticular formation and other midbrain motor structures. Furthermore, both AID neurons and neurons in primary motor cortex have activity that
persists throughout a movement. By analogy, the inputs to AId may correspond to the premotor cortex.

5.1.2 A hypothesis for descending motor control

One hypothesis for this network is that AId and its connected brain areas form a hierarchical circuit for movement planning and control. Based on my recordings, AId is most active during movement execution, not before movement onset. This temporal pattern is similar to what has been observed in medulla-projecting neurons in the primary motor cortex of mice (Economo et al., 2018). Earlier stages of movement planning might take place in the brain areas immediately upstream from AId: dNCL and LMANshell. These areas would have changes in their firing patterns before AId and before movement onset. They might also represent more abstract features than the movement representation I observed in AId. For example, LMANshell and/or dNCL may represent the goals of movement, like the location of a perch, water bowl, or seed. Under this hypothesis, the high-level representations in dNCL/LMANshell are translated into a specific sequence of motor subroutines in AId. Finally, AId coordinates motor primitives in the reticular formation and optic tectum to produce the desired movement. These predictions could be tested with single neuron recordings in zebra finch dNCL and LMANshell.

My recordings from AId fit into a framework for motor control from other species of birds. Other avian brains have similar structures to AId, dNCL, and LMANshell with matching locations and connectivity. Just like AId in zebra finches, part of the arcopallium in other bird species has also been associated with motor control. This region of the arcopallium does not have a unique name across all bird species, but for the sake of simplicity, I will call it sensorimotor arcopallium (SA). Like in zebra finches, SA in other bird species also projects to midbrain motor structures like reticular formation and optic tectum (Zeier and Karten, 1971). Lesions of SA causes akinesia
and difficulty eating in zebra finches (Mandelblat-Cerf et al., 2014) as well as pigeons (Zeier, 1971).

or the same reason, I will refer to the network of brain areas connected to SA, including parts of the basal ganglia, NCL, and other parts of nidopallium, as the forebrain motor network. Within this network, the most extensively studied brain area is NCL.

NCL has been studied extensively because anatomical and neurochemical evidence suggests that it may be analogous to the prefrontal cortex (PFC) in mammals. Like the PFC, NCL receives dense enervation from the dopaminergic midbrain (Divac and Mogensen, 1985; Metzger et al., 1996; Wynne and Güntürkün, 1995) as well as inputs from a wide variety of sensory areas (Kröner and Güntürkün, 1999; Leutgeb et al., 1996). This connectivity led to the hypothesis that NCL could have similar functions as the mammalian prefrontal cortex (Divac et al., 1985; Güntürkün, 2005).

Recordings from NCL during a wide variety of behaviors support the idea that NCL is involved in decision making and memory. In delayed response tasks, where birds must remember a visual cue for a delay period before responding, single neurons in NCL remain active during the delay period (Diekamp et al., 2002; Lengersdorf et al., 2014; Rose and Colombo, 2005; Veit et al., 2014). In crows, NCL neurons encode other cognitive variables such as abstract rules (Veit and Nieder, 2013) or the number of items present on a screen (Ditz and Nieder, 2015).

These high-level representations may be used to trigger the sequential execution of motor programs in SA. For example, in targeted pecking tasks, NCL may coordinate a series of orienting head movements towards the target before initiating a peck at the correct time to receive a reward. Consistent with this idea, lesions of NCL specifically disrupt the timing of pecking in these tasks without affecting the pecking movement itself (Helduser and Güntürkün, 2012). After NCL lesions,
pigeons still peck at the same rate, but their pecks are either not directed at the target, or come at the wrong time in the trial.

Future studies can clarify how cognitive and timing signals from NCL reach the sensorimotor arcopallium. In zebra finches and other bird species, the sensorimotor arcopallium is only enervated by a small region of NCL, and neurophysiological recordings in NCL were not specifically targeted to the SA-projecting region. To understand the information flow from NCL to SA, future studies will need to record from the SA projection neurons in NCL.

5.1.3 Relationship with the song system

AId, LMANshell, and dNCL are part of a brain network that has many anatomic similarities to the song system. In the song system, the motor pathway consists of HVC in the dorsal nidopallium and RA in the arcopallium. Similarly, the parallel pathway includes dNCL in the dorsal nidopallium and AId in the arcopallium. The nuclei of the anterior forebrain pathway (AFP) of the song system also have counterparts in the parallel pathway: LMAN is surrounded by LMANshell, Area X is surrounded by MSt, and DLM is next to DTZ. Given these anatomical similarities, it is tempting to map the functions of the song system onto their counterparts. However, the functions of the parallel pathway nuclei are unknown, at least in zebra finches. Studies of the corresponding brain areas in other bird species, particularly pigeons, has given some insight into the functions of two of these nuclei in particular. In the next few paragraphs I compare the function of HVC and LMAN in the song system to their counterparts in the pigeon, NCL and NIML.

