Dynamics of Sensory Integration of Olfactory and Mechanical Stimuli Within the Response Patterns of Moth Antennal Lobe Neurons

Harrison Tuckman

Follow this and additional works at: https://scholarworks.wm.edu/honorstheses

Part of the Applied Mathematics Commons, Biology Commons, and the Computational Neuroscience Commons

Recommended Citation
Tuckman, Harrison, "Dynamics of Sensory Integration of Olfactory and Mechanical Stimuli Within the Response Patterns of Moth Antennal Lobe Neurons" (2020). Undergraduate Honors Theses. Paper 1480. https://scholarworks.wm.edu/honorstheses/1480

This Honors Thesis is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Undergraduate Honors Theses by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.
Dynamics of Sensory Integration of Olfactory and Mechanical Stimuli Within the Response Patterns of Moth Antennal Lobe Neurons

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Mathematics from The College of William and Mary

by

Harrison Glen Tuckman

Accepted for __________________ Honors __________________
(Honors, no Honors)

Mainak Patel
Dr. Mainak Patel, Advisor

Sarah Day
Dr. Sarah Day

Greg Conradi Smith
Dr. Greg Conradi Smith

Williamsburg, VA
May 6, 2020
I would like to express my deepest gratitude for my advisor, Dr. Mainak Patel. Without his knowledge, guidance, and vision, this project would not have culminated to what it is today. I would also like to thank all of my professors, who helped me cultivate a curiosity for learning and aided me in converging on my academic goals. Finally, I would like to thank my family and friends, whose continuous support and care helped me to flourish throughout my education.
DYNAMICS OF SENSORY INTEGRATION OF OLFACTORY AND MECHANICAL STIMULI WITHIN THE RESPONSE PATTERNS OF MOTH ANTENNAL LOBE NEURONS

Abstract

by Harrison Glen Tuckman,
William and Mary
May 2020

Advisor: Mainak Patel

Odors emanating from a biologically relevant source are rapidly embedded within a windy, turbulent medium that folds and spins filaments into fragmented strands of varying sizes. Environmental odor plumes therefore exhibit complex spatiotemporal dynamics, and rarely yield an easily discernible concentration gradient marking an unambiguous trail to an odor source. Thus, sensory integration of chemical input, encoding odor identity or concentration, and mechanosensory input, encoding wind speed, is a critical task that animals face in resolving the complex dynamics of odor plumes and tracking an odor source. In insects, who employ olfactory navigation as their primary means of foraging for food and finding mates, the antennal lobe (AL) is the first brain structure that processes sensory odor information. Although the importance of chemosensory and mechanosensory integration is widely recognized, the AL itself has traditionally been viewed purely from the perspective of odor encoding, with little attention given to its role as a bimodal integrator. In this work, we seek to establish the AL as an ideal model for studying sensory integration – it boasts well-understood architecture, well-studied olfactory responses, and easily measurable cells. Experimental studies suggest that mechanosensory responses are transient and temporally
precise, while olfactory responses are long-lasting but lack temporal precision. Within this work, we develop a computational model of the AL that captures these odor response dynamics, and then examine the dynamics of our model with the inclusion of mechanosensory input. Through use of this model, we pinpoint dynamical mechanisms potentially underlying the bimodal AL responses revealed in experimental studies. Finally, we propose a novel hypothesis about the role of mechanosensory input in sculpting AL dynamics and the implications for biological odor tracking.
# TABLE OF CONTENTS

| Chapter | Title | Page |
|---------|-------|------|
| 1       | INTRODUCTION | 1 |
| 1.1     | Background | 1 |
| 1.2     | The Olfactory Pathway | 4 |
| 1.2.1   | Olfactory Transduction Pathway | 4 |
| 1.2.2   | The Antennal Lobe | 6 |
| 1.3     | Experimental Data | 7 |
| 2       | Developing the Model | 11 |
| 2.1     | Model Summary | 11 |
| 2.2     | Model Details | 13 |
| 2.2.1   | The Neuron Model | 13 |
| 2.2.2   | Network Architecture | 16 |
| 2.2.3   | Stimulus Modeling | 17 |
| 3       | Results | 21 |
| 3.1     | Model Dynamics with Olfactory Input in Isolation | 21 |
| 3.2     | Model Dynamics with Mechanosensory Input | 27 |
| 3.3     | Odor Discrimination Dynamics within the Model | 34 |
| 4       | Discussion | 39 |
| 4.1     | Biological Implications and Hypotheses | 40 |
| 4.2     | Sources of Mechanosensory Input to the AL | 42 |
| 4.3     | Future Directions | 43 |
## REFERENCES

APPENDIX

| Section | Title | Page |
|---------|-------|------|
| A | Simulation, Stability, and Data Analysis | 52 |
| B | Varying Pulse Length and Frequency | 54 |
| B.1 | Pulse Length | 54 |
| B.2 | Pulse Frequency | 56 |
| C | The Model Code | 59 |
| C.1 | Parameters | 59 |
| C.1.1 | Parameters.h | 59 |
| C.1.2 | Parameters.cpp | 61 |
| C.2 | PNs | 64 |
| C.2.1 | PN_Neuron.h | 64 |
| C.2.2 | PN_Neuron.cpp | 66 |
| C.3 | LNs | 74 |
| C.3.1 | LN_Neuron.h | 74 |
| C.3.2 | LN_Neuron.cpp | 76 |
| C.4 | Glomeruli | 82 |
| C.4.1 | Glomerulus.h | 82 |
| C.4.2 | Glomerulus.cpp | 83 |
| C.5 | Antennal Lobe | 86 |
| C.5.1 | Antennal_Lobe.h | 86 |
| C.5.2 | Antennal_Lobe.cpp | 88 |
| C.6 | Olfactory Stimuli | 96 |
| C.6.1 | Odor_Pulse.h | 96 |
| C.6.2 | Odor_Pulse.cpp | 97 |
| C.7 | Mechanosensory Stimuli | 100 |
| C.7.1 | Air_Puff.h | 100 |
| C.7.2 | Air_Puff.cpp | 101 |
| C.8 | Auxiliary Functions | 104 |
| C.8.1 | Functions.h | 104 |
| C.8.2 | Functions.cpp | 104 |
| C.9 | Simulation | 105 |
| C.9.1 | main.cpp | 105 |
LIST OF FIGURES

1.1 Schematic Diagram of Olfactory Pathway .................................................. 5
1.2 Triphasic PN response ................................................................................. 7
1.3 Experimental Data ....................................................................................... 10
2.1 Schematic Model of the AL ................................................................. 12
3.1 Model Dynamics ......................................................................................... 22
3.2 Effects of SK on Phase II firing ................................................................. 24
3.3 Mechanism for IHP and AHP ................................................................. 26
3.4 Simulations with Mechanosensory Stimulus ............................................. 29
3.5 Transient Response of Mechanosensory Stimuli ........................................ 32
3.6 Diffuse Odor and Odor Classification ..................................................... 38
A.1 Simulation Stability .................................................................................... 53
B.1 Firing Length Determination ................................................................. 55
B.2 PN Response to Varying Pulse Length .................................................... 55
B.3 AL response to Pulsed Stimuli ................................................................. 57
Chapter One

INTRODUCTION

1.1 Background

Integration of chemosensory input (encoding odor identity and concentration) and mechanosensory input (encoding wind velocity) is critical for olfactory navigation and the ability of insects to home in on and locate an odor source (Mamiya and Dickinson, 2015). One particularly crucial task for an insect is the ability to track an odor source mid-flight. This ability is critical for finding food, but poses a monumental challenge: turbulent wind eddies produce complex patterns of odor strands of different sizes and concentrations intermixed with clear media, obfuscating the odor source and rarely yielding an easily discernible ‘concentration gradient’ to follow (Vickers, 2000; Carde and M.. Willis, 2008; Murlis and Jones, 1981). To successfully track an odor, an insect must simultaneously classify odor identity and rapidly resolve the spatio-temporal dynamics of the odor plume, all while maintaining balance and bearing during flight.

Evidence strongly suggests that the ability of insects to track odors mid-flight relies on the integration of mechanical with chemical input, and, accordingly, chemosensory and mechanosensory bimodality is indeed widespread within the central nervous system of insects. In terms of sensory organs, while Johnston’s organ and Böhm’s bristles on the antenna are well known to detect wind velocity and guide flight maneuvers (Patella and Wilson, 2018;
Jarman, 2002; Sane et al., 2007), there are also less studied bimodal sensilla (Jarman, 2002; Sane et al., 2007). For example, a subtype of trichoid sensilla on the antenna of the male hawkmoth (Lee and Strausfeld, 1990) and sensilla chaetica on the honeybee antenna (Whitehead and Larsen, 1976) exhibit chemo- and mechano-sensory bimodality. A wide range of animals have been shown to be adept at intelligently sampling environmental odor plumes to home in on an odor source (Ando and Kanzaki, 2015; Pasternak, Bartumeus, and Grasso, 2009; Pyk, 2006; Vergassola, Villermaux, and Shraiman, 2007); moths, in particular, employ a strategy in which they surge upwind upon encountering odor strands and cast across wind when losing contact with odors (Vickers and T. Baker, 1994). This suggests that chemosensory and mechanosensory input may augment and corroborate each other, consistent with the close link between odor plume structure and air turbulence; the apparent convergence on this strategy across many species reveals the importance of sensory integration in tracking odor sources.

The antennal lobe (AL) is the first brain area to substantially process odor information arriving from olfactory receptor neurons (ORNs) in the sensory periphery. While the AL has been traditionally thought to function exclusively within the domain of odor (and \( CO_2 \) detection, there exist some data in moths and tadpoles suggesting that AL neurons (in moths) or olfactory bulb neurons (in tadpoles) are actually bimodal (Han, Hansson, and Anton, 2005; Brinkmann and Schild, 2016); recently, a couple of studies in mice have shown that olfactory bulb glomeruli are responsive to mechanosensory input (in the absence of odor) arising from air speed and pressure (Iwata, Kiyonari, and Imai, 2017; K. Baker et al., 2019). Unfortunately, such data have been collected only rarely and sparsely and have received little attention, and the AL has hence been investigated only from the perspective of odor encoding. The AL has therefore not yet been rigorously established as a model system for the study of sensory integration, despite the fact that sensory integration is an important aspect of odor tracking. The AL, however, is an otherwise ideal system for the study of bimodality – specifically, AL architecture is well understood, AL odor response
dynamics are well-studied, and AL neurons are easy to measure individually and in aggregate. Additionally, the anatomy and physiology of the AL is analogous to that of the olfactory bulb in vertebrates (Hildebrand and Shepherd, 1997), and hence may provide insight into chemo- and mechano-sensory integration in a broad range of species.

Experimental studies suggest that the moth AL is indeed acting as a bimodal integrator. Some preliminary data shows that projection neurons (PNs) within the AL respond to both olfactory and mechanosensory signals, and that these two sources of input may influence AL dynamics in starkly disparate ways. These data indicate that strong mechanosensory input enhances the amplitude and temporal precision of AL odor responses (causing a sharp, transient burst in AL activity at odor onset), as well as possibly contributing to the correlational structure of responses across AL neurons.

To examine the dynamics underlying experimental observations, we construct a realistic, large-scale biophysical model of the moth AL consisting of projection neurons (PNs) and local neurons (LNs) organized into glomeruli; using physiologically reasonable model components and parameters, we simulate both olfactory and mechanosensory input to the model. We first probe the responses of our model network to odor input alone, and we not only show that our model exhibits the triphasic odor responses observed experimentally in moth PNs (Christensen and Hildebrand, 1987b; Waldrop, Christensen, and Hildebrand, 1987; Christensen and Hildebrand, 1988; Christensen and Hildebrand, 1990; Christensen, Waldrop, Harrow, et al., 1993), but we also dissect model dynamics to provide plausible mechanistic explanations for how such response patterns arise. We then explore model responses to mechanosensory input alone, and show that the transient, temporally precise nature of responses to mechanosensory signals arises as a consequence of pervasive slow inhibition activated via excitation of the global LN network. Finally, we show that model dynamics are capable of capturing empirically observed response patterns to olfactory and mechanosensory input in combination, and we examine the implications of bimodality for odor discrimination and tracking.
1.2 The Olfactory Pathway

The olfactory pathways is substantially studied, and thus is relatively well understood. This section will outline what is known about how olfactory stimuli are processed, and what the physiology of the involved brain regions, particularly the AL, looks like.

1.2.1 Olfactory Transduction Pathway

Olfactory transduction begins in the antennae, sensory organs which respond to the presence of chemical stimuli comprising distinct odors. The antennae contain thousands of ORNs, which each express one of approximately 60 different odorant receptors. With only 60 unique odorant receptors, odor discrimination relies on a highly combinatorial process; each scent perceived by the antennae stimulates a subset of the odorant receptors, and thus a subset of the ORNs. By varying which subset of olfactory receptors are stimulated, the antennae are able to respond uniquely to approximately $2^{60}$ distinct odors. ORN axons bundle together to form the antennal nerve, and relay information to the first brain structure, the AL (Cazamine and Hildebrand, 1979).

The AL is composed of approximately 60 distinct bundles of neurons called glomeruli. Each glomerulus contains approximately 6 GABAergic, inhibitory LNs, and 10-15 excitatory, output PNs, leading to a total of around 360 LNs and 840 PNs within the entire AL (Homberg, Christensen, and Hildebrand, 1989; Kingan and Hildebrand, 1985; Hoskins et al., 1986; Hildebrand, Hall, and Osmond, 1979; Matsumoto and Hildebrand, 1981; Christensen and Hildebrand, 1987a; Homberg, Montague, and Hildebrand, 1988). Each glomerular subunit responds most strongly to one of the 60 types of ORNs, similarly to how each type of ORN has only a single odor receptor. This means that odor identity is relatively well preserved, as specific glomeruli will respond most strongly to specific odors. The dendritic trees of PNs are typically confined to a single glomerulus, although few PN trees span across multiple glomeruli and thus integrate input from multiple olfactory receptor types (Kanzaki...
et al., 1989). LNs serve to bridge the gap between glomerular units, as they create a diffuse inhibitory network which is able to connect multiple glomerular subunits (Reisenman et al., 2004; Lei, Christensen, and Hildebrand, 2004). The axons of PNs only exit the AL and provide input to the next brain region, the mushroom body.

The mushroom body is the first brain region believed to be responsible for odor association and memory. It is composed of thousands of Kenyon Cells (KCs), which greatly outnumber the number of input PNs. KCs have relatively high depolarization thresholds, and thus exhibit nearly no background firing. Furthermore, these neurons exhibit extreme odor selectivity, and minimal overlapping odor representations. Odor-evoked responses, while highly selective, are extremely limited, consisting of only a few spikes. Any particular odor will excite approximately 10-20% of KCs, and typically does so within the first 500 ms of odor stimulus onset (Ito et al., 2008). This entire olfactory pathways is summarized in the schematic diagram illustrated in figure 1.1.

**Figure 1.1** Olfactory transduction begins on the left with the antennae. The antennae contains ORNs which each express a single odorant receptor. These ORNs send signal to the AL, shown in the middle, which is comprised of subunits called glomeruli. Each glomerulus contains GABAergic, inhibitory LNs and excitatory, output PNs. PNs send signal to the mushroom body, shown on the right, which is the first brain region responsible for odor association and memory. The mushroom body is composed of Kenyon Cells, which have highly selective, odor-evoked responses.
1.2.2 The Antennal Lobe

Of particular focus in this research is modeling the AL, as it is the first brain region to process olfactory stimuli. To create a realistic simulation, the experimental dynamics observed within the AL must be reasonably replicated. The response patterns of PNs in particular are well studied, and will serve as a benchmark for the simulations.

PN response to olfactory stimulation is triphasic, as shown in figure 1.2. The first phase, the $I_1$ or initial hyperpolarization (IHP), is characterized by suppression of PN firing for approximately 50 ms. This phase is mechanistically well understood, and is believed to be the result of GABA$_A$ based inhibition from the LNs within the network. Application of bicuculline, a GABA$_A$ antagonist, removes the IHP phase, and thus supports the belief that fast inhibition from LNs is responsible for this phase (Waldrop, Christensen, and Hildebrand, 1987; Christensen and Hildebrand, 1988; Christensen, Waldrop, Harrow, et al., 1993; Christensen, Waldrop, and Hildebrand, 1998).

The following phase is phase II, characterized by sudden depolarization of PNs and rapid firing for the remainder of the odor stimulus. PN behavior varies widely from PN to PN within this phase, with some PNs firing more homogeneously and others firing with more burst-like patterns. Experimentally, this spectrum of behaviors is not well understood.

The final phase, $I_2$ or after hyperpolarization (AHP), is characterized by suppression of PN firing for approximately one second following the termination of olfactory stimulation (Christensen and Hildebrand, 1987b; Waldrop, Christensen, and Hildebrand, 1987; Christensen and Hildebrand, 1988; Christensen and Hildebrand, 1990; Christensen, Waldrop, Harrow, et al., 1993). Unlike the IHP, this phase is not well understood mechanistically.

This complex dynamical response of PNs to olfactory stimuli serves as a target behavior that our model will hopefully recreate. Once our model does recreate this behavior, predictions can be made as to what mechanistic components give rise to these specific behaviors that at this point are not experimentally understood.
Figure 1.2 PN response to olfactory stimulus, represented by the black bar, is triphasic. The first phase, $I_1$ or IHP, is characterized by hyperpolarization of PN voltage, and thus suppression of firing. Phase II occurs afterwards for the remainder of the olfactory stimulus, during which PNs depolarize and fire rapidly. The firing behavior of PNs during this phase varies, where some fire more homogeneously and others fire with more burst-like behavior. Finally, following olfactory stimulus termination, $I_2$ or AHP is characterized by a long lasting hyperpolarization of PNs for 1 second. PN voltage trace was taken from Heinbockel et al. (Heinbockel, Christensen, and Hildebrand, 1999)

1.3 Experimental Data

Experimental data was collected from our collaborator Dr. Hong Lei at Arizona State University. In the sphinx moth *Manduca sexta*, the cell bodies of uniglomerular AL PNs are located within the medial cluster, a convenient target for identifying PNs (figure 1.3A); dye injection into glomerular neuropil often results in partial staining at the soma. Data from such PN recordings, in response to scented and non-scented air puffs at various air flow speeds, are shown in figure 1.3. We recorded from a total of 10 PNs, most of which showed a clear increase in spike rate in response to non-scented air puffs. One representative example is shown in figure 1.3. This PN displayed clear responses to mechanical stimulation alone (figure 1.3D), with a stronger response to air puffs infused with pheromone (which provides both chemical and mechanical stimulation) (figure 1.3E). Furthermore, within our recorded PN population, increasing the puffing speed of non-scented air puffs from 0.5 liter/min (low speed) to 1 liter/min (high speed) increased the intensity of the response of the PN population, as shown in figure 1.3B,D,F. Published work in a noctuid moth (*Spodoptera*
littoralis) has reported similar phenomena (Han, Hansson, and Anton, 2005; Froese, Szyszka, and Menzel, 2014).

We further parse the dependence of the PN response on air speed from the data displayed in figure 1.3. The sample PN did not respond to non-scented air puffs at low speed (figure 1.3B), but displayed a clear, long-lasting response to pheromone-infused air puffs at low speed (figure 1.3C); the correlation coefficient of these two response patterns (based on peri-stimulus time histograms) is rather low (r=-0.19). However, at high air-puffing speed, the PN exhibited a rather robust bursting response to non-scented puffs (figure 1.3D), but showed a clear and substantial increase in response intensity to pheromone-infused air puffs at the same speed (figure 1.3E); interestingly, these two response patterns are remarkably similar (r=0.67).

Thus, the data indicate that PNs respond robustly to strong mechanosensory input (i.e., high wind speed), even in the absence of an olfactory signal. Moreover, mechanical and chemical input appear to trigger distinct response patterns in PNs – strong mechanosensory input alone elicits a transient, temporally precise burst of spikes lasting a few hundred ms, while olfactory input alone (at low wind speed, mechanical input is reduced) yields a sustained spiking response lacking temporal precision. When the two modalities are combined (i.e., odor delivered at high wind speed), both of the corresponding response patterns appear to merge and contribute to PN dynamics – PN responses exhibit a large, transient, temporally precise burst of spikes followed by a lower-intensity sustained spiking response. Finally, it is tantalizing that there is a high correlation between a PN’s response pattern to strong mechanosensory input alone and its response pattern to strong mechanosensory input plus olfactory input; this suggests that, in natural environments, strong mechanosensory input may serve to mold and shape the dynamic, correlational structure of PN responses (see Discussion for further elaboration upon this latter observation).
**Figure 1.3** Mechanosensory input shapes PN response patterns. A) Confocal microscopic images show a single PN cell body (arrow) located in the medial cluster, labeled with Lucifer yellow fluorescent dye. The square in A1 is magnified in A2. MC: medial cluster of cell bodies; LC: lateral cluster of cell bodies; G: glomerulus; D: dorsal; V: ventral; L: lateral; M: medial. B-E) Response of a representative PN to consecutive non-scented air puffs or pheromone-infused air puffs. While responses to pheromone-infused air puffs are stronger, non-scented air puffs induce robust responses as well. The peristimulus histograms were based on responses to 5 puffs of non-scented air (B1) and pheromone blend (C1) at low air speed, and responses to the same stimuli but at high air speed (D1, E1). Examples of the raw intracellular spike traces show response to single stimulus puff under each air speed-stimulus scenario (B2, C2, D2, E2). Note that a non-scented air puff at high speed evoked a response pattern similar to that evoked by pheromone at high puffing speed. The correlation coefficient between D1 and E1 is 0.69 whereas the correlation coefficient between B1 and C1 is only 0.19. F) Response to non-scented air puffs is markedly stronger in high air speed than in the low air speed. Scatter plot (F1) shows the averaged firing rate during responses to air puffs in 10 different PNs; the plot is mostly biased towards the high air speed. Mean PN response (averaged over PNs) to a non-scented air puff at low or high puffing speed (F2, n=10), however, was not statistically significant (Sign test, p=0.15).
Chapter Two

Developing the Model

2.1 Model Summary

In order to examine the mechanisms underlying AL responses to olfactory or mechanosensory input, as well as the juxtaposition of the two, we construct a realistic, large-scale, spiking-network model of the moth AL; the model consists of six glomeruli, with 10 PNs and 6 LNs per glomerulus (Homberg, Christensen, and Hildebrand, 1989; Kingan and Hildebrand, 1985; Hoskins et al., 1986; Hildebrand, Hall, and Osmond, 1979; Matsumoto and Hildebrand, 1981; Christensen and Hildebrand, 1987a; Homberg, Montague, and Hildebrand, 1988). Individual PNs and LNs are governed by integrate-and-fire spiking dynamics, with random but fixed network connectivity – LNs synapse onto other LNs within the same glomerulus and onto PNs both within and across glomeruli, while PNs synapse only onto PNs and LNs within the same glomerulus (Reisenman et al., 2004; Lei, Christensen, and Hildebrand, 2004). LNs are GABAergic and inhibit other neurons through fast GABA\textsubscript{A} receptors, and LNs also deliver a slow inhibitory current to PNs, acting over \(~500-1000~ms\) time scales, via slower metabotropic GABA receptors. PNs within the model are cholinergic and act synaptically through fast nicotinic acetylcholine receptors; PNs are also equipped with an intrinsic calcium-dependent potassium (SK) current that activates following several PN spikes and serves to curtail further spiking (Mercer and Hildebrand, 2002b; Mercer and Hildebrand,
Figure 2.1 Schematic of model AL network. Each column represents a glomerulus; squares represent PNs (10 per glomerulus) and circles represent LNs (6 per glomerulus). Arrow heads indicate excitation, while bar heads indicate inhibition. Within a glomerulus, all cell types form synapses with each other (with cell type-specific connection probabilities), while glomerular cross-talk is mediated only via LN→PN synapses. An odor is simulated via delivery of excitatory stimulus current to all cells within a subset of three glomeruli (solid incoming arrows), while strong mechanosensory input is simulated via delivery of excitatory stimulus current to all cells within all glomeruli (dashed incoming arrows).

We simulate both chemosensory and mechanosensory input to the model, both in isolation and in conjunction. In accordance with the well-established combinatorial odor code employed by olfactory receptor neurons (ORNs) (Joerges et al., 1997; Vickers and Christensen, 1998; Vickers, Christensen, and Hildebrand, 1998; Malnic et al., 1999; Ache and Young, 2005; Wang et al., 2003; Ng et al., 2002), an odor (in the absence of significant mechanosensory input) is simulated by delivering an excitatory stimulus current to all cells within a subset of model glomeruli (half of model glomeruli are designated to receive stimulus current); different odors are simulated by varying the composition of the glomerular subset receiving stimulus current. Mechanosensory input is modeled by sending stimulus current to all cells within the entire AL network, in accordance with our data indicating that PN responses to mechanosensory input are widespread and, seemingly, non-specific. Stimuli,
whether olfactory or mechanosensory, are delivered in the form of a 1 second current pulse.

We employ two paradigms to simulate both modalities in conjunction (i.e., an odor delivered within the context of strong mechanosensory input due to high wind speed): (i) Additive Paradigm – the stimulus current due to mechanosensory input and odor input are simply added together to construct the high wind speed odor stimulus; (ii) Normalized Paradigm – the stimulus current due to mechanosensory input and odor input are both halved in amplitude then added together. Paradigm (i) may be more akin to physiological reality, since PN responses from our data (figure 1.3) are suggestive of the possibility that the mechanosensory response and the odor response of a PN augment each other to obtain the response to an olfactory signal delivered at high wind speed. Paradigm (ii) allows analysis of network dynamics when net excitation to the AL is kept approximately constant in the three cases of odor input alone, mechanosensory input alone, and odor+mechanosensory input, permitting a more distilled comparison of the dynamical effects of the two modalities in isolation and in conjunction. Figure 2.1 shows a schematic of the model network.

2.2 Model Details

We construct a spiking model of the AL network that strives to attain enough architectural complexity to achieve the complex dynamics of the AL while maintaining enough simplicity to allow for investigation of core mechanistic components. Below, we elaborate the components and connectivity of our model.

2.2.1 The Neuron Model

The model is composed of two subclasses of neurons: excitatory, cholinergic PNs and inhibitory, GABAergic LNs. The membrane potential of the $j^{th}$ PN ($V_{PN}^j(t)$) or the $j^{th}$ LN ($V_{LN}^j(t)$) are modeled using integrate-and-fire dynamics by the following set of ODEs, which include both intrinsic and synaptic currents:
\[
\frac{d}{dt} V^j_{PN}(t) = -\frac{1}{\tau_V} (V^j_{PN} - V_L) - g^j_{SK}(t)(V^j_{PN} - V_{SK}) - g^j_{stim}(t)(V^j_{PN} - V_{stim}) \\
- g^j_{exc}(t)(V^j_{PN} - V_{exc}) - g^j_{inh}(t)(V^j_{PN} - V_{inh}) - g^j_{slow}(t)(V^j_{PN} - V_{inh}).
\]

\[
\frac{d}{dt} V^j_{LN}(t) = -\frac{1}{\tau_V} (V^j_{LN} - V_L) - g^j_{stim}(t)(V^j_{LN} - V_{stim}) \\
- g^j_{exc}(t)(V^j_{LN} - V_{exc}) - g^j_{inh}(t)(V^j_{LN} - V_{inh}) - g^j_{slow}(t)(V^j_{LN} - V_{inh}).
\]

PN \(j\) in the model is equipped with an intrinsic slow potassium current (SK), and receives stimulus-induced input (from external background, odor, and mechanosensory sources), fast excitatory input from other PNs, fast inhibitory input from LNs, and slow inhibitory input from LNs. LN \(j\) in the model receives stimulus-induced input (from external background, odor, and mechanosensory sources), fast excitatory input from PNs, fast inhibitory input from other LNs, and slow inhibitory input from other LNs. In these equations, \(V_L = 0\), \(V_{exc} = V_{stim} = \frac{14}{3}\), and \(V_{SK} = V_{inh} = -\frac{2}{3}\) (expressed in nondimensional units) represent reversal potentials associated with leakage, excitation, and inhibition respectively. The leakage timescale is given by \(\tau_V = 20\)ms. Upon any neuron reaching a threshold voltage of \(V_{thres} = 1\), a spike is recorded and its voltage subsequently reset to \(V_L = 0\) (and held at \(V_L = 0\) for a refractory period of \(\tau_{ref} = 2\)ms). The neuron model is based on a reduced dimensional integrate-and-fire model previously developed in the literature (Tao et al., 2004; Lei, Yu, et al., 2016).

The term \(g^j_{exc}(t)\) represents the membrane conductance of neuron \(j\) to excitatory synaptic input from PNs, and is modeled as follows:

\[
g^j_{exc}(t) = \sum_{s \in S} S_{PN} \alpha_{exc}(t|s), \text{ where } \alpha_{exc}(t|s) = \frac{H(t - s)}{\tau_{exc}} e^{-\frac{(t-s)}{\tau_{exc}}}.\]

In this equation, \(S\) represents the set of all spike times of all PNs presynaptic to neuron \(j\). The parameter \(S_{PN}\) is the coupling strength of a network PN to neuron \(j\); \(S_{PN} = .01\)
if neuron $j$ is a PN, while $S_{PN} = .006$ if neuron $j$ is an LN. The function $\alpha_{exc}(t|s)$ has instantaneous rise time and exponential decay time, with time constant $\tau_{exc} = 2$ms (whether neuron $j$ is an LN or a PN).

The other synaptic conductances, $g_{inh}^j(t)$ and $g_{slow}^j(t)$, as well as the stimulus conductance, $g_{stim}^j(t)$, are modeled similarly:

$$g_{inh}^j(t) = \sum_{s \in S} S_{inh}\alpha_{inh}(t|s), \quad \text{where} \quad \alpha_{inh}(t|s) = \frac{H(t - s)}{\tau_{inh}} e^{-\frac{(t-s)}{\tau_{inh}}},$$

$$g_{slow}^j(t) = \sum_{s \in S} S_{slow}\alpha_{slow}(t|s), \quad \text{where} \quad \alpha_{slow}(t|s) = \frac{H(t - s)}{\tau_{slow}} e^{-\frac{(t-s)}{\tau_{slow}}},$$

$$g_{stim}^j(t) = \sum_{s \in S} S_{stim}\alpha_{stim}(t|s), \quad \text{where} \quad \alpha_{stim}(t|s) = \frac{H(t - s)}{\tau_{stim}} e^{-\frac{(t-s)}{\tau_{stim}}}.$$

For the $g_{inh}^j(t)$ and $g_{slow}^j(t)$ equations, $S$ represents the set of all spike times of all LNs presynaptic to neuron $j$. For the $g_{stim}^j(t)$ equation, $S$ represents the set of all spike times of the external input delivered to neuron $j$; these stimulus-induced spike times arise from simulation of background input, odor input, and mechanosensory input as Poisson processes of incoming spike events (see Stimulus Modeling section below for details). If neuron $j$ is a PN, the coupling strengths are given by $S_{inh} = 0.0169$, $S_{slow} = .0338$, and $S_{stim} = .004$, while if neuron $j$ is an LN, the coupling strengths are given by $S_{inh} = 0.015$, $S_{slow} = .04$, and $S_{stim} = .0031$. The fast inhibition and stimulus timescales are comparable to excitation, with $\tau_{inh} = \tau_{stim} = 2$ms, while the slow inhibition time scale is dramatically slower, with $\tau_{slow} = 750$ms (whether neuron $j$ is a PN or an LN).

Finally, the SK current is an intrinsic slow potassium current, displayed by only PNs, that activates upon spiking and serves to curb further spiking activity. Rather than an instantaneous jump, the rise time of the SK current is modeled as sigmoidal; this non-instantaneous rise time allows PNs to potentially emit multiple spikes prior to suppression of firing activity by the SK current. The SK current for PN $j$ is modeled as follows:
\[ g^j_{SK}(t) = \sum_{s \in S} S_{SK} \beta(t|s). \]

\[ \beta(t|s) = \begin{cases} 
    \frac{H(t-s)}{\tau_{SK}} e^{\frac{5((t-s)-\tau_{rise})}{\tau_{rise}}}, & t \leq s + 2\tau_{rise} \\
    \frac{1}{\tau_{SK}} e^{-\frac{-((t+(s+2\tau_{rise}))}{\tau_{SK}}}, & t > s + 2\tau_{rise}
\end{cases}. \]

\( S \) represents the set of all firing times of PN \( j \). The strength \( S_{SK} \) of the SK current is a randomly determined, but fixed, parameter, and hence varies from PN to PN; the value of \( S_{SK} \) for PN \( j \) is drawn from a normal distribution with mean \( \mu = .5 \) and standard deviation \( \sigma = .2 \). While rare, it is possible for \( S_{SK} \) to be negative with this distribution, so any negative value for \( S_{SK} \) is manually set to 0. The parameter \( \tau_{SK} = 250 \text{ms} \), and the rise of the SK current is modeled as sigmoidal with a half-rise time of \( \tau_{rise} = 25 \text{ms} \).

In the above, \( H(t) \) is the standard Heaviside Step Function:

\[ H(t) = \begin{cases} 
    1, & t \geq 0 \\
    0, & t < 0
\end{cases}. \]

### 2.2.2 Network Architecture

Our AL model consists of 6 glomeruli, with each glomerulus consisting of 10 PNs and 6 LNs; connectivity within glomeruli is dense in comparison with relatively sparse connectivity across glomeruli. Synaptic connections within the model are randomly determined but fixed, with the probability of a synaptic connection varying within and across glomeruli and dependent on cell type. Within a glomerulus, the PN→PN, PN→LN, LN→PN, LN→LN connection probabilities are given by 0.75, 0.75, 0.38, 0.25, respectively. Long-range connections (i.e., connections across glomeruli) are mediated exclusively by LN→PN synapses, and the cross-glomerular LN→PN connection probability is given by 0.38.