Functional evidence suggests that NCL and HVC have different roles in controlling motor output. HVC controls the stereotyped structure of the song at all timescales. Lesions to HVC completely abolish stereotyped singing (Aronov et al., 2008), and local cooling of HVC slows the song throughout its entire duration (Long and Fee, 2008). The activity of projection neurons in
HVC are distributed across the duration of the song (Lynch et al., 2016; Picardo et al., 2016). By contrast, NCL neurons are most active during stimulus and memory periods preceding movement, but much less active during the movements themselves (Kirsch et al., 2009; Starosta et al., 2013). Furthermore, lesions of NCL do not abolish motor behaviors like pecking (Helduser and Güntürkün, 2012). These results suggest that the details of head and body movements are coordinated by activity in brain areas downstream from NCL.

Preliminary evidence also suggests that LMAN and its counterpart NIML have different functions. Lesions of LMAN in juvenile birds eliminate vocal variability and prevent song learning, but have very little effect on a well learned adult song. However, lesions of NIML impair the execution of a well-learned visually guided pecking task. In this task, pigeons peck keys in a sequence, as guided by a visual cue that appears on the next key to be pecked. After learning this task, NIML lesions decrease the success of pigeons in this task, similar to lesions of NCL (Helduser and Güntürkün, 2012). Unlike lesions of LMAN, which reduce variability, lesions of NIML seem to increase variability by causing more extraneous pecks in this task. In a second experiment using a simulated foraging task, where pigeons had to peck to discover a randomly placed hotspot that led to food delivery, NIML lesions did not affect the number of pecks, nor did the lesions affect measures of pecking variability such as the direction and speed (Helduser et al., 2014).

5.1.4 Comparative approach

In the ongoing study of motor systems, comparison of motor control circuits from different species may provide unique insights by taking advantage of the unique advantages of each species. For example, in the previous section, I drew on studies on saccade in owls and trained pecking in pigeons to gain insight into the possible functions of regions LMANshell and dNCL in zebra finches. Typically, comparative studies involving songbirds focus on the song system as a model
for skill learning. The song system and the brain circuitry underlying motor skills in other species share interesting anatomical features, such as direct connections from cortex to motor neurons (Lemon, 2008; Wang et al., 2015). Similarly, finding differences between circuits for complex motor skills and more general-purpose motor circuits may provide insight into the unique adaptations required to control complex movements.

The songbird contains a specialized circuit and its more general-purpose counterpart in the same brain. In fact, the song system is hypothesized to have evolved as a specialization of the more general purpose motor circuit (Chakraborty and Jarvis, 2015; Feenders et al., 2008). This provides a unique opportunity to compare a specialized circuit for song learning with its closest evolutionary relative, the songbird motor system. Any differences in the function of these two networks would provide a potential feature that is unique to specialized skill learning systems. My thesis provides a first step in this direction. For example, Ald neurons lack the extreme periods of high frequency bursting that are characteristic of RA neurons in the song system. Perhaps the bursty nature of RA activity is a specialization for controlling a behavior that requires very high temporal precision. Future physiological studies of Ald, LMANshell and dNCL could discover more features that are unique to skilled motor learning.

5.1.5 Organization within Ald

Another avenue for future research is to dissect the internal organization of Ald. The anatomy of Ald suggests that different parts of Ald may have slightly different functions. The connections between Ald and other regions in the cortex and basal ganglia are all arranged topographically. This layout could form many parallel circuits for controlling motion of different parts of the body.
The functional topography of A1d could also be addressed through causal experiments. Electrical or optogenetic microstimulation in A1d may evoke different movements, depending on the location of the stimulation. The SA has already been mapped in this fashion in owls. Electrical stimulation of this part of the owl arcopallium evokes saccadic head and eye movements, with the direction of the saccade dependent on the location of stimulation (Knudsen et al., 1995). Electrical stimulation of the midbrain targets of the arcopallium, the optic tectum and the reticular formation, can also evoke saccadic movements. In other bird species, stimulation of the midbrain reticular evokes beak movements and wing flaps (Dubbeldam, 1998; Dubbeldam and Den Boer-Visser, 1994). These results are similar to experiments in mammals, where delivering trains of stimulation lasting hundreds of milliseconds to motor cortex evoked entire actions from the natural behavioral repertoire, like chewing and reaching (Graziano, 2016). In some ways, this view of motor cortex as a map of ethological behaviors disagrees with the more traditional view of motor cortex as a myotopic map.

The movement classification algorithms I developed in Chapter 4 can be used to quantitatively evaluate the results of stimulation experiments. If stimulation evokes natural movements, then both natural and evoked movements should fall into the same clusters. An evoked reach should fall in the same cluster as a natural reach in the same direction, as long as the kinematics of the behaviors are sufficiently similar.

5.1.6 Conclusion

I initially recorded from A1d to find vocal error related signals, but instead I discovered neurons that modulated their firing patterns during head turns and movements such as pecking and hopping. Although few studies have examined the motor related activity A1d and its connected brain regions in zebra finches, comparable regions have been studied more extensively in other
avian species. From pigeons to owls, studies of other birds support a hypothesis in which AId controls specific movement and upstream brain areas are involved in planning and sequencing these movements. Future comparative studies of the motor system in different mammal and avian species may provide insight into the functional building blocks that are common to all motor control systems. The songbird provides a unique opportunity to use a comparative approach within a single species by the two motor systems in the songbird brain: the song system and the AId system.
Appendix A  Recording system for neural activity and movements

This appendix describes the design of a recording system for capturing neural activity and head movements simultaneously. Neural activity is recorded using sharp metal microelectrodes. The neural activity is amplified, filtered, and finally recorded on a computer. Movements are measured using a 9-axis inertial measurement unit (IMU), which includes gyroscopes, accelerometers, and magnetometers. The movement signals are recorded by a separate data logger, which writes the IMU signals to an SD card. I designed three major components of this system: microdrive, cable, and IMU data logger. The following sections describe each of these components in greater detail.

A.1  Microdrive

The microdrive is based on a custom printed circuit board (PCB), which holds the sensors for detecting both neural activity and movements. The PCB is 0.02” thick and has four copper layers. The “top” side of the PCB (Figure A.1B, A.2A) holds the IMU with some supporting components as well as the connector for the cable. The “bottom” side of the PCB (Figure A.1A, A.2D) holds the motor assembly to move the electrodes as well as instrumentation amplifiers for the neural signals. The two inner layers have additional electrical connections. The layer closest to the bottom (electrode) side is a ground plane (Figure A.2C). The layer closest to the top (IMU) side has a separate ground plane for the IMU circuitry as well as some additional wiring (Figure A.2B).

The initial design of the microdrive had very large artifacts on the recording electrodes, which I was able to solve by manually modifying the PCB. The artifacts only occurred when the IMU was communicating with the Data Logger. The artifacts disappeared when I gave the IMU
its own power supply, completely separate from the power supply for the instrumentation amplifiers. The power for the IMU comes from three AA batteries attached to the Data Logger. In constructing the microdrive and Data Logger, you will need to make some manual modifications to the PCBs, as described in the instructions in the following sections. Future designs could incorporate these modifications into the PCBs themselves.

To construct the microdrive for the first time, start by populating the PCB with all of the electrical components and the motor assembly. The motor assembly is made of a motor attached to a custom 3D printed holder (Figure A.3). The holder is necessary to prevent the shuttle from rotating when the motor spins. The back surface of the shuttle is in contact with the motor holder, so when the motor spins, the shuttle cannot spin with it and just moves up and down the threaded rod. The complete instructions for building the PCB can be found in Section A.1.1. The PCB only needs to be constructed once because it is reusable, including the motor and all the electronics. For each implant, assemble the electrodes by following the instructions in Section A.1.2. At the end of the experiment, when the drive is explanted, remove the old electrodes and prepare the microdrive to be reused by following the instructions in Section A.1.3.
High-pass filter on electrodes and reference.

Bypass capacitors for instrumentation amplifiers.