It is worth mentioning that the model itself is quite robust, with the exact input parameters provided not essential to producing reasonable behavior. Rather, we find that com-
binations of parameters, and hence the relative strength of disparate network components, are important for producing realistic behavior. For example, we find that slow inhibition must be sufficiently strong, relative to stimulus-induced inputs, to suppress PN spiking if the global LN network is activated, yet not so strong as to silence PN activity upon only focal activation of a few glomeruli. Likewise, we find that the strength of the SK current must fall within a broad range of values, relative to the strength of stimulus-induced inputs and LN inhibition, with the lower end of this range yielding homogeneous PN spiking activity and the higher end of this range yielding burst-like PN behavior. Hence, our parameter choices represent a single point drawn from a relatively large cloud (in multidimensional parameter space) of parameter combinations that produce physiologically reasonable behavior.

2.2.3 Stimulus Modeling

Rather than explicitly modeling the behavior of ORNs or the cells responsible for mechanosensory sensory inputs, input to each cell within the network is modeled as a Poisson process of incoming spikes. An incoming spike to neuron \( j \) within the network is modeled as an instantaneous jump in \( g_{\text{stim}}^j(t) \) of size .004 if neuron \( j \) is a PN, or .0031 if neuron \( j \) is an LN, followed by exponential decay with time constant \( \tau = 2 \text{ms} \) (whether neuron \( j \) is a PN or an LN). Each cell has three potential sources of input; all cells receive a background rate of \( \lambda_{\text{back}} = 3.6 \text{ spikes/ms} \), while odor input (simulating the presence of a single odor) is delivered at a maximum rate of \( \lambda_{\text{odor}}^{\text{max}} = 3.6 \text{ spikes/ms} \) and mechanosensory input is delivered at a maximum rate of \( \lambda_{\text{mech}}^{\text{max}} = 1.8 \text{ spikes/ms} \). The total rate of incoming spikes for the \( j^{th} \) cell is therefore given by:

\[
\lambda_j^{\text{tot}}(t) = \lambda_{\text{back}} + \lambda_{\text{odor}}^{\text{max}} O^j(t) + \lambda_{\text{mech}}^{\text{max}} M^j(t),
\]

where \( O^j(t) \) and \( M^j(t) \) are functions that range between 0 and 1 and serve to model the temporal dynamics of odor and mechanosensory input pulses, respectively.
To simulate background AL activity, we set $\lambda^j_{\text{odor}} = 0$ and $\lambda^j_{\text{mech}} = 0$ for all $j$. To simulate a single odor (without simulation of mechanosensory input) presented at time $t_{on}$ and removed at time $t_{off}$, we send odor-induced input to all cells within 3 out of 6 model glomeruli (with the glomerular subset signifying odor identity); we therefore set $\lambda^j_{\text{odor}} = 0$ if cell $j$ belongs to an unstimulated glomerulus and $\lambda^j_{\text{odor}} = \lambda_{\text{odor}}^{\text{max}}$ if cell $j$ belongs to a stimulated glomerulus, and set $\lambda^j_{\text{mech}} = 0$ for all $j$. To simulate a pulse of mechanosensory input (without an accompanying olfactory stimulus) from time $t_{on}$ to time $t_{off}$, we set $\lambda^j_{\text{odor}} = 0$ and $\lambda^j_{\text{mech}} = \lambda_{\text{mech}}^{\text{max}}$ for all $j$. Hence, within the model, olfactory input stimulates a focal glomerular subset, while mechanosensory input represents a global signal delivered to the entirety of the AL.

In addition to simulating olfactory and mechanosensory input in isolation, we also simulate the two in combination. To simulate a stimulus pulse (from time $t_{on}$ to time $t_{off}$) consisting of a single odor accompanied by a mechanosensory signal, we employ two distinct paradigms – the additive sensory integration and the normalized sensory integration paradigms. In the additive sensory integration paradigm, we set $\lambda^j_{\text{odor}} = 0$ if cell $j$ belongs to a glomerulus not responsive to the odor and $\lambda^j_{\text{odor}} = \lambda_{\text{odor}}^{\text{max}}$ if cell $j$ belongs to a glomerulus activated by the odor, and set $\lambda^j_{\text{mech}} = \lambda_{\text{mech}}^{\text{max}}$ for all $j$. In the normalized sensory integration paradigm, we set $\lambda^j_{\text{odor}} = 0$ if cell $j$ belongs to a glomerulus not responsive to the odor and $\lambda^j_{\text{odor}} = \frac{1}{2} \lambda_{\text{odor}}^{\text{max}}$ if cell $j$ belongs to a glomerulus activated by the odor, and set $\lambda^j_{\text{mech}} = \frac{1}{2} \lambda_{\text{mech}}^{\text{max}}$ for all $j$. Hence, the additive paradigm simply ‘adds’ together the isolated olfactory and mechanosensory signals, while the normalized paradigm scales the two signals in order to maintain a similar overall level of excitation sent to the AL network in comparison to the cases of olfactory input or mechanosensory input in isolation.

Finally, to simulate two odors simultaneously (as in figure 3.6), we simply ‘add’ the inputs resulting from the two odors in isolation. In other words, we set $\lambda^j_{\text{odor}} = 0$ if cell $j$ belongs to a glomerulus stimulated by neither odor, $\lambda^j_{\text{odor}} = \lambda_{\text{odor}}^{\text{max}}$ if cell $j$ belongs to a glomerulus stimulated by only 1 of the 2 odors, and $\lambda^j_{\text{odor}} = 2 \lambda_{\text{odor}}^{\text{max}}$ if cell $j$ belongs to a glomerulus
stimulated by both odors.

The function $O_j(t)$ represents the temporal dynamics of the olfactory component of a stimulus pulse beginning at time $t_{on}$ and ending at time $t_{off}$ (with $t_{off} - t_{on} = 1000\text{ms}$ – i.e., stimulus pulses are modeled as having a 1 sec duration). The function $O_j(t) = 0$ for $t < t_{on}$; at time $t_{on}$, $O_j(t)$ increases from 0 to 1 with a prescribed rise time, while for $t > t_{off}$, $O_j(t)$ decreases from 1 to 0 with a prescribed decay time. If neuron $j$ is a PN, rise is sigmoidal with a half-rise time of $\tau_{rise} = 35\text{ms}$, while decay is exponential with $\tau_{decay} = 384\text{ms}$:

If $j$ is a PN, $O_j(t) = \begin{cases} H(t - t_{on}) \frac{e^{5((t-t_{on})-\tau_{rise})\tau_{rise}}}{1+e^{5((t-t_{on})-\tau_{rise})\tau_{rise}}}, & t \leq t_{on} + 2\tau_{rise} \\ 1, & t_{on} + 2\tau_{rise} < t \leq t_{off} \\ -\frac{(t-t_{off})}{e^{\tau_{decay}}}, & t_{off} < t \end{cases}$.

If neuron $j$ is a LN, rise is instantaneous, while decay is exponential with $\tau_{decay} = 384\text{ms}$:

If $j$ is an LN, $O_j(t) = \begin{cases} H(t - t_{on}), & t \leq t_{off} \\ -\frac{(t-t_{off})}{e^{\tau_{decay}}}, & t_{off} < t \end{cases}$.

Similarly, the function $M_j(t)$ represents the temporal dynamics of the mechanosensory component of a stimulus pulse beginning at time $t_{on}$ and ending at time $t_{off}$ ($t_{off} - t_{on} = 1000\text{ms}$). The function $M_j(t) = 0$ for $t < t_{on}$; at time $t_{on}$, $M_j(t)$ increases from 0 to 1 with a prescribed rise time, while for $t > t_{off}$, $M_j(t)$ decreases from 1 to 0 with a prescribed decay time. If neuron $j$ is a PN, rise is instantaneous and decay is exponential with $\tau_{decay} = 384\text{ms}$:

If $j$ is a PN, $M_j(t) = \begin{cases} H(t - t_{on}), & t \leq t_{off} \\ -\frac{(t-t_{off})}{e^{\tau_{decay}}}, & t_{off} < t \end{cases}$.

If neuron $j$ is a LN, $\tau_{rise} = 300\text{ms}$ and $\tau_{decay} = 384\text{ms}$:
If $j$ is an LN, $M^j(t) = \begin{cases} 
H(t - t_{on}) \frac{e^{\frac{5((t-t_{on})-\tau_{rise})}{\tau_{rise}}}}{1+e^{\frac{5((t-t_{on})-\tau_{rise})}{\tau_{rise}}}}, & t \leq t_{on} + 2\tau_{rise} \\
1, & t_{on} + 2\tau_{rise} < t \leq t_{off} \\
\frac{-(t-t_{off})}{e^{-\tau_{decay}}}, & t_{off} < t \end{cases}$

We included a significantly longer rise time for mechanosensory input to LNs, relative to that for PNs, in order to ensure that global LN inhibition at the inception of a stimulus pulse (mediated by fast inhibitory synapses) is not overwhelmingly powerfully enough to prevent PN spiking altogether, and that substantial suppression of PN spiking must await the lengthy activation time of slow inhibitory synapses. This assumption, however, is not necessarily required to obtain physiologically reasonable dynamics – for example, weakening fast inhibitory synapses from LNs to PNs or strengthening slow inhibition while reducing the density of LN→PN synapses can yield similar dynamical effects without such a disparity in rise times. Since the dynamics of mechanosensory input to AL cells has not yet been studied within the experimental literature, we (somewhat arbitrarily, due to ignorance of the actual physiological mechanism at play) chose to include this mechanism of a disparity in rise times to ensure robust PN spiking at stimulus onset. However, we note that including a disparity in rise times does not affect the basic dynamical behavior of the model, other than delaying suppression of PN spiking at stimulus onset.
Chapter Three

Results

3.1 Model Dynamics with Olfactory Input in Isolation

We begin by probing model responses to olfactory input alone (i.e., without simulating mechanosensory input), modeled as strong, focal stimulus current delivered to a subset of AL glomeruli (glomeruli 1, 2, and 3). We verify that model responses to olfactory input alone recapitulate the salient features of experimentally observed moth AL odor dynamics, and we further dissect the dynamical mechanisms underlying model behavior. Experimentally, during prolonged odor presentation PNs within activated glomeruli tend to exhibit a characteristic triphasic response; the first phase ($I_1$) occurs at odor onset and consists of a brief membrane potential hyperpolarization that lasts $\sim 50$ ms, after which rapid depolarization accompanied by spiking ensues for the duration of the stimulus (phase II), while odor offset triggers the third phase ($I_2$, also called the AHP phase) of hyperpolarization followed by a slow decay back to background over $\sim 1$ sec (Christensen and Hildebrand, 1987b; Waldrop, Christensen, and Hildebrand, 1987; Christensen and Hildebrand, 1988; Christensen and Hildebrand, 1990; Christensen, Waldrop, Harrow, et al., 1993).
Figure 3.1 Model dynamics in response to odor stimulation alone, with no mechanosensory input. (A) Spike raster of the AL network. PNs and LNs are grouped together by glomerulus, with the bottom 10 rows in each glomerulus depicting PNs and the top 6 rows depicting LNs. A one second odor pulse (marked by the black bar on the horizontal axis) is simulated by sending stimulus current to all cells within glomeruli 1, 2 and 3. (B) Plot of the membrane potential, intrinsic SK current, and incoming synaptic excitation, slow inhibition, and fast inhibition for a single PN in a stimulated glomerulus. This PN displayed continuous firing behavior. (C) Similar plot as (B), but for a neuron which displayed a more burst-like firing pattern.
Figure 3.1A depicts a raster plot of AL spiking activity in response to odor input - PNs within stimulated glomeruli tend to exhibit robust responses to the odorant, while PNs within nonstimulated glomeruli are suppressed, due to pervasive network-wide LN inhibition induced by the stimulus (Reisenman et al., 2004; Lei, Christensen, and Hildebrand, 2004). Figure 3.1A further shows that PNs within activated glomeruli exhibit the characteristic triphasic odor response observed experimentally. Phase I (the I$_1$ phase) occurs due to the stimulus current inducing LNs to reach threshold prior to PNs; hence, odor onset yields an initial burst of synchronized LN spikes within activated glomeruli, leading to compounded fast inhibition from LNs briefly hyperpolarizing and silencing glomerular PNs. Following the initial burst of LN spikes, LN activity desynchronizes and diminishes slightly (due to LN→LN fast synaptic inhibition), allowing PN depolarization to commence and phase II PN spiking to ensue. This is consistent with experimental evidence showing that direct GABA application mimics an I$_1$-type response in PNs and that the GABA antagonist bicuculline eliminates the I$_1$ phase of odor-evoked PN activity (Waldrop, Christensen, and Hildebrand, 1987; Christensen and Hildebrand, 1988; Christensen, Waldrop, Harrow, et al., 1993; Christensen, Waldrop, and Hildebrand, 1998); indeed, blockade of fast GABAergic synapses within our model eliminates the phase I hyperpolarization of PNs following odor onset (figure 3.3A).
Figure 3.2 (A) Raster plot of model dynamics during a one second odor pulse (with no mechanosensory input), in the case that all PNs either have no SK currents (left) or very strong SK currents (right). Note that PNs without SK current exhibit more continuous firing, whereas PNs with high SK currents exhibit more burst-like behavior. (B) Spike rasters for a single stimulated PN in the model during a 1 second odor pulse, for varying levels of the strength of the PN’s SK current. As SK strength increases, firing behavior shifts from being continuous to bursting. (C) The standard deviation of interspike interval during odor stimulation was used to quantify the burstiness of a PN. A small standard deviation represents relatively homogeneous firing, while a larger standard deviation represents fluctuations between short interspike intervals within a burst and long interspike intervals between bursts. The scatter plot shows SK strength and standard deviation in interspike interval for each PN during odor presentation (only PNs within active glomeruli are included). A positive correlation of $r=0.57$ reveals that SK currents indeed play a role in the emergence of heterogeneous bursting behavior.
However, while experiments indicate that the $I_1$ response seems driven by GABA$_A$ receptors, the mechanisms underlying the phase II response and the AHP (or $I_2$) phase are empirically less well-understood. During the phase II response, PN spike patterns can vary broadly across PNs; within active glomeruli, some PNs fire spikes in sporadic or irregular bursts, while others fire more continuously at moderate (~50 Hz) firing rates, with a range of response patterns intermediate between the extremes of bursting and continuous firing. Moreover, the phase II dynamics exhibited by a PN shows no spatial dependency – in fact, a significant amount of intraglomerular variability has been observed during stimulation (Christensen, Pawlowski, et al., 2000).

Our model exhibits similar variability in phase II PN response patterns (figure 3.1A), with examples of a continuously firing PN and a bursting PN shown in figures 3.1B and 3.1C, respectively. Within our model, the phase II behavior of a PN is primarily determined by the strength of its SK current – the strength of the SK current of each PN within our model is randomly drawn from a Gaussian distribution, leading to inherent variability in SK current strength across PNs and hence the diversity in PN phase II response patterns. Figure 3.2A shows spike rasters when the SK current is removed from our network (left) or fixed at a high value for all PNs (right), suggesting that a lack of SK currents tends to yield continuously firing PNs, while potent SK currents tend to produce burst-like PN behavior. This is further corroborated by measuring the phase II behavior of a single, fixed PN as the strength of its SK current is varied – figure 3.2B shows that the phase II response pattern of a PN undergoes a clear, progressive, and gradual transition from continuous firing behavior to sporadic bursting behavior as SK strength is increased. Indeed, within the normal network (with randomly drawn SK strengths), there is a substantial correlation between the degree of burst-like behavior and SK current strength across PNs (figure 3.2C).
Figure 3.3 (A) Raster plot of the model AL during odor presentation with no mechanosensory input (left) and membrane potential, SK current, incoming synaptic excitation, slow inhibition, and fast inhibition for a sample PN (right); plots shown are in the case that fast inhibition is removed from the network. Note that removing fast inhibition diminishes the phase I hyperpolarization ($I_1$) of PNs upon odor onset. (B) Similar plots as in (A), but with slow inhibition rather than fast inhibition removed from the network. Note that removing slow inhibition eliminates the prolonged AHP phase ($I_2$) of PNs following odor offset. Black bar represents odor stimulus.
Within our model, the AHP (or $I_2$) phase of lengthy hyperpolarization following prolonged odor presentation emerges as a dynamical consequence of slow inhibition from LNs to PNs. During phase II spiking in the midst of prolonged odor presentation, the slow inhibitory current activates and exerts a damping effect on PN spiking, but the potent odor-induced current impinging upon stimulated glomeruli is still able to elicit substantial PN spiking responses. Upon odor offset, however, the odor-induced current rapidly dissipates, while slow inhibition from LNs to PNs decays over a longer 1-2 second time scale, allowing the gradually dwindling slow inhibitory current to suppress PN activity for $\sim$1 second following odor offset. Indeed, severing slow inhibitory synapses within the model virtually eliminates the AHP phase of PN odor dynamics, and PNs return to background activity levels immediately following odor offset (figure 3.3B).

3.2 Model Dynamics with Mechanosensory Input

We now examine the behavior of our model with the inclusion of strong mechanosensory input, simulated as a one second current pulse delivered to all glomeruli, though weakened in comparison to an odor-induced current pulse to a stimulated glomerulus (an odor alone, as in the previous section, is simulated as a stronger one second current pulse delivered to only three glomeruli). Thus, we construct four stimulus scenarios: (i) odor only, with no mechanosensory input, which simulates odor delivered at low wind speed; (ii) mechanosensory input only, which simulates a nonscented, high speed air puff; (iii) additive paradigm of high wind speed odor delivery, simulated by simple summation of the current pulses in the odor-only and mechanosensory-only scenarios; (iv) normalized paradigm of high wind speed odor delivery, simulated by halving then summing the current pulses in the odor-only and mechanosensory-only scenarios. The additive paradigm is likely more representative of the physiological reality of moth AL dynamics in response to high wind speed odor presentation (the moth data in figure 1.3 suggest that PN responses to high wind speed odor stimuli are
approximately given by the sum of the response to mechanosensory input alone and odor input alone). However, the normalized paradigm maintains similar net excitation to the AL as in the odor-only and mechanosensory-only scenarios, allowing direct comparison of scenarios (i), (ii), and (iv); this permits distillation of the dynamical effects of odor input and mechanosensory input alone, and the two in conjunction, without the potential confound posed by variation in the net integrated excitation impinging upon the AL.
Figure 3.4 Raster plots of the first three model glomeruli (left) and firing rates of a sample PN from one of these glomeruli over 10 ms windows and averaged over 100 trials (right) for (A) odor input only, (B) mechanosensory input only, (C) additive odor and mechanosensory integration, and (D) normalized odor and mechanosensory integration. These results closely mimic the experimental results shown in figure 1.3. Black bar represents stimulus.
Figure 3.4 shows spike rasters of the AL model (left) and the trial-averaged firing rate of a sample PN in the network (right) in response to the four stimulus scenarios. In the odor-only scenario, PNs within stimulated glomeruli exhibit substantial spiking throughout the duration of the stimulus, with only a modest differential elevation in firing rate occurring specifically at stimulus onset (figure 3.4A). In the mechanosensory-only scenario, however, PNs throughout the entire AL respond with an intense spike burst at stimulus onset, but within a few hundred milliseconds are rapidly suppressed to background or lower than background activity levels, and subsequently remain in this suppressed state for the duration of the stimulus (figure 3.4B). Thus, transient, temporally precise PN responses to purely mechanosensory input followed rapidly by suppression, as seen in our experimental data in figure 1.3, automatically emerge from model dynamics, despite the prolonged nature of the current pulse simulating mechanosensory input. The response of our model AL to strong mechanosensory input in conjunction with odor input (figure 3.4C,D) combines features of the responses to the two modalities in isolation; all AL PNs respond with a sharp burst of spikes at stimulus onset (with PNs in glomeruli receiving odor input displaying considerably higher-frequency spiking within the burst), while following the burst, for the remainder of the stimulus duration, PNs within glomeruli receiving odor input continue firing at rates substantially elevated above background (though considerably diminished in comparison to the prior spike burst) and PNs within glomeruli not receiving odor input are strongly suppressed below background. Collectively, this shows that model behavior captures the salient features of our experimental data (figure 1.3) – PN responses to odor alone (at low wind speed) are long-lasting but lack temporal precision, responses to mechanosensory input alone (high wind speed without odor) are transient and temporally precise, and responses to odor and mechanosensory input in conjunction (odor delivered at high wind speed) contain both a transient, temporally precise component as well as a long-lasting component.
Figure 3.5 (A) Raster plots of three sample glomeruli within the model AL during AL stimulation by a 1 second pulse of mechanosensory input only (black bar), in the case where slow inhibition is removed (left), fast inhibition is removed (center), or the SK current is removed (right) from the model. Slow inhibition appears to have the largest effect in causing the transient nature of the mechanosensory response. (B) Raster plots of a sample stimulated glomerulus during AL stimulation by a 1 second pulse of mechanosensory input only (left) or odor input only (right) for varying strengths of the slow inhibitory synapses from LNs to PNs. As slow inhibitory strength increases, both mechanosensory and odor responses appear more transient, though the mechanosensory response becomes transient at lower slow inhibitory strengths than the odor response. (C) Left: Bar plot for PN firing rate, averaged over all network PNs and normalized by the background firing rate, during the last 500 ms of a 1 second pulse of mechanosensory input only. Data are shown in the case of the fully intact model (control) as well as in cases where various network components are removed. Mean and standard deviations are computed over 100 trials. Note that, without slow inhibition, there is little response suppression during the last 500 ms of the stimulus pulse. Center: PN firing rate, averaged over all network PNs and normalized by the background firing rate, during the last 500 ms of a 1 second pulse of mechanosensory input only or odor input only, as a function of the strength of slow inhibitory synapses. Mean and standard deviations are computed over 100 trials. Right: Same as in the center panel, except PN firing rate is plotted as a function of the density of LN→PN synapses. Note that during the latter half of a stimulus pulse the response to mechanosensory input alone, compared to the response to odor input alone, is suppressed at lower values of slow inhibitory strength or LN→PN connection probability.
This leads to a natural query: how do model dynamics give rise to such qualitatively discordant behavior in response to the two modalities? In other words, why are responses to odor input long-lasting while those to strong mechanosensory input are transient and brief, despite the temporally prolonged nature of both stimuli? The answer lies primarily within the dynamics of the slow inhibitory current from LNs to PNs, coupled with the globally extensive nature of the glomeruli-spanning LN network. Figure 3.5A shows the AL response to strong mechanosensory input alone in the absence of slow inhibition, fast inhibition, or the SK current, and suggests that the lack of slow inhibition produces the most profound effect on eliminating the transient nature of responses to mechanosensory input; this is further quantified in figure 3.5C (left), which shows that PN firing rates during the latter half of a one second mechanosensory input pulse tend to be higher in the absence of slow inhibition versus in the absence of other network components. Since LN efferents traverse glomeruli to synapse onto PNs throughout the AL, the mechanosensory signal, which activates all glomeruli, yields global activation of the LN network, and hence globally pooled slow inhibition is delivered to all network PNs; the potency of this pooled slow inhibitory current ensures that, after a few hundred ms (once the slow inhibition activates and rises in strength) network PNs are hyperpolarized and silenced. In the case of an odor input alone, on the other hand, only a subset of glomeruli receive stimulus current, and activation of this LN subset does not generate enough pooled slow inhibition to overcome PN responses to the strong stimulus current, leading to long-lasting PN responses (though PNs within nonstimulated glomeruli are indeed suppressed by the slow inhibitory current).

This suggests that if the potency of slow inhibition were sufficiently enhanced, then PN responses to odor input alone would also begin to display a more transient quality (since, presumably, slow inhibition from even a subset of network LNs would then be sufficient to silence PNs). This is exemplified in figure 3.5B, which shows the spike response of a sample glomerulus to a one second pulse of mechanosensory input alone (left) or odor input alone (right) for varying levels of pervasive slow inhibition within the network. For very low levels
of slow inhibition, responses to both mechanosensory input alone and odor input alone are long lasting, while for very high levels of slow inhibition responses to both stimulus modalities are transient, brief, and rapidly curtailed by the rising slow inhibitory current. For mid-range levels of slow inhibition within the network, however, PN responses to mechanosensory input are transient and brief (global LN activation is sufficient to suppress PN responses), while PN responses to odor input are long-lasting (local LN activation does not produce enough network-wide slow inhibition to squash PN responses). This observation is quantified in the center and right panels of figure 3.5C, which show that slow inhibition levels within a moderate, mid-level range, during the latter half of a one second stimulus pulse, yield strong suppression of PN responses in the case of mechanosensory input alone but substantial PN spiking in the case of odor input alone.

3.3 Odor Discrimination Dynamics within the Model

Experimental studies show that simultaneous presentation of multiple odors can actually have a damping effect on responses within the moth AL, often resulting in less intense network-wide PN activation than in the case of single-odor presentation (Lei and Vickers, 2008; Silbering and Galizia, 2007; Yamagata et al., 2009). Our model exhibits behavior consistent with these empirical observations – simultaneous presentation of two odors that activate sufficiently disparate sets of glomeruli triggers network-wide inhibition that is stronger than the enhancement in excitatory ORN input to the network resulting from simulating two odors (rather than a single odor), producing a net suppressive effect on model dynamics (figure 3.6A). This is due to the diffuse, glomeruli-spanning LN network (Reisenman et al., 2004; Lei, Christensen, and Hildebrand, 2004); simultaneous presentation of two odors, provided the odors activate minimally overlapping glomerular subsets, activates a broad swath of this LN network (in comparison to a single odor), resulting in a substantial increase in the barrage of globally pervasive slow (and fast) inhibitory inputs impinging upon PNs across
glomeruli that more than offsets the greater stimulus-induced excitation.

We further assess the ability of PN activity within our model to classify different odors. Since a single odor within our model is represented by the identity of the three (out of six) glomeruli that receive odor-induced stimulus current, this implies that our model is capable of simulating a suite of 20 distinct odorants. We employ a simple linear classification scheme to test the ability of PN activity within the model to discriminate among the panel of twenty odorants – each odorant is represented by a template, given by the trial-averaged vector of PN firing rates in response to presentation of the odor (100 trials per odor), and each trial is designated as ‘correctly classified’ by the network if the distance between the vector of PN firing rates corresponding to that trial to the various odor templates is minimized for the correct odor template (see Appendix A for details).

Figure 3.6B (left) shows the correct classification rate of the network, computed in 10 ms windows, over a one second odor pulse; in general, the odor alone (with no mechanosensory input) stimulus scenario exhibits the greatest accuracy in odor classification in comparison to the stimulus scenarios which include mechanosensory input (the additive or normalized sensory integration scenarios). This matches intuitive expectations – the mechanosensory signal is nonspecific and independent of odor identity; thus, the mechanical signal cannot impart any information about odor identity, and is more likely to confound and imbue an element of ambiguity into existing odor identity information. Accordingly, odor classification is reduced most strikingly in the normalized sensory integration stimulus scenario, since in this scenario the olfactory signal is diminished in strength, along with the presence of the potentially confounding mechanosensory signal.

Interestingly, however, figure 3.6B (left) indicates that while the odor-only scenario yields more accurate odor classification during the initial few hundred ms of the odor response, the additive sensory integration scenario actually yields slightly higher classification rates during the subsequent few hundred ms of the odor response (with the two scenarios yielding comparable performance after ∼500 ms). The reason for this can be ascertained by compar-
ing spike rasters of odor-stimulated glomeruli in the two scenarios (figure 3.4A versus figure 3.4C); note that, in glomeruli receiving olfactory input, the initial burst of PN spikes at odor onset in the additive sensory integration case outlasts the initial burst in the odor-only case by a few hundred ms, which presumably enhances the separation between distinct odor templates in the 60 dimensional PN phase space for the additive sensory integration scenario in comparison to the odor-only scenario (beginning a few hundred ms after odor onset and ending \(\sim 500\) ms after odor onset). Thus, odor discrimination is better in the additive sensory integration case versus the odor-only case during this brief temporal window.

This feature of odor classification dynamics is an emergent consequence of the slow inhibitory current from LNs to PNs. In the odor-only scenario, odor-stimulated glomeruli receive less net ORN input than in the additive sensory integration scenario, since in the latter scenario odor-stimulated glomeruli receive the odor-induced signal (as in the former scenario) plus additional excitation from the mechanosensory signal. This implies that stronger slow inhibition is required to substantially dampen PN spiking in odor-stimulated glomeruli in the additive sensory integration scenario than in the odor-only scenario; however, since the activation time scale of slow inhibition is similar in the two scenarios, substantial dampening of PN responses during the initial stimulus-induced high-frequency spike burst takes a greater length of time in the additive sensory integration scenario than in the odor-only scenario. Thus, due to the temporal dynamics of the activation of the slow inhibitory current, there exists a brief time window, beginning a few hundred ms after odor onset and ending \(\sim 500\) ms after odor onset, during which PNs within odor-stimulated glomeruli spike at a substantially higher rate in the additive sensory integration scenario than in the odor-only scenario, and odor classification is hence more accurate in the former scenario during this brief epoch. Indeed, as evident from figure 3.6B (right), blockade of slow inhibitory synapses within the network leads to removal of this brief epoch, yielding odor classification rates in the odor-only scenario that surpass those in the additive sensory integration scenario for the entire first \(\sim 500\) ms of stimulus presentation. Overall, since the first few hundred ms
of an odor response are likely the most behaviorally relevant for an insect, it is reasonable to suggest that odor classification is, for practical purposes, diminished with the addition of mechanosensory input (even with the inclusion of slow inhibition).
Figure 3.6 (A) Left: Raster of model AL network spikes in response to simultaneous presentation of two odors (without mechanosensory input). Odor 1 stimulated glomeruli 1, 2, and 3; odor 2 stimulated glomeruli 3, 4, and 5. Right: Bar chart of PN firing rate, averaged over the entire AL, for simultaneous presentation of two odors with various degrees of overlap in the glomeruli activated by each odor; data show means and standard deviations computed over 100 trials. The less the overlap in stimulated glomeruli, the greater the suppression of PN responses, likely due to greater AL-wide activation of the LN network. (B) Left: Plots of the odor classification rate of net PN activity during a one second period of odor presentation for the odor only, additive odor and mechanosensory, and normalized odor and mechanosensory stimulus scenarios. A panel of 20 odors was employed, and the correct classification rate of net PN activity was computed in 10 ms windows using a simple linear classification scheme (see Appendix A for details). Right: Same plot as in the left panel, except with slow inhibition removed from the model.
Chapter Four

Discussion

In this work, we present experimental evidence showing that PNs within the moth AL respond not only to ambient olfactory stimuli, but also to mechanosensory signals arising from high-speed air flow across the antennae, firmly establishing the AL as a structure that integrates input from multiple sensory modalities (rather than simply responding to olfactory input). Additionally, our experimental work suggests that olfactory and mechanosensory signals induce starkly different response dynamics within the AL – olfactory input tends to induce long-lasting PN responses that lack temporal precision, while in contrast mechanosensory input leads to brief, temporally precise PN responses. We then develop a biophysically detailed model of the moth AL that captures many salient features of moth AL odor responses reported in the literature, and we use our model to dissect and distill the dynamical mechanisms underlying these network behaviors. Furthermore, we simulate both olfactory and mechanosensory input within our model, showing that model PN responses closely mimic our experimental observations, and we suggest that a slow inhibitory current from LNs to PNs, coupled with the more global nature of mechanosensory input (in comparison to olfactory input) and a glomeruli-spanning LN network that widely distributes inhibition throughout the AL, may be largely responsible for the remarkably disparate AL dynamics we observe in response to olfactory versus mechanosensory signals. Finally, we suggest, using our model, that mechanosensory input may actually somewhat diminish the ability of AL activity to
parse and classify a set of ambient environmental odors.

4.1 Biological Implications and Hypotheses

The tantalizing evidence of sensory integration within the AL presented in this work leads to a natural query: what are the possible biological functions of such bimodality? Insight may be gleaned from two key features of AL responses delineated in this paper: 

(i) while the PN response to scented puffs at low air speed contains a minimal mechanosensory component and tends to be long-lasting (i.e., not temporally precise), the PN response to scented puffs at high air speed is augmented by a strong mechanosensory component and exhibits a large transient burst (i.e., is quite temporally precise); 

(ii) in general, PNs exhibit larger odor responses in the presence of significant mechanosensory input (see figures 1.3 and 3.4). These starkly disparate PN response patterns in the presence versus absence of a strong mechanosensory signal suggests that the AL may alternate between two mostly distinct dynamic regimes – an odor tracking regime and an odor discrimination regime, respectively.

*Odor Tracking Regime:* Strong mechanosensory input (e.g., when the insect is mid-flight and actively tracking an odor source) may place the AL within an odor-tracking dynamic regime. Strong mechanosensory input may ‘prime’ the AL network, allowing an accompanying odor to ‘push’ the AL into a globally coherent dynamic regime; when air speed is high, strong and rapidly fluctuating mechanosensory input may induce subthreshold voltage fluctuations across PNs that are tightly correlated and large in amplitude (due to the fact that all PNs receive *the same* mechanosensory signal), and embedding an odor within the windy flow (causing odor packets to ‘ride’ atop high-speed air pulses) may then yield widespread PN spiking, tightly correlated spiking across PNs, and greater global coherence and synchrony. Widespread, globally coherent AL activity may then bring full attentional resources to bear on the source-tracking task. Furthermore, strong mechanosensory input (in addition to odor input) may enhance the temporal precision of AL responses (possibly due...
to both rapid fluctuations in the mechanical input itself as well as AL network mechanisms similar to those encapsulated in figure 3.5); this may allow the AL to both resolve odor plume dynamics by tracking pulsatile odor delivery (through the transient, precise component of the response) and ascertain odor identity (through the longer-lasting response component).

**Odor Discrimination Regime:** A relatively small level of mechanosensory input (e.g., when the insect has landed on the surface of or is hovering near a food source) may place the AL within an odor-discrimination dynamic regime (since, without strong, rapidly fluctuating mechanosensory input, tracking odor pulses is less meaningful while fine odor discrimination, enhanced by lost-lasting odor inputs, may be a more profitable endeavor). When air speed is low, the minimal mechanosensory input may fail to induce large, correlated subthreshold voltage fluctuations across PNs, and accompanying odors may then yield uncorrelated and lower intensity spiking responses across PNs within responsive glomeruli. Additionally, PN responses may be longer-lasting, sacrificing temporal precision for duration. Thus, AL activity is globally unorganized, and is instead dominated by ‘patchy’ local dynamics unfolding over prolonged time scales, placing the network within a regime devoted to parsing the multitude of environmental odors (rather than tracking rapidly fluctuating odor pulses).