Connect this jumper to use AGND as reference.
Figure A.1B: Schematic for microdrive, sheet 2 of 2
<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Purpose</th>
<th>Description</th>
<th>Manufacturer</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>Omnetics connector</td>
<td>14 pin nano, male</td>
<td>Omnetics</td>
<td>A75243-001</td>
</tr>
<tr>
<td>U1</td>
<td>1</td>
<td>9 axis inertial measurement unit</td>
<td></td>
<td>InvenSense</td>
<td>MPU-9250</td>
</tr>
<tr>
<td>U2</td>
<td>1</td>
<td>Voltage regulator for digital</td>
<td>2.5 V linear voltage regulator</td>
<td>Texas</td>
<td>TPS7152S5DCKR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>electronics</td>
<td></td>
<td>Instruments</td>
<td></td>
</tr>
<tr>
<td>U3-U4</td>
<td>2</td>
<td>Instrumentation amplifier</td>
<td></td>
<td>Analog Devices</td>
<td>AD8224</td>
</tr>
<tr>
<td>C1-C2</td>
<td>2</td>
<td>MPU-9250 regulator filter</td>
<td>0402, Ceramic, X7R, 0.1uF, 10%,</td>
<td>Murata</td>
<td>GRM155R71C104KA88D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>capacitor (C1), VDD</td>
<td>16V</td>
<td>Electronics</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>bypass capacitor (C2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>1</td>
<td>MPU-9250 VDDIO bypass capacitor</td>
<td>0201, Ceramic, X7R, 0.1uF, 10%,</td>
<td>Murata</td>
<td>GRM033R71A103KA01D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10nF, 10V</td>
<td></td>
<td>Electronics</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>1</td>
<td>MPU-9250 big bypass capacitor</td>
<td>0502, Ceramic, X5R, 10uF, 20%,</td>
<td>Samsung</td>
<td>CL05A106MP5NUNC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10V</td>
<td></td>
<td>Electro-Mechanics America, Inc.</td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td>1</td>
<td>Regulator output capacitor</td>
<td>0402, Ceramic, X5R, 0.47uF, 10%,</td>
<td>Murata</td>
<td>GRM155R60J474KE19D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.3V</td>
<td></td>
<td>Electronics</td>
<td></td>
</tr>
<tr>
<td>C6</td>
<td>1</td>
<td>Regulator input capacitor</td>
<td>0402, Ceramic, X7R, 0.1uF, 10%,</td>
<td>Murata</td>
<td>GRM155R71E104KE14D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25V</td>
<td></td>
<td>Electronics</td>
<td></td>
</tr>
<tr>
<td>C7-C10</td>
<td>4</td>
<td>Bypass capacitor for instrumentation amplifier</td>
<td>0201, Ceramic, X5R, 0.1uF, 20%, 25V</td>
<td>Murata</td>
<td>GRM033R61E104ME14D</td>
</tr>
<tr>
<td>Name</td>
<td>Qty</td>
<td>Purpose</td>
<td>Description</td>
<td>Manufacturer</td>
<td>Part Number</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>C11-C12</td>
<td>2</td>
<td>Big bypass capacitor for instrumentation amplifier</td>
<td>0603, Ceramic, X5R, 10uF, 20%, 25V</td>
<td>Murata Electronics</td>
<td>GRM188R61E106MA73D</td>
</tr>
<tr>
<td>C13-C17</td>
<td>5</td>
<td>High pass filter for electrodes and virtual ground (with R7-R11)</td>
<td>0201, Ceramic, X7R, 6800pF, 10%, 16V</td>
<td>TDK Corporation</td>
<td>CGA1A2X7R1C682K030BA</td>
</tr>
<tr>
<td>R1-R2</td>
<td>2</td>
<td>I2C pull-up resistor</td>
<td>0201, 10kOhm, 1%, 1/20W</td>
<td>Panasonic Electronic Components</td>
<td>ERJ-1GEF1002C</td>
</tr>
<tr>
<td>R3-R6</td>
<td>4</td>
<td>Gain setting resistor for instrumentation amplifier (gain=100)</td>
<td>0201, 4990hm, 1%, 1/20W</td>
<td>Panasonic Electronic Components</td>
<td>ERJ-1GEF4990C</td>
</tr>
<tr>
<td>R7-R11</td>
<td>5</td>
<td>High pass filter for electrodes and virtual ground (with C13-C17)</td>
<td>0201, 10MOhm, 1%, 1/20W</td>
<td>Rohm Semiconductor</td>
<td>MCR006YRTF1005</td>
</tr>
</tbody>
</table>
Figure A.2A: Layout for microdrive printed circuit board, top layer
Figure A.2B: Layout for microdrive printed circuit board, upper inner layer
Figure A.2C: Layout for microdrive printed circuit board, lower inner layer
Figure A.2D: Layout for microdrive printed circuit board, bottom layer
Figure A.3: Motor holder. (A) Isometric view of holder with motor. (B) Top view of holder. (C) Front view of holder, showing window for electrodes. (D) Cross-section through channel that holds the motor. (E) Side view of holder.
A.1.1 PCB assembly instructions

Materials:

Electronic components from Table 1
Motor (Faulhaber, 0206 A 001 B 021:1 Y2825)
Motor holder (Figure A.3)
Solder Paste (Zephytronics SPE-0012)
Torr Seal epoxy
Five-minute epoxy (Hardman Double Bubble Red)
Reflow oven
Tweezers
Cable (Section A.2)

Instructions:

1. On the analog side of the PCB, put solder paste on all of the solder pads
2. Using tweezers, place components on the analog side
   - U3, U4
   - C7-C17
   - R3-R11
3. Using a reflow oven, solder the components to the analog side
4. Wait for PCB to cool completely
5. On the digital side of the PCB, put solder paste on all of the solder pads
6. Using tweezers, place components on the digital side:
   - X1
   - U1
   - C1 – C5
   - R1, R2
   - Do not place: U2, C6
7. Using a reflow oven, place the PCB digital side up and solder the components. The analog components will stay attached.
8. Visually inspect all of the solder joints and rework if necessary.
9. Modify the PCB to add external power for the IMU. Replace S3 with +2.5V from the Data Logger
   - Unsolder pin the pin F of the Omnetics 14-pin connector and lift up the pin.
   - Solder a wire from this pin to the positive end of C5 (labeled with a yellow dot on Figure A.2A)
10. Test communication with the IMU
11. Test the instrumentation amplifiers
12. With five-minute epoxy, attach motor holder to analog side. Use your hands to keep the motor holder in place until the epoxy cures.
13. Put shuttle on motor
14. With Torr Seal, secure the motor to the holder. Make sure threaded rod is parallel to board
15. Wait for Torr Seal to cure
16. Solder motor to board
17. Test the motor. If direction is backwards, switch the wires soldered to IX and ICOM
A.1.2 Electrode bundle assembly instructions

Materials
Five-minute epoxy (Hardman Double Bubble Red)
Light cure acrylic (FLOW-IT ALC, clear)
Kwik-Cast Sealant (World Precision Instruments)
Very sharp razor blade (Ted Pella, Inc. Product No. 121-4)
Polyimide tubing, 0.0045" inner diameter (A-M Systems Catalog #823400)
Electrodes (MicroProbes, any platinum/iridium monopolar metal microelectrode with diameter < 100 um)
Dummy electrodes (straight pieces of damaged electrodes or leftover electrode shafts)
Lateral positioner screw 0000-160X x 5/32 flat head (Morris Precision Screws Part No. F0000160CE156)
Lateral positioner nut
Wax
Cauterizer
Tweezers
Wire, platinum iridium, 0.002” bare, 0.0040” coated (A-M Systems 776000)
Wire cutters
Solder
Soldering iron
Transparency