This picture of the biological role of the AL is suggestive, and we therefore propose three overarching, testable hypotheses about the effects of mechanosensory input on AL dynamics: (1) Strong mechanosensory input correlates activity across PNs and glomeruli and results in greater global coherence and synchrony; (2) Strong mechanosensory input enhances the temporal precision and pulse-tracking ability of PN responses; (3) Strong mechanosensory input diminishes the ability of AL activity to discriminate among similar odors (as suggested by figure 3.6).

Thus, within the insect, strong mechanosensory input may serve as a ‘switch’ that alternates the AL between odor tracking and odor discrimination dynamic regimes. In the absence of strong mechanosensory input, there is little environmental information to guide tracking of an odor source, and hence the AL may employ locally disconnected, ‘patchy’,
long-lasting dynamics to devote its resources to fine odor discrimination. In the presence of strong mechanosensory input, however, the AL may ‘switch’ to a more globally coherent regime with temporally precise responses; this allows AL activity to faithfully follow the spatiotemporal dynamics of environmental odor pulses and track an odor source, while sacrificing the ability of network dynamics to finely discriminate among similar odors.

4.2 Sources of Mechanosensory Input to the AL

While our data clearly indicate that AL neurons respond to mechanosensory input in the form of air speed and pressure, this observation leads to a natural query: what are the potential mechanisms by which both mechanosensory and chemosensory information to glomeruli within the AL are conveyed? Olfactory input to the AL is well-known to arise from antennal ORNs, yet the source of mechanosensory input is less certain. However, there are two apparent possibilities that exist. The first may be similar to a mechanism found in vertebrate ORNs. In one study (Grosmaitre et al., 2007), mouse ORNs were shown to respond to both olfactory and mechanical stimuli, and inhibiting adenylyl cyclase completely blocked both types of responses, suggesting that cyclic adenosine monophosphate (cAMP) is involved as a second messenger. Furthermore, knocking out CNG channels eliminated mechanosensitivity. The authors speculated that either some odorant receptors are sensitive to mechanical pressure or a mechanosensor in the membrane might cross-link to the cAMP cascade. In *Manduca sexta*, CNG channels are believed to be similar to those in vertebrates and are directly activated by cAMP and another second messenger, cGMP (Krannich and Stengl, 2008). The similarities in signal cascade pathways between moth and mouse ORNs may be indicative of physiological similarities as well. The second possibility may be related to the physical proximity of specialized mechanosensitive neurons and ORNs. The somata of bipolar ORNs reside below the base of the sensillum in close physical proximity to one another (Keil, 1989); the axons of mechanosensory neurons (such as those from the Johnston’s
organ) travel in the same nerve bundle as the axons of ORNs towards the brain (Keil, 1989; Sane et al., 2007), providing opportunities for ephaptic interactions between axons (Jefferys, 1995; Anastassiou et al., 2011; Hillier and Vickers, 2011). The ephaptic effects may be further augmented by the fact that invertebrate neurons lack compact myelin sheaths (Zalc, 2016).

### 4.3 Future Directions

Our future work will involve further testing, through combined experimental and computational work, of the hypotheses described above. We will examine the effects of strong mechanosensory input on the correlational structure of PN responses across glomeruli, assessing its effects on global coherence and synchrony within the AL. We will also assess the effects of high air-speed stimuli on the odor discrimination ability of AL activity in order to further refine and probe the mechanisms underlying the modeling results presented in this paper (figure 3.6).

In particular, we will investigate the ability of PN activity (individually and in aggregate) to track transient odor pulses. Odor stimuli tend to present in nature as a series of discontinuous filaments that occur with higher spatial frequency and in increasing concentration as the odor source is approached. Thus, as a moth travels towards an odor source, it encounters brief pulses of odor that tend to occur at a more rapid rate as the odor source becomes less distant (Murlis, Elkinton, and Carde, 1992). Additionally, behavioral experiments in which male moths were tested using the female pheromone blend have indicated that intermittent stimuli are more effective at prompting the male moths to exhibit source-seeking behavior than continuous odor plumes (T. Baker, M. Willis, et al., 1985; T. Baker, Hansson, et al., 1988; T. Baker, 1989; Kramer, 1992; Kaiselling and Kramer, 1990). Collectively, these results suggest that the ability of PNs to track stimuli delivered in a pulsatile fashion may be more behaviorally relevant than measuring static responses.
Indeed, experiments show that a series of short (several hundred ms) odor pulses evokes a sequence of corresponding spike trains in activated PNs; each individual pulse produces an $I_1$ hyperpolarization followed by phase II depolarization and a burst of spikes, with pulse offset eliciting abrupt truncation of spiking activity. The prolonged AHP phase, however, does not manifest until the end of the final pulse in the stimulus train (Christensen, Waldrop, and Hildebrand, 1998; Lei, Christensen, and Hildebrand, 2002). Intracellular recordings from AL neurons using 50 ms odor pulses show that moth PNs act as low-pass filters of pulse rate (each cell tracks odor pulses with bursts of spikes up to a certain cutoff frequency that varies across PNs). Remarkably, PNs have been found that are capable of tracking up to ten odor pulses per second, while pulse rates exceeding a cell’s cutoff frequency elicit responses consisting of tonic firing. Furthermore, the cutoff frequency for pulse tracking is directly related to the amplitude of the $I_1$ hyperpolarization – PNs that display large $I_1$ membrane potential deflections are capable of locking to higher pulse rates (Christensen and Hildebrand, 1997; Heinbockel, Christensen, and Hildebrand, 1999). The transiency of experimentally observed PN responses to odor pulses tantalizingly mirrors the effects of strong mechanosensory input on AL dynamics explored in this paper. Experimentally assessing the effects of strong mechanosensory input on features of PN pulse-tracking ability, while concurrently employing computational modeling to dissect and clarify the underlying dynamical mechanisms, will provide a valuable avenue for investigating the nature, function, and purpose of sensory integration within the AL.
REFERENCES

Ache, B. and J. Young (2005). “Olfaction: Diverse Species, Conserved Principles”. In: Neuron 48, pp. 417–430.

Anastassiou, C. et al. (2011). “Ephaptic coupling of cortical neurons”. In: Nat Neurosci 14, pp. 217–223.

Ando, N. and R. Kanzaki (2015). “A simple behaviour provides accuracy and flexibility in odour plume tracking – the robotic control of sensory-motor coupling in silkworms”. In: J Exp Biol 218, pp. 3845–3854.

Baker, K. et al. (2019). “Spatiotemporal dynamics of odor responses in the lateral and dorsal olfactory bulb”. In: PLoS Biology 17, e3000409.

Baker, T. (1989). “Sex pheromone communication in the Lepidoptera: new research progress”. In: Experientia 45, pp. 248–262.

Baker, T., B. Hansson, et al. (1988). “Adaptation of antannal neurons in moths is associated with cessation of pheromone-mediated upwind flight”. In: Proc Natl Acad Sci USA 85, pp. 9826–9830.

Baker, T., M. Willis, et al. (1985). “A pulsed cloud of sex pheromone elicits upwind flight in male moths”. In: Physiol Entomol 10, pp. 257–265.

Brinkmann, A. and D. Schild (2016). “One special glomerulus in the olfactory bulb of Xenopus laevis tadpoles integrates a broad range of amino acids and mechanical stimuli”. In: J Neurosci 36, pp. 10978–10989.

Carde, R. and M.. Willis (2008). “Navigational strategies used by insects to find distant, wind-borne sources of odor”. In: J Chem Ecol 34, pp. 854–866.

Cazamine, S. and J. Hildebrand (1979). “Central projections of antennal sensory neurons in mature and developing Manduca sexta”. In: Soc Neurosci Abstr 5, p. 155.

Christensen, T. and J. Hildebrand (1987a). “Functions, Organization and Physiology of the Olfactory Pathways in the Lepidopteran Brain”. In: In: Arthropod Brain. It’s Evolution, Development, Structure and Functions. (Gupta AP, ed). New York: Wiley, pp. 457–484.
Christensen, T. and J. Hildebrand (1987b). “Male-specific, sex-pheromone-selective projection neurons in the antennal lobes of the moth Manduca sexta”. In: J Comp Physiol A 160, pp. 553–569.

— (1988). “Frequency coding by central olfactory neurons in the sphinx moth Manduca sexta”. In: Chem Senses 13, pp. 123–130.

— (1990). “Representation of sex-pheromonal information in the insect brain”. In: In: Doving KB (ed) Proc Xth Internatl Symp Olfaction and Taste (ISOT X). Graphic Communication Systems, Oslo, pp. 142–150.

— (1997). “Coincident stimulation with pheromone components improves temporal pattern resolution in central olfactory neurons”. In: J Neurophysiol 77, pp. 775–781.

Christensen, T., V. Pawlowski, et al. (2000). “Multi-unit recordings reveal context-dependent modulation of synchrony in odor-specific neural ensembles”. In: Nat Neurosci 3, pp. 927–931.

Christensen, T., B. Waldrop, I. Harrow, et al. (1993). “Local interneurons and information processing in the olfactory glomeruli of the moth Manduca sexta”. In: J Comp Physiol A 173, pp. 385–399.

Christensen, T., B. Waldrop, and J. Hildebrand (1998). “Multitasking in the olfactory system: context-dependent responses to odors reveal dual GABA-regulated coding mechanisms in single olfactory projection neurons”. In: J Neurosci 18, pp. 5999–6008.

Froese, A., P. Szyszka, and R. Menzel (2014). “Effect of GABAergic inhibition on odorant concentration coding in mushroom body intrinsic neurons of the honeybee”. In: J Comp Physiol 200, pp. 183–195.

Grosmaitre, X. et al. (2007). “Dual functions of mammalian olfactory sensory neurons as odor detectors and mechanical sensors”. In: Nat Neurosci 10, pp. 348–354.

Han, Q., B. Hansson, and S. Anton (2005). “Interactions of mechanical stimuli and sex pheromone information in antennal lobe neurons of a male moth, Spodoptera littoralis”. In: J Comp Physiol A Neuroethol Sens Neural Behav Physiol. 191, pp. 521–528.

Heinbockel, T., T. Christensen, and J. Hildebrand (1999). “Temporal tuning of odor responses in pheromone-responsive projection neurons in the brain of the sphinx moth Manduca sexta”. In: J Comp Neurol 409, pp. 1–12.

Hildebrand, J., L. Hall, and B. Osmond (1979). “Distribution of binding sites for $^{125}$I-labeled alpha-bungarotoxin in normal and deafferented antennal lobes of Manduca sexta”. In: Proc Natl Acad Sci USA 76, pp. 499–503.

Hildebrand, J. and G. Shepherd (1997). “Mechanisms of Olfactory Discrimination : Converging Evidence for Common Principles Across Phyla”. In: Annu Rev Neurosci 20, pp. 595–631.
Hillier, N. and N. Vickers (2011). “Mixture interactions in moth olfactory physiology: examining the effects of odorant mixture, concentration, distal stimulation, and antennal nerve transection on sensillar responses”. In: Chemical Senses 36, pp. 93–108.

Homberg, U., T. Christensen, and J. Hildebrand (1989). “Structure and Function of the Deutocerebrum in Insects”. In: Annu Rev Entomol. 34, pp. 477–501.

Homberg, U., R. Montague, and J. Hildebrand (1988). “Anatomy of antenno-cerebral pathway in the brain of the sphinx moth Manduca sexta”. In: Cell Tissue Res 254, pp. 255–281.

Hoskins, S. et al. (1986). “Immunocytochemistry of GABA in the antennal lobes of the sphinx moth Manduca sexta”. In: Cell Tissue Res 244, pp. 243–252.

Ito, I. et al. (2008). “Sparse odor representation and olfactory learning”. In: Nat Neurosci 11, pp. 1177–1184.

Iwata, R., H. Kiyonari, and T. Imai (2017). “Mechanosensory-Based Phase Coding of Odor Identity in the Olfactory Bulb”. In: Neuron 96, pp. 1139–1152.

Jarman, A. (2002). “Studies of mechanosensation using the fly”. In: Human Mol Gen 11, pp. 1215–1218.

Jefferys, J. (1995). “Nonsynaptic modulation of neuronal activity in the brain: electric currents and extracellular ions”. In: Physiological Reviews 75, pp. 689–723.

Joerges, J. et al. (1997). “Representations of Odours and Odour Mixtures Visualized in the Honeybee Brain”. In: Nature 387, pp. 285–288.

Kaißling, K. and E. Kramer (1990). “Sensory basis of pheromone-mediated orientation in moths”. In: Verh Dtsch Zool Ges 83, pp. 109–131.

Kanzaki, R. et al. (1989). “Physiology and morphology of projection neurons in the antennal lobe of the male moth Manduca sexta”. In: J Comp Physiol A 165, pp. 427–453.

Keil, T. (1989). “Fine structure of the pheromone-sensitive sensilla on the antenna of the hawkmoth, Manduca sexta”. In: Tissue & Cell 21, pp. 139–151.

Klang, T. and J. Hildebrand (1985). “Gamma-aminobutyric acid in the central nervous system of metamorphosing and mature Manduca sexta”. In: Insect Biochem 15, pp. 667–675.

Kramer, E. (1992). “Attractivity of pheromone surpassed by time-patterned application of two nonpheromone compounds”. In: J Insect Behav 5, pp. 83–97.

Kramich, S. and M. Stengl (2008). “Cyclic Nucleotide-Activated Currents in Cultured Olfactory Receptor Neurons of the Hawkmoth Manduca sexta”. In: Journal of Neurophysiology 100, pp. 2866–2877.
Lee, J. and N. Strausfeld (1990). “Structure, distribution and number of surface sensilla and their receptor cells on the olfactory appendage of the male moth Manduca sexta”. In: *J Neurocytol* 19, pp. 519–538.

Lei, H., T. Christensen, and J. Hildebrand (2002). “Local inhibition modulates odor-evoked synchronization of glomerulus-specific output neurons”. In: *Nat Neurosci* 5, pp. 557–565.

— (2004). “Spatial and temporal organization of ensemble representations for different odor classes in the moth antennal lobe”. In: *J Neurosci* 24, pp. 11108–11119.

Lei, H. and N. Vickers (2008). “Central processing of natural odor mixtures in insects”. In: *J Chem Ecol* 34, pp. 915–927.

Lei, H., Y. Yu, et al. (2016). “Intrinsic and Network Mechanisms Constrain Neural Synchrony in the Moth Antennal Lobe”. In: *Front. Physiol.* 7. doi:10.3389/fphys.2016.00080, p. 80.

Malnic, B. et al. (1999). “Combinatorial Receptor Codes for Odors”. In: *Cell* 96, pp. 713–723.

Mamiya, A. and M. Dickinson (2015). “Antennal Mechanosensory Neurons Mediate Wing Motor Reflexes in Flying Drosophila”. In: *J Neurosci.* 35, pp. 7977–7991.

Matsumoto, S. and J. Hildebrand (1981). “Olfactory mechanisms in the moth Manduca sexta: response characteristics and morphology of central neurons in the antennal lobes”. In: *Proc R Soc Lond B* 213, pp. 249–277.

Mercer, A. and J. Hildebrand (2002a). “Developmental Changes in the Density of Ionic Currents in Antennal-Lobe Neurons of the Sphinx moth, Manduca Sexta”. In: *J Neurophysiol* 87, pp. 2664–1675.

— (2002b). “Developmental Changes in the Electrophysiological Properties and Response Characteristics of Manduca Antennal-Lobe Neurons”. In: *J Neurophysiol* 87, pp. 2650–2663.

Murlis, J., J. Elkinton, and R. Carde (1992). “Odor plumes and how insects use them”. In: *Ann Rev Entomol* 37, pp. 505–532.

Murlis, J. and C. Jones (1981). “Fine-scale structure of odor plumes in relation to insect orientation to distant pheromone and other attractant sources”. In: *Physio Ento* 6, pp. 71–86.

Ng, M. et al. (2002). “Transmission of Olfactory Information Between Three Populations of Neurons in the Antennal Lobe of the Fly”. In: *Neuron* 36, pp. 463–474.

Pasternak, Z., F. Bartumeus, and F. Grasso (2009). “Levy-taxis: a novel search strategy for finding odor plumes in turbulent flow-dominated environments”. In: *J PhyA-Math Theo* 42.
Patella, P. and R. Wilson (2018). “Functional Maps of Mechanosensory Features in the Drosophila Brain”. In: Current Biology 28, pp. 1189–1203.

Pyk, P. et al. (2006). “An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search”. In: Auto rob 20, pp. 197–213.

Reisenman, C. et al. (2004). “Enantioselectivity of Projection Neurons Innervating Identified Olfactory Glomeruli”. In: J Neurosci 24, pp. 2602–2611.

Sane, S. et al. (2007). “Antennal mechanosensors mediate flight control in moths”. In: Science 315, pp. 863–866.

Silbering, A. and C. Galizia (2007). “Processing of odor mixtures in the Drosophila antennal lobe reveals both global inhibition and glomerulus-specific interactions”. In: J Neurosci 27, pp. 11966–11977.

Tao, L. et al. (2004). “An egalitarian network model for the emergence of simple and complex cells in visual cortex”. In: Proc Natl Acad Sci USA 101, p. 366.

Vergassola, M., E. Villermaux, and B. Shraiman (2007). “‘Infotaxis’ as a strategy for searching without gradients”. In: Nature 445, pp. 406–409.

Vickers, N. (2000). “Mechanisms of animal navigation in odor plumes”. In: Biol Bull 198, pp. 203–212.

Vickers, N. and T. Baker (1994). “Reiterative responses to single strands of odor promote sustained upwind flight and odor source location by moths”. In: Proc Nat Aca Sci USA 91, pp. 5756–5760.

Vickers, N. and T. Christensen (1998). “A Combinatorial Model of Odor Discrimination using a Small Array of Contiguous, Chemically Defined Glomeruli”. In: Ann NY Acad Sci 855, pp. 514–516.

Vickers, N., T. Christensen, and J. Hildebrand (1998). “Combinatorial Odor Discrimination in the Brain: Attractive and Antagonistic Odor Blends are Represented in Distinct Combinations of Uniquely Identifiable Glomeruli”. In: J Comp Neurol 400, pp. 35–36.

Waldrop, B., T. Christensen, and J. Hildebrand (1987). “GABA-mediated synaptic inhibition of projection neurons in the antennal lobes of the sphinx moth, Manduca sexta”. In: J Comp Physiol A 161, pp. 23–32.

Wang, J. et al. (2003). “Two-Photon Calcium Imaging Reveals an Odor-Evoked Map of Activity in the Fly Brain”. In: Cell 112, pp. 271–282.

Whitehead, A. and J. Larsen (1976). “Ultrastructure of the contact chemoreceptors of Apis mellifera L. (Hymenoptera : Apidae)”. In: Int J Insect Morphol Embryol 5, pp. 301–315.
Yamagata, N. et al. (2009). “Differential odor processing in two olfactory pathways in the honeybee”. In: *Frontiers in Systems Neuroscience* 3, doi:10.3389.

Zalc, B. (2016). “The acquisition of myelin: An evolutionary perspective”. In: *Brain Research* 1641, pp. 4–10.
APPENDIX
Means and standard deviations are taken over 100 trials. Normalized firing rates (as in figure 3.5) are determined by dividing the firing rate during a period of interest (in our case, the last 500ms of an odor pulse) by the firing rate during 1 second of background activity (both averaged over 100 trials). Thus, a normalized firing rate greater than 1 represents enhanced spiking activity (over background) while a normalized firing rate less than 1 represents reduced spiking activity (relative to background).

Odor discrimination rates within the model (as in figure 3.6) are determined using a linear classification scheme. To calculate the time-dependent ability of the network to discriminate among $n$ stimuli, we split the total simulation time into nonoverlapping 10ms time bins; for stimulus $i \in [1, n]$ and time bin $k$, we construct a template for stimulus $i$ in time bin $k$ as the 60-dimensional vector of PN firing rates within time bin $k$ averaged over 100 trials of stimulus $i$ presentation. This yields a total of $100n$ odor trials and $n$ odor templates for time bin $k$. For an individual stimulus trial (for, say, stimulus $l \in [1, n]$), we designate the trial as ‘correctly classified’ within time bin $k$ if the Euclidean distance between the 60-dimensional vector of PN firing rates for that trial and each of the $n$ odor templates is smallest for the template for odor $l$; otherwise, we designate the trial as ‘incorrectly classified’. The classification rate of the network in time bin $k$ is then determined as the number of ’correctly classified’ odor trials within time bin $k$ divided by $100n$.

Numerical simulations were carried out using the Euler Method with a time step of $\Delta t = 0.1\text{ms}$. Stability was verified by observing qualitatively similar behavior with a timestep
an order of magnitude smaller, as seen in figure A.1.

Model code was written in C++ with data analysis and plotting carried out in Matlab. Model code is available in Appendix C.

Figure A.1 Raster plots for timesteps of $\Delta t = .1$ms (left) and $\Delta t = .01$ms (right). Qualitative behavior of the two simulations are very similar, representing simulation stability with a timestep of $\Delta t = .1$. 
Appendix B

Varying Pulse Length and Frequency

This section contains information on the most recent investigation of variable pulse length and pulse frequency. I did not feel this section is complete enough to include in the regular results section, but there is some interesting preliminary data that will be discussed here.

B.1 Pulse Length

Experimental studies show that PNs are able to encode the length of olfactory stimulus through the length of their own phase II firing period (Christensen, Waldrop, and Hildebrand, 1998; Lei, Christensen, and Hildebrand, 2002). To see if our model reproduces this behavior, we created simulations with variable olfactory pulse lengths. To determine the start of the PN response to olfactory stimulus, we simply recorded the first spike following the IHP. The cutoff for the firing response was assigned when the interspike interval was greater than three times the interspike interval of the first 5 spikes following the IHP. Choosing such a high cutoff for the interspike interval serves not only to create a conservative guess on the length of the PN response, but also serves to account for the fact that firing rate is highest immediately upon stimulus presentation and slowly declines throughout prolonged stimulation. The result of this cutoff methodology is displayed in figure B.1, and the ability of a single PN to encode pulse length is exemplified in figure B.2.

The high correlation between the stimulus pulse length and the firing length of the chosen PN illustrates that our model indeed replicates the experimental observation. This feature is especially important because in reality insect tracking odors are required to resolve the
Figure B.1 Raster plot in response to a 500 ms odor stimulus (represented by black bar). PN firing length was determined by subtracting the time between the final fire, labeled by a red tick, and the first fire following the IHP.

Figure B.2 Scatter plot displaying the relationship between pulse length and firing length of a single PN. A high correlation coefficient suggests that pulse length is being well encoded by this particular PN.
complex dynamics of short odor plumes and pulses, rather than single, long-lasting odors.

B.2 Pulse Frequency

In addition to encoding pulse length, we are interested in determining the PNs’s ability to follow pulses at a fixed frequency. As stated previously, insects are required to resolve the complex dynamics of pulses of stimulus rather a single, long-lasting stimulus. Because the frequency of these pulses contains information such as the distance from the odor source, accurately following pulse frequency is a requirement of the AL. To model this, we simulate short 50 ms pulses at varying frequencies. At this time we lack a methodology for quantifying a frequency cutoff, though a qualitative picture of model dynamics can be observed in figure B.3.

As can be seen in B.3A, when presented with odor pulses at a low enough frequency nearly all PNs within the model are able to track pulses precisely. However, as pulse frequency increases fewer PNs are able to precisely track frequency and instead display a response very similar to a single, long-lasting stimulus. However, when presented with only mechanosensory stimulus, as shown in figure B.3B, PNs tend to be able to track pulses at a much higher frequency than in the odor only paradigm. This is likely due to the pervasive slow inhibition throughout the model which allows for suppression of firing between the stimulus pulses. Similar to how this pervasive inhibition leads to the transient response of PNs under prolonged stimulus, this inhibition likely enhances the observed pulse frequency tracking. Interestingly, when combining both olfactory and mechanosensory stimuli, as seen in figure B.3C, it appears that PNs in glomeruli that are not stimulated by the odor actually pulse track better than those in stimulated glomeruli. This leads to the natural hypothesis that there is a division of labor within the AL; PNs within stimulated glomeruli are primarily responsible for encoding pulse length and odor identity, while those in nonstimulated glomeruli are responsible for tracking the pulse frequency. Further modeling and analysis will solidify
Figure B.3 Pictured are rasters of AL response to stimuli pulsed at 3 Hz (left) and 7 Hz (right) where the stimuli consist of odor only (A), mechanosensory only (B), odor and mechanosensory (C) and normalized odor and mechanosensory (D). With odor alone (A), most PNs are able to track pulse frequency at low rates but elicit responses similar to single, long-lasting stimuli when the rate is too high. In mechanosensory only (B), PNs tend to track better than odor alone, likely due to the pervasive slow inhibition in the neural network. When adding odor and mechanosensory (C), it appears that nonstimulated glomeruli track frequency very well, while stimulated glomeruli do not.
these predictions, and will hopefully serve to elucidate some of the mechanisms behind the complex dynamical behavior observed during pulse tracking.
Appendix C

The Model Code

In an effort to be organized, the model code is divided into various files by function. Upon compiling the code, these files are linked together to create the cohesive program. The entire code was written in C++.

C.1 Parameters

C.1.1 Parameters.h

```cpp
#pragma once

#include <random>
#include <vector>
using namespace std;

extern const int mode;
extern const int trials;
extern int cur_brain;

// Global Fires
extern FILE *fires;
extern FILE *fires;
//static default_random_engine generator;
static default_random_engine generator;

//#warning

//timing
```
extern double t;
extern const double timestep;
extern const double endtime;

// Network Architecture

extern const int PN_Neurons;
extern const int LN_Neurons;
extern const int number_glomeruli;

// Connection Probabilities

extern const double PN_to_PN_prob;
extern const double PN_to_LN_prob;
extern const double LN_to_PN_prob;
extern const double LN_to_LN_prob;
extern double inter_LN_to_PN_prob;

// Neuron Parameters

extern const double leak;
extern const double vpn;
extern const double vln;
extern const unsigned int refractory_period;
extern const unsigned int refractory_counts;
extern const double tau_excitation;
extern const double tau_fast_inhibition;
extern const double tau_slow_inhibition;
extern const double tau_sk;
extern const double tau_input;
extern const double PN_to_LN_strength;
extern const double PN_to_PN_strength;
extern const double LN_to_LN_fast_strength;
extern const double LN_to_LN_slow_strength;
extern double LN_to_PN_fast_strength;
extern double LN_to_PN_slow_strength;
extern double sk_mean;
extern double sk_std_dev;
extern const double input_to_PN_strength;
extern const double input_to_LN_strength;

//SK Calculation Parameters
extern const double sk_rise; // half rise time
extern const double sk_rise_tau;

//Odor Parameters
extern const double t_rise;
extern const double rise_tau;
extern const double background_odor_rate; // units of ms\(^{-1}\)
extern double max_odor_rate;
extern const double odor_tau;

//Air Parameters
extern const double air_t_rise; // half rise time
extern const double air_rise_t tau;
extern double max_air_rate;
extern const double air_t tau;

C.1.2 Parameters.cpp
#pragma warning ( disable : 4996 )
#include <cstdio >
#include <cstdlib >
#include <ctime >
#include <vector >
#include <string >
#include " Parameters . h"
using namespace std ;

extern const int mode = 8 ;
extern const int trials = 1 ;
extern int cur_brain = 0 ;

// Global Files
// extern FILE * fires = fopen ( "fires . txt " , "w" );
extern FILE * fires = NULL ;

// timing

double t = 0 ;
extern const double timestep = .1 ;
extern const double endtime = 4000.0 ;

// Network Architecture

extern const int PN_Neurons = 10 ;
extern const int LN_Neurons = 6 ;
extern const int number_glomeruli = 6 ;

// Connection Probabilities

extern const double PN_to_PN_prob = .75 ;
extern const double PN_to_LN_prob = .75;
extern const double LN_to_PN_prob = .38;
extern const double LN_to_LN_prob = .25;
extern double inter_LN_to_PN_prob = .38; // old value: .55

// Neuron Parameters

extern const double vpn = 14.0 / 3.0;
extern const double vln = -2.0 / 3.0;
extern const double leak = .05;
extern const unsigned int refractory_period = 2;
extern const unsigned int refractory_counts = refractory_period / timestep;
extern const double tau_excitation = 2;
extern const double tau_fast_inhibition = 2;
extern const double tau_slow_inhibition = 750;
extern const double tau_sk = 250; //400
extern const double tau_input = 2;
extern const double PN_to_LN_strength = .006; // lit .0055
extern const double PN_to_PN_strength = .01; // lit .008
extern const double LN_to_LN_fast_strength = .015; // lit .0146
extern const double LN_to_LN_slow_strength = .04; // lit .0106
extern double LN_to_PN_fast_strength = 0.0169; // old .013 lit .0134
extern double LN_to_PN_slow_strength = .0338; // old .026 lit .028
extern double sk_mean = .5;
extern double sk_std_dev = .2;
extern const double input_to_PN_strength = .0040; // lit .0023
extern const double input_to_LN_strength = .0031; // lit .0017

// SK Calculation Parameters

extern const double sk_rise = 25; // half rise time
extern const double sk_rise_tau = 5 / sk_rise;
//Odor Parameters

extern const double t_rise = 35; // half rise time
extern const double rise_tau = 5 / t_rise;
extern const double background_odor_rate = 3.6; // units of ms^{-1} lit:3.8
extern double max_odor_rate = 3.6;
extern const double odor_tau = 384;

// Air Parameters

extern const double air_t_rise = 300; // 500 half rise time
extern const double air_rise_tau = 5 / air_t_rise;
extern double max_air_rate = max_odor_rate/2;
extern const double air_tau = 384;

C.2 PNs

C.2.1 PN_Neuron.h

#pragma once
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
using namespace std;
class LN_Neuron;
class PN_Neuron
{
public:

PN_Neuron();
~PN_Neuron();

void add_PN_connection(PN_Neuron & neuron);
void add_LN_connection(LN_Neuron & neuron);

void Integrate();  // Use Euler Method to calculate voltage

void fire_to_PN();  // send signal to PN connections

void fire_to_LN();  // send signal to LN connections

void receiving_excitation(double strength);  // excitation signal received from other neurons
void receiving_fast_inhibition(double strength);  // fast inhibition signal received from other neurons
void receiving_slow_inhibition(double strength);  // slow inhibition signal received from other neurons

void receiving_input(double strength);

double calculateSk();

double get_voltage();

vector<double> get_voltage_list();
vector<double> get_excitation_list();
vector<double> get_slow_inhibition_list();
vector<double> get_fast_inhibition_list();
vector<double> get_sk_list();
vector<double> get_input_list();

void reset();

private:

double voltage;

unsigned int refractory_period_internal;

double excitation;

double t_excitation;  // decay rate of
excitation signal

double fast_inhibition;
double t_fast_inhibition;  // decay rate of fast inhibition signal
double slow_inhibition;
double t_slow_inhibition;  // decay rate of slow inhibition signal
double sk;
double t_sk;
double input;
double t_input;
double sk_strength;
vector<PN_Neuron*> PN_connections;
vector<LN_Neuron*> LN_connections;
vector<double> voltage_list;
vector<double> excitation_list;
vector<double> fast_inhibition_list;
vector<double> slow_inhibition_list;
vector<double> sk_list;
vector<double> input_list;
vector<double> sk_times;

C.2.2 PN_Neuron.cpp

#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "Parameters.h"
#include "PN_Neuron.h"
#include "LN_Neuron.h"

using namespace std;

PN_Neuron::PN_Neuron()
{
    //-------------------Euler Method Parameters
    voltage = 0; //initialize voltage of Neuron
    refractory_period_internal = 0; //initialize refractory period
    excitation = 0; //initialize excitatory current input
    t_excitation = 0; //time of most recent excitation input
    fast_inhibition = 0; //initialize fast inhibition current input
    t_fast_inhibition = 0; //time of most recent fast inhibition input
    slow_inhibition = 0; //initialize slow inhibition current input
    t_slow_inhibition = 0; //time of most recent slow inhibition input
    //initialize leak current
    sk = 0; //initialize sk inhibition (PN neurons only)
    t_sk = 0;
    sk_times = {};
    input = 0; //initialize input from
stimuli
t_input = 0;
  // static default_random_engine generator;
  // generator.seed(1);
  normal_distribution<double> distribution(sk_mean, sk_std_dev);
  sk_strength = distribution(generator);
  if (sk_strength < 0) {
    sk_strength = 0;
  }
  printf("%f\n", sk_strength);

  // Neural Network Parameters
  PN_connections = {}; // PN connections from
                          // this neuron to neurons in list
  LN_connections = {}; // Ln connections from
                        // this neuron to neurons in list
}

PN_Neuron::~PN_Neuron()
{
}

void PN_Neuron::Integrate() {
  if (refractory_period_internal == 0) {
    double dV = -excitation * exp(-(t - t_excitation) / tau_excitation) / 
                tau_excitation * (voltage - vpn) 
                - fast_inhibition * exp(-(t - t_fast_inhibition) / tau_fast_inhibition) / 
                  tau_fast_inhibition * (voltage - vln) 
                - slow_inhibition * exp(-(t - t_slow_inhibition) / tau_slow_inhibition) / 
                  tau_slow_inhibition * (voltage - vln) 
                - leak * voltage - input * (voltage - vpn) * exp(-(t - t_input) / 
                  tau_input) / tau_input - calculateSk() / sk * exp(-(t - t_sk) / tau_sk) / 
                  }
tau_sk*/ (voltage - vln); // Calculation of dV
voltage += dV * timestep; // recalculate voltage

if (voltage >= 1.0) { // condition for neuron firing
    voltage = 0;
    fprintf(fires, "1, ");
    refractory_period_internal = refractory_counts; // refractory period set for 2 ms
    fire_to_PN();
    fire_to_LN();
    // sk = sk * exp(-(t - t_sk) / tau_sk) + sk_strength;
    // t_sk = t;
    sk_times.push_back(t);
    voltage_list.push_back(1);
} else {
    voltage_list.push_back(voltage);
    fprintf(fires, "0,");
}
else {
    voltage_list.push_back(voltage);
    refractory_period_internal = 1;
    fprintf(fires, "0,");
}