1. Make polyimide bundles. These will be the guide tubes for your electrodes.
   a. Using a very sharp razor blade, cut some polyimide tubing into three equal length pieces
   b. Using five-minute epoxy, glue the three pieces together side by side. Use a piece of
      solder to spread the epoxy
   c. Use just enough epoxy to hold the tubes together. No extra!
   d. Wait 15 minutes for the epoxy to cure
   e. With the razor, cut off a 2 mm piece to be the upper polyamide
   f. With the razor, cut off a 3 mm piece to be the lower polyamide
   g. Save the rest for next time
2. Thread dummy electrodes through the upper and lower polyamide tubes. This will keep the tubes
   aligned while you put them in the microdrive.
3. Put polyamides with dummy electrodes in the drive. The upper polyamide tubes should be resting
   on the shuttle and the lower polyamide tubes should be passing through the window at the base of
   the motor holder. Some of the lower polyamide bundle (0.5 to 1.0 mm) should extend through the
   window below the bottom of the case.
4. With five-minute epoxy, secure the upper polyamides to the shuttle. Take care to keep the epoxy
   off the threads of the motor.
5. Wait for epoxy to cure (10 minutes)
6. Secure the lower polyamides to the motor holder using wax
   a. Melt a drop of wax on the tip of the cauterizer and turn off the cauterizer so that the drop
      solidifies
   b. Hold the cauterizer near the lower polyamides and turn it on just long enough to melt the
      wax
   c. Touch the ball of liquid wax to where the lower polyamides meet the window at the
      bottom of the motor holder. The wax will flow onto the holder and solidify, keeping the
      lower polyamides in place. Take care to keep the wax out of the tubes!
7. Remove dummy electrodes
8. Cut a 4 cm length of platinum/iridium wire to be the reference (virtual ground)
9. Using tweezers, put a right-angle bend in the reference wire at least 5 mm from the end.
10. With light cure, attach virtual ground wire to motor holder as close to the polyamides as possible.
11. Assemble the lateral positioner by putting the nut on the screw.
12. With light cure, attach lateral positioner nut to the bottom of the motor holder. Be careful to keep the threads of the screw clean. The tip of the screw should be in contact with the lower polyamide tubes such that turning the lateral positioner screw clockwise pushes the lower polyamide tubes.
13. Cover the threads of the lateral positioner screw with Kwik-Cast between the head of the screw and the nut. This is to keep the light cure acrylic from touching the threads during implant.
14. Calculate the length of the electrodes:
   - 7.5 mm of electrode in the case
   - + 0.5 mm of electrode in the lower polyamides extending below the case
   - + 0.5 mm of electrode between the bottom of the polyamides and the surface of the brain
   - + __ mm of electrode from the brain surface to your target area.
15. Trim virtual ground wire to length and solder to the board.
16. For each electrode, cut it to length (save the extra to use as a dummy electrode) and insert it cut end first through the lower polyamide and then thread it through the upper polyamide.
17. With five-minute epoxy, secure electrodes to the upper polyamides. Take care to keep the epoxy off of the threaded rod.
18. Lower shuttle to 1/3 from bottom. This decreases the chances that you will accidentally touch the plastic body of the motor with the soldering iron.
19. Bend the cut ends of the electrodes into a fan pattern to make soldering easier.
20. Using tweezers, strip the insulation from the electrodes above the upper polyamides.
21. Solder a platinum/iridium wire to each electrode.
22. Solder each wire to a solder pad on the PCB.
23. Cut a cover out of clear overhead transparency (Figure A.4).
24. With dental cement, secure transparency cover to the PCB.

Figure A.4: Transparency cover for microdrive. Fold along dotted lines.
A.1.3 Electrode bundle removal instructions

1. Pull off the transparency cover
2. Using wire cutters, cut the electrodes between the upper and lower polyamide tubes
3. Remove any excess wax from the window at the bottom of the motor holder
4. Using tweezers, pull out the lower polyamide tubes
5. Using a soldering iron, unsolder the electrode wires from the pads on the PCB
6. Using wire cutters, cut the dental acrylic holding the reference wire and lateral positioner in place. Save the lateral positioner for reuse.
7. Using wire cutters, cut the upper polyamide tubes off of the shuttle

A.2 Cable

This section contains instructions for constructing a thin flexible cable to connect the microdrive to the IMU data logger. Electrically, the cable is just a bundle of wires that passively relays the signals and power between the microdrive and the data logger. However, the mechanical features of the cable are must be just right to minimize the discomfort of the bird wearing the microdrive. The cable is braided to increase its stiffness, which prevents it from twisting too much when the bird moves around. The length of the cable must be chosen carefully to be just long enough for full range of motion for the bird. Any extra slack in the cable just adds weight that the bird may have difficulty supporting. A long cable also increases the risk that the cable will become tangled. Instructions for building the cable are below.

Materials:

14-pin connector with lead wires, male (Omnetics Connector Corporation, A7876-001)
14-pin connector with straight pins, female (Omnetics Connector Corporation, A7877-001)
Torr Seal epoxy
Solder
Soldering iron
Wire cutters
Tweezers

Instructions:

1. Darken the side of the connectors with a black permanent marker (Figure A.5)
2. Braid the lead wires
a. Partition the 14 lead wires into three groups. Two groups will have five wires and one
group will have four wires.
b. Braid these three groups together until the braided section is about 250 mm long.
c. Tape the bottom of the braid to hold it in place

3. Cut the lead wires so that the ends of the wires are 2 cm from the floor of the cage when the cable
   is plugged in to the data logger. This length will be different in each recording rig. On my rig, the
   length is 200 mm.
4. Tape the braid above the cut point so that the wires stay braided after cutting
5. Cut the lead wires at the mark
6. Strip the tip of each wire. Place the wire on the bench and press down on the wire with a hot
   soldering iron. The insulation will melt away. Use tweezers to pull off any remaining pieces of
   insulation.
7. On the female connector, trim the silver pins to half their original length and wet them with fresh
   solder
8. Working from one side of the connector to the other, solder each wire to its matching pin. Use a
   multimeter to determine which wire corresponds to each pin on the male connector.
9. Use a multimeter to test the electrical connectivity from one end of the cable to the other. Be sure
   there are no short circuits.
10. Cover the solder joints with Torr Seal epoxy. Use just enough to cover from the plastic of the
    connector to the ends of the solder joints.