// ---------------------- File Data ----------------------
excitation_list.push_back(excitation * exp(-(t - t_excitation) / tau_excitation) / tau_excitation);
fast_inhibition_list.push_back(fast_inhibition * exp(-(t - t_fast_inhibition) / tau_fast_inhibition) / tau_fast_inhibition);
slow_inhibition_list.push_back(slow_inhibition * exp(-(t - t_slow_inhibition) / tau_slow_inhibition);


```cpp
) / tau_slow_inhibition) / tau_slow_inhibition);
    sk_list.push_back(sk * exp(-(t - t_sk) / tau_sk) / tau_sk);
    input_list.push_back(input * exp(-(t - t_input) / tau_input) / tau_input);
}

void PN_Neuron::add_PN_connection(PN_Neuron & neuron) {
    PN_connections.push_back(&neuron);
}

void PN_Neuron::add_LN_connection(LN_Neuron & neuron) {
    LN_connections.push_back(&neuron);
}

void PN_Neuron::receiving_excitation(double strength) {
    excitation = excitation * exp(-(t - t_excitation) / tau_excitation) +
        strength;
    t_excitation = t;
}

void PN_Neuron::receiving_fast_inhibition(double strength) {
    fast_inhibition = fast_inhibition * exp(-(t - t_fast_inhibition) /
        tau_fast_inhibition) + strength;
    t_fast_inhibition = t;
}

void PN_Neuron::receiving_slow_inhibition(double strength) {
    slow_inhibition = slow_inhibition * exp(-(t - t_slow_inhibition) /
        tau_slow_inhibition) + strength;
    t_slow_inhibition = t;
}

void PN_Neuron::receiving_input(double strength) {
```
input = input * exp(- (t - t_input) / tau_input) + strength;

108  t_input = t;
109 
110 }
111 
112 void PN_Neuron::fire_to_PN() {
113     for (auto i : PN_connections) {
114         i->receiving_excitation(PN_to_PN_strength);
115     }
116 }
117 
118 void PN_Neuron::fire_to_LN() {
119     for (auto i : LN_connections) {
120         i->receiving_excitation(PN_to_LN_strength);
121     }
122 }
123 
124 double PN_Neuron::get_voltage() {
125     return voltage;
126 }
127 
128 vector<double> PN_Neuron::get_voltage_list() {
129     return voltage_list;
130 }
131 
132 vector<double> PN_Neuron::get_excitation_list() {
133     return excitation_list;
134 }
135 
136 vector<double> PN_Neuron::get_slow_inhibition_list() {
137     return slow_inhibition_list;
138 }
vector<double> PN_Neuron::get_fast_inhibition_list() {
    return fast_inhibition_list;
}

vector<double> PN_Neuron::get_sk_list() {
    return sk_list;
}

vector<double> PN_Neuron::get_input_list() {
    return input_list;
}

double PN_Neuron::calculateSk() {
    double to_return = 0;
    vector<int> to_delete = {};
    for (size_t i = 0; i < size(sk_times); i++) {
        if (t - sk_times[i] < sk_rise * 2) {
            to_return += sk_strength * exp((sk_rise_tau*((t - sk_times[i]) - sk_rise))
            / (1 + exp((sk_rise_tau*((t - sk_times[i]) - sk_rise))));
        } else {
            to_delete.push_back(i);
            sk = sk * exp(-(t - t_sk) / tau_sk) + sk_strength;
            t_sk = t;
        }
    }
    for (int i : to_delete) {
        sk_times.erase(sk_times.begin() + i);
    }
    to_return += sk * exp(-(t - t_sk) / tau_sk);
    return to_return / tau_sk;
}
void PN_Neuron::reset() {
    voltage = 0; // initialize voltage of Neuron
    refractory_period_internal = 0; // initialize refractory period
    excitation = 0; // initialize excitatory current input
    t_excitation = 0; // time of most recent excitatory input
    fast_inhibition = 0; // initialize fast inhibition current input
    t_fast_inhibition = 0; // time of most recent fast inhibition input
    slow_inhibition = 0; // initialize slow inhibition current input
    t_slow_inhibition = 0; // time of most recent slow inhibition input
    leak_current
    sk = 0; // initialize sk inhibition (PN neurons only)
    t_sk = 0;
    input = 0; // initialize input from stimuli
    t_input = 0;
    voltage_list.clear();
    excitation_list.clear();
    fast_inhibition_list.clear();
    slow_inhibition_list.clear();
    sk_list.clear();
    input_list.clear();
    sk_times.clear();}
C.3 LNs

C.3.1 LN_Neuron.h

```cpp
#pragma once
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "PN_Neuron.h"
using namespace std;

class LN_Neuron
{
public:
  LN_Neuron();  
  ~LN_Neuron();
  void add_PN_connection(PN_Neuron & neuron);
  void add_LN_connection(LN_Neuron & neuron);
  void Integrate();  // Use Euler Method to calculate voltage
  void fire_to_PN();  // send signal to PN connections
  void fire_to_LN();  // send signal to LN connections
  void receiving_excitation(double strength);  // excitation signal received from other neurons
  void receiving_fast_inhibition(double strength);  // fast
};
```
void receiving_slow_inhibition(double strength); // slow inhibition signal received from other neurons

void receiving_input(double strength);

double get_voltage();

vector<double> get_voltage_list();

vector<double> get_excitation_list();

vector<double> get_slow_inhibition_list();

vector<double> get_fast_inhibition_list();

vector<double> get_input_list();

void reset();

private:

double voltage;

unsigned int refractory_period_internal;

double excitation;

double t_excitation; // decay rate of excitation signal

double fast_inhibition;

double t_fast_inhibition; // decay rate of fast inhibition signal

double slow_inhibition;

double t_slow_inhibition; // decay rate of slow inhibition signal

double input;

double t_input;

vector<PN_Neuron*> PN_connections;

vector<LN_Neuron*> LN_connections;

vector<double> voltage_list;

vector<double> excitation_list;

vector<double> fast_inhibition_list;

vector<double> slow_inhibition_list;

vector<double> input_list;
C.3.2 LN_Neuron.cpp

```cpp
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "Parameters.h"
#include "PN_Neuron.h"
#include "LN_Neuron.h"
using namespace std;

LN_Neuron::LN_Neuron()
{
    //------------------Euler Method Parameters
    voltage = 0; //initialize voltage of Neuron
    refractory_period_internal = 0; //initialize refractory period
    excitation = 0; //initialize excitatory current input
    t_excitation = 0; //time of most recent excitation input
    fast_inhibition = 0; //initialize fast inhibition current input
    t_fast_inhibition = 0; //time of most recent fast inhibition input
```
slow_inhibition = 0; //initialize slow inhibition current input

t_slow_inhibition = 0; //time of most recent slow inhibition input

input = 0; //initialize input from stimuli

t_input = 0;

//Neural Network Parameters

PN_connections = {}; //PN connections from this neuron to neurons in list

LN_connections = {}; //Ln connections from this neuron to neurons in list

LN_Neuron::~LN_Neuron()
{
}

void LN_Neuron::Integrate() {
if (refractory_period_internal == 0) {

double dV = -excitation * exp(-(t - t_excitation) / tau_excitation) /
    tau_excitation * (voltage - vpn)
    - fast_inhibition * exp(-(t - t_fast_inhibition) / tau_fast_inhibition)
    / tau_fast_inhibition * (voltage - vln)
    - slow_inhibition * exp(-(t - t_slow_inhibition) / tau_slow_inhibition)
    / tau_slow_inhibition * (voltage - vln)
    - leak * voltage - input * (voltage - vpn)* exp(-(t - t_input) /
    tau_input) / tau_input; //Calculation of dV

voltage += dV * timestep; //recalculate
```c
if (voltage >= 1.0) {
    // condition for neuron firing
    voltage = 0;
    refractory_period_internal = refractory_counts; // refractory period set for 2 ms
    fire_to_PN();
    fire_to_LN();
    voltage_list.push_back(1);
    fprintf(fires, "1, " );
}
else {
    voltage_list.push_back(voltage);
    fprintf(fires, "0, ");
}
else {
    refractory_period_internal -= 1;
    voltage_list.push_back(voltage);
    fprintf(fires, "0, " );
}
}

// File Data
excitation_list.push_back(excitation * exp(-(t - t_excitation) / tau_excitation) / tau_excitation);
fast_inhibition_list.push_back(fast_inhibition * exp(-(t - t_fast_inhibition) / tau_fast_inhibition) / tau_fast_inhibition);
slow_inhibition_list.push_back(slow_inhibition * exp(-(t - t_slow_inhibition) / tau_slow_inhibition) / tau_slow_inhibition);
input_list.push_back(input);
```
```c++
void LN_Neuron::add_PN_connection(PN_Neuron & neuron) {
    PN_connections.push_back(&neuron);
}

void LN_Neuron::add_LN_connection(LN_Neuron & neuron) {
    LN_connections.push_back(&neuron);
}

void LN_Neuron::receiving_excitation(double strength) {
    excitation = excitation * exp(-(t - t_excitation) / tau_excitation) + strength;
    t_excitation = t;
}

void LN_Neuron::receiving_fast_inhibition(double strength) {
    fast_inhibition = fast_inhibition * exp(-(t - t_fast_inhibition) / tau_fast_inhibition) + strength;
    t_fast_inhibition = t;
}

void LN_Neuron::receiving_slow_inhibition(double strength) {
    slow_inhibition = slow_inhibition * exp(-(t - t_slow_inhibition) / tau_slow_inhibition) + strength;
    t_slow_inhibition = t;
}

void LN_Neuron::receiving_input(double strength) {
    input = input * exp(-(t - t_input) / tau_input) + strength;
    t_input = t;
}

void LN_Neuron::fire_to_PN() {
```
for (auto i : PN_connections) {
    i->receiving_fast_inhibition(LN_to_PN_fast_strength);
    i->receiving_slow_inhibition(LN_to_PN_slow_strength);
}

void LN_Neuron::fire_to_LN() {
    for (auto i : LN_connections) {
        i->receiving_fast_inhibition(LN_to_LN_fast_strength);
        i->receiving_slow_inhibition(LN_to_LN_slow_strength);
    }
}

double LN_Neuron::get_voltage() {
    return voltage;
}

vector<double> LN_Neuron::get_voltage_list() {
    return voltage_list;
}

vector<double> LN_Neuron::get_excitation_list() {
    return excitation_list;
}

vector<double> LN_Neuron::get_slow_inhibition_list() {
    return slow_inhibition_list;
}

vector<double> LN_Neuron::get_fast_inhibition_list() {
    return fast_inhibition_list;
}
vector<double> LN_Neuron::get_input_list() {
    return input_list;
}

void LN_Neuron::reset() {
    voltage = 0; // initialize voltage of Neuron
    refractory_period_internal = 0; // initialize refractory period
    excitation = 0; // initialize excitatory current input
    t_excitation = 0; // time of most recent excitation input
    fast_inhibition = 0; // initialize fast inhibition current input
    t_fast_inhibition = 0; // time of most recent fast inhibition input
    slow_inhibition = 0; // initialize slow inhibition current input
    t_slow_inhibition = 0; // time of most recent slow inhibition input
    input = 0; // initialize input from stimuli
    t_input = 0;
    voltage_list.clear();
    excitation_list.clear();
    fast_inhibition_list.clear();
    slow_inhibition_list.clear();
    input_list.clear();
}
C.4  Glomeruli

C.4.1  Glomerulus.h

```cpp
#pragma once
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "LN_Neuron.h"
#include "PN_Neuron.h"
#include "Functions.h"
using namespace std;

class Glomerulus {
public:
    Glomerulus();
    ~Glomerulus();
    void intraconnec(t());
    void update();
    void set_PN_signal(double s);
    void set_LN_signal(double s);
    void reset();

    //-------------creation-------------
    vector<PN_Neuron> PNs;
    vector<LN_Neuron> LNs;

private:
    double signalPN;
    double signalLN;
};
```
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "Glomerulus.h"
#include "Parameters.h"
using namespace std;

Glomerulus::Glomerulus() {
    signalPN = background_odor_rate;
    signalLN = background_odor_rate;
    //---creating neurons---
    PNs.resize(PN_Neurons);
    LNs.resize(LN_Neurons);
    //---connecting neurons---
    intraconnect();
}

Glomerulus::~Glomerulus() {
}

void Glomerulus::intraconnect() {

for (size_t i = 0; i < size(PNs); i++)
{
    for (size_t j = 0; j < size(PNs); j++)
    {
        if (i != j) {
            if (random_check(PN_to_PN_prob)) {
                PNs[i].add_PN_connection(PNs[j]);
            }
        }
    }
}

for (size_t j = 0; j < size(LNs); j++)
{
    if (random_check(PN_to_LN_prob)) {
        PNs[i].add_LN_connection(LNs[j]);
    }
}

for (size_t i = 0; i < size(LNs); i++)
{
    for (size_t j = 0; j < size(PNs); j++)
    {
        if (random_check(LN_to_PN_prob)) {
            LNs[i].add_PN_connection(PNs[j]);
        }
    }
    for (size_t j = 0; j < size(LNs); j++)
    {
        if (i != j) {
            if (random_check(LN_to_LN_prob)) {
                LNs[i].add_LN_connection(LNs[j]);
            }
        }
    }
}
```cpp
void Glomerulus::update() {
    for (size_t i = 0; i < size(PNs); i++)
    {
        if (random_check(signalPN*timestep)) {
            PNs[i].receiving_input(input_to_PN_strength);
        }
    }

    for (size_t i = 0; i < size(LNs); i++)
    {
        if (random_check(signalLN*timestep)) {
            LNs[i].receiving_input(input_to_LN_strength);
        }
    }

    for (size_t i = 0; i < size(PNs); i++)
    {
        PNs[i].Integrate();
    }

    for (size_t i = 0; i < size(LNs); i++)
    {
        LNs[i].Integrate();
    }

    signalPN = background_odor_rate;
    signalLN = background_odor_rate;
}

void Glomerulus::set_PN_signal(double s) {
    signalPN += s;
}
```
void Glomerulus::set_LN_signal(double s) {
    signalLN += s;
}

void Glomerulus::reset() {
    for (size_t i = 0; i < size(PNs); i++)
    {
        PNs[i].reset();
    }
    for (size_t i = 0; i < size(LNs); i++)
    {
        LNs[i].reset();
    }
}

C.5 Antennal Lobe

C.5.1 Antennal_Lobe.h

#pragma once
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "Glomerulus.h"
#include "Parameters.h"
#include "Odor_Pulse.h"
#include "Air_Puff.h"
#include "Parameters.h"
```cpp
using namespace std;

class Antennal_Lobe
{
public:
    Antennal_Lobe();
    ~Antennal_Lobe();
    void interconnect();
    void update();
    void write_times(string dir);
    void write_voltage(string dir);
    void write_excitation(string dir);
    void write_slow_inhibition(string dir);
    void write_fast_inhibition(string dir);
    void write_sk(string dir);
    void write_input(string dir);
    void write_all(string dir);
    void add_odor(Odor_Pulse & odor);
    void clear_odors();
    void add_air(Air_Puff & air);
    void clear_air();
    void reset();
    void hard_reset();

    //-------------writeup-----------------------------

    vector<double> times = {};

private:
    vector<Glomerulus> glomeruli;
    vector<Odor_Pulse*> odors;
    vector<Air_Puffs> airs;
```
C.5.2 Antennal_Lobe.cpp

```cpp
#pragma warning (disable : 4996)
#include "Antennal_Lobe.h"

Antennal_Lobe::Antennal_Lobe() {
    glomeruli.resize(number_glomeruli);
    odors = {};
    interconnect();
}

Antennal_Lobe::~Antennal_Lobe() {
}

void Antennal_Lobe::interconnect() {
    for (size_t i = 0; i < size(glomeruli); i++)
        {
            for (size_t j = 0; j < size(glomeruli); j++)
                {
                    for (size_t k = 0; k < size(glomeruli[i].LNs); k++)
                        {
                            for (size_t l = 0; l < size(glomeruli[j].PNs); l++)
```
if (random_check(inter_LN_to_PN_prob)) {
    glomeruli[i].LNs[k].add_PN_connection(glomeruli[j].PNs[l]);
}
}
}

void Antennal_Lobe::update() {
    for (auto odor : odors) {
        odor->update();
        for (int glom : odor->get_glomeruli()) {
            glomeruli[glom].set_PN_signal(odor->get_PN_signal());
            glomeruli[glom].set_LN_signal(odor->get_LN_signal());
        }
    }
    for (auto air : airs) {
        air->update();
        for (int glom = 0; glom < number_glomeruli; glom++) {
            glomeruli[glom].set_PN_signal(air->get_PN_signal());
            glomeruli[glom].set_LN_signal(air->get_LN_signal());
        }
    }
    for (size_t i = 0; i < size(glomeruli); i++)
    { 
        glomeruli[i].update();
    } 
    times.push_back(t);
}
```cpp
void Antennal_Lobe::write_times(string dir) {
    string str = dir + "time.txt";
    FILE *file = fopen(str.c_str(), "w");
    for (auto i : times) {
        fprintf(file, "%f\n", i);
    }
    fclose(file);
}

void Antennal_Lobe::add_odor(Odor_Pulse & odor) {
    odors.push_back(&odor);
}

void Antennal_Lobe::clear_odors() {
    odors = {};
}

void Antennal_Lobe::add_air(Air_Puff & air) {
    airs.push_back(&air);
}

void Antennal_Lobe::clear_airs() {
    airs = {};
}

void Antennal_Lobe::write_voltage(string dir) {
    for (size_t i = 0; i < size(glomeruli); i++)
    {
        for (size_t j = 0; j < size(glomeruli[i].PNs); j++)
        {
            string str = dir + "pn_voltage_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
        }
    }
}
```
FILE * file = fopen(str.c_str(), "w");

for (auto k : glomeruli[i].PNs[j].get_voltage_list()) {
    fprintf(file, "%.f\n", k);
}
fclose(file);

for (size_t j = 0; j < size(glmereuli[i].LN); j++)
{
    string str = dir + "/ln_voltage_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
    FILE * file = fopen(str.c_str(), "w");
    for (auto k : glomeruli[i].LN[j].get_voltage_list()) {
        fprintf(file, "%.f\n", k);
    }
    fclose(file);
}

void Antennal_Lobe::write_excitation(string dir) {
    for (size_t i = 0; i < size(glmereuli); i++)
    {
        for (size_t j = 0; j < size(glmereuli[i].PNs); j++)
        {
            string str = dir + "/pn_excitation_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
            FILE * file = fopen(str.c_str(), "w");
            for (auto k : glmereuli[i].PNs[j].get_excitation_list()) {
                fprintf(file, "%.f\n", k);
            }
            fclose(file);
        }
    }
}
for (size_t j = 0; j < size(glomeruli[i].LN); j++)
{
    string str = dir + "/ln_excitation_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
    FILE * file = fopen(str.c_str(), "w");
    for (auto k : glomeruli[i].LN[j].get_excitation_list()) {
        fprintf(file, "%f\n", k);
    }
    fclose(file);
}
}

void Antennal_Lobe::write_fast_inhibition(string dir) {
for (size_t i = 0; i < size(glomeruli); i++)
{
    for (size_t j = 0; j < size(glomeruli[i].PN); j++)
{
    string str = dir + "/pn_fast_inhibition_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
    FILE * file = fopen(str.c_str(), "w");
    for (auto k : glomeruli[i].PN[j].get_fast_inhibition_list()) {
        fprintf(file, "%f\n", k);
    }
    fclose(file);
}
    for (size_t j = 0; j < size(glomeruli[i].LN); j++)
{
    string str = dir + "/ln_fast_inhibition_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
    FILE * file = fopen(str.c_str(), "w");
    for (auto k : glomeruli[i].LN[j].get_fast_inhibition_list()) {

void Antennal_Lobe::write_slow_inhibition(string dir) {
    for (size_t i = 0; i < size(glmernuli); i++)
    {
        for (size_t j = 0; j < size(glmernuli[i].PNs); j++)
        {
            string str = dir + "/pn_slow_inhibition_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
            FILE * file = fopen(str.c_str(), "w");
            for (auto k : glmernuli[i].PNs[j].get_slow_inhibition_list()) {
                fprintf(file, "%f\n", k);
            }
            fclose(file);
        }
        for (size_t j = 0; j < size(glmernuli[i].LNs); j++)
        {
            string str = dir + "/ln_slow_inhibition_glomerulus_" + to_string(i) + "_neuron_" + to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
            FILE * file = fopen(str.c_str(), "w");
            for (auto k : glmernuli[i].LNs[j].get_slow_inhibition_list()) {
                fprintf(file, "%f\n", k);
            }
            fclose(file);
        }
    }
}
```cpp
void Antennal_Lobe::write_input(string dir) {
    for (size_t i = 0; i < size(glomeruli); i++)
    {
        for (size_t j = 0; j < size(glomeruli[i].PNs); j++)
        {
            string str = dir + "/pn_input_glomerulus_" + to_string(i) + "_neuron_" +
                          to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
            FILE * file = fopen(str.c_str(), "w");
            for (auto k : glomeruli[i].PNs[j].get_input_list()) {
                fprintf(file, "%f
", k);
            }
            fclose(file);
        }
        for (size_t j = 0; j < size(glomeruli[i].LNs); j++)
        {
            string str = dir + "/ln_input_glomerulus_" + to_string(i) + "_neuron_" +
                          to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
            FILE * file = fopen(str.c_str(), "w");
            for (auto k : glomeruli[i].LNs[j].get_input_list()) {
                fprintf(file, "%f
", k);
            }
            fclose(file);
        }
    }
}

void Antennal_Lobe::write_sk(string dir) {
    for (size_t i = 0; i < size(glomeruli); i++)
    {
        for (size_t j = 0; j < size(glomeruli[i].PNs); j++)
        {
```
string str = dir + "/pn_sk_glomerulus_" + to_string(i) + "_neuron_" +
to_string(j) + "_run_" + to_string(cur_brain) + ".txt";
FILE * file = fopen(str.c_str(), "w");
for (auto k : glomeruli[i].PNs[j].get_sk_list()) {
    fprintf(file, "%f\n", k);
}
fclose(file);
}
}

void Antennal_Lobe::write_all(string dir) {
    write_voltage(dir);
    write_excitation(dir);
    write_slow_inhibition(dir);
    write_fast_inhibition(dir);
    write_sk(dir);
    write_input(dir);
}

void Antennal_Lobe::reset() {
    times.clear();
    for (size_t i = 0; i < size(gglomeruli); i++)
    {
        glomeruli[i].reset();
    }
    for (auto odor : odors) {
        odor->reset();
    }
    for (auto air : airs) {
        air->reset();
    }
}
void Antennal_Lobe::hard_reset() {
    times.clear();
    for (size_t i = 0; i < size(glomeruli); i++)
    {
        glomeruli[i].reset();
    }
    odors.clear();
    airs.clear();
}

C.6 Olfactory Stimuli

C.6.1 Odor_Pulse.h

#pragma once
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "Parameters.h"
#include "Glomerulus.h"
using namespace std;

class Odor_Pulse
{
public:
    Odor_Pulse(vector<int> send_to, double start, double duration);
    ~Odor_Pulse();
C.6.2 Odor_Pulse.cpp

```cpp
#include "Odor_Pulse.h"

Odor_Pulse::Odor_Pulse(vector<int> send_to, double start, double duration) {
    t_on = start;
    t_off = start + duration;
    t_r = start + 2*t_rise;
    receivers = send_to;
    started = false;
    type = t_r < t_off;
    if (!type){
```
new_max = max_odor_rate * \exp((\text{rise}_\tau * ((t_{\text{off}} - t_{\text{on}}) - t_{\text{rise}}))) / (1 + \exp((\text{rise}_\tau * ((t_{\text{off}} - t_{\text{on}}) - t_{\text{rise}}))));

signal = 0;

Odor_Pulse::~Odor_Pulse()
{
}

void Odor_Pulse::update() {
    if (started) {
        if (t \geq t_{\text{on}} - \text{timestep}) {
            started = true;
        }
    }
    else {
        if (type) {
            if (t < t_{r}) {
                signal = max_odor_rate * \exp((\text{rise}_\tau * ((t - t_{\text{on}}) - t_{\text{rise}}))) / (1 + \exp((\text{rise}_\tau * ((t - t_{\text{on}}) - t_{\text{rise}}))));
            }
            else if (t < t_{\text{off}}) {
                signal = max_odor_rate;
            }
            else {
                signal = max_odor_rate * \exp(-(t - t_{\text{off}}) / \text{odor}_\tau);
            }
        }
        else {
            if (t < t_{\text{off}}) {
            }
        }
    }
}
signal = max_odor_rate * exp((rise_tau * ((t - t_on) - t_rise)) / (1 + exp((rise_tau * ((t - t_on) - t_rise))));
    }
    else {
        signal = new_max * exp(-(t - t_off) / odor_tau);
    }
}
}
}

double Odor_Pulse::get_PN_signal() {
    return signal;
}

double Odor_Pulse::get_LN_signal() {
    if (started) {
        if (t < t_off) {
            return max_odor_rate;
        }
        else {
            return max_odor_rate * exp(-(t - t_off) / odor_tau);
        }
    }
    else {
        return signal;
    }
}

vector<int> Odor_Pulse::get_glomeruli() {
    return receivers;
}
void Odor_Pulse::reset() {
    started = false;
    signal = 0;
}

C.7  Mechanosensory Stimuli

C.7.1  Air_Puff.h

#pragma once
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include "Parameters.h"
#include "Glomerulus.h"
using namespace std;

class Air_Puff
{
public:
    Air_Puff(double start, double duration);
    ~Air_Puff();
    void update();
    double get_PN_signal();
    double get_LN_signal();
    void reset();
private:
    double signal;
double t_on;
double t_off;
double t_r;
double new_max;
bool started;
bool type;
};

C.7.2 Air_Puff.cpp

#include "Air_Puff.h"

Air_Puff::Air_Puff(double start, double duration)
{
    t_on = start;
    t_off = start + duration;
    t_r = start + 2 * air_t_rise;
    started = false;
    type = t_r < t_off;
    if (!type) {
        new_max = max_air_rate * exp((air_rise_tau * ((t_off - t_on) - air_t_rise)) / (1 + exp((air_rise_tau * ((t_off - t_on) - air_t_rise))));
    }
    signal = 0;
}

Air_Puff::~Air_Puff()
void Air_Puff::update() {
    if (!started) {
        if (t >= t_on - timestep) {
            started = true;
        }
    }
    else {
        if (type) {
            if (t < t_r) {
                signal = max_air_rate * exp((air_rise_tau * ((t - t_on) - air_t_rise))
                ) / (1 + exp((air_rise_tau * ((t - t_on) - air_t_rise))));
            } else if (t < t_off) {
                signal = max_air_rate;
            } else {
                signal = max_air_rate * exp(-(t - t_off) / air_tau);
            }
        } else {
            if (t < t_off) {
                signal = max_air_rate * exp((air_rise_tau * ((t - t_on) - air_t_rise))
                ) / (1 + exp((air_rise_tau * ((t - t_on) - air_t_rise))));
            } else {
                signal = new_max * exp(-(t - t_off) / air_tau);
            }
        }
    }
}
double Air_Puff::get_PN_signal() {
    // return signal;
    if (started) {
        if (t < t_off) {
            return max_air_rate;
        } else {
            return max_air_rate * exp(-(t - t_off) / air_tau);
        }
    } else {
        return signal;
    }
}

double Air_Puff::get_LN_signal() {
    /* if (started && t < t_off) {
        return max_air_rate;
    } else {*/
        return signal;
    */
    return signal;
}

void Air_Puff::reset() {
    started = false;
    signal = 0;
}
C.8 Auxilary Functions

C.8.1 Functions.h

```cpp
#pragma once

#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>

using namespace std;

bool random_check(double prob);
```

C.8.2 Functions.cpp

```cpp
#pragma once

#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>

#include "Functions.h"

using namespace std;

bool random_check(double prob) {
    double test = prob * 10000;
    int roll = rand() % 10000;
    if (roll <= test) {
        return true;
    }
}
```
return false;
}

C.9 Simulation

C.9.1 main.cpp

#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include <direc.h>
#include "Parameters.h"
#include "Antennal_Lobe.h"
using namespace std;

int main() {
    srand(123456789);
    if (mode == 1) {
        Odor_Pulse odor({ 0, 1, 2 }, 1000, 1000);
        Air_Puff air(1000, 1000);
        Antennal_Lobe brain;
        brain.add_odor(odor);
        brain.add_air(air);
    
    for (cur_brain = 1; cur_brain < trials + 1; cur_brain++) {
        t = 0;
        printf("%d\n", cur_brain);
    }
string str = "fires" + to_string(cur_brain) + ".txt";
fires = fopen(str.c_str(), "w");
while (t < endtime) {
    brain.update();
    fprintf(fires, "\n");
    t += timestep;
}
close(fires);
if (cur_brain == 1) {
    brain.write_times("./");
}
brain.write_all("./");
brain.reset();
} 
else if (mode == 2) {
    vector<string> Odor_String = { "/0,1,2", "/0,1,3", "/0,1,4", "/0,1,5", "/0,2,3", "/0,2,4", "/0,2,5", "/0,3,4", "/0,3,5", "/0,4,5", "/1,2,3", "/1,2,4", "/1,2,5", "/1,3,4", "/1,3,5", "/1,4,5", "/2,3,4", "/2,3,5", "/2,4,5", "/3,4,5" };
    int u = 0;
    int v = 1;
    int w = 2;
    for (size_t i = 0; i < Odor_String.size(); i++) {
        mkdir(Odor_String[i].c_str());
        srand(123456789);
        Odor_Pulse odor({ u,v,w }, 1000, 1000);
        w += 1;
        if (w > 5) {
            v += 1;
            if (v > 4) {
                ...
u += 1;
v = u + 1;
}
w = v + 1;
}
Air_Puff air(1000, 1000);
Antennal_Lobe brain;

brain.add_odor(odor);
//brain.add_air(air);

for (cur_brain = 1; cur_brain < trials + 1; cur_brain++) {
t = 0;
printf("%d\n", cur_brain);
string str = Odor_String[i] + "/fires" + to_string(cur_brain) + ".txt";
fires = fopen(str.c_str(), "w");
while (t < endtime) {
    brain.update();
    fprintf(fires, "\n");
    t += timestep;
}
fclose(fires);
if (cur_brain == 1) {
    string time_str = Odor_String[i] + "/time.txt";
    FILE* file = fopen(time_str.c_str(), "w");
    for (auto i : brain.times) {
        fprintf(file, "%f\n", i);
    }
    fclose(file);
}
//brain.write_all();
else if (mode == 3) {
  LN_to_PN_slow_strength = 0;
  vector<string> Odor_String = { 
    "./000 slow", 
    "./005 slow", 
    "./010 slow", 
    "./015 slow", 
    "./020 slow", 
    "./025 slow", 
    "./030 slow", 
    "./035 slow", 
    "./040 slow", 
    "./045 slow", 
    "./050 slow", 
    "./055 slow", 
    "./060 slow", 
    "./065 slow", 
    "./070 slow", 
    "./075 slow", 
    "./080 slow", 
    "./085 slow", 
    "./090 slow", 
    "./095 slow", 
    "./100 slow" 
  };
  for (size_t i = 0; i < Odor_String.size(); i++)
  {
    mkdir(Oodor_String[i].c_str());
    srand(123456789);
    Odor_Pulse odor({ 0,1,2 }, 1000, 1000);
    Air_Puff air(1000, 1000);
    Antennal_Lobe brain;

    brain.add_odor(odor);
    //brain.add_air(air);

    for (cur_brain = 1; cur_brain < trials + 1; cur_brain++)
    {
      t = 0;
      printf("%d\n", cur_brain);
      string str = Odor_String[i] + "/fires" + to_string(cur_brain) + ".txt";
      fires = fopen(str.c_str(), "w");
      while (t < endtime) {
        brain.update();
        fprintf(fires, "\n");
        t += timestep;
      }
    }
  }
fclose(fires);

if (cur_brain == 1) {
    string time_str = Odor_String[i] + "/time.txt";
    FILE* file = fopen(time_str.c_str(), "w");
    for (auto i : brain.times) {
        fprintf(file, "%f\n", i);
    }
    fclose(file);
}
//brain.write_all();
brain.reset();
}
LN_to_PN_slow_strength += .005;
}
else if (mode == 4) {
    inter_LN_to_PN_prob = 0;
    vector<string> Odor_String = { "/0 prob", "/1 prob", "/2 prob", "/3 prob", "/4 prob", "/5 prob", "/6 prob", "/7 prob", "/8 prob", "/9 prob", "/10 prob" };
    for (size_t i = 0; i < Odor_String.size(); i++)
    {
        mkdir(Odor_String[i].c_str());
        srand(123456789);
        Odor_Pulse odor({ 0,1,2 }, 1000, 1000);
        Air_Puff air(1000, 1000);
        Antennal_Lobe brain;

        //brain.add_odor(odor);
        brain.add_air(air);

        for (cur_brain = 1; cur_brain < trials + 1; cur_brain++) {
            t = 0;
        }
    }
printf("%d\n", cur_brain);

string str = Odor_String[i] + "/fires" + to_string(cur_brain) + ".txt";

fires = fopen(str.c_str(), "w");

while (t < endtime) {
    brain.update();
    fprintf(fires, "\n");
    t += timestep;
}
fclose(fires);

if (cur_brain == 1) {
    string time_str = Odor_String[i] + "/time.txt";
    FILE* file = fopen(time_str.c_str(), "w");
    for (auto i : brain.times) {
        fprintf(file, "%f\n", i);
    }
    fclose(file);
}

//brain.write_all();
brain.reset();

}  
inter_LN_to_PN_prob += .1;

}  
return 0;
Appendix D

List of Common Abbreviations

AL: Antennal Lobe
ORN: Olfactory Receptor Neuron
PN: Projection Neuron
LN: Local Neuron
KC: Kenyon Cell
IHP: Initial Hyperpolarization
AHP: After Hyperpolarization
SK current: Calcium Dependent Potassium current
cAMP: Cyclic Adenosine Monophosphate