Figure A.5: 14-pin connector with letter labels for each contact and dark label on one side.

A.3 IMU Data Logger

The data logger receives the neural and movement signals from the microdrive and sends
them each to separate recording systems. The bottom of the data logger PCB connects to the
microdrive cable, and the top of the data logger PCB connects to the commutator. The pre-amplified neural signals pass through the data logger unaltered, while the movement signals are saved on the data logger.

The Data Logger PCB is designed as an Arduino “shield” for easy attachment to a microcontroller. The schematic for this PCB is shown in Figure A.6 and its layout is in Figure A.7. Once the PCB is constructed, an Arduino can mate with the pins on the back side of the logger. The Arduino (or other microcontroller) communicates with the IMU and saves the movement data.

This board requires major modifications to separate the analog and digital sources. As the board was originally designed, there is only one shared ground that cannot be separated. To get a separate ground, you will take a second PCB and cut off the converter part. Future studies could redesign this data logger with separate power and ground for analog and digital systems.
Table A.2: Electronic components for Data Logger PCB

<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Purpose</th>
<th>Description</th>
<th>Manufacturer</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>1</td>
<td>Bypass capacitor for voltage regulator input</td>
<td>0402, Ceramic, X7R, 0.1uF, 10%, 25V</td>
<td>Murata Electronics</td>
<td>GRM155R71E104KE14D</td>
</tr>
<tr>
<td>C4</td>
<td>1</td>
<td>Bypass capacitor for voltage regulator output</td>
<td>0402, Ceramic, X5R, 0.47uF, 10%, 6.3V</td>
<td>Murata Electronics</td>
<td>GRM155R60J474KE19D</td>
</tr>
<tr>
<td>C5-C6</td>
<td>2</td>
<td>Bypass capacitor for instrumentation amplifier</td>
<td>0805, Ceramic, X7R, 100nF, 10%, 25V</td>
<td>Murata Electronics</td>
<td>GRM21BR71E104KA01L</td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>Current limiting resistor for hall sensor</td>
<td>0805, 10k, 1%, 1/8W</td>
<td>ROHM Semiconductor</td>
<td>MCR10ERTF1002</td>
</tr>
<tr>
<td>R2-R5</td>
<td>4</td>
<td>Voltage divider for electrode signals (with R6-R9)</td>
<td>0805, 357k, 1%, 1/8W</td>
<td>ROHM Semiconductor</td>
<td>MCR10ERTF3573</td>
</tr>
<tr>
<td>U1</td>
<td>1</td>
<td>Hall sensor</td>
<td></td>
<td>Lake Shore Cryonics</td>
<td>HGT-2101-10</td>
</tr>
<tr>
<td>U2</td>
<td>1</td>
<td>I2C level translating repeater</td>
<td>PCA9517</td>
<td>NXP Semiconductors</td>
<td>PCA9517D,118</td>
</tr>
<tr>
<td>U4</td>
<td>1</td>
<td>Voltage regulator, 2.5V</td>
<td>TPS715 fixed</td>
<td>Texas Instruments</td>
<td>TPS71525DCKR</td>
</tr>
<tr>
<td>U5</td>
<td>1</td>
<td>Instrumentation amplifier</td>
<td>INA2141U</td>
<td>Texas Instruments</td>
<td>INA2141U</td>
</tr>
<tr>
<td>X1</td>
<td>1</td>
<td>Connector to commutator</td>
<td>Dale 26 pin connector, male</td>
<td>Vishay Dale</td>
<td>MMP24-26</td>
</tr>
<tr>
<td>X2</td>
<td>1</td>
<td>Connector to bird</td>
<td>16-pin header, male</td>
<td>Amphenol FCI</td>
<td>68602-116HLF</td>
</tr>
</tbody>
</table>
Figure A.6: Schematic for Data Logger

```
12C pull-up resistors

I2C Level shifter

Power supply for IMU

Hall sensor

stabilization of power supply

Amplifier for hall sensor

According to datasheet: "If channel is unused, connect inputs to ground, sense to Vo, and leave Ref open-circuit."

Title: Data Logger
Document Number: REV:
Date: 1/1/19 12:28 PM  Sheet: 1/1

174
Figure A.7A: Layout of Data Logger PCB, top layer

Figure A.7B: Layout of Data Logger PCB, upper middle layer
Figure A.7C: Layout of Data Logger PCB, lower middle layer

Figure A.7D: Layout of Data Logger PCB, bottom layer
Materials

- Electronic components (see Table A.2)
- Solder Paste (Zephytronics SPE-0012)
- Torr Seal epoxy
- 14-pin connector with leads (Omnetics Connector Corporation A8109-001)
- Aluminum stock, 0.05” thick
- Ball bearing 0.1250” x 0.2500” x 0.1094” (Dynaroll SFR144ZZ A5)
- Foam
- Dale connector (Vishay Dale MMP24-26)
- Metal tube that fits inside ball bearing
- Disk magnet, neodymium, 0.5” diameter, 0.125” thick
- Zip ties
- Hookup wire, 30 AWG

Instructions

1. Put solder paste on pads for the instrumentation amplifier and the nearby capacitors (U5, C5, C6)
2. Place U5, C5, C6
3. Reflow
4. Use a soldering iron to attach pin headers.
5. On the other side, use a soldering iron to solder R1 and the hall sensor. The hall sensor stands on its edge
   a. Hold the sensor on its side and solder pins to pads that are closer together
   b. Use silver wire to attach pins in the air to the other pads
6. Cut piece of aluminum 38 mm x 14 mm
7. Bend the aluminum like an “L”. The base of the L should be 12 mm long
8. Drill a hole in the center of the base of the L that is just large enough to hold the ball bearing
9. Use Torr Seal to attach the ball bearing to the hole. Be very careful to keep the epoxy out of the bearings!
10. Put a 3-4 mm spacer between the PCB and the Dale 26-pin connector. Then, Torr seal the Dale 26-pin connector to the PCB.
11. Wait for the Torr Seal to cure
12. Connect pins of Dale 26-pin connector to the PCB using hookup wire
13. Make sure that the ball bearing can spin easily before continuing. If not, you probably got epoxy in the ball bearing. Replace the ball bearing and try again.
14. Torr seal the aluminum L to the PCB. Make sure the ball bearing is centered on the Hall sensor. The inner side of the upright portion of the L should be Torr sealed to the side the PCB that does not have the Hall sensor.
15. Cut a tube 75 mm long
16. Put female 14-pin connector with leads into tube. Pull the lead wires all the way through
17. Use Torr Seal to attach the connector to the end of the tube. Make sure the connector is perpendicular to the tube.
18. Wait for Torr Seal to cure
19. Put tube through ball bearing and Torr Seal it in place. The tube should be parallel to the PCB and pass directly over the hall sensor. The distance from the bottom of the PCB to the bottom of the 14-pin connector should be 60 mm
20. Wait for Torr Seal to cure
21. Solder wires from 14-pin connector to the pin headers on X2, except for wires E, F, and N. Those will be soldered separately later.
22. Torr Seal the disc magnet to the tube, above the hall sensor. When the wires are resting in their natural position, the magnet should be parallel to the PCB.
23. Wait for Torr Seal to cure.
24. From a second PCB, cut out a fragment of the PCB that includes all of the I2C circuitry: U2, U4, C2, C4, R2–R5.
25. Place the components on the PCB fragment: U2, U4, C2, C4, R2–R5.
26. Solder the components by baking the fragment in the reflow oven.
27. Cut the following pins from the connectors that mate with the Arduino on the Data Logger PCB: Vin, GND, 5V, SDA, SCL.
28. Using zip ties, attach the battery pack to the Data Logger.
29. Connect the battery pack to the microcontroller. The positive terminal of the battery should go to Vin, and the negative terminal should go to GND.
30. Connect the level shifter board using hookup wire:
   a. 5V from microcontroller to V+ and +5V on level shifter board
   b. GND from microcontroller to GND on level shifter board
   c. SCL from microcontroller to SCL_RIG on level shifter board
   d. SDA from microcontroller to SDA_RIG on level shifter board
   e. Wire E from bundle to SDA_BIRD on level shifter board
   f. Wire N from bundle to SCL_BIRD on level shifter board
   g. Wire F from bundle to +2.5V on level shifter board
31. Set up wires for sync pulses:
   a. from pin 9 on microcontroller to pin d on Dale connector.
   b. GND from microcontroller to pin a on Dale connector.
Bibliography


182


